Low Density Parity Check Lattices

by

Mohammad-Reza Rafsanjani-Sadeghi

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Mathematics and Statistics
Ottawa-Carleton Institute for Mathematics and Statistics
Carleton University
Ottawa, Ontario, Canada
December, 2003

© Copyright 2003

Mohammad-Reza Rafsanjani-Sadeghi

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-89892-X

Our file Notre référence

ISBN: 0-612-89892-X

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To my wife

Mansourh

Abstract

Low density parity check codes can have a remarkable performance under iterative decoding algorithms. This idea is used to construct a class of lattices with relatively high coding gain and low decoding complexity.

The lattice construction is based on the so-called Construction D' . This lattice construction converts a set of parity checks defining a family of nested low density parity check codes into congruences for a lattice called a low density parity check lattice (LDPC lattice, for short). Bounds on the minimum distance and coding gain of the corresponding lattice in terms of the minimum distance of the underlying codes are provided. Given a parity check matrix of a lattice, a practical way of finding the lattice parameters such as cross sections and label group sizes is given.

A new approach based on iterative decoding of lattices is taken. For the decoding of a low density parity check lattice, the Min-Sum algorithm is generalized to group codes over different alphabet sizes and is applied to the Tanner graph of the lattice. An upper bound on the decoding complexity using the generalized Min-Sum algorithm is given. Also for the decoding complexity of LDPC lattices the exact number of computations among lower and upper bounds is derived. The analysis of decoding complexity confirms the use of LDPC codes for the lattice construction. It is shown that the decoding complexity grows linearly with the dimension, exponentially with row weights and is polynomial time in terms of coding gain of the lattice.

The progressive edge growth algorithm is extended to what we call the E-PEG algorithm. This algorithm is used to construct a class of nested regular binary codes to generate the corresponding lattice. A class of 2-level lattices is constructed. The performance of this class is compared with the corresponding 1-level construction and other well known constructions.

In the Name of God the Compassionate and the Merciful

Acknowledgements

I wish to express my sincere appreciation for the valuable efforts of my thesis supervisors Dr. Daniel Panario and Dr. John D. Dixon in supervising my Ph.D. program. My heartfelt thanks to Dr. Daniel Panario for his guidance, suggestions, and his support throughout the course of this research.

I am grateful to Dr. Amir H. Banihashemi for advising me during the initial phase of my Ph.D. program. I am thankful to Dr. Ian Blake for his willingness to act as the external examiner and his valuable comments. Specifically the author appreciates Dr. Brett Stevens, Dr. Halim Yanikomeroglu and Dr. Abas Yongacoglu for acting as examining committee.

The scholarship provided by Amir-Kabir University of Technology, Islamic Republic of Iran is gratefully appreciated.

I am grateful to the school of Mathematics and Statistics of Carleton University for considering me as the Most Outstanding Graduate (Ph.D.) Student and awarding me Marvah Award.

Most importantly, I wish to extend my highest appreciations to my wife Mansoureh, for her support, patience, sacrifice and assistance throughout my studies. I dedicate this thesis to her. I also thankful to my parents for their encouragements and prayers.

I wish the best to my beloved daughter and son, Faezeh and Farzad, whose love gave me strength during difficult periods.

Contents

Abstract	iii
Acknowledgements	v
Contents	vi
List of Abbreviations and Symbols	ix
1 Introduction	1
2 Background on Codes and Lattices	6
2.1 Codes	6
2.1.1 Repetition, cyclic and other simple codes	9
2.1.2 Low Density Parity Check (LDPC) Codes	11
2.2 Lattices	11
2.3 The sphere packing problem	17
2.4 The channel coding problem	19

3	Tanner Graph Construction	27
3.1	Tanner graph construction for binary linear codes	27
3.2	Tanner graph for group codes	30
3.3	Tanner graph construction for lattices	37
4	Constructions of Lattices from Codes	43
4.1	Construction A	44
4.2	Constructions B and C	47
4.3	Construction D	49
4.4	Construction D'	52
4.5	Cycle free lattices	62
5	Decoding Algorithm	65
5.1	The Max-Sum Algorithm	66
5.2	Overview of the Generalized Min-Sum Algorithm	70
5.3	Generalized Min-Sum Algorithm	72
5.3.1	Initialization	73
5.3.2	Hard Decision	74
5.3.3	Symbol node operation	75
5.3.4	Check node operation	77
5.4	Decoding Complexity	83
5.5	Convergence of Generalized Min-Sum algorithm	87

6	Construction and Performance	92
6.1	Extended Progressive Edge-Growth Algorithm	93
6.2	Construction of regular lattices	101
6.3	A class of $(3, 4)$ -2-level lattices: construction and performance	104
6.4	Semi-regular lattices: construction	113
7	Conclusion	119
	Bibliography	122

List of Abbreviations and Symbols

A	alphabet
B	generator matrix
BW_n	Barnes-Wall lattice
C	code
C^*	dual code
d_c	symbol degree of a regular code
d_{c_k}	degree of the k th check node
d_s	check degree of a regular code
d_{s_i}	degree of the i th symbol node
$d_{\min}$	minimum distance of a code
$d_{\min}(\Lambda)$	minimum distance of a lattice Λ
erfc	error function complementary
$\det(\Lambda)$	volume of Λ
E	set of edges
F_q	Galois field
g	girth
g_i	i th alphabet size
g_{ave}	average girth
g^ℓ	girth at level ℓ
G	group (code)
H	parity check matrix
I_{ave}	average iteration
L	label group

LDPC	low density parity check
$M_{ch_k \rightarrow s_i}$	outgoing values from k th check to i th symbol node
$M_{s_i \rightarrow ch_k}$	outgoing values from i th symbol to k th check node
N_D	decoding complexity
$\tilde{N}_D$	decoding complexity per two-dimensions
N_{dec}	decoding complexity pre iteration
P_e	probability of error
P_e^*	normalized probability of error
s_i	i th symbol node
SNR	signal to noise ration
V_c	set of check nodes
V_s	set of symbol nodes
$\gamma(\Lambda)$	coding gain of Λ
γ_{eff}	effective coding gain
δ	center density
Δ	density
ρ	packing radius
σ	noise variance

Chapter 1

Introduction

In this thesis we investigate the construction of high coding gain lattices with low decoding complexity. The lattice version of the Gaussian channel coding problem is to find, for a given value of SNR (signal-to-noise ratio), the n -dimensional lattice for which the error probability is minimized [12]. Shannon's fundamental results [43] guarantee the existence of such lattice codes when the dimension increases. This suggests that we should try constructing high coding gain lattices in larger dimensions. For high SNR this problem coincides with the lattice sphere packing problem [9, 10, 50]. Some high coding gain lattices in large dimensions have been constructed and the gain clearly increases with dimension.

In the 1950's and 1960's mathematicians developed some constructive dense sphere packings based on lattices in spaces of moderate to high dimensions, notably the 2^n -dimensional Barnes-Wall lattices [6], and ultra dense 24-dimension Leech lattice [37]. However the issue is not only the lattice

construction but also the decoding problem.

There are two methods to decode a lattice: the integer programming approach based on geometry of numbers and the trellis approach.

The main approach to the decoding of lattices by integer programming is based on using a reduced basis for the lattice [1, 33, 34]. The complexity of such a decoding algorithm is composed of two parts: (i) computing the reduced basis of the lattice, and (ii) finding the nearest lattice point using this reduced basis. For applications in communications, the lattice is fixed, so the basis reduction is performed just once and the resulting basis is then stored for subsequent uses. The complexity results show that even by performing some exponential-time pre-computations (for computing the reduced basis and covering radii), one cannot decode any sequence of lattices with possible applications ($\gamma \geq 1$) in polynomial time. Thus the integer programming approach is not going to be practically attractive for decoding dense lattices in high dimensions.

The trellis approach is based on representing a lattice by a trellis diagram. The trellis diagram of a lattice which was introduced by Forney [14] in 1967, is based on representing a lattice by a diagram which reflects its underlying group structure. Then to decode a lattice with finite trellis, the Viterbi algorithm is used [22, 15]. This method is the most commonly used in coding and communications. In [16] Forney derived lower bounds on the complexity and he constructed trellis diagrams for some important low-dimensional lattices like Barnes-Wall lattices $D_4, E_4, \Lambda_{16}, \Lambda_{32}$ [12].

However the complexity analysis shows that [7, 48] the decoding com-

plexity grows at least exponentially as a function of the coding gain.

Therefore there are two main issues: (i) the construction of high coding gain lattices, and (ii) finding an appropriate decoding algorithm with low decoding complexity. In this thesis we contribute to both.

For (i) we propose a construction of lattices based on low-density parity check codes. We call these lattices low-density parity check lattices, LDPC lattices for short.

Low density parity check codes (LDPC codes for short) were first discovered by Gallager [25] in 1962 and have been recently rediscovered [38, 40, 41]. These codes have attracted a great deal of attention due to their impressive performance under iterative decoding algorithms. Therefore constructing lattices based on LDPC codes seems to be a promising tool. In Chapter 4 several constructions of lattices from codes have been investigated. Among them Construction D' [5, 8] is shown to be the best choice. In Theorem 4.7 and Corollary 4.8 lower and upper bounds on the minimum distance and coding gain of a lattice in terms of the minimum distance of its underlying codes are provided. Given a parity check matrix of a lattice, in Theorem 4.13 a practical way of finding the cross sections of the lattice is given. The relation between Construction D' and Construction A is shown in Theorem 4.18. Also a generalization of Construction A to group codes is established. This generalization may be used to construct lattices based on LDPC codes over a finite field $\mathbb{F}_q$.

For the decoding problem we will describe a new approach based on iterative decoding of lattices. The proposed decoding algorithm is a generalization

to group codes of the well-known Min-Sum (or Max-Sum) algorithm [21]. In 1981 Tanner [46] generalized Gallager's LDPC codes to codes defined by general bipartite graphs, called Tanner graphs. For any linear binary block code this construction is based on a parity check matrix of the code. The study of the Tanner graph for lattices began in [47]. In [2] Tanner graphs were developed for Abelian group codes. That construction is based on the modular linear constraints imposed on the code symbols by a set of equations for the dual of the code. In Chapter 3 this Tanner graph construction is fully explained and is used for the label code [16, 23] of a lattice to find a graphical representation of the lattice in terms of its linear constraints.

In 1999 Etzion [13] proved that cycle free Tanner graphs cannot support good codes. In Chapter 4 we prove that this is in fact the case for the lattices constructed using Construction D' , i.e., for this type cycle free Tanner graphs cannot support good lattices.

To decode an LDPC lattice constructed using Construction D' , the generalized Min-Sum algorithm is applied to the Tanner graph of the lattice. Lower and upper bounds on the decoding complexity, in terms of coding gain and dimension, are also derived. We have shown that the decoding complexity of LDPC lattices using the generalized Min-Sum algorithm grows linearly with the dimension, exponentially with row weights and polynomial time in terms of coding gain of the lattice. This iterative decoding algorithm is presented in Chapter 5.

The performance of the LDPC lattices that we construct are highly dependent on the performances of the underlying LDPC codes. So the codes

should be constructed properly. There are different methods which can be used to construct LDPC codes. These methods can be divided into two main categories: algebraic methods such as group theoretic and finite geometry constructions [36] and random methods such as permutation constructions [25] and progressive edge growth constructions [31, 30, 32].

In Chapter 6 a (non-algebraic) method based on progressive edge growth (called PEG) is used. In this method the influence of the (short) cycles is reduced as much as possible. In this thesis we extend the PEG algorithm, and call it the E-PEG algorithm, to construct a class of nested regular binary LDPC codes for use in our lattice construction. This algorithm is deterministic. The bounds on the minimum distance and the girth of the Tanner graph in [30, 32, 46] are extended to bounds for the parameters of the underlying nested codes used in the LDPC lattice.

In higher dimensions a class of regular LDPC lattices is introduced, here denoted by L_n^2 , and in lower dimensions another class is introduced, here denoted by SL_n^2 . A program in C^{++} has been written to run simulations on these classes. The performance of these classes is compared with previous results [17]. The results show that these lattices, in higher dimensions, perform very well.

Finally in Chapter 7 we comment on further research.

Chapter 2

Background on Codes and Lattices

2.1 Codes

Given a discrete index set I and a family $\{A_i, i \in I\}$ where A_i is an alphabet of symbols, a *sequence space* W is defined as the set of all sequences $(c_i)_{i \in I}$, with $c_i \in A_i$ for $i \in I$. Hereafter $I = \{1, 2, \dots, n\}$. In fact W is the Cartesian product $W = A_1 \times \dots \times A_n$.

Definition 2.1. *A block code C is any subset of a sequence space $W = A_1 \times \dots \times A_n$. The elements of C are called codewords, or blocks.*

If $C = \{\mathbf{c}_1, \dots, \mathbf{c}_M\}$ has M codewords then C is called an (n, M) -code and is denoted by $C = [n, M]$.

If all code symbols (coordinates) are drawn from the same alphabet

$A = \{a_1, \dots, a_q\}$, then the code C is called a q -ary (n, M) -code over A . In particular when $A = \{0, 1\}$ the code C is called a *binary code*.

The *Hamming distance* between two codewords $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ is the number of coordinates in which the two codewords differ, i.e.

$$d_H = |\{i : a_i \neq b_i, i = 1, \dots, n\}|.$$

The *minimum distance* $d_{\min}$ of a block code C is the minimum Hamming distance between all distinct pairs of codewords in C . If C has M codewords then we denote $C = [n, M, d_{\min}]$.

A *group sequence space* W is a sequence space in which all symbol alphabets are groups G_i . Then $W = G_1 \times \dots \times G_n$ is a direct product group under the componentwise group operation

$$\mathbf{a} * \mathbf{b} \triangleq (a_1 * b_1, \dots, a_n * b_n),$$

where $a_i * b_i$ denotes the binary operation in group G_i .

Definition 2.2. A *group code* C is a subgroup of a group sequence space W .

In other words a group code is a set of codewords (sequences) that forms a group under the componentwise group operation. If all code symbols are drawn from a common group G , then $W = G^n$ and C will be called a *group code over G* .

Let G be a finite Abelian group with binary addition operation $+$ and identity element 0 . For any positive integer m , we use the notation ma to denote the m -fold addition of a with itself, i.e. $ma = \sum_{i=1}^m a$. For any negative integer m , define $ma = (-m)(-a)$, where $-a$ is the inverse of a . A

set $B = \{a_1, \dots, a_k\}$ of nonzero elements of G is said to be *independent* if $\sum_{i=1}^k m_i a_i = 0$ implies $m_i a_i = 0$ for all i . We say the set B is a *generating set* of G if every element $a \in G$ can be written as $a = \sum_{i=1}^k m_i a_i$, for some integers m_i . The set B is called a *basis* if it is an independent generating set of G .

We denote by $\mathbb{F}_q$ the Galois field of order q , where q is a prime or power of a prime.

Definition 2.3. A q -ary code C over the sequence space $\mathbb{F}_q^n$ is a linear code if and only if C forms a vector subspace over $\mathbb{F}_q$.

The dimension of a linear code C is the dimension of the corresponding subspace. A linear q -ary code C with length n and dimension k has q^k codewords, it is called an (n, k) -code and is denoted by $C = [n, k]$ (or $[n, k, d_{\min}]$). Let $\{\mathbf{g}_1, \dots, \mathbf{g}_k\}$ be a basis for a k -dimensional code C , then the matrix B whose rows are $\mathbf{g}_i$'s, is called a *generator matrix* B for the code C .

Definition 2.4. Let $\mathbf{u} = (u_1, \dots, u_n)$ and $\mathbf{v} = (v_1, \dots, v_n)$ be vectors in the vector space V over the field F . The inner product $\langle \mathbf{u}, \mathbf{v} \rangle$ is defined as $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^n u_i v_i$. Two elements $\mathbf{u}$ and $\mathbf{v}$ are said to be orthogonal if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$.

Let S be a subspace of the vector space V . Let S^* be the set of all vectors $\mathbf{v}$ in V such that for all $\mathbf{u} \in S$, $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. Then S^* is a subspace and is called the dual space of S .

Definition 2.5. Let $C = [n, k]$ be a q -ary linear code. Then the dual space of C , which has dimension $n - k$, is called the dual code of C and is denoted by C^* .

We note that the notation M in $C = [n, M]$ refers to a block code which may not be linear.

Let $\{h_1, \dots, h_{n-k}\}$ be a basis for $C^\star$, then the *parity check matrix* H of C is defined by

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{n-k} \end{bmatrix}.$$

It is trivial that a vector $c \in C$ if and only if $\mathbf{H}c^T = \mathbf{0}$. Therefore the linear code C is the null space of the parity check matrix $\mathbf{H}$.

In this thesis the notation k in $C = [n, k, d_{\min}]$ refers to the dimension and the notation r in $C = [n, r, d_{\min}]$ refers to the number of parity checks.

2.1.1 Repetition, cyclic and other simple codes

We start by listing some basic codes that will be used frequently in this thesis.

The $[n, 0, n]$ *zero code* of length n contains just the codeword $(0, \dots, 0)$.

The $[n, n, 1]$ *universal (trivial) code* $\mathbb{F}_q^n$ is the dual of the zero code.

The $[n, 1, n]$ *repetition code* contains all codewords of the form $(a, \dots, a)$, where $a \in \mathbb{F}_q$.

A code is *cyclic* if whenever $(c_1, \dots, c_n)$ is a codeword so is $(c_n, c_1, \dots, c_{n-1})$.

Let $q = p^a$, where p is a prime and let C be an $[n, k, d_{\min}]$ cyclic code over $\mathbb{F}_q$.

Then a codeword $c = (c_1, \dots, c_n) \in C$ can be represented by the polynomial $c(x) = c_1 + c_2x + \dots + c_nx^{n-1}$ in the ring $R_n[x]$ of polynomials modulo $x^n - 1$ with coefficients from $\mathbb{F}_q$. A cycle shift of C is represented by $xc(x)$, and so

a linear cyclic code is represented by an ideal in $R_n[x]$. Because the ring of polynomials over the field $\mathbb{F}_q$ is a principal ideal domain, C can be generated by a single monic polynomial $g(x)$, a *generator polynomial* for the code. Since we are working modulo $x^n - 1$ we may assume $g(x)$ has degree $< n$. It is then known that $g(x)$ is uniquely determined, $g(x)$ divides $x^n - 1$ over $\mathbb{F}_q$ and the dimension of C is $k = n - \deg g(x)$. For the universal code $g(x) = 1$ and for the repetition code $g(x) = 1 + x + \dots + x^{n-1}$.

Let the binary weight $w(i)$ of a non-negative integer i be the number of ones in the binary expansion of i . Let α be a primitive root for the field $\mathbb{F}_q$. For $1 \leq r \leq m - 2$ the r th order punctured Reed-Muller code of length $n = 2^m - 1$ is the cyclic code whose generator polynomial has as roots those α^i such that $1 \leq i \leq 2^m - 2$ and $1 \leq w(i) \leq m - r - 1$.

The r th order Reed-Muller code $R(r, m)$ of length 2^m is formed by appending a zero-check digit ¹ to the r th order punctured Reed-Muller code, for $1 \leq r \leq m - 2$. An r th order Reed-Muller code has parameters

$$R(r, m) = [n = 2^m, k = \sum_{i=0}^r \binom{m}{i}, d = 2^{m-r}].$$

The 0th and m th order Reed-Muller codes are the repetition and universal codes respectively. It is shown [45] that the dual of a Reed-Muller code is a Reed-Muller code and

$$R^*(r, m) = R(m - r - 1, m). \quad (2.1)$$

The Reed-Muller codes may also be defined in terms of Boolean functions [45].

¹Given a codeword $(c_1, \dots, c_M)$, a zero-check digit is a symbol c_{M+1} for which $c_1 + \dots + c_M + c_{M+1} = 0$

2.1.2 Low Density Parity Check (LDPC) Codes

Every linear code can be fully described by its parity check matrix. Low density parity check codes are defined as the follows.

Definition 2.6. *A Low Density Parity Check code (LDPC code) is defined as the null space of a sparse parity check matrix, i.e., a matrix that contains mostly “zeros” and relatively few “ones”.*

LDPC codes were first introduced by Gallager [25, 26]. They were soon forgotten however, due to limited computational resources available in the sixties, which made the corresponding iterative decoding algorithm look impractical. These codes have been rediscovered recently [38, 40, 41] and generalized [36]. It is shown that these codes perform extremely well under iterative decoding algorithms.

A (d_c, d_s) -regular LDPC code is a code whose parity check matrix has d_c “ones” in each column and d_s “ones” in each row. An LDPC code is called *irregular* if it is not regular.

It is shown [25, 38] that the minimum distance of the regular LDPC code and those having d_s “ones” in each column, grows linearly with the dimension n .

2.2 Lattices

Let $\mathbb{R}^m$ be the m -dimensional real space with scalar product $\langle \cdot, \cdot \rangle$ and standard basis $e_1 = (1, 0, \dots, 0), \dots, e_m = (0, \dots, 0, 1)$. The Euclidian length is

$$\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}}.$$

A set Λ of vectors is called *discrete* if there exists a positive number β such that any two vectors of Λ have distance at least β .

Definition 2.7. *An infinite discrete set $\Lambda \subseteq \mathbb{R}^m$ is called a lattice if Λ is a group under addition in $\mathbb{R}^m$. The elements of the lattice Λ are called points or vectors.*

A lattice is a discrete additive subgroup of $\mathbb{R}^m$. The notation $d_{\min} = d_{\min}(\Lambda)$ is used to denote the length of the shortest non-zero vector of the lattice Λ . In fact $d_{\min}(\Lambda)$ refers to the minimum distance between lattice points.

It can be shown that every lattice is generated by the integer combinations of some linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^m$ [42], i.e.,

$$\Lambda = \{k_1 \mathbf{b}_1 + \dots + k_n \mathbf{b}_n : k_1, \dots, k_n \in \mathbb{Z}\}.$$

The set of vectors $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ is called a *lattice basis* and n is called the *dimension* or *rank* of Λ . Since $\mathbf{b}_1, \dots, \mathbf{b}_n$ are linearly independent, $n \leq m$. If $m = n$ the lattice Λ is called *full rank (dimension)*.

By a change of basis, we may transform an n -dimensional lattice Λ to be a subgroup of $\mathbb{R}^n$. Thus, without loss of generality, we may always assume that Λ has full rank.

Definition 2.8. *Any subgroup of a lattice Λ is called a sublattice of Λ .*

Example 2.9. *The lattice $\mathbb{Z}^m$ generated by standard basis of $\mathbb{R}^m$ consists of all the vectors having integer components. It is called the integer lattice.*

Definition 2.10. Let $\mathbf{b}_i = (b_{i1}, \dots, b_{im})$, $i = 1, \dots, n$ be the lattice basis vectors. The generator matrix of Λ is defined by

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}.$$

With the above definition we can write

$$\Lambda = \{\mathbf{v} = \mathbf{x}\mathbf{B} : \mathbf{x} \in \mathbb{Z}^n\}.$$

Example 2.11. For $n \geq 0$, the lattice A_n is defined as

$$A_n = \left\{ (x_0, x_1, \dots, x_n) \in \mathbb{Z}^{n+1} : \sum_{i=0}^n x_i = 0 \right\}.$$

has an $n \times (n+1)$ generator matrix

$$\mathbf{B} = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}.$$

We will refer to A_2 as the Hexagonal lattice.

Definition 2.12. The Gram matrix of Λ is defined as $A = \mathbf{B}\mathbf{B}^T$.

Two lattices are called equivalent or similar if one of them can be obtained from the other by (possibly) a rotation, a reflection or a change of scale.

Two generator matrices B and B' define equivalent lattices if and only if they are related by

$$B' = cUBM$$

where c is a non zero constant, U is a matrix with integer entries and determinant -1 or 1 , and M is a real orthogonal matrix with $MM^T = I$.

Definition 2.13. A lattice is called orthogonal (rectangular) if it has a basis with mutually orthogonal vectors.

Definition 2.14. Given a full rank lattice Λ of $\mathbb{R}^n$, its dual lattice is defined as

$$\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}, \text{ for all } \mathbf{v} \in \Lambda\}.$$

Definition 2.15. The fundamental parallelotope of the lattice Λ is the set of points

$$\{\theta \mathbf{B} : \theta = (\theta_1, \dots, \theta_n) \in \mathbb{R}^n, 0 \leq \theta_i < 1\}.$$

The fundamental parallelotope of a lattice is not unique since it depends on the choice of the lattice basis. In the case of full rank lattices the volume of the fundamental parallelotope is equal to $|\det \mathbf{B}|$, a number independent of the lattice basis. This number is called the determinant of Λ , and is denoted by $\det(\Lambda)$.

For non full rank lattices the volume of the fundamental parallelotope is not well defined since it has zero measure in $\mathbb{R}^m$. We define the determinant of Λ by using its Gram matrix.

Definition 2.16. Let $\mathbf{B}$ be the basis matrix of the lattice Λ , the determinant (or volume) of Λ is defined by the determinant of the Gram matrix of Λ , i.e.,

$$\det(\Lambda) = \det(A)^{\frac{1}{2}} = \det(\mathbf{B}\mathbf{B}^T)^{\frac{1}{2}}.$$

The notation $\text{span}(\Lambda) = \text{span}\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ denotes the real span of the generators $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, i.e.,

$$\text{span}(\Lambda) = \{\alpha_1 \mathbf{b}_1 + \dots + \alpha_n \mathbf{b}_n : (\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n\}.$$

Definition 2.17. *A fundamental region of a lattice Λ is defined as a subset of $\text{span}(\Lambda)$ which, when translated by lattice points, partitions $\text{span}(\Lambda)$ with just one lattice point in each copy.*

The fundamental parallelotope is an example of a fundamental region, see Figure 2.1.

