UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Paulo Ricardo Branco da Silva

Multilevel LDPC Lattice Codes with Efficient Encoding and Decoding

Florianópolis, 16 de setembro de 2020.

PAULO RICARDO BRANCO DA SILVA

MULTILEVEL LDPC LATTICE CODES WITH EFFICIENT ENCODING AND DECODING

Tese submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Doutor em Engenharia Elétrica.

Orientador: Danilo Silva

FLORIANÓPOLIS
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

da Silva, Paulo Ricardo Branco
Multilevel LDPC Lattice Codes with Efficient Encoding and Decoding / Paulo Ricardo Branco da Silva ; orientador, Danilo Silva, 2020.
76 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2020.

Inclui referências.

1. Engenharia Elétrica. 2. Reticulados. 3. códigos LDPC. 4. canal AWGN ponto a ponto. 5. códigos multinível. I. Silva, Danilo. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Paulo Ricardo Branco da Silva

MULTILEVEL LDPC LATTICE CODES WITH EFFICIENT ENCODING AND DECODING

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Bartolomeu Ferreira Uchôa Filho, Ph.D.
Universidade Federal de Santa Catarina

Prof. Richard Demo Souza, Dr.
Universidade Federal de Santa Catarina

Prof. Cecílio José Lins Pimentel, Ph.D.
Universidade Federal de Pernambuco

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Doutor em Engenharia Elétrica.

Documento assinado digitalmente
Telles Brunelli Lazzarin
Data: 08/10/2020 15:01:59-0300
CPF: 023.400.719-25

Prof. Telles Brunelli Lazzarin, Dr. Eng.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Santa Catarina

Documento assinado digitalmente
Danilo Silva
Data: 06/10/2020 14:17:31-0300
CPF: 031.557.924-26

Orientador: Prof. Danilo Silva, Ph.D.
Universidade Federal de Santa Catarina

Florianópolis, 16 de setembro de 2020.

Para Carolina Ribeiro Branco,

minha sobrinha

Agradecimentos

Gostaria de agradecer a todas as pessoas que contribuíram para a conclusão deste trabalho.

Ao meu orientador. O Prof. Danilo Silva me apoiou em todas as etapas do doutorado. Ele me fez perseverar apesar dos grandes problemas que existiram ao longo do meu trabalho. Ele me deu inúmeros conselhos e, sem ele, esta tese nunca teria sido concluída. O meu maior agradecimento é, sem dúvida, a ele;

Aos meus pais. Eles foram muito compreensivos. Apoiaram-me sempre. Nunca me negaram tempo para discutir minhas questões pessoais e para me motivar. Foram sempre gentis e firmes;

À Dora. Minha amiga de tantos anos me incentivou e me orientou. Agradeço por sempre me contatar exatamente quando eu mais precisava. Ela fez dos momentos difíceis momentos leves e de distração. Obrigado.

Ao Roberto Philippi. Ele me ajudou com inúmeras simulações, com um artigo e com fórmulas e deduções. Ele foi essencial para a conclusão desta tese;

Ao João Vinholi. Assim como o Roberto, ele me ajudou com as simulações e com muitos detalhes técnicos. Ele foi imprescindível, principalmente na parte final do doutorado. Sem ele, a minha pesquisa teria levado muito mais tempo;

Ao Prof. Bartolomeu. Ele me ajudou com dúvidas quando o Prof. Danilo não estava presente.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo financiamento feito através da bolsa do programa PROEX de número 88882.344612/2019-01.

“Viele sind hartnäckig in Bezug auf den einmal eingeschlagenen Weg, wenige in Bezug auf das Ziel.”.

(Friedrich Nietzsche)

Resumo da Tese apresentada à UFSC como parte dos requisitos
necessários para obtenção do grau de Doutor em Engenharia Elétrica

MULTILEVEL LDPC LATTICE CODES WITH EFFICIENT ENCODING AND DECODING

Paulo Ricardo Branco da Silva

16 de setembro de 2020

Orientador: Danilo Silva
Área de concentração: Comunicações e Processamento de Sinais
Palavras-chave: Códigos de reticulado, reticulados, codificação multinível, decodificação multi-estágio, códigos LDPC (*low-density parity-check*) aninhados
Número de páginas: 76

Códigos de reticulado possuem uma estrutura que não só é capaz de atingir a capacidade do canal AWGN como também é essencial para vários esquemas que exploram suas propriedades lineares, tais como os esquemas com múltiplos terminais. Construir códigos de reticulado capazes de obter uma boa performance em termos de probabilidade de erro de bloco ao mesmo tempo em que se possibilita uma implementação prática com baixa complexidade computacional para a codificação e a decodificação é um problema desafiador. Nesta tese, algoritmos de codificação e decodificação eficientes são propostos para reticulados LDPC multinível binários construídos via Construção D′ cuja complexidade é linear em relação ao número total de bits codificados. Além disto, a Construção D′ é generalizada com o intuito de relaxar as restrições de aninhamento das matrizes de verificação de paridade dos códigos componentes. Isto facilita o projeto de reticulados e melhora a performance dos códigos construídos. Para provar a eficiência da generalização da Construção D′, projetamos e testamos a performance de reticulados LDPC multinível sob decodificação multi-estágio. O desempenho de decodificação se mostrou comparável à de reticulados polares e próximo daquele evidenciado para LDLCs (*low-density lattice codes*) para códigos de reticulado projetados para o canal AWGN sem restrição de potência.

Resumo Expandido

Introdução

Reticulados têm atraído um aumento de interesse nos últimos anos. Sua estrutura não só permite alcançar a capacidade de canais gaussianos aditivos brancos (AWGN), como também serve como a base para esquemas multiterminal que exploram propriedades lineares [1]. No entanto, apesar de teoricamente interessantes, a construção de códigos de reticulado confiáveis e de baixa complexidade representa ainda um problema desafiador.

Nossa ideia é fazer uso de reticulados com estrutura LDPC (*Low-Density Parity Check* - verificação de paridade de baixa densidade), os quais admitem decodificadores do tipo *Belief Propagation* cuja complexidade é linear com relação à dimensão do reticulado. Apesar de várias construções desta forma terem sido propostas recentemente e terem demonstrado atingir um desempenho notável [2–4], todas sofreram da necessidade de realizar operações sobre corpos finitos de alta cardinalidade (seja $\mathbb{F}_p$ [3, 4] ou $\mathbb{R}$ [2]). Isto torna a complexidade por bit muito maior que para códigos estritamente binários.

Construções multinível de reticulado que usam códigos binários incluem a Construção D e a Construção D′ [5, 6]. Elas dependem de uma família de L códigos binários aninhados (o código de maior cardinalidade contém todas as palavras código do código de menor cardinalidade) usados em conjunto com uma modulação 2^L-PAM. A Construção D (D′) descreve um reticulado através das matrizes geradoras (de verificação de paridade) dos códigos componentes aninhados. Quando usado em conjunto com a decodificação multi-estágio (MSD) [7, 8], cada código componente pode ser codificado e decodificado individualmente sobre um corpo binário, levando a uma redução significativa da complexidade.

Reticulados construídos com a Construção D′ usando códigos LDPC como componentes são referidos como reticulados LDPC. Eles foram originalmente introduzidos em [9] e estudados posteriormente em [10–12]. Até onde sabemos, nenhum artigo sobre a teoria ou aplicação de reticulados LDPC estritamente multinível ($L \geq 2$) usando a Construção D′ foi publicado. Esta aparente falta de interesse pode ser parcialmente explicada pelos grandes desafios que a Construção D′ apresenta.

A Construção D′ tal como definida não possui um esquema de codificação eficiente [9]. A complexidade da codificação de um reticulado LDPC é quadrática na dimensão do reticulado se realizada via "força bruta" utilizando a matriz geradora do reticulado. Para explorar métodos de codificação sistemáticos de baixa complexidade, deve-se codificar cada código componente individualmente usando sua matriz de verificação de paridade. No entanto, na definição da Construção D$'$, os níveis individuais se acoplam através de equações modulares não binárias de forma não trivial [6], tornando difícil o desenvolvimento de regras para realizar a codificação individual.

Além do problema da codificação, a definição da Construção D$'$ impõe dificuldades à formulação de regras que implementam de forma eficiente a decodificação multi-estágio. Em princípio, esta forma de decodificação requer a remoção da influência de todos os níveis passados—pela re-codificação da palavra parcialmente decodificada e sua subtração do vetor recebido—antes da decodificação do nível atual. No entanto, quando do uso da Construção D$'$, não é claro no processo de re-codificação como um nível influencia os níveis posteriores. Além disto, dada a dependência na re-codificação, um método de codificação eficiente seria exigido para que a decodificação multi-estágio fosse eficiente.

Constata-se também que projetos feitos com a Construção D$'$ apresentam uma grande diferença de desempenho com relação aos feitos com a Construção D [13]. Resultados experimentais revelam que o desempenho de comprimento finito para reticulados LDPC construídos via Construção D$'$ é tipicamente inferior, contradizendo a expectativa baseada no excelente desempenho de códigos LDPC binários de um único nível. Uma das razões é que a Construção D$'$ requer não somente códigos componentes aninhados mas também matrizes de verificação de paridade aninhadas (isto é, uma matriz contém outra como submatriz) [6]. Trata-se de uma restrição severa que gera códigos com um perfil de distribuições de graus de variável extremamente sub-ótimo em termos de desempenho de decodificação via *Belief Propagation*.

Nesta tese, os desafios previamente apresentados são resolvidos através de reinterpretações da Construção D$'$. Nossas contribuições incluem uma descrição alternativa da Construção D$'$ que permite a codificação sequencial dos códigos componentes, a demonstração que cosets de códigos LDPC podem ser codificados e decodificados em tempo linear, uma generalização da Construção D$'$ e um método baseado no particionamento de equações de paridade para a construção prática de matrizes de verificação de paridade capazes de satisfazer as restrições da Construção D$'$ Generalizada.

Objetivos

Este trabalho visa a produzir bons códigos em termos de probabilidade de erro com alta eficiência espectral — fazendo uso de codificação multinível — e estrutura algébrica de reticulado.

Mais especificamente, o objetivo é flexibilizar a construção de reticulados LDPC. Primeiramente, tenta-se realizar a redefinição da Construção D′ para tornar a codificação e a decodificação mais eficientes. Depois, por meio de uma generalização da definição clássica da Construção D′, tenta-se melhorar o projeto de códigos de reticulado. De fato, pretende-se obter projetos flexíveis com desempenho superior ao dos reticulados LDPC previamente discutidos na Literatura e semelhante ao de reticulados construídos com a Construção D.

Metodologia

Analisamos a Construção D′ e os códigos de reticulado por ela construídos. Detectamos que a Construção D′ possui uma definição restritiva que limita o desempenho de reticulados LDPC.

Sugerimos, portanto, primeiramente alterá-la de forma a comportar a codificação sequencial e a decodificação multi-estágio. Verificamos que a contribuição de níveis passados no nível atual passa simplesmente por um vetor de síndrome que a palavra-código atual deve satisfazer.

Em seguida, demonstramos como algoritmos para a codificação e decodificação em tempo linear de códigos LDPC podem ser adaptados para cosets destes códigos sem aumento da ordem de complexidade.

Analisando mais a fundo a definição da Construção D′, nota-se que ela exige o aninhamento das matrizes de verificação de paridade dos códigos componentes envolvidos. Esta exigência não é necessária; basta que os códigos componentes sejam aninhados. Propomos então uma generalização da Construção D′ que relaxa as restrições de aninhamento sobre as matrizes de verificação de paridade dos códigos componentes.

Para implementar a Construção D′ Generalizada, propomos um método de construção das matrizes de verificação de paridade dos códigos componentes com base na matriz de verificação de paridade do código LDPC de maior taxa. Também apresentamos variações deste método básico com o fim de maximizar o *girth* da matriz e de permitir a codificação em tempo linear.

Resultados e Discussão

Desenvolvemos uma definição para a Construção D′ que possibilita o uso de codificação sequencial e decodificação multi-estágio. Como para cada nível um vetor de síndrome contendo as contribuições de níveis anteriores pode ser computado diretamente das equações de verificação de paridade, evita-se a re-codificação explícita para a decodificação multi-estágio. Assim, reticulados LDPC multinível sempre podem ser codificados e decodificados com complexidade $O(Ln)$ se os códigos componentes permitirem codificação sistemática em tempo linear, onde n é a dimensão do reticulado.

Desenvolvemos uma abordagem que permite para cada nível a decodificação de códigos de coset (vetor de síndrome não nulo). Identificamos ser viável realizar esta decodificação com complexidade linear por meio do uso de decodificadores *Belief Propagation* binários padrão apenas ligeiramente adaptados.

Também verificamos que a Construção D′ possui restrições muito severas de aninhamento. Os códigos devem ser aninhados, mas não necessariamente suas matrizes de verificação de paridade. Generalizamos a Construção D′, impondo a ela apenas uma restrição de linearidade sobre as matrizes de verificação de paridade dos códigos componentes. Esta generalização aumenta consideravelmente o espaço de projeto para bons códigos componentes, possibilitando projetos de códigos LDPC multinível com melhor desempenho.

Os métodos de implementação da Construção D′ Generalizada por meio do particionamento de equações de paridade se provaram práticos e eficientes. Permitiram construir, por exemplo, reticulados LDPC de dois níveis com complexidade de codificação e decodificação linear no número total de bits codificados e com desempenho comparável ao de reticulados estado da arte construídos via Construção D [13].

Considerações Finais

Obtivemos resultados teóricos para códigos de reticulado no canal sem restrição de potência. Duas contribuições se destacam: uma descrição alternativa da Construção D′ e uma generalização da Construção D′.

A descrição alternativa da Construção D′ possibilita a codificação sequencial dos códigos componentes. Como consequência, mostramos que codificadores e decodificadores binários LDPC podem ser utilizados para produzir algoritmos de codificação sequencial e decodificação multi-estágio para reticulados LDPC com complexidade linear no número total de bits codificados. A complexidade linear não foi atingida até o momento por nenhuma outra construção de reticulado capaz de se aproximar do limite de Poltyrev (limite teórico do canal sem restrição de potência [14]).

