Electronic Colloquium on Computational Complexity, Report No. 165 (2011)

List Decoding Barnes-Wall Lattices

Elena Grigorescu* Chris Peikert†

December 7, 2011

Abstract

The question of *list decoding* error-correcting codes over finite fields (under the Hamming metric) has been widely studied in recent years. Motivated by the similar discrete structure of linear codes and *point lattices* in $\mathbb{R}^N$, and their many shared applications across complexity theory, cryptography, and coding theory, we initiate the study of list decoding for lattices. Namely: for a lattice $\mathcal{L} \subseteq \mathbb{R}^N$, given a target vector $r \in \mathbb{R}^N$ and a distance parameter d, output the set of all lattice points $w \in \mathcal{L}$ that are within distance d of r.

In this work we focus on combinatorial and algorithmic questions related to list decoding for the well-studied family of *Barnes-Wall* lattices. Our main contributions are twofold:

1. We give tight (up to polynomials) combinatorial bounds on the worst-case list size, which is polynomial in the lattice dimension, for any error radius bounded away from the lattice's minimum distance (in the Euclidean norm).
2. Building on the *unique* decoding algorithm of Micciancio and Nicolosi (ISIT '08), we give a list-decoding algorithm that runs in time polynomial in the lattice dimension and worst-case list size, for any error radius. Moreover, our algorithm is highly parallelizable, and with sufficiently many processors can run in parallel time only *poly-logarithmic* in the lattice dimension.

In particular, our results imply a polynomial-time list-decoding algorithm for error bounded away from the minimum distance, thus beating a typical barrier for error-correcting codes posed by the Johnson radius.

*School of Computer Science, Georgia Institute of Technology. Email: elena_g@csail.mit.edu. This material is based upon work supported by the National Science Foundation under Grant #1019343 to the Computing Research Association for the CIFellows Project.

†School of Computer Science, Georgia Institute of Technology. Email: cpeikert@cc.gatech.edu. This material is based upon work supported by the National Science Foundation under CAREER Award CCF-1054495 and the Alfred P. Sloan Foundation. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the National Science Foundation of the Sloan Foundation.

ISSN 1433-8092

1 Introduction

A linear error-correcting *code* $\mathcal{C}$ of block length N and dimension K over a field $\mathbb{F}$ is a K-dimensional subspace of $\mathbb{F}^N$, generated as all linear combinations of the rows of a full-rank K-by-N matrix over $\mathbb{F}$. The code's minimum distance, denoted $d(\mathcal{C})$, is the minimum Hamming weight of all nonzero codewords. Similarly, a point *lattice* of dimension N and rank K is a discrete additive subgroup of $\mathbb{R}^N$ (or $\mathbb{C}^N$), generated as all integer linear combinations of the rows of a full-rank K-by-N matrix over $\mathbb{R}$ (or $\mathbb{C}$). The lattice's minimum distance, denoted $\lambda(\mathcal{L})$, is the minimum (Euclidean) norm of all nonzero lattice points.

Codes and lattices are intensely studied objects, with many applications in computational complexity, cryptography, and coding theory. In particular, both kinds of objects can be used to encode data so that it can be sent over a noisy channel and recovered reliably. A central question associated with codes is *unique decoding*: given a received word $r \in \mathbb{F}^N$ within Hamming distance less than $d(\mathcal{C})/2$ of some codeword $w \in \mathcal{C}$, find w. Similarly, the unique (often called bounded-distance) decoding problem on lattices is: given a received word $r \in \mathbb{R}^N$ within Euclidean distance less than $\lambda(\mathcal{L})/2$ of some lattice vector $v \in \mathcal{L}$, find v.

For error-correcting codes, Elias [13] and Wozencraft [44] proposed extending the classical unique decoding problem to settings where the amount of error could cause ambiguous decoding. More precisely, the goal of *list decoding* is to find all codewords within a certain radius (typically exceeding $d(\mathcal{C})/2$) of a received word; in many cases, the list is guaranteed to contain few codewords. The first breakthrough algorithmic list decoding results were due to Goldreich and Levin [16] for the Hadamard code, and to Sudan [39] and Guruswami-Sudan [23] for Reed-Solomon codes. These results and others have had countless applications, e.g., in building hard-core predicates for one-way functions [16], in hardness amplification [41], in learning Fourier coefficients [29, 15, 2], and in constructing randomness extractors [42, 43, 25].

There are two central tasks associated with list decoding: combinatorially bounding the number of codewords within a given radius of a received word, and algorithmically finding these codewords. An important question in understanding list decodability is finding the *list-decoding radius* of the code, i.e., the maximum distance from a received word within which the number of codewords is guaranteed to be polynomial in the input parameters.

The Johnson bound. Under the Hamming metric, the *Johnson bound* gives a distance up to which list decoding is guaranteed to be combinatorially efficient. One version of the Johnson bound states that for any code $\mathcal{C}$ of relative distance $\delta = d(\mathcal{C})/N$, a Hamming ball of relative radius $J(\delta) - \epsilon$ contains at most $1/\epsilon^2$ codewords, and a ball of relative radius $J(\delta)$ contains at most $\delta N^2|\mathbb{F}|$ codewords, where $J(\delta) = 1 - \sqrt{1-\delta}$. The Johnson bound is generic since it does not use any structure of the code (not even linearity), and in many cases it is not necessarily the same as the list-decoding radius. It is, however, a barrier in the current analysis of combinatorial list decoding for many well-studied families like Reed-Solomon codes, algebraic geometry codes, Chinese remainder codes, and others. The breakthrough works of Parvaresh-Vardy [34] and Guruswami-Rudra [22] gave families of codes which could be (efficiently) list decoded beyond the Johnson bound, and were followed by several related combinatorial and algorithmic results for other codes (e.g., [8, 18, 28, 17]). For more detailed surveys on list decoding of codes we refer to [40, 19, 20, 21].

1.1 Contributions

Motivated by the common discrete structure of codes and lattices, we initiate the study of list decoding for lattices, from both a combinatorial and algorithmic perspective. The list-decoding problem for a lattice $\mathcal{L}$ is: given a received word r and an error bound, find all lattice vectors $v \in \mathcal{L}$ within the given error bound of r (in, say, the Euclidean norm). In this work we focus on the well-studied *Barnes-Wall* [3] family of lattices,

which share many connections to the Reed-Muller [32, 36] family of error-correcting codes (we elaborate below). Our main contributions are twofold:

1. We give tight (up to polynomials) combinatorial bounds on the worst-case list size, showing it to be polynomial in the lattice dimension N, for any error radius bounded away from the minimum distance of the lattice; see Theorems 1.2 and 1.3 below for precise statements. Moreover, it is already known that the list size is super-polynomial $N^{\Theta(\log N)}$ when the error radius equals the minimum distance.
2. We give a corresponding list-decoding algorithm that, for any error radius, runs in time polynomial in the lattice dimension and worst-case list size. Moreover, our algorithm is highly parallel: with sufficiently many processors it runs in only *poly-logarithmic* $O(\log^2 N)$ parallel time.