Definition 2.18. *For any point $\mathbf{p} \in \Lambda$ the Voronoi cell $\mathcal{V}(\mathbf{p})$ is defined by the set of those points of $\text{span}(\Lambda)$ that are at least as close to $\mathbf{p}$ as to any other point $\mathbf{q} \in \Lambda$, i.e.,*

$$\mathcal{V}(\mathbf{p}) = \{\mathbf{x} \in \text{span}(\Lambda) : \|\mathbf{x} - \mathbf{p}\| \leq \|\mathbf{x} - \mathbf{q}\|, \text{ for all } \mathbf{q} \in \Lambda\}.$$

The translation of a Voronoi cell by a vector in Λ gives another Voronoi cell. A Voronoi cell is a fundamental region of the lattice Λ . The Voronoi cell for the origin is denoted by $\mathcal{V}$. If $0 \neq \mathbf{u} \in \Lambda$, then $\|\mathbf{x}\| = \|\mathbf{x} - \mathbf{u}\|$ defines a hyperplane half way between $\mathbf{0}$ and $\mathbf{u}$ of equation $\langle \mathbf{x}, \mathbf{u} \rangle = \|\mathbf{u}\|^2/2$. The points $\mathbf{p}$ of the lattice for which the hyperplane between $\mathbf{0}$ and $\mathbf{p}$ contains a facet of $\mathcal{V}$ are called *Voronoi-relevant points* and the hyperplanes are called *Voronoi-relevant hyperplanes*.

If Λ' is a sublattice of Λ such that Λ' is also full rank, then the index $|\Lambda/\Lambda'|$ is finite, and $\det(\Lambda') = |\Lambda/\Lambda'| \det(\Lambda)$. In fact since only one of every $|\Lambda/\Lambda'|$ points in Λ is in Λ' , the volume of fundamental region of Λ' must be $|\Lambda/\Lambda'|$ times larger than that of Λ for its fundamental region to fill $\mathbb{R}^n$.

Figure 2.1: Voronoi cells and fundamental parallelotope

Definition 2.19. *The (fundamental) coding gain (or Hermite parameter) of the lattice Λ is defined as*

$$\gamma(\Lambda) = \frac{d_{\min}^2(\Lambda)}{(\det(\Lambda))^{\frac{2}{n}}}$$

For any n , we have $\gamma(\mathbb{Z}^n) = 1$. An uncoded system may be defined as one that uses a constellation² based on $\mathbb{Z}^n$. Thus the coding gain $\gamma(\Lambda)$ of an arbitrary lattice Λ may be considered to be the gain using constellations based on Λ over an uncoded system using constellations based on $\mathbb{Z}^n$ [17]. Therefore the coding gain of a lattice is a useful measure to assess the performance of the lattice.

Geometrically, $\gamma(\Lambda)$ measures the increase in density of Λ over the base-

²An n -dimensional constellation is any finite set of n -tuples or points in n -dimensional space. In particular a signal constellation is the set of all possible signal points that are taken from a finite subset of points lying within a translate of a lattice.

line lattice $\mathbb{Z}$ or $\mathbb{Z}^n$. The coding gain is invariant to scaling, $\gamma(c\Lambda) = \gamma(\Lambda)$. More generally this quantity is invariant to any scaled orthogonal transformation T , $\gamma(T\Lambda) = \gamma(\Lambda)$, because $\det(T\Lambda) = |\det T| \det(\Lambda)$ and $d_{\min}^2(T\Lambda) = |\det T|^{2/n} d_{\min}^2(\Lambda)$.

For every n , the maximum value of the coding gain is attainable [27] and is called Hermite parameter. It is denoted by γ_n . The maximum value of γ_n is presently known only for $n = 1, 2, \dots, 8$ [12].

2.3 The sphere packing problem

A very old problem in mathematics asks to stack a large number of identical 3-dimensional spheres in a very large box in the most efficient way, i.e., to maximize the number of spheres which can fit inside the box. Such arrangement of spheres is called a *sphere packing*. The spheres do not fill all the space in the box. The percentage of space occupied by the spheres is called the *packing density* and is denoted by Δ .

The above problem can be generalized to higher or lower dimensions.

Definition 2.20. *Let Λ be an n -dimensional (full rank) lattice with minimum distance $d_{\min}$. The packing radius ρ of Λ is defined by*

$$\rho = \frac{d_{\min}}{2}.$$

For an n -dimensional lattice Λ , assume that an n -dimensional sphere of radius ρ is centered at each lattice point. This arrangement of spheres is called a *lattice packing* or briefly a *packing* corresponding to the lattice Λ .

We observe that the sphere of radius ρ is the largest sphere inscribed in the Voronoi cell of the lattice.

The packing density Δ of a lattice packing is

$$\Delta = \frac{\text{volume of one sphere}}{\text{volume of fundamental region}} = \frac{\rho^n V_n}{\det(\Lambda)} \leq 1$$

where V_n is the volume of the n -dimensional sphere of radius 1. It is known that

$$V_n = \frac{\pi^{n/2}}{(n/2)!} = \begin{cases} \frac{\pi^k}{k!}, & n = 2k \\ \frac{2^n \pi^k k!}{n!}, & n = 2k + 1 \end{cases} \quad (2.2)$$

where $(n/2)! = \Gamma(\frac{n}{2} + 1)$ and

$$\Gamma(t) = \int_0^\infty u^{t-1} e^{-u} du$$

is Euler's gamma function.

We also define the *center density* $\delta(\Lambda)$ of the lattice packing as

$$\delta(\Lambda) = \frac{\Delta}{V_n} \quad (2.3)$$

which gives the number of centers per unit volume and enable one to compare lattices in different dimensions. It is easy to see that

$$\gamma(\Lambda) = 4\delta(\Lambda)^{\frac{2}{n}}. \quad (2.4)$$

The density and the center density are independent of the scaling factor of the lattice. The optimal lattice sphere packings are known only up to dimension 8 [12].

Another factor of a sphere packing which will be used in the estimation of the probability of decoding error, is the kissing number. The number of

spheres touching one sphere in a sphere packing is called the *kissing number* and is denoted by τ . For sphere packing the kissing number varies from one sphere to another one but for lattice packing it is the same for all spheres.

Figure 2.2: The optimal 2-dimensional lattice sphere packing

2.4 The channel coding problem

In this section we present the central problem of coding theory [12].

A source of information emits a sequence of messages (either continuous signals or discrete sets of symbols from a finite alphabet). These messages need to be transmitted over a channel impaired by noise in such a way that the probability of correct detection at the receiver side is maximized.

We assume that the channel impairments are only due to white Gaussian noise (Additive White Gaussian Noise, AWGN channel). The channel bandwidth is larger than the bandwidth of the transmitted messages so that no other type of distortion appears. AWGN can also be represented as a

random vector $\mathbf{n} = (n_1, \dots, n_n)$ in the Euclidian space whose components are independent and identically Gaussian distributed (i.i.d.) random variable with zero mean and variance σ^2 . Let the vector $\mathbf{c}$ be transmitted, then the received vector $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{c} + \mathbf{n}.$$

The channel output $\mathbf{c} + \mathbf{n}$ is decoded to $\mathbf{c}$ under *maximum likelihood decoding* if and only if $\mathbf{c} + \mathbf{n}$ belongs to $\mathcal{V}(\mathbf{c})$, the Voronoi cell of $\mathbf{c}$, in the lattice Λ . Thus the probability of correct decoding is given by

$$P_c = \frac{1}{(\sigma\sqrt{2\pi})^n} \int_{\mathcal{V}(\mathbf{c})} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} d\mathbf{x}. \quad (2.5)$$

Using the fact that the Voronoi cells of a lattice are congruent, therefore P_e the probability of error under maximum likelihood decoding of Λ is the probability that a white Gaussian n -tuple $\mathbf{n}$ with variance σ^2 per dimension falls outside the Voronoi cell $\mathcal{V}$, i.e.

$$P_e = 1 - P_c = 1 - \frac{1}{(\sigma\sqrt{2\pi})^n} \int_{\mathcal{V}} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} d\mathbf{x}. \quad (2.6)$$

Let C be a set of code points $\{c_1, \dots, c_M\}$ and let $\mathcal{V}(c_k)$ be the Voronoi cell of c_k . Assuming that all code points are equally likely to be used, then the error probability for this code is

$$P_e = 1 - \frac{1}{M} \sum_{k=1}^M \frac{1}{(\sigma\sqrt{2\pi})^n} \int_{\mathcal{V}(c_k)} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} d\mathbf{x}. \quad (2.7)$$

If all the Voronoi cells are congruent, (i.e. $\mathcal{V}(c_k) = \mathcal{V}$) as for lattices, again we write

$$P_e = 1 - \frac{1}{(\sigma\sqrt{2\pi})^n} \int_{\mathcal{V}} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} d\mathbf{x}. \quad (2.8)$$

For practical applications only a finite number M of points of the lattice within a bounded region containing the origin are used. This signal constellation is called a *lattice code* or *lattice constellation*. The error probability (2.8) is exact only for an infinite lattice. When a finite lattice constellation is considered, the assumption that the Voronoi cells are congruent is no longer valid. In fact the points laying on the edge of the constellation have unbounded Voronoi region while the inner points have lattice Voronoi regions. Since the unbounded regions contain the Voronoi cell of the lattice, Equation (2.8) gives an upper bound to the error probability of a lattice code. The larger the number of points M of the lattice code the tighter is the bound.

The general channel coding problem for the Gaussian channel asks for the code C which minimizes P_e in (2.7) under the signal energy constraint

$$\|c_k\| \leq P = \text{constant, for } k = 1, \dots, M.$$

The lattice version of the Gaussian channel coding problem is to find, for a given value of σ , the n -dimensional lattice of determinant 1 for which (2.8) is minimized [12].

For a lattice with determinant $\det(\Lambda)$ we follow the suggestion of Forney [20] and Tarokh [49], and define the signal-to-noise ratio as the following so that the error probability as a function of SNR remains independent of the lattice scaling factor.

Definition 2.21. *Let S_V denote the n -dimensional sphere of radius r about the origin, having the same volume as $\mathcal{V}$. Define the normalized radius ρ_0 of S_V by the relation $r^2 = n\rho_0^2$. Then the lattice signal-to-noise ratio is defined*

as

$$SNR = \frac{\rho_0^2}{\sigma^2}. \quad (2.9)$$

The volume of $S_{\mathcal{V}}$ is $V_n r^n = \text{vol}(\mathcal{V}) = \det(\Lambda)$, thus

$$SNR = \frac{\det(\Lambda)^{2/n}}{n\sigma^2 V_n^{2/n}}. \quad (2.10)$$

The difficulty in calculations involving the lattice coding problem lies mainly in the complexity of evaluation of the error probability of (2.8). To overcome this difficulty most investigators have replaced (2.8) by simpler expressions.

We assume that all the points are transmitted with equal probability. Due to the geometric uniformity of the lattice [19] it is enough to consider the probability of decoding a point different from $\mathbf{0}$, when $\mathbf{0}$ is transmitted. Let $\mathbf{y}$ be the received vector when $\mathbf{0}$ is transmitted. The components of $\mathbf{y}$ are Gaussian distributed random variable with zero mean and variance σ^2 . Now let $\mathbf{p}_i$ be the Voronoi-relevant points around $\mathbf{0}$. Using the union bound [39] with all the Voronoi-relevant hyperplanes only, we obtain the upper bound [12]

$$\begin{aligned} P_e &\leq P\left(\bigcup_i \{\langle \mathbf{y}, \mathbf{p}_i \rangle \geq \|\mathbf{p}_i\|^2/2\}\right) \\ &\leq \sum_i P\left(\bigcup_i \{\langle \mathbf{y}, \mathbf{p}_i \rangle \geq \|\mathbf{p}_i\|^2/2\}\right) = \sum_i \frac{1}{2} \text{erfc}\left(\frac{\|\mathbf{p}_i\|/2}{\sqrt{2}\sigma}\right) \end{aligned} \quad (2.11)$$

where

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt.$$

In general this bound is very difficult to evaluate, since it is difficult to find all the Voronoi-relevant points.

Well known upper and lower bounds [44] are given by

$$P_e < P(\mathbf{y} \notin S_\rho) = 1 - \Gamma\left(\frac{n}{2}, \frac{\rho^2}{2\sigma^2}\right) / \Gamma\left(\frac{n}{2}\right) \quad (2.12)$$

and

$$P_e > P(\mathbf{y} \notin S_\Pi) = 1 - \Gamma\left(\frac{n}{2}, \frac{r^2}{2\sigma^2}\right) / \Gamma\left(\frac{n}{2}\right) \quad (2.13)$$

where S_ρ and S_Π are n -dimensional spheres of radius ρ , and r , respectively and

$$\Gamma(a, x) = \int_0^x t^{a-1} e^{-t} dt$$

is the incomplete gamma function. By using the definitions of coding gain and SNR these bounds can be written as

$$P_e < P(\mathbf{y} \notin S_\rho) = 1 - \Gamma\left(\frac{n}{2}, \frac{nV_n^{2/n}}{8} \gamma(\Lambda) \text{SNR}\right) / \Gamma\left(\frac{n}{2}\right) \quad (2.14)$$

and

$$P_e > P(\mathbf{y} \notin S_\Pi) = 1 - \Gamma\left(\frac{n}{2}, \frac{n}{2} \text{SNR}\right) / \Gamma\left(\frac{n}{2}\right). \quad (2.15)$$

When SNR is large and $\sigma \rightarrow 0$, a simple estimation of the probability of error in (2.11) is given by [12]

$$P_e \approx \frac{\tau}{2} \text{erfc}\left(\frac{\rho}{\sigma\sqrt{2}}\right). \quad (2.16)$$

This approximation is obtained from (2.11) by considering only the τ largest terms of the sum.

If we use the definition of the packing density and substitute $\gamma(\Lambda)$ from (2.3) and (2.4), we have

$$P_e \approx \frac{\tau}{2} \text{erfc}\left(\sqrt{\frac{n}{8} V_n^{2/n} \gamma(\Lambda) \text{SNR}}\right)$$

$$= \frac{\tau}{2} \operatorname{erfc} \left(\sqrt{\frac{n}{2} \Delta^{2/n} \text{SNR}} \right).$$

These equations simply show the dependency of the probability of error on the lattice parameters, SNR and the lattice packing problem.

The following theorem provides us another lower bound which we use later to compare the performances of lattices.

Theorem 2.22. *If points of an n -dimensional lattice Λ are transmitted over an AWGN channel with variance σ^2 per dimension, the probability of error under maximum likelihood decoding is lower bounded as follows*

$$P_e \geq e^{-z} \left(1 + \frac{z}{1!} + \frac{z^2}{2!} + \cdots + \frac{z^{\frac{n}{2}-1}}{(\frac{n}{2}-1)!} \right)$$

for n even, while for odd n , we have

$$P_e \geq \operatorname{erfc}(z^{\frac{1}{2}}) + e^{-z} \left(\frac{z^{1/2}}{(1/2)!} + \frac{z^{3/2}}{(3/2)!} + \cdots + \frac{z^{n/2-1}}{(n/2-1)!} \right)$$

where

$$z = \frac{V(\Lambda)^{2/n} \Gamma(\frac{n}{2} + 1)^{2/n}}{2\pi\sigma^2}.$$

Proof. Proof is given by Tarokh [49] ■

By using equation (2.2) and the definition of SNR we have $z = (n/2)\text{SNR}$. For an n -dimensional lattice the *normalized probability of (symbol) error* is denoted by P_e^* and is defined as

$$P_e^* = \frac{2}{n} P_e.$$

If we define

$$g_k(x) = e^{-x} \left(1 + \frac{x}{1!} + \cdots + \frac{x^{k-1}}{(k-1)!} \right)$$

then for $n = 2k$ the lower bound of last theorem can be written as

$$P_e^* \geq \frac{g_k(k \times \text{SNR})}{k}. \quad (2.17)$$

We use this bound to compare the performance of lattices.

Another measure of the performance of a lattice is the effective coding gain which measures the performance of a lattice in terms of SNR over an uncoded system $c\mathbb{Z}^n$, where c is a constant.

Definition 2.23. *The effective coding gain of a lattice is the difference, in dB, between the signal-to-noise ratio required to achieve a certain target error with a coded system and the SNR required to obtain the same error with an uncoded system, i.e.,*

$$\gamma_{\text{eff}} = \text{SNR}_{\text{uncoded}} - \text{SNR}_{\text{coded}}.$$

The probability of error for the uncoded system can be computed exactly. Indeed, the Voronoi region $\mathcal{V}$ for the lattice $c\mathbb{Z}^n$ is a hypercube of side c , and therefore equation (2.6) implies that

$$P_c = \frac{1}{(\sigma\sqrt{2\pi})^n} \int_{\mathcal{V}} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}} d\mathbf{x} = \left(\frac{1}{\sigma\sqrt{2\pi}} \int_{-c/2}^{c/2} e^{-\frac{\|x\|^2}{2\sigma^2}} dx \right)^n$$

Hence

$$1 - P_e = \left(1 - \text{erfc} \left(\frac{c}{\sqrt{8}\sigma} \right) \right)^n. \quad (2.18)$$

Let $x = z(n, P_e)$ denote the unique solution of the equation

$$(1 - \text{erfc}(x))^n = 1 - P_e.$$

Then to achieve a probability of P_e in the uncoded case we need $c = \sqrt{8}\sigma z(n, P_e)$. The volume of $c\mathbb{Z}^n$ is $c^n \text{Vol}(\mathbb{Z}^n) = c^n$, and hence the corresponding lattice SNR is given by

$$\text{SNR}_{\text{uncoded}} = \frac{c^2}{\sigma^2} \cdot \frac{1}{nV_n^{2/n}} = \frac{c^2}{\sigma^2} \cdot \frac{(k!)^{1/k}}{2\pi k}$$

assuming that $n = 2k$. Therefore in the uncoded case the desired probability of error is achieved precisely at lattice SNR of

$$\text{SNR}_{\text{uncoded}} = z(n, P_e)^2 \cdot \frac{4(k!)^{1/k}}{\pi k}. \quad (2.19)$$

The $\text{SNR}_{\text{uncoded}}$ (in dB) at error probability of $P_e = 10^{-6}$ for several dimensions are provided in the following table:

n	$z(n, P_e)$	$\text{SNR}_{\text{uncoded}}$
64	3.998	9.104
256	4.163	9.209
460	4.231	9.300
1000	4.308	9.428
2000	4.320	9.430

In larger dimensions we compute the uncoded SNR in Equation 2.19 by means of the following expression

$$(k!)^{\frac{1}{k}} = \exp \left(\frac{1}{k} \sum_{i=1}^k \log(i) \right). \quad (2.20)$$

Chapter 3

Tanner Graph Construction

In order to use two-way algorithms to decode a lattice we need to know how to construct a Tanner graph for a lattice. In this chapter we survey this matter, first by presenting the Tanner graph for binary codes. Then we state the Tanner graph construction for group codes. In the last section, we present the concept of a trellis diagram for a lattice and its label code, and apply it to construct the Tanner graph for lattices.

3.1 Tanner graph construction for binary linear codes

A Tanner graph [46] is a graphical representation of a code corresponding to a set of parity checks that specify the code. Each bit (codeword coordinate) is represented by a symbol node and each parity check is represented by a

parity-check node. Each parity-check node is connected by an edge to the symbols that it checks. A Tanner graph is thus a bipartite graph in which symbol nodes are connected only to parity-check nodes and vice versa.

Let $H = [h_{ij}]$ be a parity check matrix for a linear code $C = [n, k]$. Formally a Tanner graph is denoted as (V, E) where the set V of vertices (nodes) is partitioned into two sets $V = V_s \cup V_c$ where $V_s = \{s_1, \dots, s_n\}$, $V_c = \{c_1, \dots, c_m\}$ ¹ (called the set of symbol nodes and check nodes, respectively) and E is a set of edges such that $E \subseteq V_c \times V_s$. The n symbol nodes of the graph correspond to the codeword coordinates $(x_1, \dots, x_n)$ and the $m = n - k$ (or possibly more) check nodes correspond to the m check equations. An edge (c_i, s_j) joins symbol node x_j to check node i if and only if x_j is involved the i -th check equation, i.e., if and only if $h_{ij} \neq 0$.

The i -th check equation is regarded as a “local constraint” stating the condition that $\sum_{j=1}^n h_{ij}x_j = 0$. A given configuration $(x_1, \dots, x_n)$ is a valid codeword if and only if all local constraints are satisfied.

Example 3.1. *The 3×5 parity-check matrix*

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

defines a binary code $C = [5, 2]$. The set of length-5 codewords $x =$

¹To avoid any confusion with codeword notation $c = (c_1, \dots, c_n)$, sometimes (especially in Chapter 5) we use $\{ch_1, \dots, ch_m\}$ to denote the check nodes set.

$(x_1, \dots, x_5)$ satisfy the three parity checks

$$x_1 + x_2 + x_3 = 0$$

$$x_4 + x_5 = 0$$

$$x_3 + x_5 = 0$$

The Tanner graph corresponding to this code is illustrated in Figure 3.1. Symbol nodes are represented by black circles and parity-check nodes (check-

Figure 3.1

nodes) are represented by white circles.

The following theorem provides us with a bound on the minimum distance of a regular binary code in term of its girth, the length of the shortest cycle in its Tanner graph, and its column degree.

Theorem 3.2. *Let $C = [n, k, d_{\min}]$ be a (d_c, d_s) -regular linear binary code with $d_s \geq 3$ and girth $g > 4$, then its minimum distance satisfies*

$$d_{\min} \geq 1 + \frac{d_s[(d_s - 1)^{\lfloor (g-2)/4 \rfloor} - 1]}{d_s - 2}. \quad (3.1)$$

Furthermore if $g/2$ is even, the lower bound on $d_{\min}$ can be made tighter

$$d_{\min} \geq 1 + \frac{d_s[(d_s - 1)^{\lfloor (g-2)/4 \rfloor} - 1]}{d_s - 2} + (d_s - 1)^{\lfloor (g-2)/4 \rfloor}. \quad (3.2)$$

If $d_s = 2$ then $d_{\min} = g/2$.

Proof. The proof is given by Tanner in [46]. ■

3.2 Tanner graph for group codes

Let A be a finite Abelian group with a binary addition operation $+$ and identity element 0 . Given the group A , a *character* of A is a homomorphism Ψ from A into the additive circle group $\mathbb{R}_{[0,1)}$ of real numbers modulo 1. The circle group is sometimes denoted as $\mathbb{R}/\mathbb{Z}$. It is also isomorphic to the group of complex numbers with unit magnitude under multiplication. The *character group* $\hat{A}$ of A is the Abelian group of all homomorphisms $\Psi : A \rightarrow \mathbb{R}_{[0,1)}$ under the operation defined by $(\Psi_1 \circ \Psi_2)(a) = \Psi_1(a) + \Psi_2(a), \forall a \in A$. The identity element of $\hat{A}$, also referred to as the neutral element or the zero element, is denoted by $\Psi^{(0)}$, and maps all the elements of A into 0. When A is finite, then $\hat{A}$ is isomorphic to A (see Lemma 3.3 below). The pairing $\langle, \rangle$ is defined as a function from $\hat{A} \times A$ into $\mathbb{R}_{[0,1)}$ given by $\langle \Psi, a \rangle = \Psi(a)$ for all $(\Psi, a) \in \hat{A} \times A$. The elements $\Psi \in \hat{A}$ and $a \in A$ are called *orthogonal* if $\langle \Psi, a \rangle = 0$. For each subgroup H of A we define H^* as the set of elements $h^* \in \hat{A}$ such that $\langle h^*, h \rangle = 0, \forall h \in H$. Then H^* is a subgroup of $\hat{A}$ called the *dual group* of H .

It is known that every finite Abelian group is a direct product of cyclic groups. We use $\mathbb{Z}_g = \{0, 1, \dots, g-1\}$ to denote the cyclic group of order g under addition mod g . For a direct product of cyclic groups

$$\mathbf{G} = \mathbb{Z}_{g_1} \times \mathbb{Z}_{g_2} \times \dots \times \mathbb{Z}_{g_n}$$

we have

$$\widehat{\mathbf{G}} = \widehat{Z}_{g_1} \times \widehat{Z}_{g_2} \times \cdots \times \widehat{Z}_{g_n},$$

which is also isomorphic to $\mathbf{G}$. Thus each character Ψ of $\mathbf{G}$ can be written uniquely as $(\Psi_1, \dots, \Psi_n)$ for some $\Psi_i \in \widehat{Z}_{g_i}, i = 1, \dots, n$, where

$$\langle \Psi, \mathbf{c} \rangle = \langle \Psi_1, c_1 \rangle + \langle \Psi_2, c_2 \rangle + \cdots + \langle \Psi_n, c_n \rangle$$

for $\mathbf{c} = (c_1, \dots, c_n) \in \mathbf{G}$, where additions are performed in $R_{[0,1)}$.

Now let L be a subgroup of G and

$$\mathbf{C}^* = \{\Psi_1, \dots, \Psi_r\} \tag{3.3}$$

be a generating set for L^* . Then the set of orthogonality constraints

$$\langle \Psi_i, \mathbf{c} \rangle = \langle \Psi_{i1}, c_1 \rangle + \cdots + \langle \Psi_{in}, c_n \rangle = 0, \quad i = 1, \dots, r, \tag{3.4}$$

characterizes the codewords $\mathbf{c} \in \mathbf{L}$. This construction can be used to construct a Tanner graph. In fact each constraint is associated to a check node and each symbol c_j of $\mathbf{L}$ is represented by a symbol node, and is joined by an edge to the check node corresponding to Ψ_i if the j th element of Ψ_i is not the identity, i.e. $\Psi_{ij} \neq \Psi^{(0)}$.