A Construção D′ Generalizada relaxa as restrições de aninhamento das matrizes de verificação de paridade dos códigos componentes, facilitando significativamente o projeto; especificamente, através da nova construção, apenas os códigos componentes devem ser aninhados e não suas matrizes de verificação de paridade. Seguindo este resultado, propomos um princípio generalizado para a construção de códigos aninhados baseado no particionamento das equações de verificação de paridade, assim como um método prático, inspirado pelo algoritmo PEG [15],

para a construção de códigos componentes LDPC de alto *girth* e decodificação eficiente.

Baseados na nova construção, reticulados LDPC com baixa complexidade computacional foram projetados. Demonstramos por meio de simulações de probabilidade de erro que seu desempenho é comparável ao de reticulados polares [13], acabando com a disparidade de desempenho que existia entre reticulados construídos via Construção D e Construção D′. Demonstramos também que os reticulados LDPC propostos são capaz de alcançar desempenho semelhante ao dos reticulados LDLC [29] mas com muito menor complexidade computacional.

Apesar de o desempenho alcançado ser ainda aquém do de reticulados p-ários como os GLD [4] e LDA [28], deve-se ressaltar que apenas reticulados regulares com relação ao peso dos nós de variáveis foram considerados nesta tese. Pode-se esperar um desempenho muito melhor para reticulados LDPC irregulares com distribuições de grau mais flexíveis e cuidadosamente projetadas. No entanto, para tanto, é necessário um novo procedimento de projeto que garanta o aninhamento dos códigos componentes resultantes, o que constitui um interessante desafio para trabalhos futuros.

Palavras-chave: Códigos de reticulado, reticulados, codificação multinível, decodificação multi-estágio, códigos LDPC (*low-density parity-check*) aninhados.

Abstract of Thesis presented to UFSC as a partial fulfillment
of the requirements for the degree of Doctor in Electrical Engineering

MULTILEVEL LDPC LATTICE CODES WITH EFFICIENT ENCODING AND DECODING

Paulo Ricardo Branco da Silva

September 16th, 2020

Advisor: Danilo Silva

Area of concentration: Communications and Signal Processing

Keywords: Lattice codes, lattices, multilevel coding, multistage decoding, nested low-density parity-check (LDPC) codes

Number of pages: 76

Lattice codes have a structure that is not only able to achieve the capacity of the AWGN channel but that is also essential to a number of multiterminal schemes which exploit its linear properties. To construct lattice codes capable of good performance in terms of word error probability and of a practical implementation that allows for low encoding and decoding computational complexity is a challenging problem. In this thesis efficient encoding and decoding algorithms are proposed for multilevel binary LDPC lattices constructed via Construction D′ whose complexity is linear in the total number of coded bits. In addition, Construction D′ is generalized with the intent of relaxing the nesting constraints on the parity-check matrices of the component codes. This leads to simpler lattice design and improved performance of the constructed codes. In order to prove the effectiveness of the generalized construction, we design and test the performance of multilevel LDPC lattices constructed under this framework under multistage decoding. The decoding performance is comparable to that of polar lattices and close to that of low-density lattice codes (LDLC) for lattice codes designed for the power-unconstrained AWGN channel.

List of Figures

List of Algorithms

Contents

1 Introduction

In this chapter we set out the background and scope of this thesis. Specifically, we underline why this research work is relevant, and we present its major technical contributions. We also perform a bibliographical review of the work most closely related to the concepts herein discussed. To conclude this introductory chapter, we describe how this thesis is structured.

1.1 Motivation

Given the developments in mobile technology and the transition to data-intensive media consumption, it has become extremely important to find communication strategies that deliver reliable high-rate data exchanges. In particular, it seems very promising to do so in a scenario where the computational complexity is shared amongst many mobile devices. In this way, mathematical tools and transmitting/receiving schemes addressing the requirements of such co-operative communication ensembles need to be studied and developed. One such mathematical tool are lattices.

Lattice codes, the analogue of linear codes in the Euclidean space, have attracted an increasing amount of attention in recent years. Their rich structure not only provides an elegant and powerful solution to achieving the capacity of the additive white Gaussian noise (AWGN) channel, but is also a key ingredient to many multiterminal information theory schemes that exploit linearity [1]. However, despite the potential for reliable communication with high spectral efficiency, constructing lattice codes that can realize these benefits with low complexity still poses many challenges.

A promising direction is to construct lattices with a low-density parity-check structure. These allow the use of a Belief Propagation decoder with complexity linear in the lattice dimension. While several constructions of this form have been proposed recently and shown to achieve remarkable performance [2–4], they all suffer from the need to perform operations over a large field (either $\mathbb{F}_p$ [3, 4] or $\mathbb{R}$ [2]), whose complexity per bit is much higher than that of binary codes.

Alternative lattice constructions that leverage the use of binary codes are the multilevel Construction D and Construction D′ [5, 6], which rely on a family of L nested binary linear codes used in conjunction with the 2^L-PAM modulation. These constructions use the concepts of nested codes and nested matrices. The former identifies codes whose codebooks are contained within the codebook of higher-rate codes. The latter describes matrices that are contained in higher-dimension matrices. Construction D (Construction D′) describes a lattice through the generator (parity-check) matrices of the nested component codes. When used together with multistage decoding (MSD) [7, 8], each component code can be individually encoded and decoded over the binary field, leading to a significant reduction in complexity.

Construction D′ lattices based on binary low-density parity-check (LDPC) codes, referred to as LDPC lattices, were originally introduced in [9] and subsequently studied in [10–12]. To the best of our knowledge, no papers have been published on the theory or applications of strictly multilevel ($L \geq 2$) Construction D′ LDPC lattices. The apparent lack of interest in this lattice construction may be partly explained by three major challenges of Construction D′ which have so far remained unsolved:

- **Lack of efficient encoding:** The complexity of encoding an LDPC lattice is quadratic in the lattice dimension if done via the generator matrices of the component codes, i.e., deriving them from the related parity-check matrices present in the definition of Construction D′. In order to exploit low-complexity systematic encoding methods for LDPC codes, one should be able to encode each component code individually using its parity-check matrix. However, in the definition of Construction D′ [5, 6], individual levels are coupled through non-binary modular equations in a nontrivial way, making it unclear how to perform individual encoding.

- **Lack of efficient multistage decoding:** In principle, multistage decoding requires the influence of all past levels to be removed—by re-encoding the partially decoded lattice codeword and subtracting it from the received vector—before the current level is decoded. However, when using Construction D′, it is unclear how a level influences the subsequent ones, so that re-encoding can be performed. Moreover, due to such dependence on re-encoding, an efficient encoding method would be required for multistage decoding to be efficient.

- **Performance gap:** Even if we ignore the issue of low-complexity decoding, experimental results reveal that the finite-length performance of existing Construction D′ LDPC lattices is typically much inferior to that of Construction D lattices [9, 11, 13], contradicting the expectation one might have given the excellent performance of one-level LDPC codes. Part of the reason may be that Construction D′ requires not only the component codes, but also their parity-check matrices, to be nested (i.e., one matrix must contain the other as a submatrix) [6], a stringent constraint that may degrade the overall performance of the resulting lattice.

1.2 Contributions

In this thesis, the three challenges presented previously are solved starting from a reinterpretation of Construction D’. In particular, reinterpreting and reassessing the restrictions placed on the parity-check matrices of the component codes is the key to obtaining our theoretical results and the performance improvement shown by our LDPC lattices.

Our main technical contributions are summarized as follows:

- We present an alternative description of Construction D′ which enables sequential encoding of the component codes. The contribution of past levels on the current level is subsumed by a syndrome vector that the current codeword must satisfy. Apart from that, the actual encoding of a level is done entirely over the binary field. Moreover, as the syndrome vector can be computed directly from parity-check equations, the need for explicit re-encoding is avoided for multistage decoding.

- We show how existing linear-time algorithms for encoding and decoding LDPC codes can be adapted to handle cosets of LDPC codes without increasing the order of complexity (coset codes result from the fact that a level’s codeword does not produce a null syndrome vector). It follows that multilevel LDPC lattices can always be decoded with complexity $O(Ln)$ and can be encoded also with complexity $O(Ln)$ if the component codes admit linear-time systematic encoding, where n is the lattice dimension. Moreover, encoding and decoding can be performed with off-the-shelf binary LDPC encoders and decoders.

- We propose a generalization of Construction D′ that relaxes the nesting constraints on the parity-check matrices of the component codes (which, by the traditional definition of

Construction D′ [6], must themselves remain nested). This new construction significantly enlarges the design space for good component codes, enabling the design of multilevel LDPC lattices with much better error probability performance under decoding algorithms bounded in computational complexity, i.e, for practical decoders. This contribution may also be of independent interest from a mathematical perspective.

- We propose an efficient method to construct the parity-check matrices of the remaining component codes given only the parity-check matrix of the highest-rate LDPC code, whilst respecting the conditions of the Generalized Construction D′. We also present variations of this basic method which are inspired by the PEG algorithm of [15] and which aim at maximizing girth and enabling linear-time encoding.
- We present examples of two-level LDPC lattices, with encoding and decoding complexity linear in the total number of coded bits, that achieve performance comparable to state-of-the-art Construction D lattices [13], closing the long-standing gap of such lattices to Construction D′ lattices.

It is worth mentioning that, although we have opted to use binary codes for simplicity and computational efficiency, our results can be straightforwardly generalized to a general prime p, as well as to Complex Construction D′ lattices over a complex ring [6, 16, 17].

1.3 Related Work

A lattice is a discrete additive subgroup of the Euclidean space, prized in many applications for its linear properties. A lattice code consists of the intersection of a lattice and a bounded region, also called a shaping region. While it is well-known that lattice codes can achieve the capacity of the AWGN channel (see [18] and references therein), renewed interest in the topic can be traced to the seminal paper by Erez and Zamir [19], which showed that capacity can be achieved by nested lattice codes with lattice decoding (a suboptimal decoding approach which effectively ignores the shaping region). Since then, several applications of lattice codes to multiterminal information theory have been proposed based on their properties, including: distributed source coding [20], physical-layer security [21], and communication over Gaussian networks [1]—in particular, lattice codes are essential to the compute-and-forward strategy for relay networks [22] and to integer-forcing methods for MIMO channels [23].

The main problem in the design of a lattice code is arguably the design of the underlying lattice, which must be good at rejecting noise. This problem can be formalized by means of the power-unconstrained AWGN channel model introduced by Poltyrev [14], leading to the notion of AWGN-goodness as a necessary condition to achieve capacity. As a consequence, much of the literature on the topic, as well as this thesis, is focused on the unconstrained lattice design problem. Recent work has shown that the use of a fixed shaping region is not essential to achieve capacity under lattice decoding if signal points are selected with a nonuniform (discrete Gaussian) distribution [24, 25]. In principle, the lattices designed in this thesis can be combined with a variety of shaping methods, such as trellis shaping [26, 27] or probabilistic shaping [24, 25], which are however outside the scope of this work.

A popular way of constructing lattices, which exploits the power of linear codes, is the so-called Construction A [6]. The basic (real-valued) version of this construction relies on a single p-ary linear code, where p is a prime, which is repeated across the Euclidean space (at multiples of p along each coordinate) to produce an infinite constellation. This construction is the source of most proofs of achievable rates (using asymptotically long random codes) [1, 19], but is also shown to produce lattices with excellent finite-length performance and complexity linear in the lattice dimension, provided that p is sufficiently large. This is the case with Generalized Low-Density (GLD) lattices [4, 30] and Low-Density Construction A (LDA) lattices [3, 28, 31], which are shown to be AWGN-good. However, decoding a p-ary code, for large p, is much more complex than decoding a similarly structured binary code. For instance, the Belief Propagation decoder in [3] has complexity $O(p^2 n)$, i.e., the complexity is exponential in the bit depth $\log_2 p$.

Another approach is to construct lattices that are designed and decoded directly in the Euclidean space, such as low-density lattice codes (LDLC) [2]. However, since the decoder now has to process continuous functions [2, 29], the decoding complexity is typically even higher than that of Construction A lattices.

Multilevel lattice constructions based on binary codes are potentially harder to design, but have the promise of complexity that scales linearly with the number of levels. Moreover, they are known to be AWGN-good under multistage decoding [8]. Construction D was used in [32] to produce turbo lattices and in [13] to construct polar lattices; the latter are shown to be AWGN-good with encoding and decoding complexity $O(Ln \log n)$. Construction D has also been used in [33] to construct spatially-coupled LDPC lattices, which were shown to be AWGN-good under multistage Belief Propagation decoding. However, both the encoding and the cancellation step that has to be performed at each decoding stage rely on the generator matrices of the component LDPC codes, which are generally dense, leading to an overall high complexity.

Construction D$'$ was used in [9–12] to construct multilevel LDPC lattices; however, these papers consider only joint decoding[1] of the component codes, whose complexity is exponential in L, while the encoding complexity is not addressed. Both issues can be avoided using a single-level LDPC lattice, as done in [35–38, 43], allowing the use of conventional encoding and decoding methods for LDPC codes. However, in this case the construction reduces to Construction A, which is known to result in a poor performance for $p = 2$. More precisely, since all the higher levels are uncoded, the performance in terms of word error rate quickly degrades as the block length increases and does not improve sharply as the noise level decreases.