We note that one can easily obtain a Johnson-type bound for lattices, independent of the structure of the lattice. Namely, for a lattice $\mathcal{L} \subset \mathbb{C}^N$ with *relative squared minimum distance* (abreviated as rsmd) $\delta = \lambda(\mathcal{L})^2/N$, a Euclidean ball of relative squared radius $\frac{\delta}{2} - \epsilon$ contains at most $\frac{1}{2\epsilon}$ lattice points, and a ball of relative squared radius $\frac{\delta}{2}$ contains at most $4N$ lattice points (see Lemma 2.4). Interestingly, the latter bound is tight for Barnes-Wall lattices (see Corollary 2.5). Since $\delta = 1$ for every Barnes-Wall lattice, our combinatorial and algorithmic results apply far beyond the Johnson bound.

To describe our results in more detail, we need to define Barnes-Wall lattices. Let $\mathbb{G} = \mathbb{Z}[i]$ be the ring of Gaussian integers, and let $\phi = 1 + i \in \mathbb{G}$.

Definition 1.1 (Barnes-Wall lattice)**.** The nth Barnes-Wall lattice $\mathrm{BW}_n \subseteq \mathbb{G}^N$ of dimension $N = 2^n$ is defined recursively as $\mathrm{BW}_0 = \mathbb{G}$, and for positive integer $n \geq 1$ as

$$\mathrm{BW}_n = \{[u, u + \phi v] : u, v \in \mathrm{BW}_{n-1}\}.$$

Observe that if $[u, w] \in \mathrm{BW}_n$ for $u, w \in \mathbb{C}^{N/2}$, then $[w, u] \in \mathrm{BW}_n$. One can check that BW_n is a lattice; indeed, it is easy to verify that it is generated by the rows of the n-fold Kronecker product

$$W = \begin{bmatrix} 1 & 1 \\ 0 & \phi \end{bmatrix}^{\otimes n} \in \mathbb{C}^{N \times N}.$$

The minimum distance of BW_n is $\sqrt{N}$.[1] The mathematical and coding properties of Barnes-Wall lattices have been studied in numerous works, e.g., [1, 6, 26, 27, 33, 37, 38, 31].

To simplify notation, in what follows we use the notion of *relative squared distance*, abbreviated rsd, and use this as the most convenient notion of "error rate." For $x \in \mathbb{C}^m$, define its rsd as $\delta(x) = \frac{1}{m}\|x\|^2$, and for $x, y \in \mathbb{C}^m$ define $\delta(x, y) = \delta(x - y)$. Note that the rsmd of a lattice $\mathcal{L}$ (defined above) satisfies $\delta(\mathcal{L}) = \min_{x,y \in \mathcal{L}, x \neq y} \delta(x, y) = \min_{0 \neq x \in \mathcal{L}} \delta(x)$, and that the rsmd of BW_n is 1. We use the base-2 logarithm throughout the paper.

Combinatorial bounds. Let $\ell(\delta, n)$ denote the worst-case list size (over all received words) for BW_n at rsd δ. We prove the following upper bound.

Theorem 1.2. *For any integer $n \geq 0$ and real $\epsilon > 0$, we have*

$$\ell(1 - \epsilon, n) \leq 4 \cdot (1/\epsilon)^{16n} = N^{O(\log(1/\epsilon))}.$$

[1] The fundamental volume of BW_n in $\mathbb{C}^N$ is $\det(W) = 2^{nN/2}$, so its normalized minimum distance is $\sqrt{N}/\det(W)^{1/(2N)} = \sqrt[4]{N}$. This is better than the normalized minimum distance 1 of the integer lattice $\mathbb{G}^N$, but worse that the largest possible of $\Theta(\sqrt{N})$ for N-dimensional lattices.

Moreover, we show that the above bound is tight, up to polynomials.

Theorem 1.3. *For any integer $n \geq 0$ and $\epsilon \in [2^{-n}, 1]$, we have*

$$\ell(1-\epsilon, n) \geq 2^{(n - \log \frac{1}{\epsilon}) \log \frac{1}{2\epsilon}}.$$

In particular, for any constant $\epsilon > 0$ (or even any $\epsilon \geq N^{-c}$ for $c < 1$), we have $\ell(1-\epsilon, n) = N^{\Omega(\log(1/\epsilon))}$.

It is also known that at rsd 1, the maximum list size $\ell(1, n)$ is quasi-polynomial $N^{\Theta(\log N)}$ in the lattice dimension, and is achieved by letting the received word be any lattice point (see, e.g., [6, Chapter 1, §2.2, page 24]). Because the relative minimum distance of BW_n is exactly 1, here we are just considering the number of lattice points at minimum distance from the origin, the so-called "kissing number" of the lattice.

List-decoding algorithm. We complement the above combinatorial bounds with an algorithmic counterpart, which builds upon the unique (bounded-distance) decoding algorithm of Micciancio and Nicolosi [31] for rsd up to $\frac{1}{4}$.

Theorem 1.4. *There is a deterministic algorithm that, given any received word $r \in \mathbb{C}^N$ and $\delta \geq 0$, outputs the list of all points in BW_n that lie within* rsd *δ of r, and runs in time $O(N^2 \cdot \ell(\delta, n)^2)$.*

We also remark that the algorithm can be parallelized just as in [31], and runs in only polylogarithmic $O(\log^2 N)$ parallel time on $p \geq N^2 \cdot \ell(\delta, n)^2$ processors.

Theorems 1.2 and 1.4 immediately imply the following corollary for $\delta = 1 - \epsilon$.

Corollary 1.5. *There is a deterministic algorithm that, given a received word $r \in \mathbb{C}^N$ and $\epsilon \in (0, 1)$, outputs the list of all lattice points in BW_n that lie within* rsd *$(1-\epsilon)$ of r, and runs in time $(1/\epsilon)^{O(n)} = N^{O(\log(1/\epsilon))}$.*

Given the lower bounds, our algorithm is optimal in the sense that for any constant $\epsilon > 0$, it runs in $\mathrm{poly}(N)$ time for rsd $1 - \epsilon$, and that list decoding in $\mathrm{poly}(N)$ time is impossible (in the worst case) at rsd 1.

1.2 Proof Overview and Techniques

Combinatorial bounds. Our combinatorial results exploit a few simple observations. The first is that by the Pythagorean theorem, if $\delta = \delta(r, w)$ is the error rate between a received vector $r = [r_0, r_1] \in \mathbb{C}^N$ and a lattice vector $w = [w_0, w_1] \in \mathrm{BW}_n$ (where $r_i \in \mathbb{C}^{N/2}$ and $w_i \in \mathrm{BW}_{n-1}$), then $\delta(r_b, w_b) \leq \delta$ for some $b \in \{0, 1\}$. The second observation is that BW lattices are closed under the operation of swapping the two halves of their vectors, namely, $[w_0, w_1] \in \mathrm{BW}_n$ if and only if $[w_1, w_0] \in \mathrm{BW}_n$. Therefore, without loss of generality we can assume that $\delta(r_0, w_0) \leq \delta$, while incurring only an extra factor of 2 in the final list size.

The other important fact is the relationship between the error rates for the two Barnes-Wall vectors $u = w_0, v = \frac{1}{\phi}(w_1 - w_0) \in \mathrm{BW}_{n-1}$ that determine w; namely, we have $\delta = \frac{1}{2}\delta(r_0, u) + \delta(\frac{1}{\phi}(r_1 - u), v)$. (See Lemma 2.1.) Since $\delta(r_0, u) \leq \delta$, we have must have $\delta(\frac{1}{\phi}(r_1 - u), v) = \delta - \frac{1}{2}\delta(r_0, w_0) \in [\delta/2, \delta]$.