Although this way they have a simple presentation, the check equations are not convenient for computational purposes. In the following we use an isomorphism Φ between an Abelian finite group $\mathbf{G}$ and its character group $\widehat{\mathbf{G}}$ to introduce new definitions of inner product and orthogonality for which the pairing between $\mathbf{G}$ and $\widehat{\mathbf{G}}$ is reduced to an inner product $(\cdot, \cdot)_{\Phi}$ between the elements of $\mathbf{G}$ [2].

Lemma 3.3. *Let A be an Abelian group with basis $\{a_1, \dots, a_r\}$ then $A \cong \widehat{A}$.*

Proof. Suppose a_i has order k_i . For any $\Psi \in \hat{A}$ we have

$$0 = \Psi(0) = \Psi(k_i a_i) = k_i \Psi(a_i), \quad i = 1, \dots, r,$$

with $\Psi(a_i) \in \mathbb{R}_{[0,1]}$. Therefore the order of $\Psi(a_i)$ divides k_i and hence must take one of the k_i values

$$0, \frac{1}{k_i}, \dots, \frac{k_i - 1}{k_i}.$$

It is easily verified that for $i = 1, \dots, r$ there is a unique character $\Psi_i \in \hat{A}$ (of order k_i) such that

$$\Psi_i(a_i) = \frac{1}{k_i}, \quad \Psi_i(a_j) = 0 \quad \forall j \neq i.$$

Now consider any $\Psi \in \hat{A}$. As we saw above, $\Psi(a_i) = \alpha_i/k_i$ where $\alpha_i = 0, 1, \dots, k_i - 1$. Then Ψ can be written uniquely as a linear combination of $\Psi_1, \dots, \Psi_r$, namely

$$\Psi = \alpha_1 \Psi_1 + \dots + \alpha_r \Psi_r.$$

Therefore $\{\Psi_1, \dots, \Psi_r\}$ is a basis for $\hat{A}$. Since Ψ_i and a_i both have order k_i , this shows that $A \cong \hat{A}$. ■

We note that the isomorphism between A and $\hat{A}$ is not unique.

For $A = Z_g$, we have $\hat{A} \cong Z_g$. Let us define a character $\Psi^{(1)}$ by $\Psi^{(1)}(1) = 1/g$. We identify the elements of $\hat{A}$ as multiples of $\Psi^{(1)}$, i.e.,

$$\Psi^{(i)} = i\Psi^{(1)} = \Psi^{(1)} + \dots + \Psi^{(1)} \quad \text{mod } 1, \quad i = 0, \dots, g-1.$$

An element $\Psi^{(i)}$ generates $\hat{Z}_g$ if and only if $\gcd(g, i) = 1$. Then any isomorphism $\Phi : Z_g \longrightarrow \hat{Z}_g$ has the form $a \longmapsto a\Psi^{(i)} = ai\Psi^{(1)}$ for some i with $\gcd(g, i) = 1$.

Now for any $c, a \in Z_g$ we have

$$\begin{aligned}\langle \Phi(c), a \rangle &= \langle c\Phi(1), a \rangle = ca\langle \Psi^{(i)}, 1 \rangle = ca\Psi^{(i)}(1) \\ &= cai\Psi^{(1)}(1) = \frac{cai}{g} \pmod{1}.\end{aligned}$$

Therefore we have proved the following lemma.

Lemma 3.4. *Let Φ be an isomorphism between Z_g and $\widehat{Z}_g$, for which $1 \longrightarrow \Psi^{(i)}$, for some i relatively prime to g . Then for every $c, a \in Z_g$*

$$\langle \Phi(c), a \rangle = \frac{cai}{g}.$$

Using the above lemmas we introduce new definitions for inner product and duality.

Definition 3.5. *For every pair $(c, a) \in Z_g \times Z_g$ and any isomorphism Φ between Z_g and $\widehat{Z}_g$ the inner product of c and a with respect to Φ is defined by*

$$(c, a)_\Phi = \langle \Phi(c), a \rangle.$$

By Lemma 3.4

$$(c, a)_\Phi = \frac{cai}{g} \pmod{1},$$

for some i relatively prime to g . Two elements $c, a \in Z_g$ are called orthogonal with respect to Φ if

$$(c, a)_\Phi = 0.$$

Let L be a subgroup of Z_g , we define $L_\Phi^* := \Phi^{-1}(L^*)$. In fact

$$L_\Phi^* = \{c \in Z_g : (c, a)_\Phi = 0, \quad \forall a \in L\}.$$

Then L_{Φ}^* is a subgroup of Z_g and is called dual of L with respect to Φ .

Let $\mathbf{G} = Z_{g_1} \times \cdots \times Z_{g_n}$, and $\Phi = (\Phi_1, \dots, \Phi_n)$ for which Φ_i is an isomorphism between Z_{g_i} and $\widehat{Z}_{g_i}$. For every $\mathbf{c}, \mathbf{a} \in \mathbf{G}$ based on Lemma 3.4 define the inner product of $\mathbf{c}$ and $\mathbf{a}$ with respect to Φ by:

$$\begin{aligned} (\mathbf{c}, \mathbf{a})_{\Phi} &= \langle \Phi(\mathbf{a}), \mathbf{c} \rangle \\ &= \langle \Phi_1(a_1), c_1 \rangle + \langle \Phi_2(a_2), c_2 \rangle + \cdots + \langle \Phi_n(a_n), c_n \rangle \\ &= \frac{c_1 a_1 i_1}{g_1} + \frac{c_2 a_2 i_2}{g_2} + \cdots + \frac{c_n a_n i_n}{g_n} \pmod{1}, \end{aligned} \quad (3.5)$$

for some integers $i_j \in \{1, \dots, g_j - 1\}$, $j = 1, \dots, n$, relatively prime to g_j , $j = 1, \dots, n$, respectively. We use the notation $(\mathbf{c}, \mathbf{a})_{(i_1, \dots, i_n)}$ to denote the inner product of $\mathbf{c}$ and $\mathbf{a}$ with respect to Φ .

Lemma 3.6. *The definition of L_{Φ}^* is independent of Φ .*

Proof. Let $\gcd(g, i) = 1$ then $cai/g \equiv 0 \pmod{1}$, if and only if, $cai \equiv 0 \pmod{g}$ and this is equivalent to $ca \equiv 0 \pmod{g}$. ■

Theorem 3.7. *Let $\mathbf{G} = Z_{g_1} \times \cdots \times Z_{g_n}$, and $\Phi = (\Phi_1, \dots, \Phi_n)$ for which Φ_i is an isomorphism between Z_{g_i} and $\widehat{Z}_{g_i}$. Assume that $\mathbf{L}$ is a subgroup of $\mathbf{G}$ and $\mathbf{C}^*$ is a generating set for $\mathbf{L}^*$ as in the Equation (3.3). Let $\mathbf{C}_{\Phi}^* := \Phi^{-1}(\mathbf{C}^*)$. Then the Tanner graph for $\mathbf{L}$ based on the generating set $\mathbf{C}^*$ and the pairing $\langle \cdot, \cdot \rangle$ is the same as the Tanner graph for $\mathbf{L}$ based on the generating set $\mathbf{C}_{\Phi}^*$.*

Proof. Since Φ is an isomorphism then $|\mathbf{C}_{\Phi}^*| = |\mathbf{C}^*|$, so both Tanner graphs have the same number of check nodes. Let the elements of $\mathbf{C}_{\Phi}^*$ be denoted by $\mathbf{c}_i^* = \Phi^{-1}(\Psi_i)$, $i = 1, \dots, r$. Since $\langle \Psi_i, \mathbf{c} \rangle = \langle \Phi(\mathbf{c}_i^*), \mathbf{c} \rangle = (\mathbf{c}_i^*, \mathbf{c})_{\Phi}$, then the

orthogonality constraints (3.4) are equivalent to

$$(\mathbf{c}_i^*, \mathbf{c}) = 0, \quad i = 1, \dots, r. \quad (3.6)$$

In particular $\Psi_{ij} = \Psi^{(0)}$ is equivalent to $c_{ij}^* = 0$. ■

For any subgroup $\mathbf{L}$ of $\mathbf{G}$, $\mathbf{L}_\Phi^*$ is a subgroup of $\mathbf{G}$ and is called *dual* of L with respect to Φ . From now on we choose the inner product $(\cdot, \cdot)_{(1, \dots, 1)}$

$$(\cdot, \cdot) = (\cdot, \cdot)_{(1, \dots, 1)} : G \times G \longrightarrow R_{[0,1]}$$

and use $\mathbf{L}^* = \mathbf{L}_{(1, \dots, 1)}^* \leq G$ to construct a Tanner graph for $\mathbf{L}$.

For the inner product we have

$$\langle \mathbf{c}, \mathbf{c}' \rangle = \sum_{i=1}^n \frac{c_i c'_i}{g_i} \in R_{[0,1]}, \quad \forall \mathbf{c}, \mathbf{c}' \in Z_{g_1} \times Z_{g_2} \times \dots \times Z_{g_n},$$

where the multiplications and divisions are performed in the field of real numbers. For any generating set $\mathcal{C}^* = \{\mathbf{c}_1^*, \dots, \mathbf{c}_r^*\}$ with r generators for $\mathbf{L}^*$, we have the following r check equations which fully describe codewords $\mathbf{c} = (c_1, \dots, c_n) \in L$:

$$\sum_{i=1}^n \frac{c_{ki}^* c_i}{g_i} = 0 \pmod{1}, \quad k = 1, \dots, r,$$

or equivalently;

$$\sum_{i=1}^n \frac{c_{ki}^* c_i}{g_i} \in Z, \quad k = 1, \dots, r,$$

where $\mathcal{C}^* = \{\mathbf{c}_1^*, \dots, \mathbf{c}_r^*\}$ is a generating set for $\mathbf{L}^*$ with r generators.

Example 3.8. Let $\mathbf{G} = Z_2 \times Z_6 \times Z_6 \times Z_2$, and

$$\mathbf{L} = \left\{ \begin{array}{cccccc} 0000, & 0031, & 0220, & 0251, & 0440, & 0411, \\ 1300, & 1331, & 1520, & 1551, & 1140, & 1111 \end{array} \right\}.$$

We then have

$$\mathbf{L}^* = \left\{ \begin{array}{l} 0000, 0240, 0420, 1511, 1151, 0031, \\ 1300, 1331, 0451, 1540, 1120, 0211 \end{array} \right\}.$$

It can be seen that $\{1120, 0031\}$ is a generating set for $\mathbf{L}^*$. The corresponding check equations are

$$\frac{c_1}{2} + \frac{c_2}{6} + \frac{2c_3}{6} \in \mathbb{Z}, \quad \frac{3c_3}{6} + \frac{c_4}{2} \in \mathbb{Z}. \quad (3.7)$$

These equations can be written as:

$$\begin{aligned} 3c_1 + c_2 + 2c_3 &\equiv 0 \pmod{6} \\ c_3 + c_4 &\equiv 0 \pmod{2}. \end{aligned}$$

The corresponding Tanner graph is illustrated in Figure 3.2.

Figure 3.2

3.3 Tanner graph construction for lattices

The *trellis of an n -dimensional lattice* [16, 22] Λ is an efficient graph representation of Λ in terms of the cosets of Λ' , an n -dimensional orthogonal sublattice of Λ , in Λ . Therefore it is natural to consider the construction of the Tanner graph for Λ based on Λ/Λ' . In this section we describe the trellis diagram of lattices and their label code, and then apply them to the hexagonal lattice A_2 defined in Example 2.11. Consider a nested sequence of vector spaces:

$$\{0\} \subseteq V_0 \subseteq V_1 \subseteq \dots \subseteq V_n = \mathbb{R}^n,$$

where $\dim(V_i) = i$. Let W_i be the orthogonal complement of V_{i-1} in V_i , that is

$$V_i = V_{i-1} \oplus W_i, \quad 1 \leq i \leq n.$$

For any subspace V of $\mathbb{R}^n$, the *cross section* Λ_V of Λ is defined as the set of lattice points that lie in V :

$$\Lambda_V \triangleq \Lambda \cap V.$$

It can be seen that the cross sections Λ_{V_i} and Λ_{W_i} have a lattice structure.

We use the notation P_V for the projection operator onto the vector space V . The *state space* $\Sigma_i(\Lambda)$ at level i , $0 \leq i \leq n$, is defined as the quotient group $P_{V_i}(\Lambda)/\Lambda_{V_i}$ and the *label group* $G_i(\Lambda)$ of Λ at trellis section i , $1 \leq i \leq n$ is defined as the quotient group $P_{W_i}(\Lambda)/\Lambda_{W_i}$, that is,

$$P_{V_i}(\Lambda)/\Lambda_{V_i} = \{P_{V_i}(\mathbf{x}) + \Lambda_{V_i} : \mathbf{x} \in \Lambda\}$$

$$P_{W_i}(\Lambda)/\Lambda_{W_i} = \{P_{W_i}(\mathbf{x}) + \Lambda_{W_i} : \mathbf{x} \in \Lambda\}.$$

The *trellis* T of the lattice Λ is defined as a directed graph whose nodes (states) at each level i , $0 \leq i \leq n$, are the elements of $\Sigma_i(\Lambda)$. Each edge between level $i - 1$ and i (in trellis section i) is labeled by the elements of $G_i(\Lambda)$. The set of all paths through the trellis diagram corresponds to Λ , i.e., for each lattice point $x \in \Lambda$, there is a path through T which starts from the initial state $\Sigma_0(\Lambda)$ and ends at the final state $\Sigma_n(\Lambda)$.

For every lattice point $\mathbf{x} \in \Lambda$, the *state sequence* $\sigma(\mathbf{x})$ and *label sequence* $\mathbf{g}(\mathbf{x})$ are defined:

$$\sigma(\mathbf{x}) = (\sigma_0(\mathbf{x}), \dots, \sigma_n(\mathbf{x})), \quad \mathbf{g}(\mathbf{x}) = (g_0(\mathbf{x}), \dots, g_n(\mathbf{x})),$$

where

$$\sigma_i(\mathbf{x}) = P_{V_i}(\mathbf{x}) + \Lambda_{V_i}, \quad g_i(\mathbf{x}) = P_{W_i}(\mathbf{x}) + \Lambda_{W_i}.$$

It is clear that $\sigma_0(\mathbf{x}) = \Sigma_0(\Lambda)$, $\sigma_n(\mathbf{x}) = \Sigma_n(\Lambda)$. For every $\mathbf{x} \in \Lambda$, the path passes through the state sequence $\sigma(\mathbf{x})$ and label sequence $\mathbf{g}(\mathbf{x})$.

The set of all possible state sequences

$$\sigma(\Lambda) = \{\sigma(\mathbf{x}) : \mathbf{x} \in \Lambda\}$$

is called the *state code*. The set of all possible label sequences

$$\mathbf{L} = \mathbf{g}(\Lambda) = \{\mathbf{g}(\mathbf{x}) : \mathbf{x} \in \Lambda\}$$

is called the *label code*.

It can be shown that $\sigma(\Lambda)$ and $\mathbf{g}(\Lambda)$ are isomorphic [23]. Their cardinality is denoted by $N(\Lambda)$, which is equal to the number of distinct paths in the trellis diagram of Λ .

In the above construction, the ordered system of subspaces $S = \{W_i\}_{i=1}^n$ corresponding to the sequence $\{V_i\}_{i=1}^n$, is called (trellis) *coordinate system* of Λ .

The lattice Λ is said to have a *finite trellis* if there exists a trellis diagram for Λ with a finite number of states (or edges). It can be shown [3] that Λ has a finite trellis if and only if it has an n -dimensional orthogonal sublattice

$$\Lambda' = \Lambda_{W_1} \oplus \dots \oplus \Lambda_{W_n}.$$

In this case

$$N(\Lambda) = \frac{\det \Lambda'}{\det \Lambda}.$$

Therefore the number of paths in the trellis T of Λ , which is equal to $|L|$, is the number of cosets of Λ' in Λ . Hence the trellis of Λ is an efficient way of representing Λ as the union of cosets of Λ' in Λ .

Suppose Λ has an n -dimensional orthogonal sublattice Λ' and Λ' has a set of basis vectors along the orthogonal subspaces $S = \{W_i\}_{i=1}^n$. To construct the Tanner graph for a lattice Λ we apply the Tanner graph construction for group codes to the label code L of Λ , which is an Abelian group code over the alphabet sequence space

$$G = G_1 \times \dots \times G_n.$$

Let $|G_i| = g_i$ and let v_i be the shortest vector of Λ_{W_i} , i.e., $\Lambda_{W_i} = Zv_i$, then each element of G_i can be written of the form

$$\Lambda_{W_i} + j \det(P_{W_i}) \frac{v_i}{\|v_i\|}$$

for $j = 0, 1, \dots, g_i - 1$. Then the map

$$\Phi : G_i \longrightarrow Z_{g_i}$$

defined by

$$\Lambda_{W_i} + \det P_{W_i}(\Lambda) \frac{v_i}{\|v_i\|} \mapsto 1$$

is an isomorphism between G_i and Z_{g_i} . Based on this isomorphism, we assume

$$\mathbf{G} = Z_{g_1} \times \cdots \times Z_{g_n}.$$

In general the elements of each label group G_i are of the form $(Z + a_j)v_i$ where

$$a_j = j \frac{\det(P_{W_i}(\Lambda))}{\det(\Lambda_{W_i})}, \text{ for } j = 0, 1, \dots, g_i - 1.$$

Example 3.9. Consider the hexagonal lattice A_2 [12] in Figure 3.4 that is generated by

$$B = \begin{bmatrix} 1 & 0 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}.$$

Here $V_0 = \{0\}$, $V_1 = \{(x, 0) : x \in \mathbb{R}\}$, and $V_2 = \mathbb{R}^2$. Therefore the trellis coordinate system is $S = \{W_1, W_2\}$ where

$$W_1 = V_1 = \langle (1, 0) \rangle = \{(x, 0) : x \in \mathbb{R}\}, \quad W_2 = \langle (0, 1) \rangle = \{(0, y) : y \in \mathbb{R}\}.$$

The projections of the lattice points onto W_i and V_i are

$$P_{W_1}(\Lambda) = P_{V_1}(\Lambda) = \frac{1}{2}\mathbb{Z}, \quad P_{W_2}(\Lambda) = \frac{\sqrt{3}}{2}\mathbb{Z}.$$

It is clear that $P_{V_0} = \{0\}$ and $P_{V_2} = \Lambda$.

For the cross sections we have

$$\Lambda_{W_1} = \mathbb{Z}, \quad \Lambda_{W_2} = \sqrt{3}\mathbb{Z}.$$

The label groups are:

$$G_1(\Lambda) = \frac{\frac{1}{2}\mathbb{Z}}{\mathbb{Z}} = \{\mathbb{Z}, \frac{1}{2} + \mathbb{Z}\} \cong \mathbb{Z}_2,$$

$$G_2(\Lambda) = \frac{\frac{\sqrt{3}}{2}\mathbb{Z}}{\sqrt{3}\mathbb{Z}} = \{\sqrt{3}\mathbb{Z}, \frac{\sqrt{3}}{2} + \sqrt{3}\mathbb{Z}\} \cong \mathbb{Z}_2,$$

and for the label sequences we have

$$g_1(\mathbf{x}) = P_{W_1}(\mathbf{x}) + \Lambda_{W_1} = \frac{1}{2}n + \mathbb{Z}, \quad n \in \mathbb{Z},$$

$$g_2(\mathbf{x}) = P_{W_2}(\mathbf{x}) + \Lambda_{W_2} = \frac{\sqrt{3}}{2}n + \sqrt{3}\mathbb{Z}, \quad n \in \mathbb{Z}.$$

For n even, $g_1(\mathbf{x}) = \mathbb{Z} \longleftrightarrow 0$ and $g_2(\mathbf{x}) = \sqrt{3}\mathbb{Z} \longleftrightarrow 0$ and for n odd, $g_1(\mathbf{x}) = \frac{1}{2} + \mathbb{Z} \longleftrightarrow 1$ and $g_2(\mathbf{x}) = \frac{\sqrt{3}}{2} + \sqrt{3}\mathbb{Z} \longleftrightarrow 1$. Therefore the label code $\mathbf{L}$ of the lattice is $\mathbf{L} = \{00, 11\}$. The trellis diagram is illustrated in the following figure.

Figure 3.3: Trellis diagram of the Hexagonal lattice A_2

In Figure 3.4 the orthogonal sublattice $\mathbb{Z} \oplus \sqrt{3}\mathbb{Z}$, denoted by black dots, corresponds to the top path. The other coset of this sublattice corresponds to the bottom path. The label code $\mathbf{L}$ of this lattice is $\mathbf{L} = \{00, 11\}$ which is a repetition code and has check equation

$$x_1 + x_2 = 0 \pmod{2}.$$

This results in the Tanner graph in Figure 3.5.

Figure 3.4: The Hexagonal lattice A_2

Figure 3.5: The Tanner graph of the Hexagonal lattice A_2

Chapter 4

Constructions of Lattices from Codes

In this chapter first we investigate Constructions A, B, C, D and D' of lattices from codes and compare them. These constructions are given based on binary codes. Then we generalize Construction A to group codes. This generalization can be used to construct lattices using group codes, e.g., LDPC codes over a finite field $\mathbb{F}_q$ can be used to generate lattices. Also in Example 5.2 we use this construction to produce a lattice based on the group code in Example 3.8. In these investigations we show that among all of these constructions, Construction D and D' can have higher coding gain than the others. In particular when using LDPC codes identified by a set of parity check vectors, as the underlying codes, Construction D' seems to be the best choice.

Construction D' uses a collection of codes C_ℓ with a condition on their

minimum distances, i.e., $d_{\min}^\ell \geq \alpha 4^\ell$. Here we prove a theorem which allow us to generalize this construction to any collection of nested linear binary codes without any condition on their minimum distances. In this theorem we derive lower and upper bounds on the squared minimum distance of the corresponding lattice in term of the minimum distances of each code involved in the construction. This shows that the coding gain of the corresponding lattice can improve if $d_{\min}^\ell \geq 4^{\ell+1}$.

In Theorem 4.13 we provide a practical way of finding the cross sections of the lattice given the parity check matrix of a lattice based on Construction D' . In particular we prove that all cross sections have the same determinant if the first code C_0 has minimum distance at least 2.

Cycle free graphs cannot support good codes [13]. In the last section we prove that good lattices, constructed using Construction D' , cannot be supported by cycle free Tanner graphs.

4.1 Construction A

Let $C = [n, M, d_{\min}]$ be a (not necessarily linear) binary code of length n with M codewords. We construct Λ as follows: $x \in \mathbb{R}^n$ is in Λ if and only if the n -tuple $(x_1, \dots, x_n)$ is congruent modulo 2 to a codeword of C , i.e,

$$x \in \Lambda \text{ if and only if } x \equiv c \pmod{2}, \text{ for some } c \in C.$$

On the unit cube at origin, $\{(x_1, \dots, x_n) : 0 \leq x_i \leq 1, i = 1, \dots, n\}$, the centers (lattice points) are the M codewords, each one marked on the vertices. All other centers are obtained by adding even integers to any of

the coordinates of a codeword. Hence all the centers may be obtained by repeating a building block consisting of a $2 \times 2 \times \cdots \times 2$ cube with the unit cube in one corner.

It can be seen that when $C = [n, k, d_{\min}]$ is linear, $M = 2^k$, then Λ has a lattice structure.

It is shown [12] that the packing radius of Λ is

$$\rho = \frac{1}{2} \min\{2, \sqrt{d_{\min}}\},$$

and the center density is

$$\delta = 2^k \rho^n 2^{-n}.$$

Therefore the minimum distance $d_{\min}(\Lambda)$ of Λ is

$$d_{\min}(\Lambda) = \min\{2, \sqrt{d_{\min}}\}$$

and its coding gain is

$$\gamma(\Lambda) = 4\delta^{2/n} = 4^{k/n} \frac{d_{\min}^2(\Lambda)}{4}.$$

Hence

$$\gamma(\Lambda) = \begin{cases} 4^{\frac{k}{n}} & d_{\min} \geq 4, \\ \frac{d_{\min}^2}{4} 4^{\frac{k}{n}} & d_{\min} < 4. \end{cases}$$

Figure 4.1 shows an example of the Construction A applied to the repetition code $C = \{00, 11\}$. We can imagine that the lattice is obtained by shifting the square (cube 2×2) over all the plane. The translation of the two points $(0, 0)$ and $(1, 1)$ appear once in every translated square and form a lattice. The lattice has basis vectors $v_1 = (1, 1)$ and $v_2 = (0, 2)$.

Figure 4.1: Construction A of the repetition code

Now we generalize Construction A to group codes.

Definition 4.1. Let C be a group code over $G = \mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$. Define Λ as

$$\Lambda = \mathbb{Z}^n \text{diag}(g_i) + C,$$

where $\text{diag}(g_i)$ is the diagonal matrix with diagonal $(g_1, \dots, g_n)$. Clearly

$$\Lambda = \{(z_1 g_1 + c_1, \dots, z_n g_n + c_n) : z_i \in \mathbb{Z}, c = (c_1, \dots, c_n) \in C\}.$$

Theorem 4.2. The set Λ has a lattice structure with

$$d_{\min}(\Lambda) = \min\{g_{\min}, \sqrt{d_{\min}}\}, \quad \det \Lambda = \frac{\prod_{i=1}^n g_i}{|C|}$$

where $g_{\min} = \min\{g_1, \dots, g_n\}$ and $d_{\min}$ is the minimum weight of C .