1.4 Organization

The remainder of this thesis is organized as follows. Chapter 2 reviews basic concepts on lattices, giving special emphasis to the power-unconstrained channel, Forney-Trott-Chung's two-stage decoding scheme, and the definition of Construction D′. In Chapter 3, we present a sequential description of Construction D′, which is used to derive efficient multilevel encoding and multistage decoding schemes for Construction D′ LDPC lattices. Also in Chapter 3, a generalization of Construction D′ is proposed that relaxes the nesting constraints of the original definition whilst still satisfying the requirements for sequential encoding. Essentially, the section on Generalized Construction D′ explains how the constraint of nested parity-check matrices is exchanged by the constraint of linearly dependent parity-check matrices. In addition, Chapter 3 also proposes an efficient method for the implementation of Generalized Construction D′ LDPC lattices, which is based on the partition of parity-check equations and which we call check splitting. This construction method is then formalized by the use of the PEG construction into variants that maximize girth and provide linear-time encoding. Chapter 4 presents design examples and their corresponding simulation results in terms of word error probability. Finally, Chapter 5 presents the conclusions we derived from our work and traces possible new research directions.

[1] The multistage decoder mentioned in [11] is unsuitable for Construction D′ since it relies on independent encoding of levels through the Code Formula [16], which does not generally produce lattices [34].

2
Preliminaries

In this chapter, we set forth the preliminary constructs underlying the rest of this thesis. We describe our notation and review basic concepts on lattices. In particular, we define Construction D′ (for further details, we refer the reader to [1, 6, 8]). In addition, we describe transmission over a power-unconstrained channel. We also outline the Poltyrev limit and Forney-Trott-Chung's two-stage decoding strategy for achieving this limit [8].

2.1 Notation

Let $\mathbf{0}$ denote the all-zero vector, with appropriate length implied by the context. If $\mathcal{A}$ is a set, then $\mathcal{A}^n$ and $\mathcal{A}^{m\times n}$ denote the set of length-n vectors and $m \times n$ matrices, respectively, with entries in $\mathcal{A}$. Let $\mathbb{F}_2$ be the finite field of size 2 and let $\varphi : \mathbb{Z} \to \mathbb{Z}/2\mathbb{Z} \cong \mathbb{F}_2$ be the natural reduction homomorphism, extended to vectors and matrices in a component-wise fashion.

We follow common convention for the notation "mod m" when m is an integer. For any $a, b \in \mathbb{Z}$, we use the modular congruence $a \equiv b \pmod{m}$ to denote that $a - b$ is divisible by m. For any $x \in \mathbb{R}$, we define $x \bmod m$ as the unique $r \in [0, m)$ such that $x = r + qm$, for some $q \in \mathbb{Z}$. These notations are again extended to vectors and matrices in a component-wise fashion.

In this thesis, we treat binary linear codes and associated parity-check matrices as having entries in $\{0, 1\} \subseteq \mathbb{Z}$, rather than in $\mathbb{F}_2$. This approach significantly simplifies notation when dealing with lattices. The algebraic properties of a linear code are recovered by using modular equations or the explicit mapping to $\mathbb{F}_2$. For instance, if $\mathbf{H} \in \{0, 1\}^{m\times n}$, then a binary linear code $C \subseteq \{0, 1\}^n$ is defined by the parity-check matrix $\mathbf{H}$ as

$$C = \{\mathbf{x} \in \{0, 1\}^n : \mathbf{H}\mathbf{x}^T \equiv \mathbf{0} \pmod{2}\}.$$

The dimension of C as a linear code is the dimension of the subspace $\varphi(C) \subseteq \mathbb{F}_2^n$ or, equivalently, the dimension of the null space of $\varphi(\mathbf{H}) \in \mathbb{F}_2^{m\times n}$.

2.2 Lattices

A lattice $\Lambda \subseteq \mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$. This implies that Λ is closed under integer linear combinations and may be expressed as $\Lambda = \{\mathbf{uG},\ \mathbf{u} \in \mathbb{Z}^n\}$, where $\mathbf{G} \in \mathbb{R}^{n\times n}$ is a generator matrix.

A fundamental region of Λ is a set $\mathcal{R}_\Lambda \subseteq \mathbb{R}^n$ such that any $\mathbf{x} \in \mathbb{R}^n$ can be *uniquely* expressed as $\mathbf{x} = \boldsymbol{\lambda} + \mathbf{r}$, where $\boldsymbol{\lambda} \in \Lambda$ and $\mathbf{r} \in \mathcal{R}_\Lambda$. Every fundamental region has the same volume, which is denoted by $V(\Lambda)$. A fundamental region $\mathcal{R}_\Lambda$ defines a quantizer $Q_\Lambda : \mathbb{R}^n \to \Lambda$ and a modulo-Λ operation $\mathbb{R}^n \to \mathcal{R}_\Lambda$ as $Q_\Lambda(\mathbf{x}) = \boldsymbol{\lambda}$ and $\mathbf{x} \bmod \Lambda = \mathbf{r}$, respectively, where $\mathbf{x} = \boldsymbol{\lambda} + \mathbf{r}$. In particular, the Voronoi region of Λ (around the origin) is the set $\mathcal{V}_\Lambda$ of points that are closer to $\mathbf{0}$ than to any other lattice point, with ties decided arbitrarily but such that $\mathcal{V}_\Lambda$ is a fundamental region.

A sublattice $\Lambda' \subseteq \Lambda$ is a subset of Λ which is itself a lattice. If Λ and $\Lambda' \subseteq \Lambda$ are lattices, then $C = \Lambda \cap \mathcal{R}_{\Lambda'}$ is said to be a nested lattice code. Note that $|C| = V(\Lambda')/V(\Lambda)$.

2.3 Transmission Without a Power Constraint

In the design of lattice codes, one is often interested in addressing the coding problem separately from the shaping problem. This leads to the so-called (power-)unconstrained AWGN channel studied by Poltyrev [14], over which any lattice point may be transmitted without restrictions. In that case, the main performance metric for a lattice is its probability of decoding error for a given density of lattice points.

More precisely, let $\Lambda \subseteq \mathbb{R}^n$ be a lattice. The channel output is given by $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{x} \in \Lambda$ is the transmitted vector and $\mathbf{z} \in \mathbb{R}^n$ is a white Gaussian noise vector with variance σ^2 per component. Decoding is performed via lattice decoding, i.e., by quantizing $\mathbf{y}$ to the nearest lattice point $\hat{\mathbf{x}} = Q_\Lambda(\mathbf{y})$. The probability of error, denoted by $P_e(\Lambda, \sigma^2)$, is the probability that $\mathbf{z}$ falls outside $\mathcal{V}_\Lambda$.

Given that the density of Λ is inversely proportional to $V(\Lambda)$, it is convenient to define the volume-to-noise ratio (VNR) as[1]

$$\gamma_\Lambda(\sigma) \triangleq \frac{V(\Lambda)^{2/n}}{2\pi e\sigma^2} \tag{2.1}$$

which gives a measure of the density of Λ relative to the noise level.

It is well-known [1, 8] that, if $P_e(\Lambda, \sigma^2) \approx 0$, then $\gamma_\Lambda(\sigma)$ is necessarily greater than 1,

[1] The VNR is also commonly defined [1] without the term $2\pi e$ in the denominator.

or 0 dB. This fundamental limit is known as the Poltyrev limit or the sphere bound. On the other hand, for all $\sigma^2 > 0$ and all $P_e > 0$, there exists a family of n-dimensional lattices with $P_e(\Lambda, \sigma^2) \leq P_e$ such that $\lim_{n\to\infty} \gamma_\Lambda(\sigma) = 1$. Lattices with this property are said to be *good* for AWGN coding.

In practice, nearest-neighbor lattice decoding may not be feasible to implement, so a sub-optimal lattice decoder $\mathcal{D} : \mathbb{R}^n \to \Lambda$ may be used instead. In this case, one should refer to the probability of error of the pair $(\Lambda, \mathcal{D})$, i.e., $P_e((\Lambda, \mathcal{D}), \sigma^2)$. For simplicity, we keep the notation $P_e(\Lambda, \sigma^2)$ when the decoder is clear from the context.

2.3.1 Forney-Trott-Chung's Two-Stage Decoding

A practical way to approach the Poltyrev problem, which often simplifies the decoding and the code design, is the two-level partition proposed in [8]. Given a lattice $\Lambda \subseteq \mathbb{R}^n$, a sublattice $\Lambda' \subseteq \Lambda$ is chosen such that the modulo-Λ' operation (as well as the enumeration of elements of Λ') is easy to implement. In this manner, Λ can be partitioned as $\Lambda = C + \Lambda'$, where $C = \Lambda \cap \mathcal{R}_{\Lambda'}$ is a lattice code and $\mathcal{R}_{\Lambda'}$ is a fundamental region (usually chosen as a Voronoi region) defining the modulo-Λ' operation.

In order to transmit $\mathbf{x} \in \Lambda$, vectors $\mathbf{c} \in C$ and $\boldsymbol{\lambda}' \in \Lambda'$ are chosen independently. The point $\mathbf{x}$ is transmitted as the sum of these two components, i.e., $\mathbf{x} = \mathbf{c} + \boldsymbol{\lambda}'$. Let $\mathbf{y} = \mathbf{x} + \mathbf{z}$ be the channel output, as described above. Decoding from $\mathbf{y}$ is as follows. First,

$$\mathbf{r} = \mathbf{y} \bmod \Lambda' = \mathbf{c} + \mathbf{z} \bmod \Lambda' \tag{2.2}$$

is computed, eliminating the influence of $\boldsymbol{\lambda}'$. For instance, if $\Lambda' = q\mathbb{Z}^n$, with $\mathcal{R}_{\Lambda'} = [0, q)^n$, then the modulo-$\Lambda'$ operation can be easily implemented by component-wise modulo-q reduction over $\mathbb{R}$ (see Section 2.1). Next, a decoder for code C is applied on the modulo-Λ' equivalent channel described above, from which the estimate $\hat{\mathbf{c}} \in C$ is obtained. Finally, $\hat{\mathbf{c}}$ is subtracted from $\mathbf{y}$, resulting in

$$\mathbf{y}' = \mathbf{y} - \hat{\mathbf{c}} = (\mathbf{c} - \hat{\mathbf{c}}) + \boldsymbol{\lambda}' + \mathbf{z}, \tag{2.3}$$

and a decoder for Λ' is applied, obtaining the estimate $\hat{\boldsymbol{\lambda}}' \in \Lambda'$ and, consequently, $\hat{\mathbf{x}} = \hat{\mathbf{c}} + \hat{\boldsymbol{\lambda}}'$.

It follows by the union bound that

$$P_e(\Lambda, \sigma^2) \leq P_e(C, \Lambda') + P_e(\Lambda', \sigma^2) \tag{2.4}$$

where $P_e(C, \Lambda')$ is the probability of error for the code C when used on the channel in (2.2). Note that, when $\Lambda' = q\mathbb{Z}^n$, nearest-neighbor lattice decoding can be easily implemented and $P_e(\Lambda', \sigma^2)$ can be computed exactly, i.e.,

$$P_e(q\mathbb{Z}^n, \sigma^2) = 1 - \left(1 - 2Q\left(\frac{q}{2\sigma}\right)\right)^n \tag{2.5}$$

where $Q(x) = (1/\sqrt{2\pi}) \int_x^\infty e^{-u^2/2} du$ is the Q-function.

2.4 Construction D′

Let $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{L-1} \subseteq \{0, 1\}^n$ be a family of nested binary linear codes. For $\ell = 0, \ldots, L - 1$, let k_ℓ be the dimension of C_ℓ, let $R_\ell = k_\ell/n$, and let $m_\ell = n - k_\ell$. For convenience, define $m_L = 0$. Clearly,

$$m_0 \geq m_1 \geq \cdots \geq m_{L-1} \geq m_L.$$

Let $\mathbf{h}_1, \ldots, \mathbf{h}_{m_0} \in \{0, 1\}^n$ be such that

$$\mathbf{H}_\ell = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{m_\ell} \end{bmatrix} \tag{2.6}$$

is a parity-check matrix for C_ℓ, for $\ell = 0, \ldots, L - 1$. An L-level Construction D′ lattice [6] is defined as

$$\begin{aligned} \Lambda = \{\mathbf{v} \in \mathbb{Z}^n : \mathbf{h}_j \mathbf{v}^{\mathrm{T}} \equiv 0 \pmod{2^{\ell+1}}, \\ m_{\ell+1} < j \leq m_\ell,\ 0 \leq \ell \leq L - 1\}. \end{aligned} \tag{2.7}$$

We can also express $\Lambda = C + 2^L\mathbb{Z}^n$, where

$$C = \Lambda \cap [0, 2^L)^n \tag{2.8}$$

is a lattice code. In particular, $V(\Lambda) = 2^{n(L-R)}$, where

$$R = R(C) \triangleq \frac{1}{n} \log_2 |C|. \tag{2.9}$$

If matrices $\mathbf{H}_0, \ldots, \mathbf{H}_{L-1}$ are sparse, i.e., if $C_0, \ldots, C_{L-1}$ are LDPC codes, then Λ is said to be an L-level LDPC lattice [9].

It is worth emphasizing that the number of levels in the construction, L, refers to the number of *coded* levels—there is always an additional uncoded level[2] ($\ell = L$) corresponding to the lattice $2^L\mathbb{Z}^n$, which tiles the lattice code C into an infinite constellation. Note that, for $L = 1$, the construction is "degenerate" in the sense that it reduces to Construction A [6]. Thus, strictly multilevel Construction D′ lattices require $L \geq 2$. For the remainder of this thesis, we assume $L \geq 2$ when referring to Construction D′.

Remark: When an L-level Construction D′ lattice $\Lambda \subseteq \mathbb{Z}^n$ is used for the Poltyrev channel under the approach of Section 2.3.1, we naturally choose $\Lambda' = 2^L\mathbb{Z}^n$ and $\mathcal{R}_{\Lambda'} = [0, 2^L)^n$, so that the mod Λ' operation becomes simply mod 2^L over $\mathbb{R}$. In this case, decoding of Λ essentially reduces to decoding of the lattice code C over the mod-2^L channel $\mathbf{y} = \mathbf{c} + \mathbf{z} \bmod 2^L$, where $\mathbf{c} \in C$.

[2] The uncoded level is not shown explicitly in equation (2.7). It is implicit in the operation $(\bmod\ 2^{\ell+1})$.