Our critical insight in analyzing the list size is to carefully partition the lattice vectors in the list according to their distances from the respective halves of the received word. Informally, a larger error rate on the left half (i.e., between r_0 and u) allows for a larger list of u's, but also implies a smaller error rate on the right half (i.e., between $\frac{1}{\phi}(r_1 - u)$ and v), which limits the number of corresponding v's. We bound the total list size using an inductive argument for various carefully chosen ranges of error rates at lower dimensions. Remarkably, this technique along with the Johnson bound allows us to obtain tight combinatorial bounds on the list size for error rates all the way up to the minimum distance.

As a warm-up example, which also serves as an important step when analyzing larger error rates, Lemma 2.6 gives a bound of $\ell(\frac{5}{8}, n) \leq 4 \cdot 24^n = \mathrm{poly}(N)$ for error rate $\delta = \frac{5}{8}$. This bound is obtained by partitioning according to the two cases $\delta(r_0, u) \in [0, \frac{5}{12})$ and $\delta(r_0, u) \in [\frac{5}{12}, \frac{5}{8}]$, which imply that the error rate between v and $\frac{1}{\phi}(r_1 - u)$ is at most $\frac{5}{8}$ and $\frac{5}{12}$, respectively. When bounding the corresponding number of u's and v's, the error rates up to $\frac{5}{12} < \frac{1}{2}$ are handled by the Johnson bound, and error rates up to $\frac{5}{8}$ are handled by induction.

To extend the argument to error rates up to $\delta = 1 - \epsilon$, we need to partition into three cases, including ones which involve error rates $1 - \frac{3\epsilon}{2}$ and $\frac{3}{4}$. In turn, the bound for error rate $\frac{3}{4}$ also uses three cases, plus the above bound for error rate $\frac{5}{8}$. Interestingly, all our attempts to use fewer cases or a more direct analysis resulted in qualitatively worse list size bounds, such as $N^{O(\log^2(1/\epsilon))}$ or worse.

Lastly, our lower bounds from Theorem 1.3 are obtained by using a representation of BW lattices in terms of RM codes (see Fact 2.8), and by adapting the lower bounds from [18] for RM codes to BW lattices.

List-decoding algorithm. A natural approach to devising a list-decoding algorithm using the above facts (also used in the context of Reed-Muller codes [18]) is to first list decode the left half r_0 of the received word to get a list of u's, and then sequentially run through the output list to decode the right half $\frac{1}{\phi}(r_1 - u)$ and get a corresponding list of v's for each value of u. However, because the recursion has depth n, this algorithm runs in super-polynomial time roughly ℓ^n, where $\ell = \ell(\delta, n-1)$ is the list size for the recursive calls.

Instead, our list decoding algorithm is based on the bounded-distance (unique) decoding algorithm for BW lattices of Micciancio and Nicolosi [31]. They give an elegant divide-and-conquer algorithm that decodes from error up to half the minimum distance (i.e., $\delta = \frac{1}{4}$) in quasi-linear $\tilde{O}(N)$ time, or even poly-logarithmic $O(\log^c N)$ parallel time on a sufficiently large $\mathrm{poly}(N)$ number of processors. This was the first known unique decoding algorithm for BW_n lattices, for all n.

The main feature of the Micciancio-Nicolosi algorithm, which we exploit in our algorithm as well, is the use of a distance-preserving linear automorphism $\mathcal{T}$ of the BW lattice, i.e., $\mathcal{T}(\mathrm{BW}_n) = \mathrm{BW}_n$. In particular, a lattice vector $w \in \mathrm{BW}_n$ can be reconstructed from just *one* arbitrary half of each of $w = [w_0, w_1]$ and $\mathcal{T}(w) = [\mathcal{T}_0(w), \mathcal{T}_1(w)]$. Recall that for a received word $r = [r_0, r_1]$ (where $r_i \in \mathbb{C}^{N/2}$), we are guaranteed that $\delta(r_b, w_b) \leq \delta(r, w)$ for some $b \in \{0, 1\}$, and similarly for $\mathcal{T}(r)$ and $\mathcal{T}(w)$. These facts pretty straightforwardly yield a divide-and-conquer, parallelizable list-decoding algorithm that recursively list decodes each of the four halves $r_0, r_1, T_0(r), T_1(r)$ and reconstructs a list of solutions by combining appropriate pairs from the sub-lists. The runtime of this algorithm is only quadratic in the worst-case list size, times a $\mathrm{poly}(N)$ factor (see Section 3).

1.3 Comparison with Reed-Muller Codes

Here we discuss several common and distinguishing features of Barnes-Wall lattices and Reed-Muller codes.

Definition 1.6 (Reed-Muller code)**.** For integers $d, n \geq 0$, the Reed-Muller code of degree d in n variables (over $\mathbb{F}_2$) is defined as

$$\mathrm{RM}_n^d = \{\langle p(\alpha) \rangle_{\alpha \in \mathbb{F}_2^n},\ p \in \mathbb{F}_2[x_1, \ldots, x_n] \text{ and } \deg(p) \leq d\}.$$

An equivalent recursive definition is $\mathrm{RM}_n^0 = \{\vec{0}, \vec{1}\} \subseteq \mathbb{F}_2^{2^n}$ for any integer $n \geq 0$, and $\mathrm{RM}_n^d = \{[u, u + v] : u \in \mathrm{RM}_{n-1}^d, v \in \mathrm{RM}_{n-1}^{d-1}\}$. Here if $u \in \mathrm{RM}_{n-1}^d, v \in \mathrm{RM}_{n-1}^{d-1}$ correspond to polynomials $p_u, p_v \in \mathbb{F}_2[x_1, \ldots x_{n-1}]$ respectively, then the codeword $[u, u + v] \in \mathrm{RM}_n^d$ corresponds to the polynomial $p = p_u + x_n \cdot p_v \in \mathbb{F}_2[x_1, \ldots, x_n]$.

The recursive definition of RM codes already hints at structural similarities between BW lattices and RM codes. Indeed, BW lattices can be equivalently defined as evaluations modulo ϕ^n of (Gaussian) integer multilinear polynomials in n variables over the domain $\{0, \phi\}^n$. Recall that an integer multilinear polynomial $p \in \mathbb{G}[x_1, \ldots, x_n]$ is one whose monomials have degree at most one in each variable (and hence total degree at most n), i.e.,

$$p(x_1, \ldots, x_n) = \sum_{S \in \{0,1\}^n} a_S \cdot \prod_{i \in S} x_i$$

where each $a_S \in \mathbb{G}$. A simple inductive argument proves the following lemma.

Lemma 1.7. *$\mathrm{BW}_n = \phi^n\mathbb{G} + \{\langle p(x)\rangle_{x \in \{0,\phi\}^n} : p \in \mathbb{G}[x_1, \ldots, x_n]$ is multilinear$\}$.*

Thus, while RM_n^d codewords correspond to low-degree polynomials (when d is small), BW lattice points correspond to possibly high-degree polynomials.

As an immediate application, our main theorems imply the following corollary regarding the set of integer multilinear polynomials that approximate a function $f\colon \{0, \phi\}^n \to \mathbb{C}$.