Proof. The code C has group structure, so for any two elements x_1 and x_2 in Λ , $x_1 + x_2$ is in Λ . Consider two distinct lattice points x_1, x_2 in Λ . If x_1

and x_2 are congruent to the same codeword, i.e., for some $c \in C$

$$x_1 = (g_1 z_1, \dots, g_n z_n) + c, \quad x_2 = (g_1 z'_1, \dots, g_n z'_n) + c$$

then their distance apart is at least $g_{\min}$.

If x_1 and x_2 are congruent to different codewords then they differ by at least 1 in at least $d_{\min}$ places. Therefore their distance apart is at least $\sqrt{d_{\min}}$. Hence

$$d_{\min}(\Lambda) = \min\{g_{\min}, \sqrt{d_{\min}}\}.$$

Let $\Lambda' = g_1 \mathbb{Z} \oplus \dots \oplus g_n \mathbb{Z}$ then $\Lambda = \Lambda' + C$ where Λ' is the orthogonal sublattice of Λ with $\det \Lambda' = \prod_{i=1}^n g_i$. The lattice Λ is the union of $|C|$ cosets of Λ' in Λ , so

$$|C| = \frac{\det \Lambda'}{\det \Lambda}.$$

Therefore

$$\det \Lambda = \frac{\prod_{i=1}^n g_i}{|C|}.$$

■

4.2 Constructions B and C

Construction B [37]. Let $C = [n, k, d_{\min} = 8]$ be a binary linear code with the property that the weight of each codeword is even. The lattice Λ is constructed in the following way: $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ is in Λ if and only if x is congruent (modulo 2) to a codeword of C and $\sum_{i=1}^n x_i$ is divisible by 4.

The number of centers is half that of Construction A, so

$$\delta = 2^k \rho^n 2^{-n-1}.$$

Here $\rho = \frac{1}{2}\sqrt{8}$ therefore the coding gain is

$$\gamma(\Lambda) = 4\delta^{\frac{2}{n}} = 2 \times 4^{\frac{k-1}{n}}.$$

Since $1 \leq k \leq n$ then $1 \leq \gamma(\Lambda) \leq 24^{(n-1)/n} < 8$. Hence for n large we cannot gain more than 8, while much denser lattices exist. Also we are restricted to the class of linear codes with $d_{\min} = 8$. Therefore for the codes with large length n , this construction cannot be successful.

Before introducing Construction C we comment on the coordinate array of a point. The coordinate array of a point $x = (x_1, \dots, x_n)$ with integer coordinates is obtained by writing the binary expansion of the coordinates x_i in columns, beginning with the least significant digit. Complementary notation is used for negative numbers.

Let $C_\ell = [n, M_\ell, d_{\min}^\ell]$, $\ell = 0, 1, \dots, a$, be a family of $a + 1$ codes (not necessarily linear) with M_ℓ codewords and $d_{\min}^\ell = \alpha 4^{a-\ell}$, where $\alpha = 1$ or 2 . A sphere packing in $\mathbb{R}^n$ is given by the following construction.

Construction C [37]. A point x with integer coordinates is a center if and only if the 2^ℓ th row of the coordinate array of x is in C_ℓ , for $\ell = 0, 1, \dots, a$. It can be shown [12] that the radius of the spheres is $\rho = \sqrt{\alpha} 2^{a-\ell}$ and the center density is

$$\delta = M_0 M_1 \cdots M_a \alpha^{\frac{2}{n}} 2^{-2n}.$$

Therefore we can show that if the codes are linear with $\dim(C_\ell) = k_\ell$, then

$$\delta = 2^{\sum_{\ell=0}^a k_\ell - 2n} \quad \text{if } \alpha = 1$$

$$\delta = 2^{\sum_{\ell=0}^a k_\ell - \frac{3n}{2}} \quad \text{if } \alpha = 2.$$

Hence the coding gain is

$$\gamma = 4^{\sum_{\ell=0}^a \frac{k_\ell}{n} - 1} \quad \text{if } \alpha = 1$$

$$\gamma = \frac{1}{2} 4^{\sum_{\ell=0}^a \frac{k_\ell}{n}} \quad \text{if } \alpha = 2.$$

In general a nonlattice packing is obtained, in fact the multiplication of a center by an integer may not be a center. A modification of this construction (Construction D) is given in the following section. This one always produces a lattice packing. Construction *C* is also restricted to a class of codes with given minimum distance.

4.3 Construction *D*

This construction, first given in [5], uses a nested family of binary codes to produce a lattice packing Λ in $\mathbb{R}^n$.

Let $\alpha = 1$ or 2 and let

$$C_0 \supseteq C_1 \supseteq \dots \supseteq C_a$$

be a family of $a + 1$ linear codes where $C_\ell = [n, k_\ell, d_{\min}^\ell]$ with $d_{\min}^\ell \geq \frac{4^\ell}{\alpha}$, for $\ell = 1, \dots, a$ and C_0 be the $[n, n, 1]$ trivial code $\mathbb{F}_2^n$. Choose a basis $\{\mathbf{c}_1, \dots, \mathbf{c}_n\}$ for $\mathbb{F}_2^n$ such that

$$C_\ell = \langle \mathbf{c}_1, \dots, \mathbf{c}_{k_\ell} \rangle.$$

For any element $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_2^n$ and for $\ell = 1, \dots, a$ consider

$$\frac{1}{2^{\ell-1}} \mathbf{x} = \left(\frac{x_1}{2^{\ell-1}}, \dots, \frac{x_n}{2^{\ell-1}} \right)$$

as a vector in $\mathbb{R}^n$. Define $\Lambda \subseteq \mathbb{R}^n$ to consist of all vectors of the form

$$\mathbf{z} + \sum_{\ell=1}^a \sum_{j=1}^{k_\ell} \alpha_j^{(\ell)} \frac{1}{2^{\ell-1}} \mathbf{c}_j$$

where $\mathbf{z} \in (2\mathbb{Z})^n$ and $\alpha_j^{(\ell)} = 0$ or 1 .

Theorem 4.3. *The set Λ is a lattice, with minimal norm at least $4/\alpha$, determinant*

$$\det \Lambda = 2^{n - \sum_{\ell=1}^a k_\ell},$$

center density

$$\delta \geq \alpha^{-\frac{n}{2}} 2^{\sum_{\ell=1}^a k_\ell - n}$$

and coding gain

$$\gamma(\Lambda) \geq \alpha^{-1} 4^{\sum_{\ell=1}^a \frac{k_\ell}{n}}.$$

An integral basis for Λ is given by the vectors

$$\frac{1}{2^{\ell-1}} \mathbf{c}_j \text{ for } \ell = 1, \dots, a \quad j = k_{\ell+1} + 1, \dots, k_\ell,$$

plus $n - k_1$ vectors of the form $(0, \dots, 0, 2, 0, \dots, 0)$, see [5].

When $a = \alpha = 1$, Construction D reduces to the linear case of Construction A .

When $a = \alpha = 2$, C_1 is the $[n, n-1, 2]$ even weight code, C_2 has minimum distance 8 and the weight of every codeword of C_2 is a multiple of 4, then Construction D reduces to Construction B .

Whenever Construction C is applied to codes that are linear and nested, Construction D produces a lattice packing with the same density.

Example 4.4. *Barnes-Wall Lattices BW_n*

Let $C_{a-r} = R(2r, m)$, where $R(2r, m)$ is the $2r$ th Reed-Muller code of length $n = 2^m$. If m is even, take $\alpha = 1$ and $a = m/2$, while if m is odd, take $\alpha = 2$ and $a = (m - 1)/2$. In both cases the radius is $2^{(m-1)/2}$. Using properties of Reed-Muller codes, the density and coding gain of the corresponding lattice are

$$\delta = 2^{\frac{-5n}{4}} n^{\frac{n}{4}}, \quad \gamma(\Lambda) = \sqrt{\frac{n}{2}}.$$

Example 4.5. *Barnes-Wall lattice BW_{16}*

Let $a = 2, \alpha = 1$ and consider Reed-Muller codes

$$C_0 = R(4, 4), \quad C_1 = R(2, 4), \quad C_3 = R(0, 4).$$

In this case we have $n = 16, k_0 = 16, k_1 = 11, k_2 = 1$ and $d_1 = 4, d_2 = 16$.

Determinant of Λ is

$$\det \Lambda = 2^{16-(11+1)} = 16.$$

The center density and coding gain are

$$\delta = 2^{-4}, \quad \gamma(\Lambda) = \sqrt{2^3}.$$

The generator matrix of Λ is given in the following. The rows of this matrix are a basis for $C_0 = \mathbb{F}_2^{16}$. The first 11 rows are the generators of C_1 which are divided by 1 and the first row is the generator of C_2 which is divided by 2. The last 5 rows are the part of generators of $\mathbb{F}_2^{16}$ that are not included in any set of code generators hence are multiplied by 2.

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

4.4 Construction D'

As we saw Construction D converts a set of generators for a family of nested codes into a set of generators for a lattice. Construction D' [8] converts a set of parity checks defining a family of codes into congruences for a lattice.

Let $\alpha = 1$ or 2 , and let

$$C_0 \supseteq C_1 \supseteq \cdots \supseteq C_a$$

be a family of binary linear codes, where C_ℓ has parameters $[n, k_\ell, d_{\min}^\ell]$ such that

$$d_{\min}^\ell \geq \alpha 4^\ell \quad \ell = 0, 1, \dots, a. \quad (4.1)$$

Let $\mathbf{h}_1, \dots, \mathbf{h}_n$ be linear independent vectors in $\mathbb{F}_2^n$ such that for $\ell = 0, 1, \dots, a$ the code C_ℓ is defined by the $r_\ell = n - k_\ell$ parity check vectors

$\mathbf{h}_1, \dots, \mathbf{h}_{r_\ell}$ and let $r_{-1} = 0$. Consider vectors $\mathbf{h}_j$ as integral vectors in $\mathbb{R}^n$, with components 0 or 1. Define the new lattice Λ to consist of those $\mathbf{x} \in \mathbb{Z}^n$ that satisfy the congruences

$$\mathbf{h}_j \cdot \mathbf{x} \equiv 0 \pmod{2^{\ell+1}} \quad (4.2)$$

for all $\ell = 0, 1, \dots, a$ and $r_{a-\ell-1} + 1 \leq j \leq r_{a-\ell}$.

It is enough to see that the points satisfying (4.2) do form a lattice.

Theorem 4.6. *The minimal norm $d_{\min}^2(\Lambda)$ is at least $\alpha 4^a$, and*

$$\det \Lambda = 2^{\sum_{\ell=0}^a r_\ell}. \quad (4.3)$$

Proof. The proof is given by Bos [8]. We prove a more general result in the next theorem. ■

Now we prove the following theorem, which generalizes Construction D' to any collection family of nested codes without the condition $d_{\min}^\ell \geq \alpha 4^\ell$. This theorem also relates the performance of the lattice to the performance of its underlying subcodes.

Theorem 4.7. *Let $C_0 \supseteq C_1 \supseteq \dots \supseteq C_a$ be a family of linear binary codes with $C_\ell = [n, k_\ell, d_{\min}^\ell]$ and let $\{\mathbf{h}_1, \dots, \mathbf{h}_n\}$ be linear independent vectors in $\mathbb{F}_2^n$ such that*

$$C_\ell^* = \langle \mathbf{h}_1, \dots, \mathbf{h}_{r_\ell} \rangle, \quad \ell = 0, 1, \dots, a.$$

Let Λ be the corresponding lattice given by Construction D' . Then for the minimal norm $d_{\min}^2(\Lambda)$ we have

$$\min\{d_{\min}^a, 4d_{\min}^{a-1}, 4^2d_{\min}^{a-2}, \dots, 4^a d_{\min}^0, 4^{a+1}\} \leq d_{\min}^2(\Lambda) \leq 4^{a+1}.$$

Proof. For any $\mathbf{x} \in \Lambda$ we can write the modular equations as

$$\mathbf{x} \cdot \mathbf{h}_j \equiv 0 \pmod{2^{a-k+1}}, \quad r_{k-1} \leq j \leq r_k$$

for $k = 0, 1, \dots, a$. Since $2^{a-k+1} \mid 2^{a-i+1}$, for $i = 0, 1, \dots, k-1$, we have

$$\mathbf{x} \cdot \mathbf{h}_j \equiv 0 \pmod{2^{a-k+1}}, \quad 1 \leq j \leq r_k.$$

Define

$$\Lambda_k = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{x} \cdot \mathbf{h}_j \equiv 0 \pmod{2^{a-k+1}}, \quad 1 \leq j \leq r_k\}.$$

Then each Λ_k is a lattice and by the definition of Λ we have

$$\Lambda = \bigcap_{k=0}^a \Lambda_k. \quad (4.4)$$

In the next step we prove that if $\mathbf{x} \in \mathbb{Z}^n$ has at least one coordinate which is an odd number and

$$\mathbf{x} \cdot \mathbf{h}_j \equiv 0 \pmod{2}, \quad 1 \leq j \leq r_k, \quad (4.5)$$

then

$$\|\mathbf{x}\|^2 \geq d_{\min}^k. \quad (4.6)$$

To prove this, let $\mathbf{x} = \mathbf{c} + 2\mathbf{z}$, where $\mathbf{z} \in \mathbb{Z}^n$ and $\mathbf{c} = (c_1, \dots, c_n)$, with $c_i = 0$ or 1. Since $\mathbf{x}$ has at least one odd coordinate, then $\mathbf{c} \neq 0$. Also $\mathbf{h}_j \cdot \mathbf{c} \equiv 0 \pmod{2}$, therefore $\mathbf{c} \in C_k$ and $\|\mathbf{c}\|^2 \geq d_{\min}^k$. Hence $\|\mathbf{x}\|^2 \geq d_{\min}^k$.

Now let $0 \neq \mathbf{x} \in \Lambda$. Since $\mathbf{x} \neq 0$ we can choose $k \geq 0$ such that $2^{-k}\mathbf{x} \in \mathbb{Z}^k$ and $2^{-k-1}\mathbf{x} \notin \mathbb{Z}^k$. If $k < a+1$ then

$$\frac{\mathbf{x}}{2^k} \cdot \mathbf{h}_j \equiv 0 \pmod{2}, \quad \text{for } 1 \leq j \leq r_{a-k}.$$

So

$$\| \mathbf{x} \|^2 \geq 4^k d_{\min}^{a-k}.$$

If $k \geq a + 1$ then $\| \mathbf{x} \|^2 \geq (2^{a+1})^2$. Therefore

$$\| \mathbf{x} \|^2 \geq d_{\min}^a \text{ or } \| \mathbf{x} \|^2 \geq 4d_{\min}^{a-1} \text{ or } \dots \text{ or } \| \mathbf{x} \|^2 \geq 4^a d_{\min}^0 \text{ or } \| \mathbf{x} \|^2 \geq 4^{a+1}.$$

This complete the proof for the lower bound on $d_{\min}^2(\Lambda)$. It is trivial that the vector $2^{a+1}\mathbf{e}_i \in \Lambda$ which proves the upper bound on $d_{\min}^2(\Lambda)$. $\blacksquare$

Corollary 4.8. *The coding gain of Construction D' is*

$$\gamma(\Lambda) \geq \frac{\min\{d_{\min}^a, 4d_{\min}^{a-1}, 4^2d_{\min}^{a-2}, \dots, 4^a d_{\min}^0, 4^{a+1}\}}{4^{\sum_{\ell=0}^a \frac{r_{\ell}}{n}}}.$$

Furthermore when $d_{\min}^{\ell} \geq 4^{\ell+1}$, $\ell = 0, 1, \dots, a$, then

$$\gamma(\Lambda) = \frac{4^{a+1}}{4^{\sum_{\ell=0}^a \frac{r_{\ell}}{n}}}. \quad (4.7)$$

Proof. It follows from the last theorem and the definition of coding gain. $\blacksquare$

We note that by using the bound that is given by Bos (Theorem 4.6) we have the following bound on coding gain

$$\gamma(\Lambda) \geq \frac{\alpha 4^a}{4^{\sum_{\ell=0}^a \frac{r_{\ell}}{n}}} \quad (4.8)$$

where $\alpha = 1$ or 2 .

Corollary 4.9. *Let $C_0 = [n, n, 1]$ be the trivial code, then Construction D' produces a lattice with the same density (and coding gain) as Construction D does. In this case*

$$\delta \geq \alpha^{\frac{n}{2}} 2^{\sum_{\ell=1}^a k_{\ell} - n},$$

and coding gain

$$\gamma(\Lambda) \geq \alpha 4^{\sum_{\ell=1}^a \frac{k_\ell}{n}},$$

where $\alpha = 1, 2, 3$ or 4 .

Proof. By using the fact that $r_0 = 0$, the proof is straightforward. ■

By multiplying the modular equations by appropriate powers of 2, we can restate Construction D' . Indeed $\mathbf{x} \in \Lambda$ if

$$\begin{aligned} \mathbf{h}_j \cdot \mathbf{x} &\equiv 0 \pmod{2^{a+1}} & 1 \leq j \leq r_0, \\ 2\mathbf{h}_j \cdot \mathbf{x} &\equiv 0 \pmod{2^{a+1}} & r_0 + 1 \leq j \leq r_1, \\ &\vdots & \vdots \\ 2^\ell \mathbf{h}_j \cdot \mathbf{x} &\equiv 0 \pmod{2^{a+1}} & r_{\ell-1} + 1 \leq j \leq r_\ell, \\ &\vdots & \vdots \\ 2^{a-1} \mathbf{h}_j \cdot \mathbf{x} &\equiv 0 \pmod{2^{a+1}} & r_{a-2} + 1 \leq j \leq r_{a-1}, \\ 2^a \mathbf{h}_j \cdot \mathbf{x} &\equiv 0 \pmod{2^{a+1}} & r_{a-1} + 1 \leq j \leq r_a. \end{aligned} \tag{4.9}$$

Therefore we have proved the following lemma.

Lemma 4.10. *Let $C_\ell = [n, k_\ell, d_{\min}^\ell]$, $\ell = 0, 1, \dots, a$ and Λ be as in the previous theorem. Let*

$$\widehat{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_{r_0}, 2\mathbf{h}_{r_0+1}, \dots, 2\mathbf{h}_{r_1}, \dots, 2^a \mathbf{h}_{r_{a-1}+1}, \dots, 2^a \mathbf{h}_{r_a}\}$$

and let $\mathbf{H}$ be the matrix whose rows are the elements of $\widehat{H}$, i.e.,

$$\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_{r_0} \ 2\mathbf{h}_{r_0+1} \cdots 2\mathbf{h}_{r_1} \cdots 2^a \mathbf{h}_{r_{a-1}+1} \cdots 2^a \mathbf{h}_{r_a}]^{tr}.$$

Then $\mathbf{x} \in \mathbb{Z}^n$ is in Λ if

$$\mathbf{H} \cdot \mathbf{x}^T \equiv \mathbf{0} \pmod{2^{a+1}}. \tag{4.10}$$

These constitute the check equations of the lattice. Therefore the row vectors of $\mathbf{H}$, i.e., the elements of $\widehat{H}$, form the generators for the dual of the label code of Λ .

Example 4.11. Let $a = 2$ and $C_0 = \langle 1111, 0010, 0100 \rangle$, $C_1 = \langle 1111, 0010 \rangle$ and $C_2 = \langle 1111 \rangle$. Then $C_0^* = \langle 1001 \rangle$, $C_1^* = \langle 1001, 0101 \rangle$ and $C_2^* = \langle 1001, 0101, 0011 \rangle$. Hence $h_1 = (1001)$, $h_2 = (0101)$ and $h_3 = (0011)$. Let

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 4 & 4 \end{bmatrix}$$

Then $\mathbf{x} \in \mathbb{Z}^n$ belongs to Λ if and only if

$$H \cdot \mathbf{x}^T \equiv \mathbf{0} \pmod{2^3}.$$

In other words, $\mathbf{x} = (x_1, x_2, x_3, x_4) \in \Lambda$ if and only if

$$\begin{aligned} x_1 + x_4 &\equiv 0 \pmod{8} \\ 2x_2 + 2x_4 &\equiv 0 \pmod{8} \\ 4x_3 + 4x_4 &\equiv 0 \pmod{8}. \end{aligned}$$

Definition 4.12. A lattice Λ constructed based on Construction D' is called *Low Density Parity Check lattice (LDPC lattice)* if its parity check matrix is a sparse matrix.

It is trivial that if the underlying nested codes C_ℓ are LDPC codes then the corresponding lattice is an LDPC lattice and vice versa.

The following theorem provides us a computational method of finding cross sections Λ_{W_j} , $j = 1, \dots, n$.

Theorem 4.13. *Let H be defined as in the previous lemma. For any $1 \leq j \leq n$, for $r_{k-1} + 1 \leq s_j \leq r_k$, let s_j be the smallest number such that*

$$[H]_{s_j, j} \neq 0.$$

Let Λ_{W_j} be the cross section of Λ in the coordinate system $W_j = \langle \mathbf{e}_j \rangle$. Then

$$\Lambda_{W_j} = 2^{a-k+1}\mathbb{Z}.$$

Proof. Let $\mathbf{x} \in \Lambda_{W_j}$, then $\mathbf{x} = (0, \dots, 0, x_j, 0, \dots, 0)$ where x_j is in the j th coordinate of $\mathbf{x}$. Therefore $\mathbf{x} \in \Lambda$ and we have

$$2^k \mathbf{h}_{s_j} \cdot \mathbf{x} \equiv 0 \pmod{2^{a+1}}.$$

Thus $2^k x_j = q 2^{a+1}$, for some $q \in \mathbb{Z}$. Hence $x_j = 2^{a-k+1}q$ and we can conclude that

$$\Lambda_{W_j} \subseteq 2^{a-k+1}\mathbb{Z}.$$

Now let $x_j \in 2^{a-k+1}\mathbb{Z}$, then $x_j = q 2^{a-k+1}$, for some $q \in \mathbb{Z}$. Let $\mathbf{x} = (0, \dots, 0, x_j, 0, \dots, 0)$ where x_j is the j th coordinate of $\mathbf{x}$. In the following we show that $\mathbf{x}$ is in Λ .

By the definition of s_j the j th coordinate of any $\mathbf{h}_i$ for which $i = 1, \dots, s_j - 1$, is zero. So

$$\mathbf{x} \cdot \mathbf{h}_i = 0, \quad 1 \leq i \leq s_j - 1. \quad (4.11)$$

By multiplying both sides of the Equation (4.11) by any integer m we have $m \mathbf{h}_i \cdot \mathbf{x} \equiv 0 \pmod{2^{a+1}}$, $1 \leq i \leq s_j - 1$. In particular, since $r_{k-1} \leq s_j - 1$ then

$$2^\ell \mathbf{h}_i \cdot \mathbf{x} \equiv 0 \pmod{2^{a+1}}, \quad r_{\ell-1} + 1 \leq i \leq r_\ell, \quad \ell = 0, 1, \dots, k-1. \quad (4.12)$$

On the other hand $\mathbf{h}_i \cdot \mathbf{x} = 0$ or $\mathbf{h}_i \cdot \mathbf{x} = q2^{a-k+1}$, $s_j \leq i \leq r_a$, thus

$$2^k \mathbf{h}_i \cdot \mathbf{x} \equiv 0 \pmod{2^{a+1}}, \quad s_j \leq i \leq r_a. \quad (4.13)$$

We can multiply both sides of the Equation (4.13) by any integer. Since $r_k \geq s_j$ then

$$2^\ell \mathbf{h}_i \cdot \mathbf{x} \equiv 0 \pmod{2^{a+1}}, \quad r_{\ell-1} + 1 \leq i \leq r_\ell, \quad \ell = k+1, \dots, a. \quad (4.14)$$

Equations (4.12) and (4.14) along with the fact that $x_j = q2^{a+1-k}$ imply that $\mathbf{x} \in \Lambda$. This complete the proof. $\blacksquare$

Corollary 4.14. *Let Λ be a lattice constructed using Construction D' . Then the sizes of the alphabet sequences of the label code of Λ are powers of 2. In fact for every $i = 1, \dots, n$,*

$$g_i = \frac{2^{a+1}}{2^{g(i)}} \quad (4.15)$$

for some $g(i) \in \{0, 1, \dots, a\}$.

Corollary 4.15. *Suppose C_0 is a code for which $d_{\min}^0 \geq 2$, then*

$$\Lambda_{W_i} = 2^{a+1}\mathbb{Z}^n, i = 1, \dots, n. \quad (4.16)$$

Proof. If the j th column of the parity check matrix of C_0 was zero, then $(0, \dots, 0, 1, 0, \dots, 0)$, with 1 in its j th position, would be a codeword, contradicting the hypothesis. Thus every column of this matrix is a nonzero vector. Therefore in Theorem 4.13 $k = 0$ and so $\Lambda_{W_i} = 2^{a+1}\mathbb{Z}^n$. $\blacksquare$

Example 4.16. *By applying Theorem 4.13 to the previous example, we have $s_1 = 1$ ($k = 0$), $s_2 = 2$ ($k = 1$), $s_3 = 3$ ($k = 2$), $s_4 = 1$ ($k = 0$). Therefore*

$$\Lambda_{W_1} = 8\mathbb{Z}, \quad \Lambda_{W_2} = 4\mathbb{Z}, \quad \Lambda_{W_3} = 2\mathbb{Z}, \quad \Lambda_{W_4} = 8\mathbb{Z}.$$

It is easy to see that two vectors $(1, 1, 1, 7)$ and $(7, 3, 1, 1)$ belong to the lattice Λ , therefore $\det P_{W_i} = 1$, for $i = 1, \dots, 4$. Hence the sizes of the alphabet sequence of the label code of Λ are

$$g_1 = 8, \quad g_2 = 4, \quad g_3 = 2, \quad g_4 = 8.$$

The corresponding Tanner graph is given in Figure 4.2.