3

Efficient Encoding and Decoding and a Generalization of Construction D’

In this chapter, an alternative description of Construction D$'$ is proposed that enables sequential multilevel encoding over the construction’s component coset codes. Based on this description, we also present a multistage decoder adapted to LDPC coset codes. We show that it is unnecessary to continuously re-encode the partially decoded codeword in between levels during decoding, and that the influence of past levels can be subsumed in the current level’s syndrome vector. We also adapt off-the-shelf binary encoders and decoders for the component LDPC codes to take into account that the codebook actually consists of an LDPC coset code, i.e., the syndrome of the codewords is not necessarily the null vector.

The analysis starts from rewriting (2.7) in matrix form (see Proposition 3.1). This step is not only useful for the sequential description of Construction D$'$ but also for this chapter’s latter half, which presents a generalization of Construction D$'$. Firstly, Construction D$'$ is reworked in order to describe it in terms of component vectors having syndromes which synthesize the effects of previous levels. Then, Construction D$'$ is reworked to eliminate the need for component codes to have nested parity-check matrices. Instead, we require only that the codes be nested, which is ensured by linearity rules between the component codes’ parity-check matrices. Furthermore, we present an implementation for the Generalized Construction D$'$ based on the partition of parity-check equations. This set-up is applied to an adapted PEG matrix construction algorithm, allowing for girth-maximizing parity-check matrices and also for a linear-time encoding rule.

3.1 A Sequential Description of Construction D’

In this section, we define Construction D$'$ using coset component codes with syndromes that contain the information of prior coset component codes. In particular, this alternative definition is important, because it affords the possibility of using sequential encoding and multistage decoding for Construction D′ lattices.

Proposition 3.1. *Let* Λ *be a lattice defined by* (2.7)*. Then*

$$\Lambda = \left\{ \mathbf{v} \in \mathbb{Z}^n : \mathbf{H}_\ell \mathbf{v}^{\mathrm{T}} \equiv \mathbf{0} \pmod{2^{\ell+1}},\ 0 \le \ell \le L-1 \right\}. \tag{3.1}$$

Proof. The proof is straightforward and is omitted. □

Lemma 3.1. *If* $\mathbf{H}_0, \ldots, \mathbf{H}_{L-1}$ *are matrices satisfying* (2.6)*, then these matrices have the property that, for all* $\mathbf{v} \in \mathbb{Z}^n$ *and all* $1 \le \ell \le L-1$*,*

$$\mathbf{H}_{\ell-1}\mathbf{v}^T \equiv \mathbf{0} \pmod{2^\ell} \implies \mathbf{H}_\ell \mathbf{v}^T \equiv \mathbf{0} \pmod{2^\ell}. \tag{3.2}$$

Proof. The proof follows immediately since, by definition, $\mathbf{H}_\ell$ is a submatrix of $\mathbf{H}_{\ell-1}$. □

We state the following result with slight generality since this will be useful later on. For convenience, define an empty summation as the all-zero vector with the appropriate size.

Theorem 3.1 (Sequential Encoding)**.** *Let* $C = \Lambda \cap [0, 2^L)^n$ *be a lattice code carved from a lattice* Λ *defined by* (3.1)*. Suppose that the corresponding matrices* $\mathbf{H}_0, \ldots, \mathbf{H}_{L-1}$ *have the property described in Lemma 3.1 and are such that* $\varphi(\mathbf{H}_\ell) \in \mathbb{F}_2^{m_\ell \times n}$*,* $\ell = 0, \ldots, L-1$*, are full-rank. Consider the following procedure:*

1. *Sequentially, for* $\ell = 0, 1, \ldots, L-1$*, obtain some vector* $\mathbf{c}_\ell \in C_\ell(\mathbf{s}_\ell)$*, where*

$$C_\ell(\mathbf{s}_\ell) \triangleq \left\{ \mathbf{v} \in \{0,1\}^n : \mathbf{H}_\ell \mathbf{v}^{\mathrm{T}} \equiv \mathbf{s}_\ell \pmod 2 \right\} \tag{3.3}$$

is a coset of the linear code $C_\ell = C_\ell(\mathbf{0})$ *and* $\mathbf{s}_\ell \in \{0,1\}^{m_\ell}$ *is such that*

$$\mathbf{s}_\ell \equiv \frac{-\mathbf{H}_\ell \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^{\mathrm{T}}}{2^\ell} \pmod 2. \tag{3.4}$$

2. *Finally, compute* $\mathbf{c} = \sum_{\ell=0}^{L-1} 2^\ell \mathbf{c}_\ell$*.*

The procedure described above is well-defined. Moreover, let C_{seq} *be the set of all possible vectors* $\mathbf{c} \in \mathbb{Z}^n$ *produced by this procedure. Then* $C_{seq} = C$*.*

Proof. First, we prove that the procedure is well-defined, i.e., that

$$\mathbf{H}_\ell \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^{\mathrm{T}} \equiv \mathbf{0} \pmod{2^\ell} \tag{3.5}$$

for $\ell > 0$, so that $\mathbf{s}_\ell$ can always be computed. Note that a solution for $\mathbf{c}_\ell$ always exists, since $\mathbf{H}_\ell$ is full-rank modulo 2. We proceed by induction. The base case $\ell = 0$ is already established by definition, since $\mathbf{s}_0 = \mathbf{0}$. Let $\ell > 0$ and suppose that $\mathbf{c}_0, \ldots, \mathbf{c}_{\ell-1}$ and $\mathbf{s}_{\ell-1}$ have been computed. Thus,

$$2^{\ell-1}\mathbf{s}_{\ell-1} + \mathbf{H}_{\ell-1} \sum_{i=0}^{\ell-2} 2^i \mathbf{c}_i^{\mathrm{T}} = 2^\ell \mathbf{a}$$

for some $\mathbf{a} \in \mathbb{Z}^{m_{\ell-1}}$. But $\mathbf{H}_{\ell-1}\mathbf{c}_{\ell-1}^T \equiv \mathbf{s}_{\ell-1} \pmod 2$, i.e., $\mathbf{H}_{\ell-1}\mathbf{c}_{\ell-1}^T = \mathbf{s}_{\ell-1} + 2\mathbf{b}$, for some $\mathbf{b} \in \mathbb{Z}^{m_{\ell-1}}$. It follows that

$$\begin{aligned} 2^\ell \mathbf{a} &= 2^{\ell-1}(\mathbf{H}_{\ell-1}\mathbf{c}_{\ell-1}^T - 2\mathbf{b}) + \mathbf{H}_{\ell-1} \sum_{i=0}^{\ell-2} 2^i \mathbf{c}_i^{\mathrm{T}} \\ &= -2^\ell \mathbf{b} + \mathbf{H}_{\ell-1} \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^{\mathrm{T}} \end{aligned}$$

or simply

$$\mathbf{H}_{\ell-1} \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^{\mathrm{T}} \equiv \mathbf{0} \pmod{2^\ell}.$$

Now (3.5) follows by Lemma 3.1, completing the induction.

Since the procedure is well-defined, the set $\mathcal{C}_{\mathrm{seq}}$ is also well-defined. We now prove that $\mathcal{C}_{\mathrm{seq}} = \mathcal{C}$. First, let $\mathbf{c} \in \mathcal{C}$. Then, for all $0 \leq \ell \leq L-1$,

$$\begin{aligned} \mathbf{0} &\equiv \mathbf{H}_\ell \mathbf{c}^T \pmod{2^{\ell+1}} \\ &\equiv \mathbf{H}_\ell \sum_{i=0}^{L-1} 2^i \mathbf{c}_i^T \pmod{2^{\ell+1}} \\ &\equiv \mathbf{H}_\ell 2^\ell \mathbf{c}_\ell^T + \mathbf{H}_\ell \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^T \pmod{2^{\ell+1}} \end{aligned}$$

and therefore

$$\begin{aligned} \mathbf{H}_\ell \mathbf{c}_\ell^T &\equiv \frac{-\mathbf{H}_\ell \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^T}{2^\ell} \pmod 2 \\ &\equiv \mathbf{s}_\ell \pmod 2. \end{aligned}$$

Thus, $\mathbf{c}$ can be generated by the procedure described, which implies $C \subseteq C_{\text{seq}}$. Now, let $\mathbf{c} \in C_{\text{seq}}$. For all $0 \leq \ell \leq L-1$, we have that

$$\mathbf{H}_\ell \mathbf{c}_\ell^{\mathrm{T}} \equiv \mathbf{s}_\ell \equiv \frac{-\mathbf{H}_\ell \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^{\mathrm{T}}}{2^\ell} \pmod 2$$

which implies

$$2^\ell \mathbf{H}_\ell \mathbf{c}_\ell^{\mathrm{T}} + \mathbf{H}_\ell \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i^{\mathrm{T}} = 2^{\ell+1}\mathbf{a}$$

for some $\mathbf{a} \in \mathbb{Z}^{m_\ell}$, and therefore

$$\begin{aligned}\mathbf{H}_\ell \mathbf{c}^T &\equiv \mathbf{H}_\ell \sum_{i=0}^{\ell} 2^i \mathbf{c}_i^T \pmod{2^{\ell+1}} \\ &\equiv \mathbf{0} \pmod{2^{\ell+1}}.\end{aligned}$$

It follows that $\mathbf{c} \in C$. This proves that $C_{\text{seq}} \subseteq C$ and thus $C = C_{\text{seq}}$, completing the proof. □

Remark: In the proof of Theorem 3.1, we have only assumed the *conclusion* of Lemma 3.1 (property (3.2)), not the *hypothesis* (definition (2.6)). This will also prove useful when generalizing Construction D′.

The essence of sequential encoding for Construction D′ is that, when encoding level ℓ, rather than using the original linear code C_ℓ, we encode using the *coset code* $C_\ell(\mathbf{s}_\ell)$ defined by the *syndrome* $\mathbf{s}_\ell$, which is computed based on the codewords from the previous levels. Another important point is that we operate on mod 2, and not on mod $2^{\ell+1}$, which is why non-trivial cosets are created. Thus, at each level, encoding can be performed entirely using a binary code.

The following result has been used in the literature (e.g., [9, 10]) without an explicit proof.[1] Here, the proof follows immediately from Theorem 3.1.

Corollary 3.1. *Let C satisfy the conditions of Theorem 3.1, and let C_ℓ be the null space of $\varphi(\mathbf{H}_\ell)$, for $\ell = 0, \ldots, L-1$. Then*

$$|C| = |C_0| \cdot \cdots \cdot |C_{L-1}| \tag{3.6}$$

[1] The classical result in [6, Chapter 8, Theorem 14] assumes that "some rearrangement of $\mathbf{h}_1, \ldots, \mathbf{h}_{m_0}$ forms the rows of an upper triangular matrix," which allows the linear congruences to be independently solved. However, this assumption is omitted in the definition of Construction D' that commonly appears in the literature, such as in [9, 10, 34] and here. While it is always possible to find some $\mathbf{h}_1, \ldots, \mathbf{h}_{m_0}$ of this form given a family of nested codes, an explicit proof was lacking that the result still holds even for specific $\mathbf{h}_1, \ldots, \mathbf{h}_{m_0}$ that are not of this form.

and therefore

$$R(C) = R(C_0) + \cdots + R(C_{L-1}). \tag{3.7}$$

Example 3.1. *Let C be the code described in Theorem 3.1 for $L = 3$ and matrices*

$$\mathbf{H}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}.$$

We construct $\mathbf{c} \in C$ by sequential encoding through the vectors $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2 \in \{0, 1\}^4$. First, we select some $\mathbf{c}_0$ satisfying

$$\mathbf{H}_0\mathbf{c}_0^{\mathrm{T}} \equiv \mathbf{0} \pmod 2$$

for instance, $\mathbf{c}_0 = (1, 1, 1, 1)$. Then, we compute $\mathbf{s}_1 = -\mathbf{H}_1\mathbf{c}_0^{\mathrm{T}}/2 \bmod 2 = (0, 1)^{\mathrm{T}}$ and select some $\mathbf{c}_1$ satisfying

$$\mathbf{H}_1\mathbf{c}_1^{\mathrm{T}} \equiv \mathbf{s}_1 \pmod 2$$

for instance, $\mathbf{c}_1 = (0, 1, 1, 0)$. Next, we compute $\mathbf{s}_2 = -(\mathbf{H}_2 2\mathbf{c}_1^{\mathrm{T}} + \mathbf{H}_2\mathbf{c}_0^{\mathrm{T}})/4 \bmod 2 = 0$ and select some $\mathbf{c}_2$ satisfying

$$\mathbf{H}_2\mathbf{c}_2^{\mathrm{T}} \equiv \mathbf{s}_2 \pmod 2$$

for instance, $\mathbf{c}_2 = (0, 0, 1, 1)$. Finally, we obtain

$$\begin{aligned} \mathbf{c} &= \mathbf{c}_0 + 2\mathbf{c}_1 + 4\mathbf{c}_2 \\ &= (1, 1, 1, 1) + (0, 2, 2, 0) + (0, 0, 4, 4) \\ &= (1, 3, 7, 5). \end{aligned}$$

Since $\mathbf{c} = (1, 3, 7, 5)$ satisfies all conditions in (3.1), we confirm that $\mathbf{c} \in C$. □

3.2 Encoding and Decoding

In this section, we discuss conditions under which an LDPC lattice code can be encoded and decoded with constant per-bit complexity. Throughout the section, let C be an L-level LDPC lattice code with component codes C_ℓ defined by parity-check matrices $\mathbf{H}_\ell \in \{0,1\}^{m_\ell \times n}$, $\ell = 0, \ldots, L-1$.