Corollary 1.8. *Given a map $f : \{0, \phi\}^n \to \mathbb{C}$ (represented as a lookup table) and $\epsilon = \Omega(N^{-c})$ for some $c < 1$, there exists an algorithm that outputs in time $N^{O(\log(1/\epsilon))}$ all the integer multilinear polynomials $g\colon \{0, \phi\}^n \to \mathbb{C}$ such that $\|f - g\|^2 \leq (1 - \epsilon)N$.*

Just as in our algorithmic results for BW lattices, the recursive structure of RM codes is critically used in list-decoding algorithms for these codes, but in a different way than in our algorithm. The list-decoding algorithm for RM_n^d given in [18] recursively list decodes one of the halves of a received word, and then for each codeword in the list it recursively list decodes the other half of the received word. The recursion has depth d and thus has a total running time of $\mathrm{poly}(N) \cdot \ell(\delta)^d$, where $\ell(\delta)$ is the list size at relative (Hamming) distance δ. As mentioned above, a similar algorithm can work for BW lattices, but the natural analysis implies a super-polynomial $\ell(\delta)^n$ lower bound on the running time, since now the recursion has depth n. The reason we can overcome this potential bottleneck is the existence of the linear automorphism $\mathcal{T}$ of BW_n, which allows us to make only a *constant* number of recursive calls (independently of each other), plus a $\mathrm{poly}(N) \cdot \ell(\delta)^2$-time combining step, which yields a runtime of the form $O(1)^n \cdot \mathrm{poly}(N) \cdot \ell(\delta)^2 = \mathrm{poly}(N) \cdot \ell(\delta)^2$.

We note that RM_n^d codes are efficiently list decodable up to a radius larger than the minimum distance [18]. We also remark that while RM codes are some of the oldest and most intensively studied codes, it was not until recently that their list-decoding properties have been very well understood [35, 18, 28].

1.4 Other Related Work

Cohn and Heninger [5] study a list-decoding model on polynomial lattices, under both the Hamming metric and certain 'non-Archimedian' norms. Their polynomial analogue of Coppersmith's theorem [7] implies, as a special case, Guruswami and Sudan's result on list decoding Reed-Solomon codes [23].

Decoding and list decoding in the Euclidean space has been also considered for embeddings into real vector spaces of codes classically defined over finite fields. These embeddings can give rise to so-called spherical codes, where the decoding problem has as input a received vector on the unit sphere, and is required to output the points in the code (also on the unit sphere) that form a small angle with the given target. Another decoding model that is well-suited to real vectors is soft-decision decoding. Here, for each position in the vector, and for each symbol, it is assigned a weight representing the confidence that the received symbols was the actual transmitted symbol. Soft decision unique decoding for RM codes was studied in [10, 12, 11], and list-decoding algorithms were shown in [9, 14].

Further, the question of decoding lattices is related to the well-studied *vector quantization* problem. In this problem, vectors in the ambient space need to be rounded to nearby points of a discrete lattice; for further details on this problem see, for example, [6].

Organization. In Section 2 we prove our combinatorial upper and lower bounds for BW lattices. In Section 3 we present and analyze our main list-decoding algorithm. We conclude with several open problems in Section 4.

2 Combinatorial Bounds

We start with a few basic definitions. For a lattice $\mathcal{L}$, a vector $r \in \mathbb{C}^m$ (often called a received word) and any $\delta \geq 0$, define $L_{\mathcal{L}}(r, \delta) = \{x \in \mathcal{L} : \delta(r, x) \leq \delta\}$ to be the list of lattice points $w \in \mathcal{L}$ such that $\delta(r, w) \leq \delta$. We often omit the subscript $\mathcal{L}$ when the lattice is clear from context. For $\delta \geq 0$ and nonnegative integer n with $N = 2^n$, we define $\ell(\delta, n) = \max_{r \in \mathbb{C}^n} |L(r, \delta)|$ to be the maximum list size for rsd δ.

2.1 Helpful Lemmas

We start with two simple but important observations about Barnes-Wall lattices. The first relates the rsd's between the respective "left" and "right" halves of a received word and a lattice point. The second relates the list sizes for the same rsd but different dimensions.

Lemma 2.1. *Let $r = [r_0, r_1] \in \mathbb{C}^N$ with $r_0, r_1 \in \mathbb{C}^{N/2}$, and $w = [u, u + \phi v] \in BW_n$ for $u, v \in BW_{n-1}$. Let $\delta = \delta(r, w)$, $\delta_0 = \delta(r_0, u)$ and $\delta_1 = \delta(\frac{1}{\phi}(r_1 - u), v)$. Then $\delta = \frac{\delta_0}{2} + \delta_1$.*

Proof. We have

$$\delta(r, w) = \frac{\delta(r_0, u) + \delta(r_1, u + \phi v)}{2} = \frac{\delta_0}{2} + \frac{\|\phi\|^2 \cdot \delta(\frac{1}{\phi}(r_1 - u), v)}{2} = \frac{\delta_0}{2} + \delta_1. \qquad \square$$

Lemma 2.2. *For any $\delta \geq 0$ and $n \geq 1$, we have $\ell(\delta, n-1) \leq \ell(\delta, n)$.*

Proof. Let $r \in \mathbb{C}^{N/2}$ and $w \in L(r, \delta) \subseteq \mathrm{BW}_{n-1}$. Then $\delta([r, r], [w, w]) = \delta(r, w)$, and since $[w, w] \in \mathrm{BW}_n$ (because $w \in \mathrm{BW}_{n-1}$) it follows that $[w, w] \in L([r, r], \delta)$. $\square$

2.2 Johnson Bound

Here we give a Johnson-type bound on the list size, which applies to arbitrary lattices.

Lemma 2.3 (Adapted from [24])**.** *Let $x_1, \ldots, x_t \in \mathbb{C}^m$ and $\epsilon > 0$.*

1. *If $\Re\langle x_i, x_j \rangle \leq 0$ for all $i \neq j$, then $t \leq 4m$.*
2. *If $\|x_i\|^2 \leq \frac{1}{2} - \epsilon$ and $\Re\langle x_i, x_j \rangle \leq -\epsilon$ for all $i \neq j$, then $t \leq \frac{1}{2\epsilon}$.*

Proof. Proofs of Item 1 appear in [24] and [4, Chapter 10, page 71]. Actually, these sources refer to the inner product space $\mathbb{R}^m$ and give a bound of $t \leq 2m$ when $\langle x_i, x_j \rangle \leq 0$ for all $i \neq j$. Our form of the statement follows immediately, because the inner product space $\mathbb{C}^m$ is a dimension-$2m$ inner product space over $\mathbb{R}$, and the inner product on $\mathbb{R}^{2m}$ corresponds to the real part of the inner product on $\mathbb{C}^m$.

For Part 2 we have

$$\begin{aligned} 0 \le \left\|\sum_i x_i\right\|^2 = \left\langle \sum x_i, \sum x_i \right\rangle &\le \sum_i \langle x_i, x_i\rangle + \sum_{i\neq j} \langle x_i, x_j\rangle \\ &= \sum_i \|x_i\|^2 + \sum_{i\neq j} \Re\langle x_i, x_j\rangle \\ &\le t(\tfrac{1}{2} - \epsilon) - t(t-1)\epsilon. \end{aligned}$$

Rearranging, we have $t \le \frac{1}{2\epsilon}$. □

An immediate application of the above lemma yields a Johnson-type bound for lattices.