Figure 4.2

By changing the scale and relabelling the codes, Construction D may be restated in such a way that the norm and determinant of the corresponding lattices agree. Then if C_i are Reed-Muller codes, the two lattices coincide. We illustrate this in the following example.

Example 4.17. Consider BW_{16} given in Example 4.6. Let $a = 1$ and consider Reed-Muller codes $C_0 = R(2, 4)$ and $C_1 = R(0, 4)$, as we chose in Example 4.6. By using the properties of Reed-Muller codes we have

$$C_0^* = R(1, 4) \quad \text{and} \quad C_1^* = R(3, 4).$$

The code C_0^* consists of $r_0 = 5$ generators and C_1^* consists of $r_1 = 15$ generators. By using Lemma 4.11, we put all 15 vectors of C_1^* in the rows of $\mathbf{H}$ in

such a way that the first 5 rows of $\mathbf{H}$ are the generators of C_0^* and the next 11 generators are multiplied by 2. Therefore the matrix H that defines the modular equations is

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 \end{bmatrix}$$

The lattice Λ consists of all vectors $\mathbf{x} \in \mathbb{Z}^n$ such that

$$H.\mathbf{x} \equiv \mathbf{0} \pmod{4}.$$

It is clear that the vector $(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$ is in Λ , therefore $\det P_{W_i} = 1$, for $i = 1, 2, \dots, 16$. Also by using the previous theorem we conclude that $\Lambda_{W_j} = 4$, $j = 1, \dots, 16$. Therefore the sizes of alphabet sequence are $g_i = 4$, for $i = 1, 2, \dots, 16$. Here $r_0 = 5$ and $r_1 = 15$ so $\det \Lambda = 2^{20}$. By multiplying the generator matrix of the Barnes-Wall lattice of Example 4.6 by 2, the two lattices coincide.

We note that in general the two constructions D and D' produce inequivalent lattices with the same density [12].

Theorem 4.18. *Let $\Lambda_{D'}$ and Λ_A be the two lattices that are constructed from a code $C = [n, k]$ by Constructions D' and A , respectively. Then $\Lambda_{D'}$ and Λ_A coincide.*

Proof. Let $\{h_1, \dots, h_r\}$ be the parity check vectors of a code C . Let $x \in \Lambda_{D'}$, then by using the Euclidian algorithm we have $\mathbf{x} = \mathbf{c} + 2\mathbf{z}$ where $\mathbf{c} = (c_1, \dots, c_n)$, with $c_i = 0$ or 1 and $\mathbf{z} = (z_1, \dots, z_n) \in \mathbb{Z}^n$. Since

$$\mathbf{x}.h_j \equiv 0 \pmod{2} \text{ and } 2\mathbf{z}.h_j \equiv 0 \pmod{2}, j = 1, \dots, r,$$

then $\mathbf{c}.h_j \equiv 0 \pmod{2}$. Hence $\mathbf{c} \in C$.

Let $\mathbf{x} \in \Lambda_A$ then $\mathbf{x} = \mathbf{c} + 2\mathbf{z}$, where $\mathbf{z} \in \mathbb{Z}^n, \mathbf{c} \in C$. We have

$$\mathbf{x}.h_j = \mathbf{c}.h_j + 2\mathbf{z}.h_j \equiv 0 \pmod{2},$$

and hence $\mathbf{x} \in \Lambda_{D'}$. ■

4.5 Cycle free lattices

In [13] Etzion proved that cycle free Tanner graphs cannot support good codes. In fact he showed that any cycle free linear code $C = [n, k, d_{\min}]$ of rate $R = k/n \geq 0.5$ has minimum distance $d_{\min} \leq 2$. For codes of rate $R \leq 0.5$ we have the following theorem.

Theorem 4.19. *Let C be an $[n, k, d_{\min}]$ cycle free linear code over $\mathbb{F}_q$ of rate $k/n \leq 0.5$. Then up to scaling by constants in $\mathbb{F}_q$ at certain positions, C is obtained by repeating symbols in a cycle free code of rate $k/n > 0.5$.*

Proof. The proof is given in [13]. ■

Definition 4.20. A lattice is called cycle free if its Tanner graph $\mathcal{G} = (V, E)$ has no cycle. In particular the incident matrix H of a cycle free Tanner graph is called a cycle free matrix. The weight $wt(H)$ of H is defined as the total number of nonzero entries in H .

The following lemma is used in the main theorem of this section (Theorem 4.22).

Lemma 4.21. A graph $\mathcal{G} = (V, E)$ is cycle free if and only if $|E| = |V| - \kappa(\mathcal{G})$, where $\kappa(\mathcal{G})$ is the number of connected components in $\mathcal{G}$.

Proof. Let $\mathcal{G} = (V, E)$ be connected (i.e. $\kappa(\mathcal{G}) = 1$). It is known [51] pp. 52, that $\mathcal{G}$ is cycle free if and only if $\mathcal{G}$ has $|V| - 1$ edges. We can apply this property to a forest $\mathcal{G}$ with $\kappa(\mathcal{G})$ components. ■

Theorem 4.22. Let Λ be an $(a + 1)$ -level cycle free lattice constructed using Construction D' whose underlying codes are $C_\ell = [n, k_\ell, d_{\min}^\ell]$, with r_ℓ parity checks, $\ell = 0, 1, \dots, a$. Assume that C_a has rate $k_a/n \geq 0.5$ and $d_{\min}^0 > 1$. Then $d_{\min}^2(\Lambda) = 2$.

Proof. Let H be the $(a + 1)$ -level cycle free parity check matrix of Λ and $\mathcal{G} = (V, E)$ be its corresponding Tanner graph. We can remove the linearly dependent rows of H while preserving the cycle free property. So we assume that H has full row-rank and $r_a = n - k_a$. Let μ_i denote the number of columns of weight i in H . Then $\mu_0 = 0$, otherwise $d_{\min}^0 = 1$.

It is trivial that $n = \mu_1 + \cdots + \mu_{r_a}$. So

$$\begin{aligned} \mu_1 + 2(n - \mu_1) &= \mu_1 + 2(\mu_2 + \cdots + \mu_{r_a}) \\ &\leq \sum_{i=1}^{r_a} i\mu_i = \text{wt}(H) = |E|. \end{aligned}$$

Since $\mathbb{G}$ is a forest then $\kappa(\mathbb{G}) \geq 1$. So by Lemma 4.21 $|E| \leq |V| - 1$. But $|V| = n + r_a$, and hence

$$\mu_1 + 2(n - \mu_1) \leq n + r_a - 1. \quad (4.17)$$

By substituting $r_a = n - k_a$ in (4.17), we have $\mu_1 \geq k_a + 1$. Since $k_a/n \geq 0.5$ then $k_a \geq r_a$ and

$$\mu_1 \geq r_a + 1.$$

The parity check matrix H is full rank so it contains at least two columns, say the first and the second one, of weight one that are the same. The corresponding nonzero entries of these columns are in the first level of H (i.e. the sub-matrix of H consisting the first r_0 rows), otherwise $d_{\min}^0 = 1$. This implies that the vector $(1, 1, 0, \dots, 0)$ with two ones belongs to the lattice. Therefore $d_{\min}(\Lambda) = \sqrt{2}$.

■

Chapter 5

Decoding Algorithm

In [46] Tanner generalized Gallager's LDPC codes [25, 26] to codes on general bipartite graphs, here called "Tanner Graphs". Tanner also developed the so-called "max-sum" and "sum-product" algorithms and proved that they converge on cycle-free graphs.

In this chapter we first present the max-sum algorithm and then we generalize this algorithm to an iterative decoding algorithm for lattices. We call this new iterative decoding algorithm for lattices the "Generalized Min-Sum algorithm".

In Lemma 5.1 we provide a practical way to compute the "initial weights", (the inputs to the algorithm).

For an analysis of the decoding complexity we provide a (close) upper bound on the decoding complexity of the Generalized Min-Sum algorithm. We give the exact number of computations in each iteration for the lattice constructed using Construction D' . Also some lower and upper bounds are

given on the decoding complexity based on coding gain, label group sizes and other factors involved in Construction D' .

The analysis of the decoding complexity of the Generalized Min-Sum algorithm shows that, for the fixed check node degree and symbol node degree, the decoding complexity grows linearly with the dimension n . Therefore this algorithm can be applied to decode high dimension lattices using Construction D' with LDPC codes as their underlying codes.

In [52] Wiberg showed that the Max-Sum algorithm converges on cycle free Tanner graphs. Here we adapt his proof to the case of group codes and prove that the Generalized Min-Sum algorithm converges for cycle free Tanner graphs. However for graphs with cycles there is no general result.

5.1 The Max-Sum Algorithm

Let C be a binary code with length n . Each possible value a_k of each symbol x_k is assigned a cost $w_k(a_k)$, a log likelihood

$$w_k(a_k) = \log p(y_k | x_k = a_k), \quad k = 1, \dots, n,$$

where y_k is the received symbol corresponding to the code symbol x_k . On a memoryless channel, for which the probability distribution of the output depends only on the input at that time and is conditionally independent of previous channel inputs or outputs, with log likelihood costs we have

$$\log p(\mathbf{y}|\mathbf{x}) = \log(p(y_1|x_1 = a_1) \times \dots \times p(y_n|x_n = a_n)) = \sum_{k=1}^n \log p(y_k|x_k).$$

So the total cost $w(\mathbf{x})$ of a valid codeword $\mathbf{x}$ is the sum of its symbol costs

$$w(\mathbf{x}) = \sum_{k=1}^n w_k(a_k).$$

The problem is to compute for each possible value a_k of each symbol node x_k the maximum cost of any valid codeword $\mathbf{x} \in C$ such that $x_k = a_k$, i.e.,

$$\max\{w(\mathbf{x}) : \mathbf{x} \in C, x_k = a_k\}.$$

Consider the edges of the Tanner graph. Each edge connects a symbol node to a check node. The set of values of an edge may be taken to be equal to the set of values of the adjoining symbol nodes. The objective is to find $\max\{w(\mathbf{x}) | \mathbf{x} \in C, x_k = a_k\}$ for each of the possible values a_k of each edge. A cut through any edge of a cycle free graph cuts the graph into two parts, say “upstream” and “downstream”. The cost $w(\mathbf{x})$ of any codeword $\mathbf{x}$ may be expressed as the sum of the upstream and downstream contributions $w(\mathbf{x}^+)$ and $w(\mathbf{x}^-)$, respectively, i.e., $w(\mathbf{x}^+) + w(\mathbf{x}^-)$.

The rules for computations are as follows:

for an edge downstream from a symbol node, for each possible value a_k , the upstream contribution is the sum of the upstream contributions coming into the symbol node plus the local contribution (cost) of the node itself (Figure 5.1);

for an edge downstream from a check node, for each possible value a_k , the upstream contribution is the maximum of the sum of all upstream contributions over configurations consistent with a_k (Figure 5.2).

For example, we shall work through an example of the max-sum algorithm for the Tanner graph of Figure 5.3. (This example is taken directly from [21].)

Figure 5.1: Symbol node operation

Figure 5.2: Check node operation

In step 1, for each of the six leaf nodes, the symbol weights are simply transferred to the neighboring edge.

In step 2, for each of the two outgoing values of each of the three check nodes, an add-compare-select (ACS) operation is performed to find the maximum of the sums of the weights of the incoming configurations consistent with that value (i.e. the incoming configurations $\{00, 11\}$ are consistent with the outgoing value 0, and $\{01, 10\}$ are consistent with 1.)

In step 3, for each of the two outgoing values of each of the three edges out of the central node, the two incoming weights are summed with the central node weight to give the outgoing weight.

In step 4, for each of the two outgoing values of each of the two remaining edges out of each of the three parity check nodes, an ACS operation is performed. This determines all upstream and downstream weights for all values of all edges.

In step 5, the upstream and downstream weights are summed to give the final weights of each value of each edge, and thus of its associated symbol node.

Figure 5.3

Figure 5.4

5.2 Overview of the Generalized Min-Sum Algorithm

Given a vector $\mathbf{x} \in \mathbb{R}^n$ and an n -dimensional lattice $\Lambda \subset \mathbb{R}^n$, the *lattice decoding problem* is to find a lattice vector $\mathbf{v}$ such that $\|\mathbf{x} - \mathbf{v}\|$ is minimized.

In the corresponding trellis coordinate system $S = \{W_i\}_{i=1}^n$, let $\mathbf{v}_i$ be a

shortest vector of Λ_{W_i} , i.e., $\Lambda_{W_i} = \mathbb{Z}\mathbf{v}_i$. Also let $\mathbf{x} = \sum_{i=1}^n \bar{x}_i \mathbf{v}_i$, where

$$\bar{x}_i = \frac{\langle \mathbf{x}, \mathbf{v}_i \rangle}{\langle \mathbf{v}_i, \mathbf{v}_i \rangle}.$$

As in Section 3.3, let $(\mathbb{Z} + a_j)\mathbf{v}_i$, $0 \leq j \leq |G_i| - 1$, be the elements of the label group $G_i(\Lambda)$ and $d_{ij}\mathbf{v}_i$ denote the closest vector in $(\mathbb{Z} + a_j)\mathbf{v}_i$ to $\bar{x}_i \mathbf{v}_i$. Define the weight $w_{\mathbf{x}}^{(i)}(j)$ as $\| (d_{ij} - \bar{x}_i)\mathbf{v}_i \|^2$, i.e., the squared distance between the elements of $(\mathbb{Z} + a_j)\mathbf{v}_i$ and $\bar{x}_i \mathbf{v}_i$. Let L be the label code of Λ over the alphabet sequence $\mathbb{Z}_{g_1} \times \cdots \times \mathbb{Z}_{g_n}$. By using the isomorphism between $G_i(\Lambda)$ and $\mathbb{Z}_{g_i}$, ($g_i = |G_i(\Lambda)|$), we rename j to c_i , where $c_i \in \mathbb{Z}_{g_i}$. The total weight $\mathbf{x}$ of any valid codeword $\mathbf{c}$ in the label code L is defined by

$$W_{\mathbf{x}}(\mathbf{c}) = \sum_{i=1}^n w_{\mathbf{x}}^{(i)}(c_i), \quad \mathbf{c} = (c_1, \dots, c_n) \in L. \quad (5.1)$$

In the corresponding Tanner graph of Λ , each possible value of each symbol node c_i is assigned the weight $w_{\mathbf{x}}^{(i)}(c_i)$.

Now the problem is to compute for each possible value c_i of $\mathbb{Z}_{g_i}$ the minimum weight of $w_{\mathbf{x}}(\mathbf{c})$ of any valid codeword $\mathbf{c}$, i.e.,

$$\min \sum_{i=1}^n w_{\mathbf{x}}^{(i)}(c_i),$$

where the minimum is taken over all codewords $\mathbf{c} = (c_1, \dots, c_n) \in L$.

Let $\mathbf{c} \in L$ be the codeword for which the total weight, $w_{\mathbf{x}}(\mathbf{c})$, is minimized and take $a_{c_i} = c_i \frac{\det(P_{W_i})}{\det(\Lambda_{W_i})}$. Assume

$$d_i = d_{ic_i} = a_{c_i} + \lceil \bar{x}_i - a_{c_i} \rceil,$$

where $\lceil u \rceil$ denotes the closest integer to u . Then it is clear that the vector

$$\mathbf{v} = \sum_{i=1}^n d_i \mathbf{v}_i$$

is the desired lattice vector.

5.3 Generalized Min-Sum Algorithm

The proposed algorithm is based on an iterative decoding algorithm, called the “message-passing algorithm”. This algorithm is performed by exchanging messages between check nodes and symbol nodes through the edges of the Tanner graph, in both directions and iteratively.

In its conventional form, this iterative decoding algorithm starts by creating initial weights at symbol nodes from observations at the input $\mathbf{x} \in \mathbb{R}^n$. It then passes these messages to check nodes through the edges of the graph. Iteration then starts. Each iteration consists of two steps:

(1) Symbol node processing: the incoming messages sent from check nodes are processed and the updated information about their values are sent back. Note that at the first iteration all incoming messages from check node are zero. The message passing from symbol node i to check node k is denoted by $M_{s_i \rightarrow ch_k}$

(2) Check node processing: the incoming messages sent from symbol nodes are processed and the updated information are sent back to symbol nodes. The message passing from check node k to symbol node i is denoted by $M_{ch_k \rightarrow s_i}$.

The message passing between symbol nodes and check nodes is sometimes called an *intermediate weight function*. In each iteration, a hard decision on the values of each symbol is made at symbol nodes.

In the following we explain each part of the algorithm separately.

Figure 5.5: Generalized min-sum algorithm

5.3.1 Initialization

Let $\mathbf{x} = \sum_{i=1}^n \bar{x}_i \mathbf{v}_i = \sum_{i=1}^n x_i \mathbf{e}_i \in \mathbb{R}^n$ be given. Each symbol node i is assigned an initial weight as a vector with g_i components

$$w_{\mathbf{x}}^{(i)} = (w_{\mathbf{x}}^i(0), w_{\mathbf{x}}^i(1), \dots, w_{\mathbf{x}}^i(g_i - 1)), \quad i = 1, 2, \dots, n,$$

where $w_{\mathbf{x}}^{(i)}(j)$ is defined as the squared distance between the i th coordinate of $\mathbf{x}$ and the j th element of its corresponding label group G_i .

Lemma 5.1. *For a given $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ we have*

$$w_{\mathbf{x}}^i(j) = (\det(\Lambda_{W_i}))^2 \left(\left(\frac{x_i}{\det(\Lambda_{W_i})} - \frac{j}{g_i} \right) - \left\lceil \frac{x_i}{\det(\Lambda_{W_i})} - \frac{j}{g_i} \right\rceil \right)^2,$$

for $i = 1, 2, \dots, n$, and $j = 0, 1, \dots, g_i - 1$.

Proof. The vector $\mathbf{v}_i$ is the shortest vector of Λ_{W_i} , and so $\mathbf{v}_i = \det(\Lambda_{W_i}) \mathbf{e}_i$.

We have

$$\bar{x}_i = \frac{\langle \mathbf{x}, \mathbf{v}_i \rangle}{\langle \mathbf{v}_i, \mathbf{v}_i \rangle} = \frac{1}{\det(\Lambda_{W_i})} x_i.$$

By the definition of $w_{\mathbf{x}}^{(i)}(j)$ we have

$$\begin{aligned} w_{\mathbf{x}}^{(i)}(j) &= \min_{z \in \mathbb{Z}} \|\bar{x}_i \mathbf{v}_i - (z + a_j) \mathbf{v}_i\|^2 \\ &= \min_{z \in \mathbb{Z}} |\bar{x}_i - (z + a_j)|^2 \|\mathbf{v}_i\|^2 \\ &= \min_{z \in \mathbb{Z}} \left| \frac{x_i}{\det(\Lambda_{W_i})} - \left(z + j \frac{\det(P_{W_i})}{\det(\Lambda_{W_i})} \right) \right|^2 \|\mathbf{v}_i\|^2. \end{aligned}$$

However, $\frac{\det(\Lambda_{W_i})}{\det(P_{W_i})} = g_i$ so

$$\begin{aligned} w_{\mathbf{x}}^{(i)}(j) &= \min_{z \in \mathbb{Z}} \left| \frac{x_i}{\det(\Lambda_{W_i})} - \left(z + \frac{j}{g_i} \right) \right|^2 (\det(\Lambda_{W_i}))^2 \\ &= (\det(\Lambda_{W_i}))^2 \left(\left(\frac{x_i}{\det(\Lambda_{W_i})} - \frac{j}{g_i} \right) - \left\lfloor \frac{x_i}{\det(\Lambda_{W_i})} - \frac{j}{g_i} \right\rfloor \right)^2. \end{aligned}$$

■

Initialization. Send the initial weights as the outgoing messages from symbol nodes through all edges of the Tanner graph,

$$M_{s_i \rightarrow ch_k}(j) = w_{\mathbf{x}}^{(i)}(j), \quad M_{ch_k \rightarrow s_i}(j) = 0,$$

for $i = 1, 2, \dots, n$, $j = 0, 1, \dots, g_i - 1$ and $k = 1, \dots, m$ where m is the number of check equations of the lattice Λ .

5.3.2 Hard Decision

For every symbol i , we define the *final weights* functions

$$\mu_{\mathbf{x}}^{(i)} : \mathbb{Z}_{g_i} \longrightarrow \mathbb{R}$$

by adding all incoming messages together plus the initial (original) weights

$$\mu_{\mathbf{x}}^{(i)}(j) = w_{\mathbf{x}}^{(i)}(j) + \sum_{k_i \in \mathcal{N}_{s_i}} M_{ch_{k_i} \rightarrow s_i}(j), \quad j \in \{0, 1, \dots, g_i - 1\},$$

where $\mathcal{N}_{s_i}$ denotes the set of all neighbors of the symbol node i .

The Min-Sum algorithm only computes the final weights and no decision is made. In the case of the lattice decoding problem we also want to find a valid codeword (configuration) that minimizes the total weight $W_{\mathbf{x}}(\mathbf{c})$. Therefore in each iteration we select c_i such that

$$\mu_{\mathbf{x}}^{(i)}(c_i) = \min_{0 \leq j \leq g_i - 1} \mu_{\mathbf{x}}^{(i)}(j), \quad i = 1, 2, \dots, n.$$

We observe that if for another c'_i , $\mu_{\mathbf{x}}^{(i)}(c'_i) = \mu_{\mathbf{x}}^{(i)}(c_i)$ then the selection between c_i or c'_i is done at random.

If the selected codeword $\mathbf{c} = (c_1, \dots, c_n)$ is a valid codeword, i.e., $\mathbf{c}$ is satisfied in all parity check equations, then the iteration stops. In this case

$$d_i = c_i \frac{\det(P_{W_i})}{\det(\Lambda_{W_i})} + \left[\frac{x_i}{\det(\Lambda_{W_i})} - c_i \frac{\det(P_{W_i})}{\det(\Lambda_{W_i})} \right]$$

and therefore

$$\mathbf{v} = \sum_{i=1}^n d_i \mathbf{v}_i = \sum_{i=1}^n d_i \det(\Lambda_{W_i}) \mathbf{e}_i$$

is the desired lattice vector.

5.3.3 Symbol node operation

For every $i = 1, \dots, n$, we denote d_{s_i} to be the degree of the symbol node s_i , i.e., the number of edges that are connected to symbol node i . Let $\mathcal{N}_{s_i} :=$

$\{i_1, i_2, \dots, i_{d_{s_i}}\}$ be the set all check nodes that are connected to the symbol node i in the corresponding Tanner graph. Also let $\mathcal{N}_{s_i} - ch_{i_k}$ be the set of all neighbors of the symbol node i excluding the one coming from check node i_k . Then the outgoing messages (updating rules) from symbol node s_i are defined by

$$M_{s_i \rightarrow ch_{i_k}}(j) = w_{\mathbf{x}}^{(i)}(j) + \sum_{i_l \in \mathcal{N}_{s_i} - ch_{i_k}} M_{ch_{i_l} \rightarrow s_i}(j)$$

for $j = 0, 1, \dots, g_i - 1$. The message sent from symbol node s_i to check node ch_k can be considered as a vector $M_{s_i \rightarrow ch_{i_k}}$ with g_i components $M_{s_i \rightarrow ch_{i_k}}(j)$

$$M_{s_i \rightarrow ch_{i_k}} = (M_{s_i \rightarrow ch_{i_k}}(0), \dots, M_{s_i \rightarrow ch_{i_k}}(g_i - 1)).$$

The message passing $M_{s_i \rightarrow ch_{i_k}}$ is in fact a function from $\mathbb{Z}_{g_i}$ to $\mathbb{R}$ called the *intermediate weight function*. We note that in the first symbol node operation, all incoming messages from check nodes to symbol nodes are set to zero, i.e.,

$$M_{s_i \rightarrow ch_{i_k}} = w_{\mathbf{x}}^{(i)}(j), \quad i = 1, \dots, n, \quad \text{and} \quad j = 0, \dots, g_i - 1.$$

Figure 5.6

5.3.4 Check node operation

For every $k = 1, \dots, m$ we denote d_{c_k} to be the degree of the check node k . Let $\mathcal{N}_{ch_k} := \{k_1, \dots, k_{d_{c_k}}\}$ be the set of all symbol nodes that are connected to the check node k in the Tanner graph. Also let $\mathcal{N}_{ch_k} - l$ be the set of all neighbors of the check node k excluding the one coming from symbol node l .

Then the outgoing messages (updating rule) from check node k are defined by

$$M_{ch_k \rightarrow s_{k_i}}(c_{k_i}) = \min \sum_{k_l \in \mathcal{N}_{ch_k} - k_i} M_{s_{k_l} \rightarrow ch_k}(c_{k_l}), \quad i = 1, \dots, d_{c_k}$$

where the minimum is taken over all locally valid configurations that satisfy the k th check equation, i.e., all possible values $c_{k_l} \in \mathbb{Z}_{g_{k_l}}$, $l = 1, \dots, d_{c_k}$, $l \neq i$, such that the codewords

$$(c_{k_1}, \dots, c_{k_{d_{c_k}}}) \in \mathbb{Z}_{g_{k_1}} \times \dots \times \mathbb{Z}_{g_{k_{d_{c_k}}}}$$

are satisfied in the k th check equation.