3.2.1 Systematic Encoding

Encoding of C consists of bijectively mapping a tuple of message vectors $(\mathbf{u}_0, \ldots, \mathbf{u}_{L-1}) \in \{0,1\}^{k_0} \times \cdots \times \{0,1\}^{k_{L-1}}$ to a codeword $\mathbf{c} = \sum_{\ell=0}^{L-1} 2^\ell \mathbf{c}_\ell \in C$. We say that the encoding is systematic if, for each ℓ, there exists some permutation matrix $\mathbf{T}_\ell$ such that, for all $\mathbf{u}_\ell$, we can express $\mathbf{c}_\ell = \begin{bmatrix} \mathbf{u}_\ell & \mathbf{p}_\ell \end{bmatrix} \mathbf{T}_\ell$, for some $\mathbf{p}_\ell \in \{0,1\}^{m_\ell}$.

Let $0 \leq \ell < L$. From Theorem 3.1, we know that we must have $\mathbf{c}_\ell \in C_\ell(\mathbf{s}_\ell)$, where $\mathbf{s}_\ell$ is computed from $\mathbf{c}_0, \ldots, \mathbf{c}_{\ell-1}$ using (3.4). Note that $\mathbf{s}_\ell$ can always be computed in $O(n)$ operations, since $\mathbf{H}_\ell$ is sparse. Thus, we focus on encoding the coset code $C_\ell(\mathbf{s}_\ell)$, for any $\mathbf{s}_\ell$.

Let $\mathbf{H}_\ell$ be denoted by $\mathbf{H}_\ell = \begin{bmatrix} \mathbf{H}_\ell^u & \mathbf{H}_\ell^p \end{bmatrix} \mathbf{T}_\ell$, where $\mathbf{H}_\ell^p \in \{0,1\}^{m_\ell \times m_\ell}$ is invertible over $\mathbb{F}_2$. Note that this can always be enforced by properly choosing $\mathbf{T}_\ell$, since $\mathbf{H}_\ell$ is assumed to be full-rank over $\mathbb{F}_2$. Finding $\mathbf{c}_\ell \in C_(\mathbf{s}_\ell)$ amounts to finding $\mathbf{p}_\ell \in \{0,1\}^{m_\ell}$ such that

$$\mathbf{H}_\ell^u \mathbf{u}_\ell^\mathrm{T} + \mathbf{H}_\ell^p \mathbf{p}_\ell^\mathrm{T} = \mathbf{H}_\ell \mathbf{c}_\ell^\mathrm{T} \equiv \mathbf{s}_\ell \pmod 2$$

or, equivalently, such that

$$\mathbf{H}_\ell^p \mathbf{p}_\ell^\mathrm{T} \equiv \mathbf{s}_\ell - \mathbf{H}_\ell^u \mathbf{u}_\ell^\mathrm{T} \pmod 2.$$

Note that $\mathbf{H}_\ell^u \mathbf{u}_\ell^\mathrm{T}$ can always be computed in $O(n)$, since $\mathbf{H}_\ell$ is sparse. Thus, we have proved the following result.

Proposition 3.2. *Let $\mathbf{s}'_\ell = \mathbf{s}_\ell - \mathbf{H}_\ell^u \mathbf{u}_\ell^\mathrm{T}$. Encoding of C can be done in $O(Ln)$ operations if the system $\mathbf{H}_\ell^p \mathbf{p}_\ell^\mathrm{T} \equiv \mathbf{s}'_\ell \pmod 2$ can be solved in $O(n)$ for all ℓ.*

One example situation where the condition of Proposition 3.2 holds, as shown in [39], is when each $\mathbf{H}_\ell^p$ is in approximate lower triangular (ALT) form, i.e.,

$$\mathbf{H}_\ell^p = \begin{bmatrix} \mathbf{B} & \mathbf{L} \\ \mathbf{D} & \mathbf{E} \end{bmatrix} \tag{3.8}$$

where $\mathbf{L} \in \{0,1\}^{(m_\ell - g)\times(m_\ell - g)}$ is lower triangular with ones along the diagonal and g, called the *gap* of the ALT form, is $O(1)$.

Thus, if each $\mathbf{H}_\ell$ satisfies (3.8) (up to row/column permutations), then the lattice code C admits systematic encoding with complexity $O(Ln)$.

3.2.2 Multistage Decoding

Let $\mathbf{c} = \sum_{\ell=0}^{L-1} 2^\ell \mathbf{c}_\ell \in C$ be the transmitted codeword, where $\mathbf{c}_\ell \in \{0,1\}^n$, and let $\mathbf{r} = \mathbf{c} + \mathbf{z} \bmod 2^L\mathbb{Z}^n$ be the received vector, where $\mathbf{z}$ is a noise vector of variance σ^2 per component.

Multistage decoding of C is inspired by [8]. Suppose that the vectors $\mathbf{c}_\ell$ have been correctly decoded for $i = 0, 1, \ldots, \ell - 1$. We compute

$$\mathbf{r}_\ell = \frac{\mathbf{r} - \sum_{i=0}^{\ell-1} 2^i \mathbf{c}_i}{2^\ell} \bmod 2 \tag{3.9}$$

$$= \mathbf{c}_\ell + \frac{\mathbf{z}}{2^\ell} \bmod 2. \tag{3.10}$$

This may be interpreted as the transmission of $\mathbf{c}_\ell \in C_\ell(\mathbf{s}_\ell)$ through a modulo-2 channel subject to additive noise $\mathbf{z}/2^\ell$. In particular, maximum-likelihood decoding on this channel would be given by $\hat{\mathbf{c}}_\ell = \operatorname{argmax}_{\mathbf{c}_\ell \in C_\ell(\mathbf{s}_\ell)} p(\mathbf{r}_\ell|\mathbf{c}_\ell)$. It should be emphasized that re-encoding is not needed for multistage decoding, only the ability to decode a coset code. Thus, provided that each $\mathbf{s}_\ell$ can be efficiently computed from the previous levels (which is always the case for LDPC lattices), efficient encoding is *not* needed for efficient multistage decoding.

To obtain low complexity decoding with near optimum performance, the iterative Belief Propagation algorithm can be used, which has $O(n)$ complexity for sparse matrices and a limited number of iterations. The algorithm has as its input a vector $\mathbf{LLR} \in \mathbb{R}^n$ with the log-likelihood ratio (LLR) of each component $r_{\ell j}$ of the received vector $\mathbf{r}_\ell$, defined as

$$\mathrm{LLR}_j = \ln\left(\frac{p(r_{\ell j}|c_{\ell j} = 0)}{p(r_{\ell j}|c_{\ell j} = 1)}\right), \quad j = 1, \ldots, n. \tag{3.11}$$

However, this algorithm assumes codewords $\mathbf{c}_\ell$ belonging to a linear code, as opposed to an affine code $C_\ell(\mathbf{s}_\ell)$.

We can exploit this algorithm for the problem at hand using the lengthened linear code $C'_\ell \subseteq \mathbb{F}_2^{n+m_\ell}$ defined by the parity-check matrix $\mathbf{H}'_\ell = \begin{bmatrix} -\mathbf{I} & \mathbf{H}_\ell \end{bmatrix}$, which remains sparse. In this case, the admissible codewords must be restricted to the form $\mathbf{c}'_\ell = \begin{bmatrix}\mathbf{s}_\ell & \mathbf{c}_\ell\end{bmatrix}$, so that

$$\mathbf{H}'_\ell \mathbf{c}'^{\mathrm{T}}_\ell \equiv \mathbf{0} \pmod 2. \tag{3.12}$$

In order to impose this constraint, it suffices to provide as an input LLR vector the vector given as

$$\mathbf{LLR}' = \begin{bmatrix}(1-2\mathbf{s}_\ell)\cdot\infty & \mathbf{LLR}\end{bmatrix} \tag{3.13}$$

where the LLR value ∞ $(-\infty)$ indicates certainty that the corresponding codeword symbol is equal to 0 (1).

We conclude that the decoding of C can be realized with complexity $O(Ln)$. By means of the union bound, the probability of error satisfies

$$\begin{aligned} P_e(C,\sigma^2) \leq P_e(C_0,\sigma^2) + P_e\left(C_1,(\sigma/2)^2\right) + \cdots + \\ + P_e\left(C_{L-1},(\sigma/2^{L-1})^2\right) \end{aligned} \tag{3.14}$$

where, for $0 \leq \ell \leq L-1$, $P_e(C_\ell,(\sigma/2^\ell)^2)$ is the probability of error of C_ℓ on channel (3.10).

3.2.3 Multistage Decoding with Re-encoding

Linear-time decoding can also be proved in a more general way, without relying on a specific algorithm for the coset codes, under the assumption of linear-time re-encoding of the all-zero vector.

Proposition 3.3. *Decoding of C can be done in $O(Ln)$ operations if each linear code C_ℓ admits linear-time decoding and the condition of Proposition 3.2 is satisfied.*

Proof. For $\ell = 0,\ldots,L-1$, let $\mathbf{v}_\ell \in C_\ell(\mathbf{s}_\ell)$ be the coset codeword corresponding to systematic encoding of the all-zero message vector $\mathbf{0}$ of length k_ℓ and let $\mathbf{c}'_\ell \in C_\ell$ be the codeword corresponding to systematic encoding of the message vector $\mathbf{u}_\ell \in \{0,1\}^{k_\ell}$. It follows that $\mathbf{c}_\ell \in C_\ell(\mathbf{s}_\ell)$, the coset codeword corresponding to the systematic encoding of $\mathbf{u}_\ell$, is given by

$$\mathbf{c}_\ell = \mathbf{c}'_\ell + \mathbf{v}_\ell \bmod 2. \tag{3.15}$$

Now, we can modify the multistage decoding procedure to compute

$$\mathbf{r}'_\ell = \mathbf{r}_\ell - \mathbf{v}_\ell \bmod 2 \tag{3.16}$$

$$= \mathbf{c}'_\ell + \frac{\mathbf{z}}{2^\ell} \bmod 2 \tag{3.17}$$

then decode $\mathbf{c}'_\ell \in \mathcal{C}_\ell$ and finally obtain $\mathbf{c}_\ell$ with (3.15). Note that $\mathbf{v}_\ell$ can be computed by choosing $\mathbf{s}'_\ell = \mathbf{s}_\ell$ in Proposition 3.2. Since all the steps involved are $O(n)$, the result follows.

3.3 A Generalization of Construction D′

A significant limitation of Construction D′ is the requirement that not only the component codes but also their corresponding parity-check matrices $\mathbf{H}_\ell$ be nested, i.e., that $\mathbf{H}_\ell$ be a submatrix of $\mathbf{H}_{\ell-1}$. This constraint complicates the design of LDPC codes as it requires, for instance, that the average column weight of $\mathbf{H}_{\ell-1}$ be strictly (and often significantly) higher than that of $\mathbf{H}_\ell$, conflicting with the optimum design of LDPC codes for their corresponding target rates. In principle, one could eliminate this constraint entirely and redefine Construction D′ by means of expression (3.1). However, with that approach there would be no guarantee of the cardinality of $\mathcal{C}$ (as given by (3.6)), let alone the possibility of sequential encoding, thus compromising essential properties of the construction. The reason is that a congruence modulo $2^{\ell+1}$ in (3.1) applies not only to level ℓ but also to all levels $i < \ell$ through a reduction modulo 2^{i+1}. Thus, sequential encoding is not possible unless these new congruences for previous levels are completely redundant. This idea is captured by Lemma 3.1, which is the fundamental ingredient enabling sequential encoding and the guarantee of cardinality as an immediate consequence.

However, requiring that $\mathbf{H}_\ell$ be a submatrix of $\mathbf{H}_{\ell-1}$ is simple, but it is not the only way of satisfying Lemma 3.1. Instead, we can relax the nesting constraint on matrices $\mathbf{H}_\ell$ by enforcing the more general constraint

$$\mathbf{H}_\ell \equiv \mathbf{F}_\ell \mathbf{H}_{\ell-1} \pmod{2^\ell} \tag{3.18}$$

for some integer matrix $\mathbf{F}_\ell$, from which Lemma 3.1 immediately follows. Clearly, requiring that $\mathbf{H}_\ell$ be a submatrix of $\mathbf{H}_{\ell-1}$ is a special case of this constraint.

Definition 3.1 (Generalized Construction D′). *Let the matrices* $\mathbf{H}_\ell \in \mathbb{Z}^{m_\ell \times n}$, $\ell = 0, \dots, L-1$, *be such that*

1. $\varphi(\mathbf{H}_\ell)$ *is full-rank, for* $\ell = 0, \dots, L-1$*;*

2. $\mathbf{H}_\ell \equiv \mathbf{F}_\ell\, \mathbf{H}_{\ell-1} \pmod{2^\ell}$*, for some* $\mathbf{F}_\ell \in \mathbb{Z}^{m_\ell \times m_{\ell-1}}$*, for* $\ell = 1, \dots, L-1$.

The lattice

$$\Lambda = \left\{ \mathbf{v} \in \mathbb{Z}^n : \mathbf{H}_\ell \mathbf{v}^{\mathrm{T}} \equiv 0 \pmod{2^{\ell+1}},\ 0 \le \ell \le L-1 \right\}$$

is said to be obtained by the Generalized Construction D′ applied to $\mathbf{H}_0, \dots, \mathbf{H}_{L-1}$. *Equivalently, we can express* Λ *as* $\Lambda = \mathcal{C} + 2^L\,\mathbb{Z}^n$*, where* $\mathcal{C} = \Lambda \cap [0, 2^L)^n$ *is a lattice code.*

It follows immediately that the set Λ defined above is indeed a lattice.

The emphasis of Definition 3.1 is on the parity-check matrices $\mathbf{H}_\ell$, rather than on the component codes. The interpretation based on nested component codes can be reestablished by taking $\mathcal{C}_\ell \subseteq \{0,1\}^n$ to be such that $\varphi(\mathcal{C}_\ell) \subseteq \mathbb{F}_2^n$ is the null space of $\varphi(\mathbf{H}_\ell) \in \mathbb{F}_2^{m_\ell \times n}$. Clearly, as a consequence of (3.18), we have $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_{L-1}$.

The main result of this section is given by the following theorem.