Lemma 2.4 (Johnson bound)**.** *Let $\mathcal{L} \subset \mathbb{C}^N$ be a lattice of* rsmd *$\delta(\mathcal{L})$, and let $r \in \mathbb{C}^N$ be arbitrary. Then*

1. *$|L(r, \frac{\delta(\mathcal{L})}{2})| \le 4N$, and*
2. *$|L(r, \frac{\delta(\mathcal{L})}{2} - \epsilon)| \le \frac{1}{2\epsilon}$ for any $\epsilon > 0$.*

(In reading these bounds, recall that $\frac{\delta(\mathcal{L})}{2}$ is not the relative unique-decoding distance of $\mathcal{L}$; instead, $\frac{\delta(\mathcal{L})}{4}$ is, because $\delta(\mathcal{L})$ is the relative *squared* minimum distance of the lattice.)

Proof. Let $L(r, \delta) = \{b_1, \ldots, b_t\} \subset \mathcal{L}$, where $\delta = \frac{\delta(\mathcal{L})}{2}$ or $\delta = \frac{\delta(\mathcal{L})}{2} - \epsilon$ is the rsd of interest above. Let $x_i = \frac{1}{\sqrt{N}}(b_i - r)$ for each i, which implies that $\|x_i\|^2 = \frac{1}{N}\|b_i - r\|^2 \le \delta$. Since $b_i, b_j \in \mathcal{L}$ are distinct for $i \neq j$, it follows that $\|x_i - x_j\|^2 = \frac{1}{N}\|b_i - b_j\|^2 \ge \delta(\mathcal{L})$. Since

$$\|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - \langle x_i, x_j\rangle - \langle x_j, x_i\rangle = \|x_i\|^2 + \|x_j\|^2 - 2\Re\langle x_i, x_j\rangle,$$

we have (for $i \neq j$)

$$\Re\langle x_i, x_j\rangle = \tfrac{1}{2}(\|x_i\|^2 + \|x_j\|^2 - \|x_i - x_j\|^2) \le \delta - \frac{\delta(\mathcal{L})}{2}.$$

By substituting $\delta = \frac{\delta(\mathcal{L})}{2}$ we get that $\Re\langle x_i, x_j\rangle \le 0$, and the first claim follows from Item 1 of Lemma 2.3. Similarly, by substituting $\delta = \frac{\delta(\mathcal{L})}{2} - \epsilon$ we get that $\Re\langle x_i, x_j\rangle \le -\epsilon$, and so the second claim follows from Item 2 of Lemma 2.3. □

Corollary 2.5. *For the lattice $BW_n \subseteq \mathbb{C}^N$ and any $\epsilon > 0$, we have $\ell(\frac{1}{2}, n) = 4N$ and $\ell(\frac{1}{2} - \epsilon, n) \le \frac{1}{2\epsilon}$.*

Proof. Since $\delta(\mathrm{BW}_n) = 1$, the upper bounds follow immediately by Lemma 2.4. For the equality $\ell(\frac{1}{2}, n) = 4N$, an easy inductive argument shows that $|L(r, \frac{1}{2})| = 4N$ for the received word $r = (\frac{\phi}{2}, \ldots, \frac{\phi}{2}) \in \mathbb{C}^N$. □

2.3 Beyond the Johnson Bound

In this section we prove our main combinatorial bounds on the list size for Barnes-Wall lattices $\mathrm{BW}_n \subseteq \mathbb{G}^N$. Our main result is that the list size at rsd $(1-\epsilon)$ is $(1/\epsilon)^{O(n)} = N^{O(\log(1/\epsilon))}$ for any $\epsilon > 0$. The proof strategy is inductive, and is based on a careful partitioning of the lattice vectors in the list according to the distances of their left and right halves from the respective halves of the received word. Intuitively, the larger the distance on one half, the smaller the distance on the other (Lemma 2.1 above makes this precise). The total list size can therefore be bounded using list bounds for various carefully chosen distances at lower dimensions. Our

analysis relies on a $\mathrm{poly}(N)$ list-size bound for $\mathsf{rsd}\ \frac{3}{4}$, which in turn relies on a $\mathrm{poly}(N)$ bound for $\mathsf{rsd}\ \frac{5}{8}$. We first prove these simpler bounds, also using a partitioning argument. (Note that the concrete constants appearing below are chosen to simplify the analysis, and are likely not optimal.)

Lemma 2.6. *For any integer $n \geq 0$, we have $\ell(\frac{5}{8}, n) \leq 4 \cdot 24^n$.*

Proof. The claim is clearly true for $n = 0$, so suppose $n \geq 1$ with $N = 2^n$. Let $r = [r_0, r_1] \in \mathbb{C}^N$ with $r_0, r_1 \in \mathbb{C}^{N/2}$ be an arbitrary received word, and let $w = [u, u + \phi v] \in L(r, \frac{5}{8})$ for $u, v \in \mathrm{BW}_{n-1}$. Let $\delta = \delta(r, w) \leq \frac{5}{8}$, $\delta_0 = \delta(r_0, u)$ and $\delta_1 = \delta(\frac{1}{\phi}(r_1 - u), v)$.

Note that $\delta = \frac{1}{2}(\delta(r_0, u) + \delta(r_1, u + \phi v)) \leq \frac{5}{8}$. Without loss of generality, we can assume that $\delta_0 = \delta(r_0, u) \leq \frac{5}{8}$. For if not, then we would have $\delta(r_1, u+\phi v) \leq \frac{5}{8}$, and since $[a, b] \in \mathrm{BW}_n$ implies $[b, a] \in \mathrm{BW}_n$ for $a, b \in \mathbb{G}^{N/2}$, we could instead work with the received word $r' = [r_1, r_0]$ and $w' = [u + \phi v, u] \in L(r', \frac{5}{8})$. This incurs a factor of at most 2 in the total list size, which we account for in the analysis below.

Assuming $\delta_0 \leq \frac{5}{8}$, we now split the analysis into two cases: $\delta_0 \in [0, \frac{5}{12})$, and $\delta_0 \in [\frac{5}{12}, \frac{5}{8}]$. By Lemma 2.1, these cases correspond to $\delta_1 \leq \frac{5}{8}$ and $\delta_1 \leq \frac{5}{12}$, respectively. Since $u \in L(r_0, \delta_0)$ and $v \in L(\frac{1}{\phi}(r_1 - u), \delta_1)$, after incorporating the factor of 2 from the argument above we have

$$\begin{aligned}
\ell(\tfrac{5}{8}, n) &\leq 2 \cdot \big(\ell(\tfrac{5}{12}, n-1) \cdot \ell(\tfrac{5}{8}, n-1) + \ell(\tfrac{5}{8}, n-1) \cdot \ell(\tfrac{5}{12}, n-1)\big) \\
&= 4 \cdot \ell(\tfrac{5}{12}, n-1) \cdot \ell(\tfrac{5}{8}, n-1) \\
&\leq 4 \cdot 6 \cdot \ell(\tfrac{5}{8}, n-1) \\
&\leq 24^n \cdot \ell(\tfrac{5}{8}, 0),
\end{aligned}$$

where the penultimate inequality is by Item 2 of Corollary 2.5, and the final inequality is by unwinding the recurrence. □

Lemma 2.7. *For any integer $n \geq 0$, we have $\ell(\frac{3}{4}, n) \leq 4 \cdot 24^{2n}$.*

Proof. The claim is clearly true for $n = 0$, so suppose $n \geq 1$; we proceed by induction on n. Define the same notation as in the proof of Lemma 2.6, using rsd bound $\frac{3}{4}$ instead of $\frac{5}{8}$.