The message sent from check node ch_k to symbol node s_i can be considered as a vector with g_i components $M_{ch_k \rightarrow s_i}(j)$, $j \in \mathbb{Z}_{g_i}$, i.e.,

$$M_{ch_k \rightarrow s_i} = (M_{ch_k \rightarrow s_i}(0), \dots, M_{ch_k \rightarrow s_i}(g_i - 1)).$$

This message passing is also a function from $\mathbb{Z}_{g_k}$ to $\mathbb{R}$ called an *intermediate weight function*.

Example 5.2. By using Construction A, let

$$\Lambda = \mathbb{Z}^4 \text{diag}(2, 6, 6, 2) + C$$

Figure 5.7

where C is the group code from Example 3.8. It is clear that $\det(P_{W_i}) = 1, i = 1, 2, 3, 4$, and $\det(\Lambda_{W_1}) = \det(\Lambda_{W_4}) = 2$ and $\det(\Lambda_{W_2}) = \det(\Lambda_{W_3}) = 6$. Let the input $\mathbf{x} = (4, 8, 15, 3)$ be given. We decode $\mathbf{x}$ in the following steps.

Step 1. Weight initialization. For the first symbol we have $g_1 = 2$ and $w_{\mathbf{x}}^{(i)}(j)$ is the squared minimum distance between $x_1 = 4$ and the cosets $2\mathbb{Z} + j$, $j = 0, 1$. So $w_{\mathbf{x}}^{(1)}(0) = 0$ and $w_{\mathbf{x}}^{(1)}(1) = 1$. Hence $w_{\mathbf{x}}^{(1)} = (0, 1)$. For the second symbol we have $g_2 = 6$ and $w_{\mathbf{x}}^{(2)}(j)$ is the squared minimum distance between $x_2 = 8$ and the cosets $6\mathbb{Z} + j$, $j = 0, 1, \dots, 5$. Hence $w_{\mathbf{x}}^{(2)} = (4, 1, 0, 1, 4, 9)$.

In the same way we have $w_{\mathbf{x}}^{(3)} = (9, 4, 1, 0, 1, 4)$ and $w_{\mathbf{x}}^{(4)} = (1, 0)$. Now these messages will be passed to check nodes through the edges of the Tanner graph. Thus

$$M_{s_i \rightarrow ch_1} = w_{\mathbf{x}}^{(i)}, i = 1, 2, 3 \text{ and } M_{s_i \rightarrow ch_2} = w_{\mathbf{x}}^{(i)}, i = 3, 4.$$

Step 2. Hard decision. For the first symbol c_1 we have

$$w_{\mathbf{x}}^{(1)}(0) = \min\{w_{\mathbf{x}}^{(1)}(0), w_{\mathbf{x}}^{(1)}(1)\} \text{ so } c_1 = 0.$$

Also

$$w_{\mathbf{x}}^{(2)}(2) = \min\{w_{\mathbf{x}}^{(2)}(0), \dots, w_{\mathbf{x}}^{(2)}(5)\} \text{ so } c_2 = 2.$$

Similarly $c_3 = 3$ and $c_4 = 1$. However the vector $\mathbf{c} = 0231$ does not satisfy the first check equation. Therefore iteration starts.

Step 3. Symbol node operation. In the first symbol node operation (Figure 5.8) the outgoing messages from each symbol node i are

$$M_{s_i \rightarrow ch_{i_k}}(j) = w_{\mathbf{x}}^{(i)}(j), \quad j = 0, 1, \dots, g_i - 1,$$

for $i = 1, \dots, 4$. Therefore

$$M_{s_1 \rightarrow ch_1} = (0, 1), \quad M_{s_2 \rightarrow ch_1} = (4, 1, 0, 1, 4, 9)$$

$$M_{s_3 \rightarrow ch_1} = M_{s_3 \rightarrow ch_2} = (9, 4, 1, 0, 1, 4), \quad M_{s_4 \rightarrow ch_2} = (1, 0).$$

Figure 5.8: Step 3

Step 4. Check node operation. The codewords

$$\{000, 003, 022, 025, 041, 044, 111, 114, 130, 133, 152, 155\}$$

satisfy the first check equation $3c_1 + c_2 + 2c_3 \equiv 0 \pmod{6}$. So the message passing form the first check node can be calculated as follows

$$M_{ch_1 \rightarrow s_1}(0) = \min\{w_{\mathbf{x}}^{(2)}(0) + w_{\mathbf{x}}^{(3)}(0), w_{\mathbf{x}}^{(2)}(0) + w_{\mathbf{x}}^{(3)}(3), w_{\mathbf{x}}^{(2)}(2) + w_{\mathbf{x}}^{(3)}(2), w_{\mathbf{x}}^{(2)}(0) + w_{\mathbf{x}}^{(3)}(5), w_{\mathbf{x}}^{(2)}(4) + w_{\mathbf{x}}^{(3)}(1), w_{\mathbf{x}}^{(2)}(4) + w_{\mathbf{x}}^{(3)}(4)\} = 1.$$

Similarly

$$M_{ch_1 \rightarrow s_1}(1) = \min\{w_{\mathbf{x}}^{(2)}(1) + w_{\mathbf{x}}^{(3)}(1), w_{\mathbf{x}}^{(2)}(1) + w_{\mathbf{x}}^{(3)}(4), w_{\mathbf{x}}^{(2)}(3) + w_{\mathbf{x}}^{(3)}(0), w_{\mathbf{x}}^{(2)}(3) + w_{\mathbf{x}}^{(3)}(3), w_{\mathbf{x}}^{(2)}(5) + w_{\mathbf{x}}^{(3)}(2), w_{\mathbf{x}}^{(2)}(5) + w_{\mathbf{x}}^{(3)}(5)\} = 1.$$

Hence $M_{ch_1 \rightarrow s_1} = (1, 1)$.

The codewords 000 and 003 are the only codewords, with 0 in the second symbol, that satisfy the first check equation. So

$$M_{ch_1 \rightarrow s_2}(0) = \min\{w_{\mathbf{x}}^{(1)}(0) + w_{\mathbf{x}}^{(3)}(0), w_{\mathbf{x}}^{(1)}(0) + w_{\mathbf{x}}^{(3)}(3)\} = 0$$

Similarly

$$M_{ch_1 \rightarrow s_2}(1) = \min\{w_{\mathbf{x}}^{(1)}(1) + w_{\mathbf{x}}^{(3)}(1), w_{\mathbf{x}}^{(1)}(1) + w_{\mathbf{x}}^{(3)}(4)\} = 2,$$

$$M_{ch_1 \rightarrow s_2}(2) = \min\{w_{\mathbf{x}}^{(1)}(0) + w_{\mathbf{x}}^{(3)}(2), w_{\mathbf{x}}^{(1)}(0) + w_{\mathbf{x}}^{(3)}(5)\} = 1,$$

$$M_{ch_1 \rightarrow s_2}(3) = \min\{w_{\mathbf{x}}^{(1)}(1) + w_{\mathbf{x}}^{(3)}(0), w_{\mathbf{x}}^{(1)}(1) + w_{\mathbf{x}}^{(3)}(3)\} = 1,$$

$$M_{ch_1 \rightarrow s_2}(4) = \min\{w_{\mathbf{x}}^{(1)}(0) + w_{\mathbf{x}}^{(3)}(1), w_{\mathbf{x}}^{(1)}(0) + w_{\mathbf{x}}^{(3)}(4)\} = 1,$$

$$M_{ch_1 \rightarrow s_2}(5) = \min\{w_{\mathbf{x}}^{(1)}(1) + w_{\mathbf{x}}^{(3)}(2), w_{\mathbf{x}}^{(1)}(1) + w_{\mathbf{x}}^{(3)}(5)\} = 2.$$

Therefore $M_{ch_1 \rightarrow s_2} = (0, 2, 1, 1, 1, 2)$. Similarly $M_{ch_1 \rightarrow s_3} = (2, 2, 0, 2, 2, 0)$.

The only codewords that satisfy the second check equation $c_3 + c_4 \equiv 0 \pmod{2}$, are

$$\{00, 20, 40, 11, 31, 51\}.$$

Therefore

$$M_{ch_2 \rightarrow s_3}(0) = w_{\mathbf{x}}^{(4)}(0) = 1, \quad M_{ch_2 \rightarrow s_3}(1) = w_{\mathbf{x}}^{(4)}(1) = 0$$

$$\begin{aligned} M_{ch_2 \rightarrow s_3} &= (w_{\mathbf{x}}^{(4)}(0), w_{\mathbf{x}}^{(4)}(1), w_{\mathbf{x}}^{(4)}(0), w_{\mathbf{x}}^{(4)}(0), w_{\mathbf{x}}^{(4)}(0), w_{\mathbf{x}}^{(4)}(5)) \\ &= (1, 0, 1, 0, 1, 0) \end{aligned}$$

Similarly $M_{ch_2 \rightarrow s_4} = (1, 0)$, see Figure 5.9.

Figure 5.9: Step 4

Step 5. Hard decision. By adding all incoming messages to the initial weights, for each symbol, we have

$$\begin{aligned} \mu_{\mathbf{x}}^{(1)} &= (1, 0) + (1, 1) = (0, 2) \\ \mu_{\mathbf{x}}^{(2)} &= (4, 1, 0, 1, 4, 9) + (0, 2, 1, 1, 1, 0) = (4, 3, 1, 2, 5, 9) \\ \mu_{\mathbf{x}}^{(3)} &= (9, 4, 1, 0, 1, 4) + (2, 2, 0, 2, 2, 0) = (12, 6, 2, 1, 5, 4) \\ \mu_{\mathbf{x}}^{(4)} &= (1, 0) + (1, 0) = (2, 0). \end{aligned}$$

By doing hard decision we have $\mathbf{c} = (0, 2, 3, 1)$ which is not a valid codeword.

Step 6. Second iteration, symbol node operation. Since symbol nodes 1, 2 and 4 have degree one then

$$M_{s_i \rightarrow ch_1} = w_{\mathbf{x}}^{(i)}, \quad i = 1, 2, 4.$$

For the third symbol the outgoing messages are

$$M_{s_3 \rightarrow ch_1} = w_{\mathbf{x}}^{(3)} + M_{ch_2 \rightarrow s_3}, \quad M_{s_3 \rightarrow ch_2} = w_{\mathbf{x}}^{(3)} + M_{ch_1 \rightarrow s_3}.$$

Therefore the outgoing messages are

$$\begin{aligned} M_{s_1 \rightarrow ch_1} &= (0, 1) & M_{s_2 \rightarrow ch_1} &= (4, 1, 0, 4, 9) \\ M_{s_3 \rightarrow ch_1} &= (10, 4, 2, 0, 2, 4) & M_{s_3 \rightarrow ch_2} &= (11, 6, 1, 2, 3, 4) \\ M_{s_4 \rightarrow ch_2} &= (1, 0). \end{aligned}$$

Figure 5.10: Step 6

Step 7. Check node operation. By doing the same operations as in step 4, the outgoing messages from the check nodes are

$$\begin{aligned} M_{ch_1 \rightarrow s_1} &= (2, 1), & M_{ch_1 \rightarrow s_2} &= (0, 3, 2, 1, 2, 3) \\ M_{ch_1 \rightarrow s_3} &= (2, 2, 0, 2, 2, 0), & M_{ch_2 \rightarrow s_3} &= (1, 0, 1, 0, 1, 2) \\ M_{ch_2 \rightarrow s_4} &= (1, 2). \end{aligned}$$

Step 8. Hard decision. By adding all incoming messages to the initial weights we have

$$\mu_{\mathbf{x}}^{(1)} = (2, 2), \quad \mu_{\mathbf{x}}^{(2)} = (4, 4, 2, 2, 6, 12), \quad \mu_{\mathbf{x}}^{(3)} = (12, 6, 2, 2, 4, 4) \quad \mu_{\mathbf{x}}^{(4)} = (2, 2).$$

After hard decision we get $\mathbf{c} = (0331)$, which is a valid codeword. Therefore

$$d_1 = 2, d_2 = \frac{3}{2}, d_3 = \frac{15}{6}, d_4 = \frac{3}{2}.$$

Hence $\mathbf{v} = (4, 9, 15, 3)$ is the decoded lattice vector.

We note that the selection of c_1, c_2, c_3 and c_4 were at random between $\{0, 1\}$, $\{2, 3\}$, $\{2, 3\}$ and $\{0, 1\}$ respectively.

5.4 Decoding Complexity

Hard Decision. In each hard decision and at each symbol node i , having label group size g_i , the number of comparisons is $g_i - 1$. Therefore the total number of comparisons in each hard decision is

$$\sum_{i=1}^n (g_i - 1). \quad (5.2)$$

Symbol node operation. For each symbol node i and for each outgoing edge i_k , we do $g_i(d_{s_i} - 1)$ additions. There are d_{s_i} outgoing edges from each symbol node i and total of n symbol nodes, then the number of additions required is

$$\sum_{i=1}^n g_i(d_{s_i} - 1)d_{s_i}. \quad (5.3)$$

Check node operation. For the k th check equation we do, at most, $g_{k_1} \times \cdots \times g_{k_{d_{c_k}}}$ comparisons. For each outgoing message $M_{ch_k \rightarrow s_{k_i}}(j)$ and for each j , we do $d_{c_k} - 1$ additions. Since there are d_{c_k} outgoing edges we do at most $(g_{k_1} \times \cdots \times g_{k_{d_{c_k}}})(d_{c_k} - 1)d_{c_k}$ additions at check node k . Hence the number

of additions in each check node operation is

$$\sum_{k=1}^m (g_{k_1} \times \cdots \times g_{k_{d_{c_k}}}) (d_{c_k} - 1) d_{c_k}. \quad (5.4)$$

Equations (5.2), (5.3) and (5.4) shows how decoding complexity depends on the size of label groups, the check degrees, the symbol degrees and the number of edges in the Tanner graph. In each iteration the total number of operations (including additions and comparisons) is at most

$$\sum_{i=1}^n g_i (d_{s_i} - 1) d_{s_i} + \sum_{k=1}^m (g_{k_1} \times \cdots \times g_{k_{d_{c_k}}}) (d_{c_k} - 1) d_{c_k} + \sum_{i=1}^n g_i - n. \quad (5.5)$$

We denote as N_{dec} the decoding complexity per iteration.

Corollary 5.3. (*Bounds on the decoding complexity.*) *Let Λ^* be the dual of Λ and $\gamma = (\gamma(\Lambda)\gamma(\Lambda^*))^{1/2}$. Assume there exist g , d_s , $d_s^{\max}$, d_c and $d_c^{\max}$ for which*

$$g_i \leq g, \text{ and } d_s \leq d_{s_i} \leq d_s^{\max}, i = 1, \dots, n,$$

$$d_c \leq d_{c_k} \leq d_c^{\max}, k = 1, \dots, m.$$

Then N_{dec} is bounded below by

$$\gamma d_s (d_s - 1) n + \gamma^{d_c} d_c (d_c - 1) m + (\gamma - 1) n,$$

and bounded above by

$$g d_s^{\max} (d_s^{\max} - 1) n + g^{d_c^{\max}} (d_c^{\max} - 1) d_c^{\max} m + (g - 1) n.$$

Proof. The proof is the direct result of Equation (5.5) and the fact that $g_i \geq \gamma$, which has been shown in [2]. ■

In the following, we give the exact number of comparisons which is required in each check node operation for lattices constructed based on Construction D' .

Lemma 5.4. *Assume the matrix H be as in Equation (4.10). Let g_i , $i = 1, \dots, n$ be the sizes of the label groups of the lattice Λ constructed using Construction D' . Then the number of solutions of the congruence*

$$x_{k_1} + \dots + x_{k_{d_{c_k}}} \equiv 0 \pmod{\frac{2^{a+1}}{2^j}}, \quad (5.6)$$

where $(x_{k_1}, \dots, x_{k_{d_{c_k}}}) \in \mathbb{Z}_{k_1} \times \dots \times \mathbb{Z}_{k_{d_{c_k}}}$ is

$$\frac{2^j}{2^{a+1}}(g_{k_1} \times \dots \times g_{k_{d_{c_k}}}).$$

Proof. Let $A = 2^{a+1}$, and consider the congruence

$$x_{k_1} + \dots + x_{k_{d_{c_k}}} \equiv 0 \pmod{A}, \quad (5.7)$$

where $(x_{k_1}, \dots, x_{k_{d_{c_k}}}) \in \mathbb{Z}_A \times \dots \times \mathbb{Z}_A$. By induction on d_{c_k} it can be seen that the number of all possible solutions of Equation (5.7) is $A^{d_{c_k}-1}$. Replacing $\mathbb{Z}_A$ by $\mathbb{Z}_{A/2}$, results in halving the number of solutions of the Equation (5.7). Replacing “mod A ” by “mod $A/2$ ”, results in doubling the number of solutions of the Equation (5.7). Therefore by Corollary 4.14 the number of all possible solutions of the congruence in Equation (5.6) is

$$\frac{1}{2^{g(k_1)+\dots+g(k_{d_{c_k}})}} A^{(d_{c_k}-1)2^j},$$

where $g(k_i)$ is as in Equation (4.15). This completes the proof. ■

Corollary 5.5. *Let Λ be the lattice constructed using Construction D' by using the congruence equations (4.9). Then the number of operations (addition or comparison) required in each symbol node and check node operation is*

$$\frac{1}{2^{a+1}} \sum_{j=0}^a 2^j \sum_{k=r_{j-1}+1}^{r_j} (g_{k_1} \times \cdots \times g_{k_{d_{c_k}}}) d_{c_k} (d_{c_k} - 1) + \sum_{i=1}^n g_i (d_{s_i} - 1) d_{s_i}. \quad (5.8)$$

Proof. The proof is the direct result of Equation (5.3) and the previous lemma. ■

Corollary 5.6. *Let Λ be the lattice constructed using Construction D' and the matrix H be as in Equation (4.10). Assume that C_0 has minimum distance at least 2 and let $\det(P_{W_i}) = 1$, $i = 1, \dots, n$. Then the decoding complexity in each iteration is*

$$\frac{1}{2^{a+1}} \sum_{j=0}^a 2^j \sum_{k=r_{j-1}+1}^{r_j} (2^{a+1})^{d_{c_k}} d_{c_k} (d_{c_k} - 1) + 2^{a+1} \sum_{i=1}^n d_{s_i} (d_{s_i} - 1) + n 2^{a+1} - n.$$

In particular if H is regular with d_c “ones” in each row and d_s “ones” in each column, then the decoding complexity is

$$2^{(a+1)(d_c-1)} d_c (d_c - 1) \sum_{j=0}^a 2^j (r_j - r_{j-1}) + n 2^{a+1} (d_s (d_s - 1) + 1) - n.$$

Proof. By Corollary 4.15, $\det(\Lambda_{W_i}) = 2^{a+1}$. Using the fact that $g_i = \frac{\det(\Lambda_{W_i})}{\det(P_{W_i})}$, we have $g_i = 2^{a+1}$. ■

The previous corollary shows that to have lattices with low decoding complexity, it is natural to use LDPC codes as the underlying nested codes in Construction D' .

5.5 Convergence of Generalized Min-Sum algorithm

Let L be a group code, and let $N = \{1, 2, \dots, n\}$ and $K = \{k_1, \dots, k_l\} \subseteq N$. For any $\mathbf{c} \in L$ we select all symbols (coordinates) of $\mathbf{c}$ whose indexes belong to K and denote it by $\mathbf{c}_K$. In fact $\mathbf{c}_K = (c_{k_i})_{k_i \in K}$. We also write $L_K = \{\mathbf{c}_K : \mathbf{c} \in L\}$.

Theorem 5.7. *For a given input $\mathbf{x} \in \mathbb{R}^n$ and a lattice Λ with the label code L assume that the corresponding Tanner graph structure is finite and cycle free. Then the weight functions converge after finitely many iterations of the Generalized Min-Sum algorithm and the final weight functions become*

$$\mu_{\mathbf{x}}^{(i)}(j) = \min_{\mathbf{c} \in L: c_i=j} W(\mathbf{c}). \quad (5.9)$$

Proof. For every $\mathbf{c}_K \in L_K$ we define

$$W_K(\mathbf{c}_K) = \sum_{i \in K} w_{\mathbf{x}}^{(i)}(c_i)$$

where $w_{\mathbf{x}}^{(i)}(c_i)$ are as in Lemma 5.1.

The idea of the proof is to start with the claimed expression in Equation (5.9) for the final weights $\mu_{\mathbf{x}}^{(i)}$. Then, we break down this expression recursively, working backwards through the algorithm, until only the initial weight functions $w_{\mathbf{x}}^{(i)}$ remain. We follow this process for the specific Tanner graph of Fig 5.11.

Final weight functions $\mu_{\mathbf{x}}^{(i)}$

We start with the final weight $\mu_{\mathbf{x}}^{(i)}(j)$ for some value $j \in \mathbb{Z}_{g_i}$ at the symbol

Figure 5.11: Final weight functions

node s_1 . Let $m^{(1)}(j)$ be the claimed value for the final weight, i.e., $m^{(1)}(j) = \min_{\mathbf{c} \in L: c_1=j} W(\mathbf{c})$. Then we have

$$\begin{aligned} m^{(1)}(j) &= \min_{\mathbf{c} \in L: c_1=j} W(\mathbf{c}) = \min_{\mathbf{c} \in L: c_1=j} \{w^{(1)}(j) + W_{K_1}(\mathbf{c}_{K_1}) + W_{K_2}(\mathbf{c}_{K_2}) + W_{K_3}(\mathbf{c}_{K_3})\} \\ &= w^{(1)}(j) + \min_{\mathbf{c}_{K_1 \cup \{1\}} \in L_{K_1 \cup \{1\}}: c_1=j} W_{K_1}(\mathbf{c}_{K_1}) + \min_{\mathbf{c}_{K_2 \cup \{2\}} \in L_{K_2 \cup \{2\}}: c_1=j} W_{K_2}(\mathbf{c}_{K_2}) \\ &\quad + \min_{\mathbf{c}_{K_3 \cup \{3\}} \in L_{K_3 \cup \{3\}}: c_1=j} W_{K_3}(\mathbf{c}_{K_3}). \end{aligned}$$

Define $m_{ch_l \rightarrow s_1}(j) = \min_{\mathbf{c}_{K_l \cup \{1\}} \in L_{K_l \cup \{1\}}: c_1=j} W_{K_l}(\mathbf{c}_{K_l})$, for $l = 1, 2, 3$, then

$$m^{(1)}(j) = w_x^{(i)}(j) + m_{ch_1 \rightarrow s_1}(j) + m_{ch_2 \rightarrow s_1}(j) + m_{ch_3 \rightarrow s_1}(j).$$

The last expression has the same structure as the updating rule for the final weights, therefore the final weight function is equal to $m^{(1)}(j)$ if

$$M_{ch_l \rightarrow s_1}(j) = m_{ch_l \rightarrow s_1}(j), \text{ for } l = 1, 2, 3.$$

Due to symmetry it is sufficient to consider $M_{ch_l \rightarrow s_1}(j)$.

The check to symbol functions $M_{ch \rightarrow s}(j)$

Consider Figure 5.12, we have

$$\begin{aligned}
m_{ch_1 \rightarrow s_1}(j) &= \min_{\mathbf{c}_{K_1 \cup \{1\}} \in L_{K_1 \cup \{1\}}: c_1=j} W_{K_1}(\mathbf{c}_{K_1}) \\
&= \min_{\mathbf{c}_{K_1 \cup \{1\}} \in L_{K_1 \cup \{1\}}: c_1=j} \{W_{K_2}(\mathbf{c}_{K_2}) + W_{K_3}(\mathbf{c}_{K_3})\} \\
&= \min_{\mathbf{c}_{ch_1} \in L_{ch_1}: c_1=j} \left\{ \min_{\mathbf{c}'_{K_2} \in L_{K_2}: c'_2=c_2} W_{K_2}(\mathbf{c}'_{K_2}) + \min_{\mathbf{c}'_{K_3} \in L_{K_3}: c'_3=c_3} W_{K_3}(\mathbf{c}'_{K_3}) \right\} \\
&= \min_{\mathbf{c}_{ch_1} \in L_{ch_1}: c_1=j} \{m_{s_2 \rightarrow ch_1}(c_2) + m_{s_2 \rightarrow ch_1}(c_3)\},
\end{aligned}$$

where L_{ch_l} denotes the set of all locally valid codewords that satisfy in the l th check equation.

Again this expression has the same structure as the updating rule for $M_{ch_1 \rightarrow s_1}$. To prove that this updating rule computes $m_{ch_1 \rightarrow s_1}$ we have to show that $M_{s_l \rightarrow ch_1} = m_{s_l \rightarrow ch_1}$, $l = 1, 2$. Due to symmetry, it is enough to consider the special case $M_{s_2 \rightarrow ch_1}$.

The symbol to check node functions $M_{s \rightarrow ch}$

Consider Figure 5.13, we have

$$\begin{aligned}
m_{s_2 \rightarrow ch_1}(j) &= \min_{\mathbf{c}_{K_2} \in L_{K_2}: c_2=j} W_{K_2}(\mathbf{c}_{K_2}) \\
&\quad \min_{\mathbf{c}_{K_2} \in L_{K_2}: c_2=j} \{w_{\mathbf{x}}^{(i)}(j) + W_{K_4}(\mathbf{c}_{K_4}) + W_{K_5}(\mathbf{c}_{K_5})\} \\
&= w_{\mathbf{x}}^{(i)}(j) + \min_{\mathbf{c}_{K_4 \cup \{1\}} \in L_{K_4 \cup \{1\}}: c_2=j} W_{K_4}(\mathbf{c}_{K_5}) + \min_{\mathbf{c}_{K_5 \cup \{2\}} \in L_{K_5 \cup \{2\}}: c_2=j} W_{K_5}(\mathbf{c}_{K_5}) \\
&= w_{\mathbf{x}}^{(i)}(j) + m_{ch_4 \rightarrow s_2}(j) + m_{ch_3 \rightarrow s_2}(j).
\end{aligned}$$

Figure 5.12: Check to symbol fuctions

The last expression again has the same structure as the updating rule for $M_{s_2 \rightarrow ch_1}$ and we need to show that

$$M_{ch_3 \rightarrow s_2}(j) = m_{ch_3 \rightarrow s_2}(j) \quad \text{and} \quad M_{ch_4 \rightarrow s_2}(j) = m_{ch_4 \rightarrow s_2}(j).$$

To prove this we have to go back to the previous step.