Theorem 3.2. *Let* $\mathcal{C}$ *be a lattice code satisfying Definition 3.1. Then* $\mathcal{C}$ *admits sequential encoding according to Theorem 3.1. Moreover,* $|\mathcal{C}| = |\mathcal{C}_0| \cdot \cdots \cdot |\mathcal{C}_{L-1}|$.

Proof. The proof follows immediately since Theorem 3.1 only relies on (3.1) and Lemma 3.1. □

It is easy to see that all of the results yielded from redescribing Construction D′ in order to make it amenable to sequential encoding are valid for the Generalized Construction D′.

Example 3.2. *Let*

$$\mathbf{F}_1 = \begin{bmatrix} 2 & 7 & 4 \\ 11 & 9 & 6 \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} 3 & 5 \end{bmatrix}$$

be arbitrarily chosen integer matrices, and let

$$\mathbf{H}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{H}_1 = \mathbf{F}_1\mathbf{H}_0 \bmod 2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{H}_2 = \mathbf{F}_2\mathbf{H}_1 \bmod 4 = \begin{bmatrix} 3 & 1 & 3 & 1 \end{bmatrix}.$$

It is easy to check that $\varphi(\mathbf{H}_0)$, $\varphi(\mathbf{H}_1)$ and $\varphi(\mathbf{H}_2)$ are full-rank. Generalized Construction D′ applied to matrices $\mathbf{H}_0$, $\mathbf{H}_1$, and $\mathbf{H}_2$ produces a lattice code C with $L = 3$ levels and rate $R = \frac{1}{n}\log_2|C| = \frac{1}{4}\log_2(2^{1+2+3}) = 1.5$ bits per dimension.

Note that $\mathbf{H}_2$ is non-binary, as may be any of the matrices $\mathbf{H}_\ell$, for $\ell \geq 2$. Nevertheless, all the encoding and decoding operations are still performed over $\mathbb{F}_2$ with $\varphi(\mathbf{H}_\ell)$, except for the computation of the syndrome $\mathbf{s}_\ell$ in (3.4). □

The following theorem shows that the only real requirement for Generalized Construction D′ is that the linear component codes be nested. Any (full-rank) parity-check matrices for these codes, nested or not, may be used in the construction, although we may need to lift them to non-binary integers.

Theorem 3.3. *Let $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{L-1} \subseteq \{0,1\}^n$ be nested linear codes with parity-check matrices $\bar{\mathbf{H}}_\ell \in \{0,1\}^{m_\ell \times n}$, $\ell = 0, \ldots, L-1$, respectively. Then there exist matrices $\mathbf{H}_\ell \in \mathbb{Z}^{m_\ell \times n}$, $\ell = 0, \ldots, L-1$, satisfying* (3.18) *and such that $\mathbf{H}_\ell \equiv \bar{\mathbf{H}}_\ell \pmod 2$.*

Proof. Let $\mathbf{H}_0 = \bar{\mathbf{H}}_0$. For $\ell = 1, \ldots, L-1$, we proceed by induction. Assume that $\mathbf{H}_{\ell-1} \equiv \bar{\mathbf{H}}_{\ell-1} \bmod 2$, which is true for $\ell = 1$. Since $C_{\ell-1} \subseteq C_\ell$, we have that $C_\ell^\perp \subseteq C_{\ell-1}^\perp$, where $C_\ell^\perp$ denotes the dual code of C_ℓ. This implies that there exists some $\mathbf{F}_\ell \in \{0,1\}^{m_\ell \times m_{\ell-1}}$ such that $\bar{\mathbf{H}}_\ell \equiv \mathbf{F}_\ell \bar{\mathbf{H}}_{\ell-1} \bmod 2$. Let $\mathbf{H}_\ell = \mathbf{F}_\ell \mathbf{H}_{\ell-1} \bmod 2^\ell$, which automatically satisfies (3.18). It follows that

$$\begin{aligned} \mathbf{H}_\ell &\equiv \mathbf{F}_\ell \mathbf{H}_{\ell-1} \bmod 2 && (3.19)\\ &\equiv \mathbf{F}_\ell \bar{\mathbf{H}}_{\ell-1} \bmod 2 && (3.20)\\ &\equiv \bar{\mathbf{H}}_\ell \bmod 2 && (3.21) \end{aligned}$$

completing the induction. □

3.3.1 Comparison with Construction D′

In the remainder of this section, we compare Construction D′ and Generalized Construction D′ under two common perspectives, which differ essentially on whether complexity is taken into account.

As a Codebook Construction

From a purely theoretical (or geometric) perspective, a code or lattice is defined as a set of points in some space, i.e., as a codebook. In this case, one is concerned with geometric properties of the codebook such as minimum distance or probability of error under minimum distance decoding. This is the approach implicit in classical descriptions of linear codes and lattices [6] and, for instance, in the comparison between Constructions D and D′ in [34].

From this perspective, Generalized Construction D′ is strictly more general than Construction D′, as there exist examples of the former that cannot be described by the latter. Considering an L-level Generalized Construction D′ lattice with matrices $\mathbf{H}_0, \dots, \mathbf{H}_{L-1}$, such examples can always be produced if $L > 2$ or if $\mathbf{H}_1 \bmod 4$ is *non-binary*. Otherwise, if $L = 2$ and $\mathbf{H}_1 \bmod 4$ is binary, then the same codebook can be produced using Construction D′ with matrices $\bar{\mathbf{H}}_0$ and $\bar{\mathbf{H}}_1$, where $\bar{\mathbf{H}}_1 = \mathbf{H}_1 \bmod 2$ and $\bar{\mathbf{H}}_0$ is any binary matrix that defines C_0 and contains $\mathbf{H}_1$ as a submatrix. This follows since, as can be seen from Proposition 3.1 and Theorem 3.1, both constructions depend only on C_0 and $\mathbf{H}_\ell \bmod 2^{\ell+1}$, $\ell = 1, \dots, L-1$.

Examples of the non-equivalent cases are shown next.

Example 3.3. *Consider again Examples 3.1 and 3.2, but let all matrices, syndromes, codewords and lattice code of Example 3.1 be denoted with an overline, such as* $\bar{\mathbf{H}}_\ell$, $\bar{\mathbf{s}}_\ell$, $\bar{\mathbf{c}}_\ell$ *and* $\bar{C}$, *to distinguish them from those of Example 3.2. Clearly, the underlying binary codes are the same, namely,*

$$C_0 = \langle (1,1,1,1) \rangle$$

$$C_1 = \langle (1,1,1,1), (1,0,1,0) \rangle$$

$$C_2 = \langle (1,1,1,1), (1,0,1,0), (0,0,1,1) \rangle.$$

However, in contrast to $\bar{\mathbf{H}}_0$, $\bar{\mathbf{H}}_1$ *and* $\bar{\mathbf{H}}_2$, *neither* $\mathbf{H}_1$ *is a submatrix of* $\mathbf{H}_0$, *nor* $\mathbf{H}_2$ *is a submatrix of* $\mathbf{H}_1$. *We will construct a codeword* $\mathbf{c} = \mathbf{c}_0 + 2\mathbf{c}_1 + 4\mathbf{c}_2 \in C$ *such that* $\mathbf{c} \notin \bar{C}$.

Let $\mathbf{c}_0 = \bar{\mathbf{c}}_0 = (1,1,1,1)$. *Then* $\mathbf{s}_1 = (1,1)^T$, *from which we may select* $\mathbf{c}_1 = (1,1,0,0) \in C_1(\mathbf{s}_1)$. *However, as noted before,* $\bar{\mathbf{s}}_1 = (0,1)^T \neq \mathbf{s}_1$, *which implies that* $\bar{\mathbf{c}}_1$ *must be chosen from a different coset of* C_1 *and therefore it cannot equal* $\mathbf{c}_1$. *Thus, necessarily* $\mathbf{c} \notin \bar{C}$, *regardless of*

$\mathbf{c}_2$.

Since $\mathbf{H}_1$ is binary, one may attempt to work around this problem by redefining $\bar{\mathbf{H}}_1 = \mathbf{H}_1$ and

$$\bar{\mathbf{H}}_0 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

so that $\bar{\mathbf{H}}_1$ is still a submatrix of $\bar{\mathbf{H}}_0$, but now $\bar{\mathbf{s}}_1 = \mathbf{s}_1$. Thus, we can select $\bar{\mathbf{c}}_1 = \mathbf{c}_1$. However, $\bar{\mathbf{H}}_2$ must be chosen as a submatrix of $\bar{\mathbf{H}}_1$ and no such choice can produce C_2; for instance, the vector $(0, 0, 1, 1) \in C_2$ will never be in the null space of $\varphi(\bar{\mathbf{H}}_2)$.

More generally, the code C_2 defined by $\mathbf{H}_2$ can only be produced with $\bar{\mathbf{H}}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$, which in turn implies that $(1, 1, 1, 1)$ must be a row of $\bar{\mathbf{H}}_1$. Hence, $\bar{\mathbf{c}}_0$ must produce a syndrome $\bar{\mathbf{s}}_1$ with at least one zero entry, and thus $\bar{\mathbf{s}}_1 \neq \mathbf{s}_1$. It follows that necessarily $\bar{C} \neq C$, i.e., the lattice code C cannot be produced by Construction D′. □

Example 3.4. *Let $L = 2$ and*

$$\mathbf{F}_1 = \begin{bmatrix} 3 & 1 \end{bmatrix}$$

$$\mathbf{H}_0 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\mathbf{H}_1 = \mathbf{F}_1\mathbf{H}_0 \bmod 4 = \begin{bmatrix} 0 & 1 & 0 & 3 \end{bmatrix}.$$

Clearly, we still have $\mathbf{H}_1 \equiv \mathbf{F}_1\mathbf{H}_0 \pmod 2$. Let $C \subseteq [0, 4)^4$ be the lattice code produced by Generalized Construction D′ with matrices $\mathbf{H}_0$ and $\mathbf{H}_1$. The underlying nested codes are

$$C_0 = \langle (0, 0, 1, 0), (1, 1, 0, 1) \rangle$$

$$C_1 = \langle (0, 0, 1, 0), (1, 1, 0, 1), (1, 0, 0, 0) \rangle.$$

Let $\bar{\mathbf{H}}_0 \in \{0, 1\}^{2\times4}$ and $\bar{\mathbf{H}}_1 \in \{0, 1\}^{1\times4}$ be nested matrices that define the same codes C_0 and C_1, respectively, and let $\bar{C} \subseteq [0, 4)^4$ be the corresponding lattice code produced by Construction D′

with $\bar{\mathbf{H}}_0$ *and* $\bar{\mathbf{H}}_1$. *Clearly, there is a single possibility for* $\bar{\mathbf{H}}_1$, *namely,*

$$\bar{\mathbf{H}}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}.$$

Let $\mathbf{c}_0 = (1, 1, 1, 1)$. *Then the corresponding syndrome at level* 1, *when computed with* $\mathbf{H}_1$, *is equal to* $\mathbf{s}_1 = 0$ *but, when computed with* $\bar{\mathbf{H}}_1$, *it is equal to* $\bar{\mathbf{s}}_1 = 1$. *Thus, for any valid choice of* $\bar{C}$, *we have* $\mathbf{c} = (1, 1, 1, 1) \in C$ *but* $\mathbf{c} \notin \bar{C}$. □

It is worth pointing out that the greater flexibility of Generalized Construction D′ does not offer any advantage in terms of *theoretical* performance (with unbounded complexity) under multistage decoding. This is because the error probability in (3.14) depends solely on the component codes $C_0, \ldots, C_{L-1}$, and we can always produce a Construction D′ lattice with the same component codes. In this case, as shown in the examples above, the only difference between the two constructions is in the syndrome calculation at each level.

More generally, any multilevel codes that share the same component codes (including Construction D lattices) will also have the same error probability under multistage decoding if we allow unbounded complexity.

As a Construction of a Coding Scheme

From a practical (or complexity-constrained) perspective, a coding scheme consists of a codebook, an encoding function and a decoding function, and one is concerned, in particular, with the performance of the scheme under a certain complexity. The decoder thus plays a key role in the construction of the scheme. This is the approach implicit in descriptions of modern codes such as Turbo and LDPC codes [40]; in particular, an LDPC code is described not simply as a set of codewords, but through some specific parity-check matrix that induces a convenient decoder structure.

This second perspective is the focus of this thesis and is what motivates our definition of Generalized Construction D′. From this perspective, assuming that the decoder is specified by the parity-check matrices used in the lattice construction, Generalized Construction D′ is indeed more general than Construction D′, since it introduces fewer constraints on the choice of the parity-check matrices. This increased flexibility can translate into a better performance if the decoder is sensitive to the choice of the parity-check matrices, which is the case of LDPC lattices under a multistage Belief Propagation decoder. Numerical examples of their difference in performance are shown in Section 4.2.

One way to artificially match the performance of the two constructions may be to first design a Generalized Construction D′ lattice with matrices $\mathbf{H}_0, \ldots, \mathbf{H}_{L-1}$ and then create a Construction D′ lattice with nested matrices $\bar{\mathbf{H}}_0, \ldots, \bar{\mathbf{H}}_{L-1}$ that correspond to the same component codes. Then, use $\bar{\mathbf{H}}_0, \ldots, \bar{\mathbf{H}}_{L-1}$ for encoding and demapping from a codeword to a message vector, whilst using $\mathbf{H}_0, \ldots, \mathbf{H}_{L-1}$ solely for "denoising," i.e., for decoding from the received vector to a codeword. A clear disadvantage of this approach is that, for each component code, two distinct parity-check matrices must be stored and used, whereas, with Generalized Construction D′, a single matrix can be used in all encoding and decoding steps. Moreover, it is unclear whether the nested matrices $\bar{\mathbf{H}}_0, \ldots, \bar{\mathbf{H}}_{L-1}$ (of which we have less control) will have a convenient structure for efficient encoding and demapping. In other words, requiring the parity-check matrices to be nested is an unnecessary constraint of Construction D′, both in theory and in practice.