As before, we assume that $\delta_0 \leq \frac{3}{4}$ and account for the accompanying factor of 2 in the list size. This time we split the analysis into three cases: $\delta_0 \in [0, \frac{1}{4})$, $\delta_0 \in [\frac{1}{4}, \frac{5}{8})$, and $\delta_0 \in [\frac{5}{8}, \frac{3}{4}]$. By Lemma 2.1, these correspond to $\delta_1 \leq \frac{3}{4}$, $\delta_1 \leq \frac{5}{8}$, and $\delta_1 \leq \frac{7}{16}$, respectively.

Using Corollary 2.5 and Lemma 2.6, we therefore have

$$\begin{aligned}
\ell(\tfrac{3}{4}, n) &\leq 2 \cdot \big(\ell(\tfrac{1}{4}, n-1) \cdot \ell(\tfrac{3}{4}, n-1) + \ell(\tfrac{5}{8}, n-1) \cdot \ell(\tfrac{5}{8}, n-1) + \ell(\tfrac{3}{4}, n-1) \cdot \ell(\tfrac{7}{16}, n-1)\big) \\
&\leq 2 \cdot (2 + 8) \cdot \ell(\tfrac{3}{4}, n-1) + 2 \cdot \ell(\tfrac{5}{8}, n-1)^2 \\
&\leq 20 \cdot 4 \cdot 24^{2(n-1)} + 32 \cdot 24^{2(n-1)} \\
&\leq 4 \cdot 24^{2n}.
\end{aligned}$$

□

We are now ready to prove our main combinatorial bound.

Proof of Theorem 1.2. Recall that we need to show that, for any $n \geq 0$ and $\epsilon > 0$, we have $\ell(1 - \epsilon, n) \leq 4 \cdot (1/\epsilon)^{16n}$. The claim is clearly true for $n = 0$. We proceed by induction on n; namely, we assume that for all $\eta > 0$ it holds that $\ell(1 - \eta, n - 1) \leq 4 \cdot (1/\eta)^{16(n-1)}$. Define the same notation as in the proof of Lemma 2.6, using rsd bound $1 - \epsilon$ instead of $\frac{5}{8}$.

As in earlier proofs, we assume that $\delta_0 \leq 1-\epsilon$ and account for the accompanying factor of 2 in the list size. We split the analysis into 3 cases: $\delta_0 \in (0, \frac{1}{2}-\epsilon)$, $\delta_0 \in [\frac{1}{2}-\epsilon, 1-\frac{3\epsilon}{2})$, and $\delta_0 \in [1-\frac{3\epsilon}{2}, 1-\epsilon]$. By Lemma 2.1, these correspond to $\delta_1 \leq 1-\epsilon$, $\delta_1 \leq \frac{3}{4}-\frac{\epsilon}{2} < \frac{3}{4}$, and $\delta_1 \leq \frac{1}{2}-\frac{\epsilon}{4}$, respectively.

Using Corollary 2.5, Lemma 2.7 and the inductive hypothesis, it follows that

$$\begin{aligned}
\ell(1-\epsilon,n) &\leq 2\left(\ell(1-\epsilon,n-1)\cdot\left(\ell(\tfrac{1}{2}-\epsilon,n-1)+\ell(\tfrac{1}{2}-\tfrac{\epsilon}{4},n-1)\right)+\ell(1-\tfrac{3\epsilon}{2},n-1)\cdot\ell(\tfrac{3}{4},n-1)\right)\\
&\leq 2\left(\ell(1-\epsilon,n-1)(\tfrac{1}{2\epsilon}+\tfrac{2}{\epsilon})+\ell(1-\tfrac{3\epsilon}{2},n-1)\cdot 4\cdot 24^{2(n-1)}\right)\\
&\leq \tfrac{5}{\epsilon}\cdot\ell(1-\epsilon,n-1)+8\cdot 24^{2(n-1)}\cdot\ell(1-\tfrac{3\epsilon}{2},n-1)\\
&\leq \tfrac{20}{\epsilon}\cdot(\tfrac{1}{\epsilon})^{16(n-1)}+32\cdot 24^{2(n-1)}\cdot(\tfrac{2}{3\epsilon})^{16(n-1)}\\
&\leq (\tfrac{1}{\epsilon})^{16(n-1)}\cdot(\tfrac{20}{\epsilon}+32\cdot(24^2\cdot(\tfrac{2}{3})^{16})^{(n-1)})\\
&\leq (\tfrac{1}{\epsilon})^{16(n-1)}\cdot(\tfrac{52}{\epsilon})\\
&\leq 4\cdot(\tfrac{1}{\epsilon})^{16n}
\end{aligned}$$

when $\epsilon \leq \frac{4}{5}$. If $\epsilon > \frac{4}{5}$ then $\ell(1-\epsilon,n) = 1 \leq 4\cdot(\frac{1}{\epsilon})^{16n}$, and the proof is complete. □

Notice that in the above proof, it is important to use an upper bound like $\delta_0 \leq 1-\frac{3\epsilon}{2}$ in one of the cases, so that the factor $(\frac{2}{3})^{16(n-1)}$ in the inductive list bound can cancel out the corresponding factor of $24^{2(n-1)}$ for the corresponding rsd bound $\delta_1 \leq \frac{3}{4}$. This allows the recurrence to be dominated by the term

$$\ell(1-\epsilon,n-1)\cdot\ell(\tfrac{1}{2}-\epsilon,n-1) = O(\tfrac{1}{\epsilon})\cdot\ell(1-\epsilon,n-1),$$

yielding a solution of the form $(1/\epsilon)^{O(n)}$.

2.4 Lower Bounds

For our lower bounds we make use of a relationship between Barnes-Wall lattices and Reed-Muller codes, and then apply known lower bounds for the latter.

Fact 2.8 ([26, §IV.B])**.**

$$BW_n = \{\sum_{d=0}^{n-1}\phi^d c_d + \phi^n c_n : c_d \in RM_n^d, 0 \leq d \leq n-1; c_n \in \mathbb{G}^N\},$$

where the embedding of $\mathbb{F}_2$ into $\mathbb{C}$ is given by $0 \mapsto 0$ and $1 \mapsto 1$. In particular, any codeword $c_d \in RM_d^n$ gives rise to a lattice point $\phi^d \cdot c_d \in BW_n$,

Fact 2.9 ([30, Chap. 13, §4])**.**

1. *The minimum distance of RM_n^d is 2^{n-d}. In particular, the characteristic vector $c_V \in \mathbb{F}_2^{2^n}$ of any subspace $V \subseteq \mathbb{F}_2^n$ of dimension $k \geq n-d$ is a codeword of RM_n^d.*

   (The characteristic vector $c_S \in \mathbb{F}_2^{2^n}$ of a set $S \subseteq \mathbb{F}_2^n$ is defined by indexing the coordinates of $\mathbb{F}_2^{2^n}$ by elements $\alpha \in \mathbb{F}_2^n$, and letting $(c_S)_\alpha = 1$ if and only if $\alpha \in S$.)