This process is repeated recursively until the leaf symbols are reached, i.e., those symbols belonging to a single check nodes. However we observe that the very first updating step computes the desired expression $m_{s \rightarrow ch}$

$$m_{s_l \rightarrow ch_p} = \min_{c_l \in L_{\{l\}}} W_{\{l\}}(c_l) = w_{\mathbf{x}}^{(l)}(c_l).$$

■

Figure 5.13: Symbol to check functions

Chapter 6

Construction and Performance

Analysis of the decoding complexity of the Generalized Min-Sum algorithm showed that the class of LDPC codes can be the best option for the underlying class of nested codes of the lattice Λ where using Construction D' . Theorem 4.7 relates the performance of Λ to the performance of these codes which is very good in high dimensions [38, 40].

On the other hand in the last section we saw that cycle free Tanner graphs provide optimal decoding. Thus it is natural to try to minimize the influence of cycles in the iterative decoding process. This approach is taken for LDPC codes in [26]. By making the cycles large enough the algorithm may run several iterations without being affected by them.

In graph theory, the girth g refers to the length of the shortest cycle in the graph. An efficient method for constructing a Tanner graph having a large girth by progressively establishing edges between symbol and check nodes in an edge-by-edge manner is given in [30, 32]. This method is called the

Progressive Edge-Growth (PEG) algorithm.

In this chapter we extend the PEG algorithm to generate Tanner graphs with a multilevel incidence matrix H , and we call it the Extended PEG algorithm (E-PEG). Then each level of the matrix H is multiplied by appropriate powers of 2 as in (4.10) to define the corresponding lattice.

In Section 6.2 we introduce a class of lattices constructed using this method. Simulation results show that the 2-level lattices have better performance than 1-level lattices (Construction A). Also we have studied how the performance of this class of lattices improves when we increase the dimension.

6.1 Extended Progressive Edge-Growth Algorithm

Assume that we would like to construct an $a + 1$ -level lattice Λ in dimension n . Let H be the corresponding parity check matrix as in Equation (4.10) and $C_0 \supseteq C_1 \supseteq \dots \supseteq C_a$ be the underlying nested codes where $C_\ell = [n, r_\ell]$, $\ell = 0, 1, \dots, a$, and $m = r_a$.

The Tanner graph corresponding to H is called an $(a + 1)$ -level *Tanner Graph*. This Tanner graph was denoted before by (V, E) such that $V = V_c \cup V_s$, where $V_c = \{c_1, \dots, c_m\}$ and $V_s = \{s_1, \dots, s_n\}$. We define V_c^ℓ as the set of all r_ℓ check nodes of C_ℓ , i.e.

$$V_c^\ell = \{c_1, \dots, c_{r_\ell}\}, \ell = 0, 1, \dots, a.$$

It is clear that $V_c^\ell \subseteq V_c^{\ell+1}$ and $V_c^a = V_c$. For any $\ell = 0, 1, \dots, a$, $d_{s_j}^\ell$ will

denote the degree of the symbol node s_j in the parity check matrix of C_ℓ , i.e., the number of “ones” in the j th column of the parity check matrix of C_ℓ (the submatrix of H which contains the first r_ℓ rows of H). We put $d_{s_j} = d_{s_j}^a$, $j = 1, \dots, n$, and, as in Section 5.3.4, use d_{c_i} to denote the degree of the check node c_i , that is the number of edges incidence to c_i , $i = 1, \dots, m$.

Definition 6.1. *An $(a + 1)$ -level Tanner graph is called (d_s, d_c) -($a + 1$)-regular if*

(i) *for each $\ell = 0, 1, \dots, a$ there is a constant d_s^ℓ such that $d_{s_j}^\ell = d_s^\ell$ for $j = 1, \dots, n$, and $d_s^a = d_s$;*

(ii) $d_{c_i} = d_c$ for $i = 1, \dots, m$.

A (d_s, d_c) -1-regular Tanner graph is simply called (d_s, d_c) -regular. A Tanner graph which is not regular is called irregular.

Let the set of edges E be partitioned in terms of V_s as $E = E_{s_1} \cup \dots \cup E_{s_n}$ with E_{s_j} containing all edges incident on the symbol node s_j . Denote the k th edge incident on s_j by $E_{s_j}^k$, $1 \leq k \leq d_{s_j}$.

For a given symbol node s_j , define its *neighbor within depth p at level ℓ* , $\mathcal{N}_{s_j}^{p,\ell}$, to be the set consisting of all check nodes in V_c^ℓ reached by a tree spreading from the symbol node s_j within depth p , as shown in Figure 6.1. Its complementary set $\overline{\mathcal{N}}_{s_j}^{p,\ell}$ is defined as $V_c^\ell - \mathcal{N}_{s_j}^{p,\ell}$.

Finally, for each symbol node s_j , the *local girth $g_{s_j}^\ell$ at level ℓ* is defined as the length of the shortest cycle passing through that symbol node at level ℓ . The *girth distribution* at level ℓ is denoted by $(g_{s_1}^\ell, \dots, g_{s_n}^\ell)$, and the *average girth distribution* at level ℓ is defined by

$$g_{ave}^\ell = \frac{\sum_{i=1}^n g_i^\ell}{n}.$$

Figure 6.1

For every ℓ the *girth* of the code C_ℓ is $g^\ell = \min\{g_{s_1}^\ell, \dots, g_{s_n}^\ell\}$, which is the length of the shortest cycle in the Tanner graph of C_ℓ . We put $g_{s_i} = g_{s_i}^a$, and define g as the girth of the Tanner graph of Λ . It is clear that $g = g^a = \min\{g_{s_1}, \dots, g_{s_n}\}$.

In a progressive edge growth manner, the local girth of a symbol node at each level is maximized whenever a new edge is added to that node. Suppose that the construction at the first $(j - 1)$ th symbol nodes has been finished, i.e., edges $E_{s_1} \cup E_{s_2} \cup \dots \cup E_{s_{j-1}}$ have been established. Then for each level ℓ , $d_{s_j}^\ell - d_{s_j}^{\ell-1}$ edges of E_{s_j} are added to the current graph on an edge-by-edge basis and the length of the shortest cycle passing through the symbol node s_j is maximized whenever a new edge originating in s_j is being added. In fact at each level ℓ , the tree originating in symbol node s_j is first expanded to depth p each time a new edge of E_{s_j} is being determined, such that

$$\overline{\mathcal{N}}_{s_j}^{p,\ell} \neq \emptyset \text{ and } \overline{\mathcal{N}}_{s_j}^{p+1,\ell} = \emptyset,$$

or the cardinality of $\mathcal{N}_{s_j}^{p,\ell}$ stops increasing but is smaller than r_ℓ . Then

placing an edge between s_j and a check node c_i selected from $\overline{\mathcal{N}}_{s_j}^{p,\ell}$ for which $r_{\ell-1} + 1 \leq i \leq r_\ell$.

The Extended PEG algorithm is summarized as follows.

```

input  $a, n, r_0, r_1, \dots, r_a$ 
for  $j = 1$  to  $n$ 
.   for  $\ell = 0$  to  $a$ 
.       input  $d_{s_j}^\ell$ 
.   end
end
for  $j = 1$  to  $n$ 
.   for  $\ell = 0$  to  $a$ 
.       for  $k = d_{s_j}^{\ell-1} + 1$  to  $d_{s_j}^\ell$ 
.           if  $k = 1$ 
.                $E_{s_j}^0 \leftarrow \text{edge}(c_i, s_j)$ , where  $c_i$ ,  $r_{\ell-1} \leq i \leq r_\ell$ , is a check node
.               having the lowest check degree under the current graph
.               setting  $E_{s_1} \cup \dots \cup E_{s_{j-1}}$ .
.           else
.               expanding a tree from symbol  $s_j$  to depth  $p$  under the
.               current graph setting such that  $\overline{\mathcal{N}}_{s_j}^{p+1,\ell} \neq \emptyset$  or the
.               cardinality of  $\mathcal{N}_{s_j}^{p,\ell}$  stops increasing but is less than
.                $r_\ell$ , then  $E_{s_j}^k \leftarrow \text{edge}(c_i, s_j)$  where  $c_i$ ,  $r_{\ell-1} \leq i \leq r_\ell$ 
.               is the one check node picked from the set  $\overline{\mathcal{N}}_{s_j}^{p,\ell}$  having the
.               lowest check degree.
.           end
.       end
.   end

```

We note that in the case of the existence of multiple check nodes in $\overline{\mathcal{N}}_{s_j}^{p,\ell}$, we select the check node with lowest degree in the current graph setting. In this case we also select the most left position in order $c_1, c_2, \dots, c_m$.

[illegible][illegible]

The girth of the Tanner graph is 4.

As we saw in the previous example, a class of regular Tanner graphs may be constructed if the input parameters of the algorithm are selected properly. In fact for $\ell = 0, 1, \dots, a$ let

$$d_{s_j}^\ell = d_s^\ell, \quad j = 1, \dots, n.$$

This guarantees the uniformity of the column degrees at each level ℓ . Now select a divisor d_c of n such that $d_c > d_s$ and also select r_ℓ such that

$$nd_s^\ell = r_\ell d_c, \quad \ell = 0, \dots, a.$$

Then, by using the E-PEG algorithm, a regular Tanner graph at each level can be constructed. The method of construction implies that $g^0 \geq g^1 \geq \dots \geq g^a$. Hence by Theorem 3.2 we conclude the following.

Theorem 6.3. *For given n, d_c and $3 \leq d_s^0 < d_s^1 < \dots < d_s^a = d_s$, assume that $d_{s_j}^\ell = d_s^\ell$, $j = 1, \dots, n$, $d_s < d_c$ and d_c is a divisor of n . Let $r_0, \dots, r_a$ be selected such that*

$$r_\ell = \frac{n}{d_c} d_s^\ell, \quad \ell = 0, \dots, a. \quad (6.1)$$

Suppose that the regular $(a+1)$ level Tanner graph (V, E) constructed using the E-PEG algorithm has girth $g > 4$ and let the $(a+1)$ -level matrix H be its incidence matrix. Assume that $C_\ell = [n, r_\ell]$, $\ell = 0, \dots, a$ are the corresponding codes constructed from the first r_ℓ rows of H . Then the minimum distance of the code C^ℓ satisfies

$$d_{\min}^\ell \geq 1 + \frac{d_s^\ell [(d_s^\ell - 1)^{\lfloor (g^\ell - 2)/4 \rfloor} - 1]}{d_s^\ell - 2}. \quad (6.2)$$

Furthermore if $g^\ell/2$ is even, then the lower bound on $d_{\min}^\ell$ can be made tighter

$$d_{\min}^\ell \geq 1 + \frac{d_s^\ell[(d_s^\ell - 1)^{\lfloor (g^\ell - 2)/4 \rfloor} - 1]}{d_s^\ell - 2} + (d_s^\ell - 1)^{\lfloor (g^\ell - 2)/4 \rfloor}. \quad (6.3)$$

Moreover if $d_s^0 = 2$ then $d_{\min}^0 = g^0/2$.

Example 6.4. Let $a = 2, n = 20, r_0 = 8, r_1 = 15, r_2 = 18$ and $d_s^0 = 2, d_s^1 = 3, d_s^2 = 4$ then the E-PEG algorithm gives the following 3-level matrix:

$$\begin{bmatrix} 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . \\ 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . \\ . & 1 & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . \\ . & 1 & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . \\ . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & 1 & . & . & . & 1 & . & . & . \\ . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & 1 & . & . & . & 1 & . & . & . \\ . & . & . & 1 & . & . & 1 & . & . & . & 1 & . & . & 1 & . & . & 1 & . & . & . \\ . & . & . & 1 & . & . & 1 & . & . & 1 & . & . & . & 1 & . & . & 1 & . & . & . \\ * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * \\ 1 & . & . & . & . & . & . & . & . & . & 1 & . & . & . & 1 & . & . & . & . & . \\ . & 1 & . & . & . & . & . & 1 & . & . & . & . & . & . & . & . & 1 & . & . & . \\ . & . & 1 & . & . & . & . & . & 1 & . & . & . & . & . & . & . & . & 1 & . & . \\ . & . & . & 1 & . & . & . & . & 1 & . & . & . & . & . & . & . & 1 & . & . & . \\ . & . & . & . & 1 & . & . & . & . & . & 1 & . & . & . & 1 & . & . & . & . & . \\ . & . & . & . & . & 1 & . & . & . & . & 1 & . & . & . & . & . & . & . & 1 & . \\ . & . & . & . & . & . & 1 & . & . & . & . & . & 1 & . & . & . & . & . & . & . \\ * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * & * \\ 1 & . & . & 1 & . & . & . & 1 & . & . & 1 & . & . & 1 & . & . & 1 & . & . & 1 \\ . & 1 & . & . & . & 1 & . & 1 & . & 1 & . & . & 1 & . & . & 1 & . & . & 1 & . \\ . & . & 1 & . & 1 & 1 & . & . & . & . & 1 & . & . & 1 & . & . & 1 & . & . & . \end{bmatrix}$$

The following theorem gives bounds on the the girth of general graphs.

Theorem 6.5. Let (V, E) be a (d_c, d_s) -regular Tanner graph with m check nodes. Then the girth of this graph is bounded by

$$2(\lfloor t_l \rfloor + 2) \leq g \leq 4\lfloor t_u \rfloor + 2, \quad (6.4)$$

where

$$t_l = \frac{\log(md_c - \frac{md_c}{d_s} - m + 1)}{\log((d_s - 1)(d_c - 1))} - 1 \quad \text{and} \quad t_u = \frac{\log[(m - 1)(1 - \frac{d_s}{d_c(d_s - 1)}) + 1]}{\log[(d_c - 1)(d_s - 1)]}. \quad (6.5)$$

Proof. The proof is based on an idea of Gallager [25] (pp. 80-82) and it can be found in [30]. ■

Corollary 6.6. *For given $d_c, d_s \geq 3$ and b , let $C = [n, m, d_{\min}]$ be a (d_c, d_s) -regular code, with m parity checks, for which the girth g of the Tanner graph of C satisfies*

$$g \geq 4 \left(\frac{\log \left(\frac{(d_s-2)(b-1)}{d_s} + 1 \right)}{\log(d_s - 1)} \right) + 2. \quad (6.6)$$

Then $d_{\min} \geq b$. Moreover we have

$$n_l \leq n \leq n_u, \quad (6.7)$$

where

$$n_l = \left(\frac{[(d_s - 1)(d_c - 1)]^{\frac{g-2}{4}} - 1}{1 - \frac{d_s}{d_c(d_s-1)}} + 1 \right) \frac{d_c}{d_s}, \quad (6.8)$$

and

$$n_u = d_c \left(\frac{[(d_s - 1)(d_c - 1)]^{\frac{g}{2}-1} - 1}{d_s(d_c - 2)} \right). \quad (6.9)$$

Proof. Since g satisfies Equation (6.6) then

$$\lfloor (g-2)/4 \rfloor \geq \frac{\log \left(\frac{(d_s-2)(b-1)}{d_s} + 1 \right)}{\log(d_s - 1)},$$

which is equivalent to

$$1 + \frac{d_s \lfloor (d_s - 1)^{\lfloor (g-2)/4 \rfloor} - 1 \rfloor}{d_s - 2} \geq b.$$

Thus by Theorem 3.2 we have $d_{\min} \geq b$. Let t_u be as in Theorem 6.5, so $\lfloor t_u \rfloor \geq (g-2)/4$, and hence $t_u \geq (g-2)/4$. Using some algebraic manipulations and

Theorem 6.5 we have

$$m \geq \frac{[(d_s - 1)(d_c - 1)]^{\frac{g-2}{4}} - 1}{1 - \frac{d_s}{d_c(d_s-1)}} + 1. \quad (6.10)$$

Since the code is regular $nd_s = md_c$, and this proves the first inequality in (6.7). Let t_l be as in Theorem 6.5. Practically the bound given in this theorem is not tight enough so we can assume that $\lfloor t_l \rfloor < g/2 - 2$. Since the girth is an even number $g/2 - 2$ is an integer and thus $t_l \leq g/2 - 2$. Simple algebraic manipulations complete the proof. ■

Example 6.7. Let $d_s = 3$ and $d_c = 4$. Suppose we want to construct a code $C = [n, m, d_{\min}]$ for which $d_{\min} \geq 10$. Putting $b = 10$ in the Equation (6.6) we get $g \geq 10$. Then substituting $g = 10$ in (6.7), we have $n_i = 76$ and $n_u = 776$. A code with $n = 460$ and $m = 345$ with girth $g = 10$ and average girth $g_{ave} = 10.55$ is constructed.

We note that those bounds given in the last theorem are not tight. For example, we have constructed a code with $n = 400$ and $m = 300$ with the same girth $g = 10$ and $g_{ave} = 10$.

6.2 Construction of regular lattices

A lattice is called *regular* if its parity check matrix H is (d_c, d_s) regular.

Before constructing a class of regular lattices, we introduce the following theorem which provides a practical tool to compute the parameters of a lattice.

Theorem 6.8. *Let H be an $(a + 1)$ -level matrix as in Equation (4.10) for which*

$$d_{c_i} = z_\ell 2^{a-\ell+1}, \quad r_{\ell-1} + 1 \leq i \leq r_\ell, \quad \ell = 0, \dots, a, \quad z_\ell \in \mathbb{Z}. \quad (6.11)$$

For each $j = 1, \dots, n$, let k_j be the smallest number satisfying $r_{\ell-1} + 1 \leq k_j \leq r_\ell$, such that $[H]_{k_j, j} \neq 0$.

Then in the trellis lattice coordinate $W_i = \langle \mathbf{e}_i \rangle, i = 1, \dots, n$, the size of the label group G_i is $g_i = 2^{a-\ell+1}$, the orthogonal projection P_{W_i} onto W_i has determinant 1, and the cross section of the lattice Λ has determinant $\det(\Lambda_{W_i}) = 2^{a-\ell+1}$.

Proof. Let $\mathbf{u} = (1, \dots, 1)$ be the n -dimensional vector whose coordinates are 1. For every $k = 1, \dots, r_a$, let $\mathbf{h}_k$, $r_{\ell-1} + 1 \leq k \leq r_\ell$, be the k th row of H . Then $\mathbf{h}_k$ has $z_\ell 2^{a-\ell+1}$ “ones” and hence $2^\ell \mathbf{h}_k \cdot \mathbf{u} = z_\ell 2^{a+1}$. Therefore $H \cdot \mathbf{u} \equiv 0, \text{ mod } 2^{a+1}$, so $\mathbf{u} \in \Lambda$. In the trellis coordinate system $\langle \mathbf{e}_i \rangle$, $\det(P_{W_i}) = 1$, and Theorem 4.13 completes the proof. ■

Corollary 6.9. *Let H be as in the previous theorem. Moreover assume that $d_{s_j}^0 \geq 1, j = 1, \dots, n$. Then $g_i = \det(\Lambda_{W_i}) = 2^{a+1}$.*

Proof. In the previous theorem, for every $j = 1, \dots, n$, we have $1 \leq k_j \leq r_0$, so in the last theorem we have $\ell = 0$. ■

Example 6.10. *Consider the 3-level matrix*

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & . & . & . & . & . & 2 & . & . & . & . & . & . & . \\ . & . & . & 2 & 2 & 2 & . & . & . & 2 & . & . & . & . & . & . \\ . & . & . & . & . & . & 2 & 2 & . & . & 2 & 2 & . & . & . & . \\ 2 & . & . & . & . & . & . & . & . & . & . & . & 2 & 2 & 2 & . \\ . & 4 & . & . & . & . & . & . & . & . & . & . & . & . & . & 4 \\ . & . & 4 & . & . & . & . & . & . & 4 & . & . & . & . & . & . \\ . & . & . & 4 & . & . & . & . & 4 & . & . & . & . & . & . & . \\ . & . & . & . & 4 & . & . & . & . & 4 & . & . & . & . & . & . \\ . & . & . & . & . & 4 & . & . & . & . & 4 & . & . & . & . & . \\ . & . & . & . & . & . & 4 & . & . & . & . & 4 & . & . & . & . \end{bmatrix}.$$

Here $a = 2$ and $d_{c_i} = 8$, $i = 1, 2$, $d_{c_i} = 4$, $i = 3, 4, 5, 6$ and $d_{c_i} = 2$, $i = 7, \dots, 12$. Moreover $d_{s_j}^0 \geq 1$, $i = 1, \dots, 16$. Therefore $g_i = 2^3 = 8$.

To construct a class of regular $(a+1)$ -level lattices put $z_\ell = 2^\ell$ in Equation (6.11) so $d_c = 2^{a+1}$. Then given $d_s^0 < d_s^1 < \dots < d_s^a < 2^{a+1}$ and using Equation (6.1), we have

$$r_\ell = \frac{d_s^\ell}{2^{a+1}} n, \quad \ell = 0, 1, \dots, a.$$

Hence Corollary 6.9 implies that

$$\det(\Lambda_{W_i}) = g_i = 2^{a+1}, \quad i = 1, \dots, n.$$

Theorem 6.11. Assume that a , $d_c = 2^{a+1}$ and

$$d_s^0 < d_s^1 < \dots < d_s^a < 2^{a+1},$$

are given. Let H be the $(a+1)$ -level matrix with these parameters and Λ be its corresponding lattice. Then for sufficiently large n , Λ is an LDPC lattice with maximum coding gain

$$\gamma(\Lambda) = 4^{a+1 - \frac{1}{2a+1} \sum_{i=0}^a d_s^i}. \quad (6.12)$$

Proof. Since the lattice Λ is regular $r_\ell = \frac{d_s^\ell}{2^{a+1}}n$. By Theorem 4.6 we have

$$(\det(\Lambda))^{\frac{2}{n}} = 4^{\sum_{\ell=0}^a \frac{r_\ell}{n}} = 4^{\frac{1}{2^{a+1}} \sum_{\ell=0}^a d_s^\ell}, \quad (6.13)$$

which is independent of n . On the other hand r_ℓ grows (linearly) with n . Substituting $m = r_\ell$ in Theorem 6.5, then t_ℓ grows (logarithmic) with n , and so does g^ℓ . Thus Theorem 6.3 implies that $d_{\min}^\ell$ grows (linearly) with n . Hence we can assume that $d_{\min}^\ell \geq 4^{\ell+1}$. Now by Corollary 4.8 proof is completed. $\blacksquare$

The relation (6.12) shows that the coding gain of such lattices can grow exponentially with a . However Corollary 5.6 implies that the decoding complexity grows exponentially with a . This will have an important impact on the performance of the Generalized Min-Sum algorithm.

6.3 A class of $(3, 4)$ -2-level lattices: construction and performance

In this section we introduce a class of regular LDPC lattices. In this class $d_s^0 = 2$, $d_s^1 = 3$, and we put $z_0 = 1$ and $z_1 = 2$ in Theorem 6.8. Then $d_c = 4$. Using Equation (6.1), we have $r_0 = \frac{1}{2}n$ and $r_1 = \frac{3}{4}n$. Thus Corollary 6.9 implies that $\det(\Lambda_{W_i}) = g_i = 4$ for $i = 1, \dots, n$. These class of lattices are denoted by L_n^2 .

Equation (6.13) shows that $(\det(L_n^2))^{2/n} = 4^{5/4} = 5.65$. Hence by Corol-

lary 4.8 the coding gain is at least

$$\frac{\min\{4d_{\min}^0, d_{\min}^1, 16\}}{5.65}, \quad (6.14)$$

where $d_{\min}^0$ and $d_{\min}^1$ are the minimum distances of the two underlying codes C_0 and C_1 .