More fundamentally, Construction D′ was originally defined [5, 6] with a focus on minimum distance, i.e., on packing density, regardless of the availability of an efficient decoder. Therefore, allowing for a flexible choice of parity-check matrices was unnecessary. In contrast, the approach of Generalized Construction D′ allows the decoder structure, embodied by specific parity-check matrices, to be taken into account as part of the design. This is important for the design of LDPC component codes with good error correction performance.

3.4 Nested LDPC Codes by Check Splitting

In this section, we propose a method to construct suitable *binary* matrices $\mathbf{H}_0, \ldots, \mathbf{H}_{L-1}$ that satisfy the conditions of Generalized Construction D′. Our approach is to sequentially construct matrix $\mathbf{H}_{\ell-1}$ based on matrix $\mathbf{H}_\ell$, for $\ell = L-1, \ldots, 1$, assuming we are given the parity-check matrix $\mathbf{H}_{L-1}$ of the highest-rate code and the desired number of rows for the remaining matrices, namely, $m_{L-2}, \ldots, m_0$.

Our method guarantees that, for each new level, the column weights of the initial matrix are preserved, i.e., all component codes will share the same variable-node degree distribution. While this approach may not lead to an optimal design for all rates, it allows us to choose, for instance, all component codes to be variable-regular LDPC codes with variable-node degree $d_v = 3$, which we can expect to exhibit at least a reasonable performance. This choice is simply not available with the original Construction D′.

We also present variations of the basic method aimed at maximizing girth and at allowing linear-time encoding, inspired by the progressive edge growth (PEG) algorithm [15].

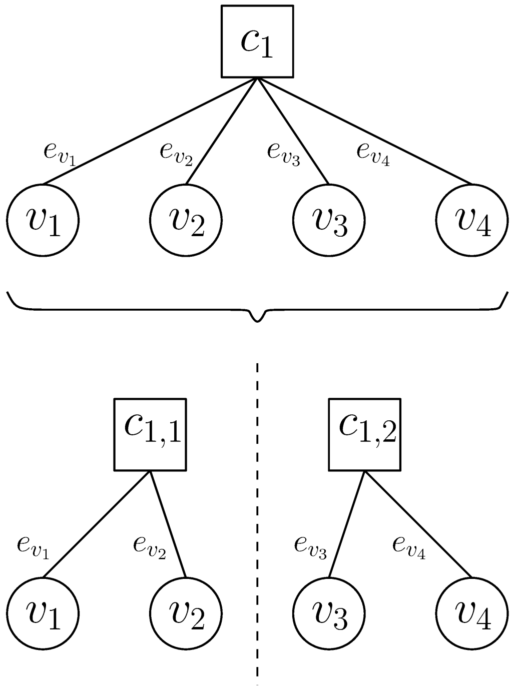

Figure 3.1: Example of check splitting. Check node c_1 is split into two check nodes $c_{1,1}$ and $c_{1,2}$. The number of edges is preserved and each edge remains incident on the same variable node.

For the remainder of the section, it suffices to consider the following problem: given a full-rank matrix $\mathbf{B} \in \{0,1\}^{b\times n}$ and a desired number of rows $m > b$, it is desired to construct a full-rank matrix $\mathbf{H} \in \{0,1\}^{m\times n}$ with the same sequence of column weights and such that $\mathbf{B} = \mathbf{FH}$, for some $\mathbf{F} \in \mathbb{Z}^{b\times m}$. Note that we require equality over $\mathbb{Z}$, which immediately implies the observance of equation (3.18) for any ℓ.

Since an equivalent representation of a binary matrix $\mathbf{H}$ is a Tanner graph (a bipartite graph, with m check nodes and n variable nodes, having $\mathbf{H} \in \{0,1\}^{m\times n}$ as incidence matrix), we use the two concepts interchangeably.

3.4.1 Check Splitting

We first describe a general method based on the partitioning (splitting) of parity-check equations, which we refer to as *check splitting*[2] for short. The basic idea is illustrated in Fig. 3.1, where a check node is split in two without changing the variable nodes on which the corresponding edges are incident.

For all $\mathbf{H} = [\mathbf{H}(i,j)] \in \{0,1\}^{m\times n}$, let $\mathcal{J}(\mathbf{H}) = (\mathcal{J}_1(\mathbf{H}),\dots,\mathcal{J}_m(\mathbf{H}))$, where each $\mathcal{J}_i(\mathbf{H}) = \{j \in \{1,\dots,n\} : \mathbf{H}(i,j) \neq 0\}$ is a set containing the indices of the nonzero entries of the ith

[2] Our definition of check splitting differs from that in [41], which introduces a variable node connecting the check nodes produced from splitting.

row of $\mathbf{H}$. Similarly, let $\mathcal{I}(\mathbf{H}) = (\mathcal{I}_1(\mathbf{H}), \ldots, \mathcal{I}_n(\mathbf{H}))$, where each $\mathcal{I}_j(\mathbf{H}) = \{i \in \{1, \ldots, m\} : \mathbf{H}(i, j) \neq 0\}$ is a set containing the indices of the nonzero entries of the jth column of $\mathbf{H}$.

Let $m \geq b$ and let $p : \{1, \ldots, m\} \to \{1, \ldots, b\}$ be a surjective mapping. A matrix $\mathbf{H} \in \{0, 1\}^{m \times n}$ is said to be obtained from $\mathbf{B} \in \{0, 1\}^{b \times n}$ by check splitting based on the parent mapping p if, for all $i = 1, \ldots, m$, the set $\{\mathcal{J}_i(\mathbf{H}) : i \in p^{-1}(k)\}$ forms a partition of $\mathcal{J}_k(\mathbf{B})$, where $p^{-1}(k) = \{i \in \{1, \ldots, m\} : p(i) = k\}$ is the preimage of k under p.

Clearly, check splitting preserves column weights, since every nonzero entry of $\mathbf{B}$ appears in $\mathbf{H}$ in the same column, although possibly in a different row. Moreover, it is easy to see that $\mathbf{B} = \mathbf{F}\mathbf{H}$, where $\mathbf{F} \in \{0, 1\}^{b \times m}$ is such that $\mathcal{J}(\mathbf{F}) = (p^{-1}(1), \ldots, p^{-1}(b))$, i.e., adding the rows of $\mathbf{H}$ with indices in $p^{-1}(k)$ gives precisely the kth row of $\mathbf{B}$.

Example 3.5. *Starting with*

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

we can partition it into

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

which, in turn, can be partitioned into

$$\mathbf{H}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

It is plain to check that $\mathbf{H}_1 = \mathbf{F}_1\mathbf{H}_0$ *and* $\mathbf{H}_2 = \mathbf{F}_2\mathbf{H}_1$*, where*

$$\mathbf{F}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

Note that all matrices are binary and that each column has the same weight, namely 1. □

A useful property of check splitting, which follows from [42, Lemma 8], is that it cannot

Algorithm 3.1 PEG-based check splitting

```
Input: $\mathbf{B} \in \{0,1\}^{b\times n}$, $m$
Output: $\mathbf{H} \in \{0,1\}^{m\times n}$
 1: Create the parent mapping $p$.
 2: Initialize $\mathbf{H} \leftarrow \mathbf{0}$.
 3: for $j = 1, \ldots, n$ do
 4:     for $k \in \mathcal{I}_j(\mathbf{B})$ do
 5:         $\mathcal{I} \leftarrow p^{-1}(k)$
 6:         $\mathcal{I} \leftarrow \{i \in \mathcal{I} : d_{\mathbf{H}}(i,j) = \max_{i'\in\mathcal{I}} d_{\mathbf{H}}(i',j)\}$
 7:         $\mathcal{I} \leftarrow \{i \in \mathcal{I} : |\mathcal{J}_i(\mathbf{H})| = \min_{i'\in\mathcal{I}} |\mathcal{J}_{i'}(\mathbf{H})|\}$
 8:         Choose some $i \in \mathcal{I}$ and set $\mathbf{H}(i,j) \leftarrow 1$.
 9:     end for
10: end for
```

reduce girth. Thus, in particular, **H** is guaranteed to be free of 4-cycles if **B** is so.

3.4.2 PEG-Based Check Splitting

While girth preservation is a desirable feature, one typically expects a lower-rate code to have a better cycle distribution than a higher-rate code of the same length, preferably a larger girth. This is possible in the check splitting procedure if checks are split in a way that breaks short cycles in which they are involved. More generally, it is conceivable that a judicious choice of check splits may increase the performance of the resulting code.

In the following, we propose a check splitting algorithm, inspired by the PEG algorithm [15], that attempts to maximize the girth of the resulting matrix. For all $i = 1, \ldots, m$ and all $j = 1, \ldots, n$, let $d_{\mathbf{H}}(i,j)$ denote the distance from check node i to variable node j in the Tanner graph induced by **H**, where $d_{\mathbf{H}}(i,j) = \infty$ if there is no path joining these nodes. As shown in Algorithm 3.1, the proposed method greedily processes each (k,j) non-zero entry of **B**, adding a corresponding entry in **H** in the same column j but in some child row $i \in p^{-1}(k)$ satisfying two goals: first, it should maximize the distance to variable node j of the resulting Tanner graph; second, if multiple possibilities remain, a check of lowest degree is chosen. The algorithm can be interpreted as a generalization of the PEG algorithm where, for each iteration, the set of allowed checks is restricted to $p^{-1}(k)$, as shown in line 5. It is worth noting that the original PEG algorithm is essentially recovered if this line is replaced by $\mathcal{I} \leftarrow \{1, \ldots, m\}$.

A secondary goal of the algorithm is to make the weights of the resulting rows as uniform as possible, which is also a desirable feature for a good LDPC code. However, some care must be taken to ensure that the parent mapping p is indeed suitable to result in concentrated weights,

Algorithm 3.2 Create the parent mapping for Algorithm 3.1

```
Input: $\mathbf{B} \in \{0,1\}^{b\times n}$, $m$
Output: $p : \{1,\ldots,m\} \to \{1,\ldots,b\}$
1: Set $p(i) = i$ for $i = 1,\ldots,b$.
2: for $i = b+1,\ldots,m$ do
3:     $\mathcal{K} \leftarrow \{1,\ldots,b\}$
4:     $\mathcal{K} \leftarrow \{k \in \mathcal{K} : \mu_p(k) = \max_{k'\in\mathcal{K}} \mu_p(k')\}$
5:     Choose some $k \in \mathcal{K}$ and set $p(i) \leftarrow k$.
6: end for
```

which is accomplished by Algorithm 3.2. Specifically, for each ith row of $\mathbf{H}$, this algorithm chooses as its parent row in $\mathbf{B}$ one that maximizes the metric

$$\mu_p(k) \triangleq \frac{|\mathcal{J}_k(\mathbf{B})|}{|p^{-1}(k)|+1} \tag{3.22}$$

which can be interpreted as the average weight of a corresponding child row after the parent assignment is made (i.e., a row is chosen that maximizes the resulting weight after a further split).

3.4.3 Triangular PEG-Based Check Splitting

A drawback of Algorithm 3.1 is that it generally does not produce matrices that have an approximate triangular structure. We propose a simple adaptation that ensures such a structure, thereby enabling efficient encoding. The algorithm takes a base matrix $\mathbf{B}$ in ALT form with gap g and returns a check-split matrix $\mathbf{H}$ in the same form and with the same gap g. Thus, $\mathbf{B}$ and $\mathbf{H}$ will have exactly the same encoding complexity.

For ease of notation, we assume that $\mathbf{B}$ is actually in approximate *upper* triangular form, which can be easily accomplished by left-right and up-down flipping of a matrix in ALT form. The resulting matrix $\mathbf{H}$ is similarly given in the same form.

The proposed method, which again takes inspiration from [15], is given in Algorithm 3.3. The differences to Algorithm 3.1 are essentially the inclusion of lines 5–9, which add a 1 in the gth subdiagonal, and the modification in line 11, which restricts the valid choices of child rows to those above the gth subdiagonal. The crucial assumption is that $p(g+j) \in \mathcal{I}_j(\mathbf{B})$ for all $j = 1,\ldots,m-g$, since a 1 can only be added in position $(g+j, j)$ of $\mathbf{H}$ if a 1 exists in position (k_0, j) of $\mathbf{B}$, where $k_0 = p(g+j)$ is the corresponding parent row. This property can be ensured during the creation of the parent mapping by a simple modification in line 3 of Algorithm 3.2,

Algorithm 3.3 Triangular PEG-based check splitting

```
Input: $\mathbf{B} \in \{0,1\}^{b \times n}$, $g$, $m$
Output: $\mathbf{H} \in \{0,1\}^{m \times n}$
Create the parent mapping $p$.
Initialize $\mathbf{H} \leftarrow \mathbf{0}$.
for $j = 1, \ldots, n$ do
    $\mathcal{K} \leftarrow \mathcal{I}_j(\mathbf{B})$
    if $j \le m - g$ then
        $\mathbf{H}(g + j, j) \leftarrow 1$
        $k_0 \leftarrow p(g + j)$
        $\mathcal{K} \leftarrow \mathcal{K} \setminus \{k_0\}$
    end if
    for $k \in \mathcal{K}$ do
        $\mathcal{I} \leftarrow p^{-1}(k) \cap \{1, \ldots, \min\{g + j - 1, m\}\}$
        $\mathcal{I} \leftarrow \{i \in \mathcal{I} : d_{\mathbf{H}}(i, j) = \max_{i' \in \mathcal{I}} d_{\mathbf{H}}(i', j)\}$
        $\mathcal{I} \leftarrow \{i \in \mathcal{I} : |\mathcal{J}_i(\mathbf{H})| = \min_{i' \in \mathcal{I}} |\mathcal{J}_{i'}(\mathbf{H})|\}$
        Choose some $i \in \mathcal{I}$ and set $\mathbf{H}(i, j) \leftarrow 1$.
    end for
end for
```

Algorithm 3.4 Create the parent mapping for Algorithm 3.3

```
Input: $\mathbf{B} \in \{0,1\}^{b \times n}$, $m$
Output: $p : \{1, \ldots, m\} \to \{1, \ldots, b\}$
Set $p(i) = i$ for $i = 1, \ldots, b$.
for $i = b + 1, \ldots, m$ do
    $\mathcal{K} \leftarrow \mathcal{I}_{i-g}(\mathbf{B})$
    $\mathcal{K} \leftarrow \{k \in \mathcal{K} : \mu_p(k) = \max_{k' \in \mathcal{K}} \mu_p(k')\}$
    Choose some $k \in \mathcal{K}$ and set $p(i) \leftarrow k$.
end for
```

as shown in Algorithm 3.4.