2. *There are $2^d \cdot \prod_{i=0}^{n-d-1}\frac{2^{n-i}-1}{2^{n-d-i}-1} > 2^{d(n-d)}$ subspaces of dimension $n-d$ in $\mathbb{F}_2^n$.*

Proof of Theorem 1.3. Let $k \geq 0$ be an integer such that $2^n\epsilon \leq 2^k \leq 2^{n+1}\epsilon$. Let the received word be $r = \phi^k \cdot [1, 0, \ldots, 0] \in \mathbb{G}^N$, where we assume that the first coordinate is indexed by $0^n \in \mathbb{F}_2^n$. By Fact 2.9 and Fact 2.8, for any subspace $H \subseteq \mathbb{F}_2^n$ of dimension $n-k$, we have $\phi^k \cdot c_H \in \mathrm{BW}_n$. Notice that $\|r - \phi^k \cdot c_H\|^2 = |\phi^k|^2 \cdot \|c_H - [1, 0, \ldots, 0]\|^2 = 2^k \cdot (2^{n-k} - 1) = 2^n - 2^k \leq 2^n(1-\epsilon)$. By Fact 2.9, there are at least $2^{k(n-k)} \geq 2^{(n - \log \frac{1}{\epsilon}) \log \frac{1}{2\epsilon}}$ subspaces $H \subset \mathbb{F}_2^n$ of dimension $n-k$, which completes the proof. □

3 List-Decoding Algorithm

In this section we prove Theorem 1.4 by giving a list-decoding algorithm that runs in time polynomial in the list size; in particular, by Theorem 1.2 it runs in time $N^{O(\log(1/\epsilon))}$ for rsd $(1-\epsilon)$ for any fixed $\epsilon > 0$. The runtime and error tolerance are optimal (up to polynomial overhead) in the sense that the list size can be $N^{\Omega(\log(1/\epsilon))}$ by Theorem 1.3, and can be super-polynomial in N for rsd 1 or more.

The list-decoding algorithm is closely related to the highly parallel Bounded Distance Decoding algorithm of Micciancio and Nicolosi [31], which outputs the unique lattice point within rsd $\delta = \frac{1}{4}$ of the received word (if it exists). In particular, both algorithms work by recursively (and independently) decoding four words of dimension $N/2$ that are derived from the received word, and then combining the results appropriately. In our case, the runtime is strongly influenced by the sizes of the lists returned by the recursive calls, and so the combinatorial bounds from Section 2 are critical to the runtime analysis.

We need the following easily-verifiable fact regarding the symmetries (automorphisms) of BW_n.

Fact 3.1. *For $N = 2^n$, the linear transformation $\mathcal{T} : \mathbb{C}^N \to \mathbb{C}^N$ given by $\mathcal{T}([u, v]) = \frac{\phi}{2} \cdot [u + v, u - v]$ is a distance-preserving automorphism of BW_n, namely $\mathcal{T}(BW_n) = BW_n$ and $\delta(x) = \delta(\mathcal{T}(x))$ for all $x \in \mathbb{C}^N$.*

Algorithm 1 LISTDECODEBW: List-decoding algorithm for Barnes-Wall lattices.

Input: $r \in \mathbb{C}^N$ (for $N = 2^n$) and $\delta \geq 0$.
Output: The list $L(r, \delta) \subset \mathrm{BW}_n$.

1: **if** $n = 0$ **then**
2: output $L(r, \delta) \subset \mathbb{G}$ by enumeration.
3: **end if**
4: parse $r = [r_0, r_1]$ for $r_0, r_1 \in \mathbb{C}^{N/2}$, and let $r_+ = \frac{\phi}{2}(r_0 + r_1)$ and $r_- = \frac{\phi}{2}(r_0 - r_1)$, so $[r_+, r_-] = \mathcal{T}(r)$.
5: **for all** $j \in \{0, 1, +, -\}$ **do**
6: let $L_j = \text{LISTDECODEBW}(r_j, \delta)$.
7: **end for**
8: for each $(b, s) \in \{0, 1\} \times \{+, -\}$ and each pair $(w_b, w_s) \in L_b \times L_s$, compute the corresponding candidate vector $w = [w_0, w_1] \in \mathrm{BW}_n$ from among

$$[w_0, \tfrac{2}{\phi} w_+ - w_0], \quad [w_0, w_0 - \tfrac{2}{\phi} w_-], \quad [\tfrac{2}{\phi} w_+ - w_1, w_1], \quad [\tfrac{2}{\phi} w_- + w_1, w_1].$$

9: **return** the set L of all the candidate vectors w such that $\delta(w, r) \leq \delta$.

Proof of Theorem 1.4. We need to show that on input $r \in \mathbb{C}^N$ and $\delta \geq 0$, Algorithm 1 runs in time $O(N^2 \cdot \ell(\delta, n)^2)$ and outputs $L = L(r, \delta)$.

We first prove correctness, by induction. The algorithm is clearly correct for $n = 0$; now suppose that $n \geq 1$ and the algorithm is correct for $n-1$. Adopt the notation from Algorithm 1, and let $w = [w_0, w_1] \in L(r, \delta)$ for $w_0, w_1 \in \mathrm{BW}_{n-1}$ be arbitrary. Since $\delta(w, r) \leq \delta$, we have $\delta(r_0, w_0) \leq \delta$ or $\delta(r_1, w_1) \leq \delta$ or both, so $w_0 \in L(r_0, \delta)$ or $w_1 \in L(r_1, \delta)$ or both. A key observation is that the same is true about the corresponding vectors after applying the automorphism $\mathcal{T}$. Namely, letting $[w_+, w_-] = \mathcal{T}(w) \in \mathrm{BW}_n$ for $w_+, w_- \in \mathrm{BW}_{n-1}$, we have $[w_+, w_-] \in L([r_+, r_-], \delta)$ and so $w_+ \in L(r_+, \delta)$ or $w_- \in L(r_-, \delta)$ or both.

By the inductive hypothesis and the above observations, we will have $(w_b, w_s) \in L_b \times L_s$ for at least one choice of $(b, s) \in \{0, 1\} \times \{+, -\}$. The algorithm calculates the vector $w = [w_0, w_1]$ as a candidate, simply by solving for w_0, w_1 using w_b, w_s and the definition of $\mathcal{T}$. Therefore, w will appear in the output list L. And because $L \subseteq L(r, \delta)$, the claim follows.

We now analyze $T(n)$, the number of operations over $\mathbb{C}$ for an input of dimension $N = 2^n$, which is easily seen to satisfy the recurrence

$$T(n) \leq 4T(n-1) + 4 \cdot \ell(\delta, n-1)^2 \cdot O(2^{n-1}) \leq 4^n \cdot T(0) + \sum_{i=1}^{n} 4^i \cdot \ell(\delta, n-i)^2 \cdot O(2^{n-i})$$

$$\leq O(N^2) + O(2^n) \cdot \ell(\delta, n-1)^2 \cdot \sum_{i=1}^{n} 2^i = O(N^2 \cdot \ell(\delta, n-1)^2). \qquad \square$$

Remark 3.2. We note that the above algorithm, like the unique decoder of [31], can be highly parallelized. The parallel time on p processors (counting the number of operations in $\mathbb{C}$) satisfies the recurrence

$$T(n, p) = \begin{cases} T(n) & \text{if } n = 0 \text{ or } p < 4 \\ T(n-1, p/4) + O(N \cdot \ell(\delta, n-1)^2/p + \log N) & \text{otherwise,} \end{cases}$$

where $T(n)$ is the sequential time computed in Theorem 1.4. This is because it takes $O(N \cdot \ell(\delta, n-1)^2/p)$ time per processor to combine the lists in Step 8 of the algorithm, and computing the $\ell(\delta, n-1)^2$ distances in Step 9 requires computing sums of N terms in $\mathbb{C}$, and takes a total of $O(N \cdot \ell(\delta, n-1)^2/p + \log N)$ parallel time. Notice that when $p \geq N^2 \cdot \ell(\delta, n-1)^2$, the algorithm runs in only polylogarithmic $O(\log^2 N)$ parallel time. Note that when the list size $\ell(\delta, n-1) = 1$, our analysis specializes exactly to that of [31].