For these lattices Corollary 5.6 implies that the decoding complexity is

$$N_{\text{dec}} = 795n - 768.$$

Definition 6.12. *Let n_i be the number of lattice points (codewords) that are correctly decoded in the i th iteration. Then the average iteration is defined as*

$$I_{\text{ave}} = \frac{\sum_{i=1}^{\max} i \times n_i}{\sum_{i=1}^{\max} n_i}$$

where $\max$ is the maximum number of iterations. The decoding complexity is denoted by N_D and is defined as

$$N_D = I_{\text{ave}} \times N_{\text{dec}}.$$

The notation $\tilde{N}_D$ denotes the decoding complexity per two dimension, i.e.,

$$\tilde{N}_D = \frac{2N_D}{n}.$$

Example 6.13. *As the first example, for $n = 16$, $r_0 = 8$ and $r_1 = 12$, we use the E-PEG algorithm to produce a $(3, 4)$ -2-level lattice with parity check matrix*

$$\begin{bmatrix} 1 & . & . & . & 1 & . & . & . & 1 & . & . & 1 & . & . & . & . \\ 1 & . & . & . & . & 1 & . & . & . & 1 & . & . & 1 & . & . & . \\ . & 1 & . & . & 1 & . & . & . & . & 1 & . & . & . & 1 & . & . \\ . & 1 & . & . & . & . & 1 & . & 1 & . & . & . & . & . & 1 & . \\ . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & 1 & . & . \\ . & . & . & 1 & . & . & . & 1 & . & . & . & 1 & . & . & . & 1 \\ . & . & . & 1 & . & . & . & 1 & . & . & . & . & 1 & . & . & . \\ 1 & . & . & . & . & . & 1 & 1 & . & . & . & . & . & 1 & . & . \\ . & 1 & . & . & . & 1 & . & . & . & . & 1 & 1 & . & . & . & . \\ . & . & 1 & . & 1 & . & . & . & . & . & . & . & 1 & . & 1 & . \\ . & . & . & 1 & . & . & . & . & 1 & 1 & . & . & . & . & . & 1 \end{bmatrix}.$$

The resulting lattice is denoted by L_{16}^2 . The girth distribution of the corresponding Tanner graph is $(6, 6, 4, 4, 6, 4, 4, 6, 6, 6, 4, 6, 6, 4, 4, 4)$ with the average girth $g_{ave} = 5$

Now consider BW_{16} (Example 4.17). The girth distribution of its Tanner graph is $(4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4)$, so the average girth distribution is 4. Some factors that play a role in the decoding algorithm are tabulated in the following table. Note that in the table E and $|G|$ denote the number of edges in the Tanner graph and the label group sizes, respectively.

	g_{ave}	E	$d_c^{\max}$	$d_s^{\max}$	$ G $	$\det$	N_{dec}
BW_{16}	4	80	16	15	4	2^{20}	1.6109×10^{10}
L_{16}^2	5	48	4	3	4	2^{20}	1.3488×10^4

The performance curve of BW_{16} and L_{16}^2 along the upper bound given in Equation (2.17) is provided in Figure 6.2.

Both lattices have the same volume (determinant) which means that they have the same noise at any SNR. The performance of BW_{16} is weaker than

Figure 6.2: Normalized error probabilities of BW_{16} and L^2_{16}

the performance of L^2_{16} . This shows the effect of the factors, in particular d_c , that play an important role in the decoding complexity (Corollary 5.6). Another factor is the Tanner graph structure itself. Although both lattices have the same girth 4, L^2_{16} has higher average girth distribution than BW_{16} .

Example 6.14. Let $n = 256$ then $r_0 = 128$ and $r_1 = 192$. When applying the E-PEG algorithm, the two underlying codes are $C_0 = [256, 128]$ with 128 parity checks and $C_1 = [256, 64]$ with 192 parity checks. The girth of the corresponding Tanner graph is $g = 6$ and $g_{ave} = 7.25$.

SNR	σ	max.itr.	I_{ave}	P_e	P_e^*
1	0.51	23	13.59	9.51×10^{-1}	7.439×10^{-3}
2	0.45	21	10.42	3.41×10^{-1}	2.666×10^{-3}
3	0.40	23	5.88	7.35×10^{-3}	5.754×10^{-5}
3.5	0.38	22	4.53	5.42×10^{-4}	4.234×10^{-6}
4	0.36	17	3.66	7.30×10^{-6}	5.703×10^{-8}
4.21	0.35	18	3.29	1.00×10^{-6}	7.810×10^{-9}

performance of $\mathbf{L}_{256}^2$

Consider the 1-level lattice corresponding to L_{256}^2 , i.e., the lattice constructed from C_0 . Its performance is given in the following table.

SNR	σ	max.itr.	I_{ave}	P_e	P_e^*
1	0.31	35	9.51	5.807×10^{-1}	4.536×10^{-3}
2	0.27	36	4.70	1.209×10^{-1}	9.453×10^{-4}
3	0.24	35	2.30	1.759×10^{-2}	1.375×10^{-4}
3.5	0.23	30	1.78	6.06×10^{-3}	4.735×10^{-5}
4	0.21	35	1.35	2.396×10^{-3}	1.872×10^{-5}
5	0.19	32	1.15	2.46×10^{-4}	1.922×10^{-6}
6	0.17	28	1.11	1.20×10^{-5}	9.375×10^{-8}

performance of $\mathbf{L}_{256}^1$

Comparison of those tables show that the 2-level L_{256}^2 performs better than the 1-level L_{256}^1 . Their curve performances together with the bound on maximum likelihood decoding are given in Figure 6.3.

Figure 6.3: Normalized error probabilities of L_{256}^1 and L_{256}^2

Example 6.15. Back to Example 6.7. In this case $C_0 = [460, 230]$ with $r_0 = 230$ parity checks and $C_1 = [460, 155]$ with $r_1 = 345$ parity checks. For the code C_0 the girth is $g^0 = 10$ and $g_{ave}^0 = 12.44$ and for C_1 , the corresponding Tanner graph has girth $g = 10$ with $g_{ave} = 10.55$. Thus using Theorem 6.3 we have $d_{\min}^0 > 5$ and $d_{\min}^1 > 10$. This implies that the minimal norm of L_{460}^2 is at least 10 and therefore its coding gain is at least 1.9 and at most 2.8. The performance of this lattice is given in the following table.

SNR	σ	max.itr	I_{ave}	P_e	P_e^*
1	0.51	20	14.50	9.87×10^{-1}	4.290×10^{-3}
2	0.45	21	11.55	3.14×10^{-1}	1.360×10^{-3}
2.5	0.43	21	8.25	3.16×10^{-2}	1.380×10^{-4}
3	0.40	21	6.02	6.99×10^{-4}	3.039×10^{-6}
3.5	0.38	19	4.71	5.67×10^{-6}	4.435×10^{-8}
3.66	0.37	17	4.40	1.10×10^{-6}	4.782×10^{-9}

performance of L_{460}^2

For large dimensions we can get a good approximation to the signal-to-noise ratio (Equation (2.10)) by replacing $(n/2)!$ in V_n by $(n/2e)^{n/2}$ using Stirling's approximation. So

$$\text{SNR} = \frac{\det(\Lambda)^{2/n}}{2\pi e \sigma^2}. \quad (6.15)$$

This is compatible with Forney's definition of Volume-to-Noise-Ratio [24].

We have used this approximation to SNR for L_{460}^2 . Its curve performances together with L_{256}^2 is depicted in Figure 6.4.

The last two examples show that if we increase the dimension then the performance of the lattices improves while the decoding complexity grows linearly with the dimension. This suggest the use of high dimensional lattices.

Figure 6.4: Normalized error probabilities of L^2_{256} and L^2_{460}

Example 6.16. In dimensions $n = 800$, $n = 1000$ and $n = 2000$ using *E-PEG* algorithm, the three lattices L^2_{800} , L^2_{1000} and L^2_{2000} are constructed. Their parameters are given in the following table.

n	lattice	r_0, r_1	E	girth	g_{ave}	N_{dec}
800	L^2_{800}	400, 600	2400	10	11.08	635232
1000	L^2_{1000}	500, 750	3000	10	11.94	794232
2000	L^2_{2000}	1000, 1500	6000	12	12.002	1589232

The curve performances of these lattices along the curve performances of the previous lattices are provided in the following figure.

Figure 6.5: Error performances of L^2_{256} , L^2_{460} , L^2_{1000} and L^2_{2000}

In the following table the decoding complexity and the effective coding gain of the class of L^2_n lattices constructed in the previous examples are compared with those given by Forney [17].

lattice	$d_{\min}^2(\Lambda)$	N_D	$\tilde{N}_D$	γ_{eff}
D_4/RD_4V	4	$2^{9.54}$	$2^{6.50}$	3.75
$D_{16}/H_{16}\text{VI}$	4	$2^{10.6}$	$2^{7.6}$	4.04
$\Lambda_{16}/R\Lambda_{16}\text{II}$	16	$2^{14.4}$	$2^{11.3}$	4.61
H_{16}/Λ_{16}	8	2^{19}	2^{16}	5.36
$\Lambda_{16}/R\Lambda_{16}\text{VI}$	16	2^{22}	2^{19}	5.73
L_{256}^2	> 4	$2^{20.3}$	$2^{12.3}$	4.99
L_{460}^2	> 10	$2^{20.6}$	$2^{12.7}$	5.64
L_{800}^2	> 10	$2^{21.6}$	$2^{12.9}$	5.85
L_{1000}^2	> 10	$2^{22.0}$	$2^{13.0}$	6.01
L_{2000}^2	> 14	$2^{23.2}$	$2^{13.2}$	6.23

These results show that in higher dimensions LDPC lattices can perform very well. In these dimensions a gain up to 6.23 dB is achieved.

6.4 Semi-regular lattices: construction

If we put $z_\ell = 1$ in Theorem 6.8 then we have a fewer number of “ones” in the parity check matrix H and can still compute the lattice parameters.

Definition 6.17. *An $(a+1)$ -level matrix H is called semi-regular if*

$$d_{c_i} = 2^{a-\ell+1}, \quad r_{\ell-1} + 1 \leq i \leq r_\ell, \quad \ell = 0, 1, \dots, a.$$

The corresponding lattice is called $(a+1)$ -level semi-regular lattice.

The matrix presented in Example 6.10 is a 3-level semi-regular matrix.

To construct a semi-regular $(a + 1)$ -level matrix H , we use the original PEG algorithm to construct the transpose of H . The resulting algorithm is called *Transpose PEG* (T-PEG for short). This process is summarized as follows.

```

input  $a, n, r_a, d_{c_1}, \dots, d_{c_{r_a}}$ 
for  $j = 1$  to  $r_a$ 
.    $d_{s_j} = d_{c_j}$ 
end
for  $j = 1$  to  $r_a$ 
.   for  $k = 1$  to  $d_{s_j}$ 
.       if  $k = 1$ 
.            $E_{s_j}^0 \leftarrow \text{edge}(c_i, s_j)$ , where  $c_i$  is a check node having the lowest
.           check degree under the current graph setting  $E_{s_1} \cup \dots \cup E_{s_{j-1}}$ .
.       else
.           expanding a tree from symbol  $s_j$  to depth  $p$  under the current
.           graph setting such that  $\overline{\mathcal{N}}_{s_j}^{p+1,a} \neq \emptyset$  or the cardinality of  $\mathcal{N}_{s_j}^{p,a}$ 
.           stops increasing but is less than  $r_a$ , then  $E_{s_j}^k \leftarrow \text{edge}(c_i, s_j)$ 
.           where  $c_i$  is the one check node picked from the set  $\overline{\mathcal{N}}_{s_j}^{p,a}$  having
.           the lowest check degree.
.       end
end

```

Example 6.18. For $n = 16$, $r_0 = 4$, $r_1 = 12$, using T-PEG algorithm with $d_{c_i} = 4$, $i = 1, \dots, 4$ and $d_{c_i} = 2$, $i = 5, \dots, 12$ we have $d_s^0 = 1$, $d_s^1 = 2$. Then H , the transpose of the resulting matrix, with the average girth=10 is

$$\begin{bmatrix} 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & 1 & 1 & 1 & 1 & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & 1 & 1 & 1 & 1 & . \\ 1 & . & . & . & 1 & . & . & . & . & . & . & . & . & . & . & . & . \\ . & 1 & . & . & . & . & . & 1 & . & . & . & . & . & . & . & . & . \\ . & . & 1 & . & . & . & . & . & . & . & . & 1 & . & . & . & . & . \\ . & . & . & 1 & . & 1 & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & 1 & . & . & 1 & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & 1 & . & . & . & . & 1 & . & . & . & . \\ . & . & . & . & . & . & . & . & . & 1 & . & . & . & 1 & . & . & . \\ . & . & . & . & . & . & . & . & . & . & 1 & . & . & . & 1 & . & . \end{bmatrix},$$

The corresponding lattice is denoted by SL_{16}^2 and its performance together with the performance of L_{16}^2 and BW_{16} is given in Figure 6.6.

Figure 6.6: Normalized error probabilities of L_{16}^2 , BW_{16} and SL_{16}^2

Compared to L_{16}^2 , we see an improvement in the performance in this dimension. This suggests the use of semi-regular lattices when we have high girth distribution in the lower dimensions. In this case the symbol node degrees reduce and this allows the decoding algorithm to run more iterations without facing overflow. The iteration bar-chart of this lattice together with L_{16}^2 , at SNR=2, is given in Figure 6.7.

Figure 6.7

However in this dimension we do not have (effective coding) gain. In the following examples we increase the dimension to have gain.

Definition 6.19. For a given dimension n let $r_0 = n/4$, $r_1 = 3n/4$, and

$$d_{c_i} = 4, i = 1, \dots, r_0 \quad \text{and} \quad d_{c_i} = 2, i = r_0 + 1, \dots, r_2.$$

Then by using the T-PEG algorithm the resulting lattice is a 2-level semi-regular lattice and is denoted by SL_n^2 .

Corollary 6.20. The decoding complexity of SL_n^2 using Generalized Min-Sum algorithm is $N_{dec} = 211n$.

Proof. When the T-PEG algorithm starts to run, it fills the first four entries of the first row of the incident matrix H . Then it moves to the second row and to make the girth as large as possible it fills the four successive entries starting from the fifth column. This process will go on until the $n/4$ th row is filled. At this stage the algorithm has filled the last four entries of this row. Thus $d_{s_i}^0 = 1, i = 1, \dots, n$. Hence by Corollary 6.9 the label group sizes $g_i = 4$. The number of remaining rows is $3n/4 - n/4 = n/2$ with $d_c^i = 2$, so $d_s^1 = 2$. Corollary 5.6 completes the proof. ■

Example 6.21. In dimensions $n = 32$ and $n = 64$ using T-PEG algorithm, the two lattices SL_{32}^2 and SL_{64}^2 are constructed. Their parameters are given in the following table.

n	lattice	r_0, r_1	E	girth	g_{ave}	N_{dec}
32	SL_{32}^2	8, 24	64	14	14	6752
64	SL_{64}^2	16, 48	128	14	14	13505

The performance of these lattices are depicted in the following figure. Both lattices have (effective coding) gain. These results show that even in

Figure 6.8: Error probabilities of SL_{32}^2 and SL_{64}^2

low dimensions the class of LDPC lattices can achieve some gains.

In the following table the performances of the last two lattices are compared with some well known lattices given by Forney [17].

lattice	$\tilde{N}_D$	γ_{eff}
D_4/RD_4I	13	2.32
D_4/RD_4II	16	2.69
SL_{32}^2	443	2.53
SL_{64}^2	459	2.69

Chapter 7

Conclusion

This thesis investigates the construction of high coding gain lattices with low decoding complexity using LDPC codes.

Several constructions from codes such as Construction A , B , C , D and D' have been investigated. Among all of them, Construction D' has been chosen as the best. With a slight modification on the modular equations, a set of linear constraints, the lattice can be presented in terms of its parity check matrix, as was done previously for LDPC codes. Theorem 4.7 provides lower and upper bounds on the minimum distance of the lattice in terms of the minimum distance of its underlying codes. Theorem 4.13 and its corollaries provide some useful tools to compute the lattices parameters. Corollary 4.8 relates the coding gain of the lattice to its underlying codes parameters.

The Generalized Min-Sum algorithm can be applied to decode any lattice. The bounds on the decoding complexity, Equation (5.3), show the dependency of the decoding complexity on the lattice parameters such as

label group sizes, coding gain, and check node and symbol node degrees of the Tanner graph. These factors along with the girth distribution of the Tanner graph affect the performance of the lattice under the iterative decoding algorithm. This is why some well-known lattices such as Barnes-Wall lattices do not perform well under this iterative decoding algorithm in spite of having high coding gain. For lattices constructed using Construction D' the exact number of computations along some (tight) bounds are derived. For these lattices the decoding complexity grows linearly with the dimension n but exponentially with check node degrees (row weights of the parity check matrix). This supports the idea of constructing lattices using LDPC codes.

The extended PEG algorithm provides a class of regular LDPC lattices. Given a specific check node and symbol node degree, and a desired minimum distance, the bounds given in Theorem 6.3 provides us an estimation about the input parameters to the E-PEG algorithm. So this algorithm is a useful tool to generate regular LDPC lattices.

The simulation results confirm that in the case of higher SNR the 2-level construction performs better than the 1-level construction and their performances get better when n increases. A class of L_n^2 is constructed and in higher dimensions a gain up to 6.23 dB is achieved.

In lower dimensions a class of Semi-regular lattices, with almost the same gain as some well know lattices, are constructed. However the decoding complexities of this class are larger than the well know lattices.

Ways in which the current work might be extended

The performance of the LDPC lattices depends on the performance of their

underlying LDPC codes. Therefore it would be worthwhile searching for good LDPC codes based on other constructions such as irregular PEG constructions or algebraic constructions of LDPC codes based Reed-Solomon codes.

In order to run the algorithm the label group sizes must be known. The LDPC lattices introduced in this thesis are constructed in such a way that the all-one vector is in the lattice. This enables us to find the label group sizes. In this class, when the number of levels (number of underlying codes) increase (e.g. 3-level or more) then the check node degrees also increases, which implies a higher decoding complexity for the lattice. Therefore investigating higher level LDPC lattices with low check node degrees for which we can find the label group sizes would be another possible continuation of this thesis.

In the Generalized Min-Sum algorithm it is important to handle numerical issues properly. Our practical experiments running the simulations in C^{++} show that perhaps more tuning of the program may help to gain better performance.

We have proved that the class of lattices investigated in this thesis cannot be supported by cycle free Tanner graphs. It would be interesting to prove that cycle free Tanner graphs cannot support any lattice, in general.

Bibliography

- [1] E. Agrell, "Closest point search in lattices", IEEE Trans. Inform. Theory, vol. 48, pp. 2201-2214, 2002.
- [2] A. H. Banihashemi and F. R. Kschinschang, "Tanner graphs for group codes and lattices: construction and complexity", IEEE Trans. Inform. Theory, vol. 47, pp. 824-882, 2001.
- [3] A. H. Banihashemi and I. F. Blake, "Trellis complexity and minimal trellis diagram of lattices", IEEE Trans. Inform. Theory, vol. 44, pp. 1829-1847, 1998.
- [4] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem", Combinatorica 6, pp. 1-13, 1986.
- [5] E. S. Barnes and N. J. A. Sloan, "New lattice packings of spheres", Can. J. Math., vol. 35, pp. 117-130, 1983.
- [6] E. S. Barnes and G. E. Wall, "Some extreme forms defined in terms of Abelian groups", J. Australian Math. Soc., vol. 1, pp. 47-63, 1959.

- [7] I. F. Blake and V. Tarokh, "On the trellis complexity of the densest lattice packings in $\mathbb{R}^n$ ", SIAM J. Discrete Math., vol. 9, no. 4, pp. 597-601, Nov. 1996.
- [8] A. Bos, J. H. Conway and N. J. A. Sloan, "Further lattice packings in high dimensions", Mathematica, vol. 29, pp. 171-180, 1982.
- [9] R. de Buda, "The upper error bound of a new near-optimal codes", IEEE Trans. Inform Theory, vol. IT-21, pp. 441-445, 1975.
- [10] R. de Buda and W. Kassem, "About lattices and the random coding theorem", IEEE Inter. Symp. Inform. Theory, pp. 145, 1981.
- [11] T. M. Cover and J. A. Thomas, Elements of Information Theory, New York: John Wiley, 1991.
- [12] J. H. Conway and N. J. A. Sloan, Sphere Packings, Lattices and Groups, 3rd ed., New York: Springer-Verlag, 1998.
- [13] T. Etzion, A. Trachtenberg and A. Vardy, "Which codes have cycle-free Tanner graphs", IEEE Trans. Inform. Theory, vol. 45, pp. 2173-2181, 1999.
- [14] G. D. Forney Jr., "Final report on a coding system design for advanced solar missions", Contract NAS2-3637, NASA Ames Research Center, Moffet Field, CA, 1967.
- [15] G. D. Forney, Jr., "The Viterbi algorithm", Proc. IEEE, vol. 61, pp. 268-278, 1973.

- [16] G. D. Forney Jr., "Density/length profile and trellis complexity of lattices", *IEEE Trans. Inform. Theory*, vol. 40, pp. 1753-1772, 1988.
- [17] G. D. Forney Jr., "Coset codes. Part I: introduction and geometrical classification", *IEEE Trans. Inform. Theory*, vol. 34, pp. 1123-1151, 1988.
- [18] G. D. Forney Jr., "Coset codes. Part II: binary lattices and related codes", *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152-1187, 1988.
- [19] G. D. Forney Jr., "Geometrically uniform codes", *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1241-1260, Sept. 1991.
- [20] G. D. Forney Jr., "Approaching AWGN channel capacity with coset codes and multilevel coset codes", *IEEE Inform. Theory, Proceedings International Symposium*, pp. 164-166, 1997.
- [21] G. D. Forney Jr., "On iterative decoding and two-way algorithm", in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 12-25, Sept. 1997.
- [22] G. D. Forney Jr., "Modulation and coding for linear Gaussian channels", *IEEE Trans. Inform. Theory*, vol. 44, pp. 2304-2414, 1998.
- [23] G. D. Forney Jr., and M. D. Trott, "The dynamics of group codes: state spaces, trellis digrams and canonical encoders", *IEEE Trans. Inform. Theory*, vol. IT-39, no. 9, pp. 1491-1513, 1993.
- [24] G. D. Forney Jr. and M. D. Trott, "Sphere-bound-achieving coset codes and multilevel coset codes", *IEEE Trans. Inform. Theory*, vol. 46, no. 3, pp. 820-850, 2000.

- [25] R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA: MIT Press, 1963.
- [26] R.G. Gallager, "Low-density parity-check codes", IRE Trans Inform. Theory, vol. IT-8, pp. 21-28, 1962.
- [27] P. M. Gruber and C. G. Lekkerkerker, Geometry of Numbers, 2nd ed. North-Holland, Amsterdam: Elsevier Science Publishers B.V., 1987.
- [28] M. Hall, The Theory of Groups. New York: Macmillan, 1959.
- [29] J. Hastel, "Dual vectors and lower bounds for the nearest lattice point problem", Combinatorica 8, pp. 75-81, 1988.
- [30] X-Y Hu, E. Eleftherou and D. Arnold, "Progressive edge-growth Tanner-graphs", Global Communications, 2001, GLOBCOM, IEEE, vol. 2 25-29, pp. 995-1001, 2001.
- [31] X-Y Hu, E. Eleftherou and D. Arnold, "Irregular progressive edge-growth Tanner-graphs", Proc. of IEEE Intel. Symp. Inform. Theory (ISIT), Lausanne, Switzerland, 2002.
- [32] X-Y Hu, E. Eleftherou and D. Arnold, "Regular and irregular progressive edge-growth Tanner-graphs", submitted to IEEE Trans. Inform. Theory.
- [33] R. Kannan, "Improved algorithm on integer programming and related lattice problem", Proc. 15th Annual ACM Symp. on Theory of Computing, pp. 193-206, 1983.

- [34] R. Kannan, "Minkowski's convex body theorem and integer programming", *Mathematics of Optimizations Research*, vol. 12, no. 3, pp. 415-440, 1987.
- [35] T. Kasami, "Optimal Lattice Codes for the Gaussian Channel", PhD Dissertation, McMaster Univ., Hamilton, Ontario, 1981.
- [36] Y. Kou, S. Lin and P.C. Fossorier, "Low-Density Parity-Check codes based on finite geometries: A Rediscovery and New Results", *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711-2738, 2001.
- [37] J. Leech, "Some sphere packings in higher space", *Can. J. Math.*, vol. 16, pp. 657-682, 1964.
- [38] S. Litsyy and V. Shevelev, "On ensembles of low-density parity-check codes: asymptotic distance distributions", *IEEE Trans. Inform. Theory*, vol. 48, no. 4, pp 887-908, 2002.
- [39] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. New York: Addison Wesley, 1994.
- [40] D.J.C. MacKay, "Good error correcting codes based on very sparse matrices", *IEEE Trans. Inform. Theory*, vol. 27, no. 2, pp. 399-431, 1999.
- [41] D.J.C. MacKay, and R. M. Neal, "Near Shannon limit performance of low density parity check codes", *Electron. Lett*, vol. 32, no. 18, pp. 1645-1671, 1996.
- [42] P. Samuel, *Algebraic Theory of Numbers*. Paris Hermann, 1971.

- [43] C. E. Shannon, "A mathematical theory of communication", The Bell System Technical Journal, vol. 27, pp. 379-423, 1948.
- [44] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel", The Bell System Technical Journal, vol. 38, no. 3, 1959.
- [45] S. B. Wicker, Error Control Systems for Digital Communication and Storage. New Jersey, Prentice Hall, 1995.
- [46] R. M. Tanner, "A recursive approach to low complexity codes", IEEE Trans. Inform. Theory, vol. 27, pp. 533-547, 1981.
- [47] V. Tarokh, "Minimal Tanner graphs for block codes and lattices", Proc. 5th Can. Workshop on Inform. Theory, Toronto, pp. 9-10, 1997.
- [48] V. Tarokh and I. F. Blake, "Trellis complexity vesus the coding gain of lattices I", IEEE Trans. Inform Theory, vol. 42, no. 6, pp. 1796-1807, 1996.
- [49] V. Tarokh and A. Vardy, "Universal bound on the performance of lattice codes", IEEE Trans. Inform. Theory, vol. 45, pp. 670-681, 1999.
- [50] E. Viterbo, "Tecniche matematiche computazionali per l'analisi ed il progtto di costellazioni a reticolo", PhD. thesis, Politecnico di Torino, Italy, 1995.
- [51] D. B. West, Introduction To Graph Theory, Englewood Cliffs, NJ: Prentice-Hall, 1996.

- [52] N. Wiberg, H-A. Loeliger, R. Kotter, "Codes and iterative decoding on graphs", *European Trans. on Telecom*, vol. 6, no. 5, pp. 513-526, Sept. 1995.