4 Discussion and Open Problems

An important variant of the list-decoding problem for codes is *local* list decoding. In this model, the algorithm is required to run in time polylogarithmic in the block length, and output succinct representations of all the codewords within a given radius. Defining a meaningful notion of local decoding for lattices (and BW lattices in particular) would require additional constraints, since lattice points do not in general admit succinct representations (since one needs to specify an integer coefficient for each basis vector). While by the Johnson bound we have a $\mathrm{poly}(n)$ list size for rsd up to $1/2 - \mathrm{poly}(1/n)$, defining the right notion of local decoding in this context could be an interesting direction.

Another interesting direction to explore is to find (or construct) more asymptotic families of lattices with nice list-decoding properties. In particular, are there generic operations that when applied to lattices guarantee good list-decoding properties? For codes, list decodability has been shown to behave well under the tensoring and interleaving operations, as demonstrated in [17]. Since at least tensoring is also well-defined for lattices, understanding its effect in the context of list decoding is a natural further direction.

Finally, it would be interesting and potentially useful to consider list decoding for norms other than the Euclidean norm, such as the ℓ_∞ or ℓ_0 norms.

4.1 Acknowledgments

We thank Eli Ben-Sasson, Daniele Micciancio, Madhu Sudan, and Santosh Vempala and for helpful discussions and comments.

References

[1] Dakshi Agrawal and Alexander Vardy. Generalized minimum distance decoding in euclidean space: Performance analysis. *IEEE Transactions on Information Theory*, 46(1):60–83, 2000.

[2] Adi Akavia, Shafi Goldwasser, and Schmuel Safra. Proving hard-core predicates using list decoding. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2003.

[3] E. S. Barnes and G. E. Wall. Some extreme forms defined in terms of abelian groups. *Journal of the Australian Mathematical Society*, 1(01):47–63, 1959.

[4] Bela Bollobás. *Combinatorics*. Cambridge University Press, Cambridge, U.K, 1986.

[5] Henry Cohn and Nadia Heninger. Ideal forms of coppersmith's theorem and guruswami-sudan list decoding. *ITCS*, 2010.

[6] John H. Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, 1998.

[7] Don Coppersmith. Finding small solutions to small degree polynomials. In *CaLC*, pages 20–31, 2001.

[8] Irit Dinur, Elena Grigorescu, Swastik Kopparty, and Madhu Sudan. Decodability of group homomorphisms beyond the Johnson bound. In *STOC*, pages 275–284, 2008.

[9] Ilya Dumer, Gregory A. Kabatiansky, and Cédric Tavernier. List decoding of biorthogonal codes and the Hadamard transform with linear complexity. *IEEE Transactions on Information Theory*, 54(10):4488–4492, 2008.

[10] Ilya Dumer and Rafail E. Krichevskiy. Soft-decision majority decoding of Reed-Muller codes. *IEEE Transactions on Information Theory*, 46(1):258–264, 2000.

[11] Ilya Dumer and Kirill Shabunov. Recursive error correction for general Reed-Muller codes. *Discrete Applied Mathematics*, 154(2):253–269, 2006.

[12] Ilya Dumer and Kirill Shabunov. Soft-decision decoding of Reed-Muller codes: recursive lists. *IEEE Transactions on Information Theory*, 52(3):1260–1266, 2006.

[13] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[14] Rafaël Fourquet and Cédric Tavernier. An improved list decoding algorithm for the second order Reed-Muller codes and its applications. *Des. Codes Cryptography*, 49(1-3):323–340, 2008.

[15] A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier representations via sampling. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2002.

[16] Oded Goldreich and Leonid Levin. A hard-core predicate for all one-way functions. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, May 1989.

[17] Parikshit Gopalan, Venkatesan Guruswami, and Prasad Raghavendra. List decoding tensor products and interleaved codes. In *STOC*, pages 13–22, 2009.

[18] Parikshit Gopalan, Adam R. Klivans, and David Zuckerman. List-decoding Reed-Muller codes over small fields. In *STOC*, pages 265–274, 2008.

[19] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition)*, volume 3282 of *Lecture Notes in Computer Science*. Springer, 2004.

[20] Venkatesan Guruswami. Algorithmic results in list decoding. *Foundations and Trends in Theoretical Computer Science*, 2(2), 2006.

[21] Venkatesan Guruswami. Bridging Shannon and Hamming: List error-correction with optimal rate, 2010.

[22] Venkatesan Guruswami and Atri Rudra. Explicit capacity-achieving list-decodable codes. In *STOC*, pages 1–10, 2006.

[23] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999. Preliminary version appeared in Proc. of FOCS 1998.

[24] Venkatesan Guruswami and Madhu Sudan. Extensions to the Johnson bound. Manuscript, February 2001. Available from `http://people.csail.mit.edu/madhu/papers/johnson.ps`.

[25] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *J. ACM*, 56(4), 2009.

[26] G. David Forney Jr. Coset codes-II: Binary lattices and related codes. *IEEE Transactions on Information Theory*, 34(5):1152–1187, 1988.

[27] G. David Forney Jr. and Alexander Vardy. Generalized minimum-distance decoding of euclidean-space codes and lattices. *IEEE Transactions on Information Theory*, 42(6):1992–2026, 1996.

[28] Tali Kaufman, Shachar Lovett, and Ely Porat. Weight distribution and list-decoding size of Reed-Muller codes. In *ICS*, pages 422–433, 2010.

[29] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SICOMP: SIAM Journal on Computing*, 22, 1993.

[30] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier/North-Holland, Amsterdam, 1981.

[31] Daniele Micciancio and Antonio Nicolosi. Efficient bounded distance decoder for Barnes-Wall lattices. In *IEEE ISIT*, 2008.

[32] D. E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers*, 3:6–12, 1954.

[33] Gabriele Nebe, Eric M. Rains, and Neil J. A. Sloane. The invariants of the Clifford groups. *Des. Codes Cryptography*, 24(1):99–122, 2001.

[34] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *FOCS*, pages 285–294. IEEE Computer Society, 2005.

[35] Ruud Pellikaan and Xin-Wen Wu. List decoding of q-ary Reed-Muller codes. *IEEE Transactions on Information Theory*, 50(4):679–682, 2004.

[36] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4:38–49, 1954.

[37] Amir J. Salomon and Ofer Amrani. Augmented product codes and lattices: Reed-Muller codes and Barnes-Wall lattices. *IEEE Transactions on Information Theory*, 51(11):3918–3930, 2005.

[38] Ba-Zhong Shen, Kenneth K. Tzeng, and Chun Wang. Generalised minimum distance decoding of Reed-Muller codes and Barnes-Wall lattices. *IEEE International Symposium on Information Theory*, page 186, 1995.

[39] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997.

[40] Madhu Sudan. List decoding: algorithms and applications. *SIGACT News*, 31(1):16–27, 2000.

[41] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[42] Amnon Ta-Shma and David Zuckerman. Extractor codes. In *ACM Symposium on Theory of Computing*, pages 193–199, 2001.

[43] Luca Trevisan. Extractors and pseudorandom generators. *J. ACM*, 48(4):860–879, 2001.

[44] John M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de