4 Simulation Results

This chapter discusses the design and word error probability performance of LDPC lattices with $L = 2$ coded levels. The analysis is performed for LDPC lattices constructed using check splitting, i.e., built with Generalized Construction D′. To serve as a benchmark, lattice design using the traditional definition of Construction D′ [5, 6] is also employed, emulating what was done in [9].

We use three different design requirements in order to assess the performance of the constructed lattices. We design lattices for:

1. an error probability $P_e \leq 10^{-5}$ and with code length $n = 1024$;
2. $P_e \leq 10^{-2}$ and $n = 1000$;
3. $P_e \leq 10^{-2}$ and $n = 10000$.

We compare the error performance of our codes to state-of-the-art, high-performance lattices built using either Construction D, Construction A, or a construction directly over $\mathbb{R}$, namely: polar lattices [13], LDA [28], GLD [4], and LDLC [29].

4.1 General Simulation Set-up

In all scenarios considered, we construct LDPC lattices $\Lambda = C + 4\mathbb{Z}^n$ with the Generalized Construction D′ applied to matrices $\mathbf{H}_0 \in \{0, 1\}^{m_0 \times n}$, and $\mathbf{H}_1 \in \{0, 1\}^{m_1 \times n}$, corresponding to the nested codes $C_0 \subseteq C_1$ of rates R_0 and R_1, respectively. Both codes are chosen to be variable-regular LDPC codes with variable-node degree $d_v = 3$. Matrix $\mathbf{H}_1$ is constructed via the triangular version of the PEG algorithm [15], modified to allow for a gap g, which is chosen to be $g = 22$, while $\mathbf{H}_0$ is obtained from $\mathbf{H}_1$ via triangular PEG-based check splitting with the same gap (Algorithm 3.3 with $g = 22$). Thus, C is guaranteed to have efficient encoding.

Transmission over a power-unconstrained AWGN channel with noise variance σ^2 is simulated using the approach of Subsection 2.3.1. Decoding of the lattice code C is performed as described in Subsection 3.2.2, where for each component code the Belief Propagation decoder performs a maximum of 50 iterations. Each simulation point is obtained after the occurrence of at least 100 word errors, except for points with word error rate (WER) below 10^{-6}, which were obtained with at least 50 word errors.

Decoding of the sublattice $\Lambda' = 4\mathbb{Z}^n$ is not simulated; instead, $P_e(4\mathbb{Z}^n, \sigma^2)$ is computed analytically from (2.5) and applied to (2.4), which is assumed to hold with equality. Note that, by combining (2.4) and (3.14), we have

$$P_e(\Lambda, \sigma^2) \leq P_e(C_0, \sigma^2) + P_e(C_1, (\sigma/2)^2) + P_e(4\mathbb{Z}^n, \sigma^2). \tag{4.1}$$

4.2 Design for $P_e \leq 10^{-5}$ and $n = 1024$

For our first example, we have used the same design parameters as the 2-level polar lattice of [13], namely $P_e \leq 10^{-5}$ and $n = 1024$. For the choice of m_0 and m_1, or, equivalently, R_0 and R_1, we have not made any attempt at optimizing for a target error probability, but simply adopted the same values obtained in [13], namely $R_0 = 0.23$ ($m_0 = 788$) and $R_1 = 0.90$ ($m_1 = 103$), in order to allow for a simpler comparison.

The rate design in [13] was done using the equal error probability rule [7] applied to the union-bound estimate of equation (4.1). Under this rule, the goal is to make the error probability of the three levels equal (in this case, equal to $10^{-5}/3$). Using (2.5), uncoded level 2 achieves $P_e = 10^{-5}/3$ at $\sigma = 0.3380$, yielding a design VNR of 2.34 dB.

Fig. 4.1 shows the word error rate as a function of the VNR for the Generalized Construction D′ LDPC lattice and for the polar lattice. As can be seen, at the design VNR, the performance of both lattices is comparable. In particular, the LDPC lattice attains WER = 10^{-5} at VNR = 2.2865 dB.

In order to better understand the performance of the LDPC lattice, Fig. 4.1 also shows the performance of each individual level computed without error propagation. As we can see, the fact that the slope of the curve becomes less steep after WER = 10^{-5} is due to the performance of level 1 and, especially, of level 2. Thus, the error rate of level 2 induces a lower bound on the performance of the LDPC lattice. Note that the polar lattice has the exact same uncoded level and the same fundamental volume, being therefore limited by the same lower bound. As depicted

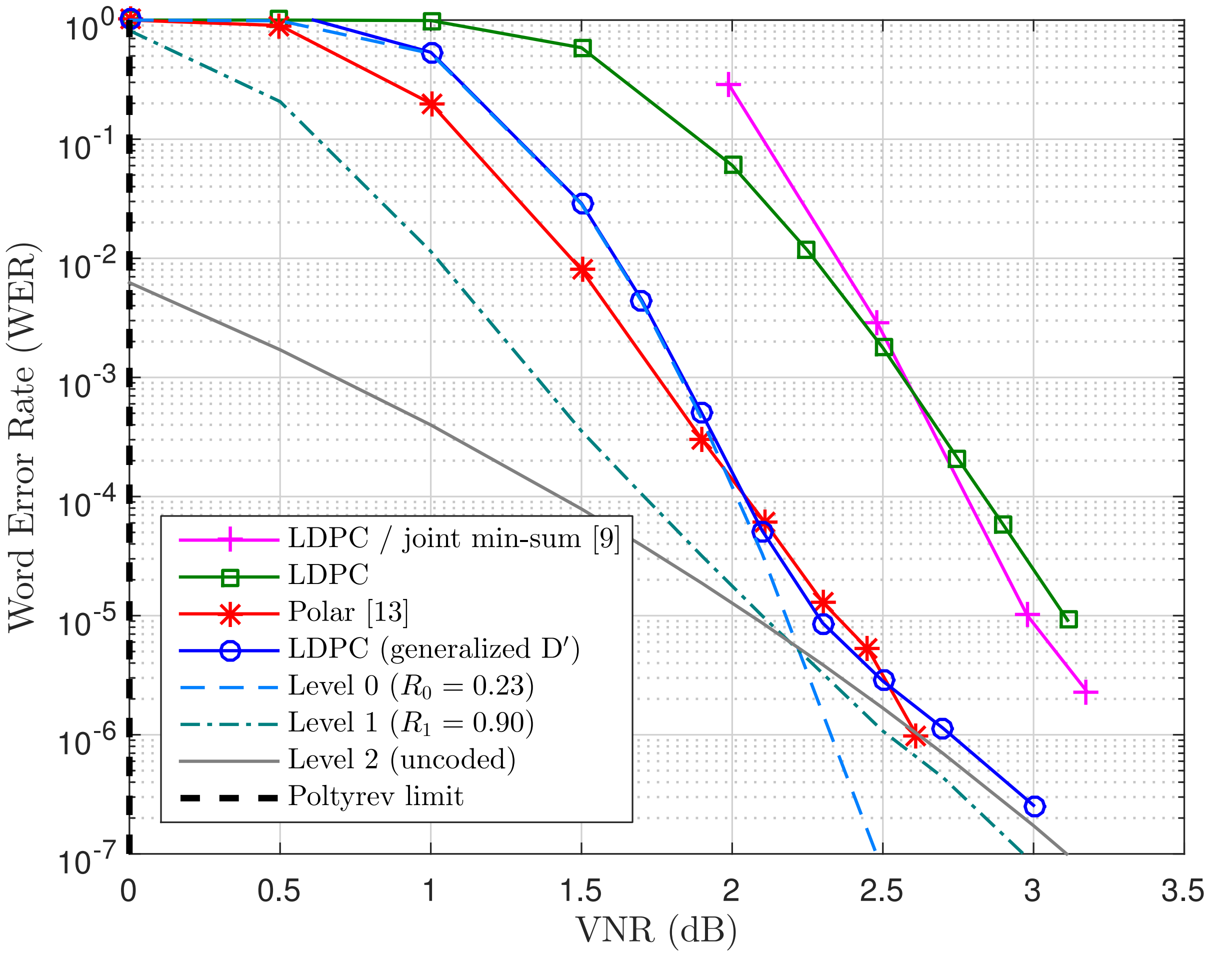

Figure 4.1: Performance of 2-level LDPC lattices of dimension 1024 via Generalized Construction D′ and via the original Construction D′, designed to achieve $P_e \leq 10^{-5}$ under multistage decoding. For comparison, the performance of a 2-level polar lattice with $n = 1024$ [13], a 2-level LDPC lattice with $n = 1000$ decoded with joint min-sum [9], and the Poltyrev limit are also shown.

in the next section, this dependence on the uncoded level can be mitigated by an improved rate design procedure.

To illustrate the motivation for the Generalized Construction D′, Fig. 4.1 shows the performance of a $(2,3;4)$-regular LDPC lattice with $n = 1000$ from [9], where all levels are decoded jointly using the min-sum algorithm. Fig. 4.1 also shows the performance of our best attempt at designing an LDPC lattice via the original Construction D′, where $\mathbf{H}_0$ is constrained to be a submatrix of $\mathbf{H}_1$, for which we used a low-complexity multistage decoder. In this case, both matrices were constructed together via the extended PEG algorithm from [9]. We used the same design criterion as before (equal error probability under multistage decoding). Assuming variable-regular degree distributions, our best design was found with degrees $d_v^0 = 6$ and $d_v^1 = 3$ and rates $R_0 = 0.0967$ ($m_0 = 925$) and $R_1 = 0.9043$ ($m_1 = 98$). Compared to our previous design, the design of C_1 remained almost unchanged, while R_0 had to be significantly decreased in order for C_0 to meet the desired error rate. The poor performance of C_0 may be explained by the highly suboptimal degree distribution, which was constrained by $\mathbf{H}_1$.

As we can see, using the sequential approach of Section 3.2 allows us to a obtain a performance similar to that of [9], but with a lower decoding complexity. On the other hand, both lattices display a significant performance gap as compared to polar lattices, as well as to LDPC lattices constructed with Generalized Construction D′.

4.3 Design for $P_e \leq 10^{-2}$ and $n = 1000$

For our second design example, we use $P_e \leq 10^{-2}$ and $n = 1000$ as design parameters. However, we now adopt an optimized rate design procedure.

Let $C(R) \subseteq \{0,1\}^n$ denote a family of LDPC codes parameterized by their rate R. Define

$$f(R,\sigma) \triangleq P_e(C(R),\sigma^2).$$

as a function of the rate R and noise level σ. Our proposed design rule selects the code rates that minimize VNR such that the error probability is kept less than or equal to P_e.

Using the fact that

$$\text{VNR} = \frac{2^{n\left(L-\sum_{\ell=0}^{L-1} R_\ell\right)}}{2\pi e\sigma^2}$$

this optimization problem can be rewritten as

$$\begin{aligned} \{R_0^*, R_1^*, \sigma^*\} = \underset{R_0, R_1, \sigma^2}{\operatorname{argmax}} \quad & R_0 + R_1 + \log_2 \sigma \\ \text{s.t.} \quad & f(R_0, \sigma) + f(R_1, \sigma/2) + P_e(4\mathbb{Z}^n, \sigma^2) \leq P_e. \end{aligned} \tag{4.2}$$

The function $f(R, \sigma)$ is computed numerically by constructing a code $C(R)$ and estimating its error probability at noise level σ via simulation. To alleviate the complexity of this estimation, simulation is used only for certain values of R and σ and linear regression is used to interpolate between any other values required by the optimization algorithm.

With this design rule, rates $R_0 = 0.5$ ($m_0 = 500$) and $R_1 = 0.978$ ($m_1 = 22$) are obtained, yielding VNR = 1.356 dB for $P_e = 10^{-2}$.

Fig. 4.2 shows the WER as a function of the VNR for our proposed LDPC lattice, as well as for the individual levels used in its multilevel construction. Note that the error probability of the uncoded level is so low that it does not appear in Fig. 4.2. It is also interesting to point out that, at the design VNR, code C_0 displays WER $= 6.9 \cdot 10^{-3}$, whereas code C_1 displays WER $= 3.2 \cdot 10^{-3}$. This suggests that the optimal design criterion under multistage decoding may not be the equal error probability rule, even if only the coded levels are considered.

For the sake of comparison, Fig. 4.2 also shows the performance of other state-of-the-art lattices with dimension around 1000, namely: a one-level QC-LDPC lattice with $n = 1190$ and rate $R = 0.786$ ($m = 935$) [37]; an LDLC lattice with degree 7 and $n = 1000$ decoded with the three/two Gaussian parametric decoder [29]; an LDA lattice with $n = 1000$ based on an $(2, 5)$-regular LDPC code over $\mathbb{F}_{11}$ [28]; and a GLD lattice with $n = 1000$ based on a $[4, 3, 2]$ linear code over $\mathbb{F}_{11}$ [4].

As we can see, the performance of the proposed LDPC lattice is only slightly inferior to that of the LDLC lattice, but it achieves this result with a much lower decoding complexity.

On the other hand, the QC-LDPC lattice has a much worse performance. This can be attributed to the fact that it contains a single coded level, suffering from the low performance of the uncoded level at these design parameters.

Fig. 4.2 also shows that the LDA and GLD lattices have a significantly better performance compared to our LDPC lattice. However, these lattices rely on linear codes defined over $\mathbb{F}_{11}$, leading to a much higher decoding complexity.

It can be seen in Fig. 4.2 that our proposed LDPC lattice displays an error floor caused by

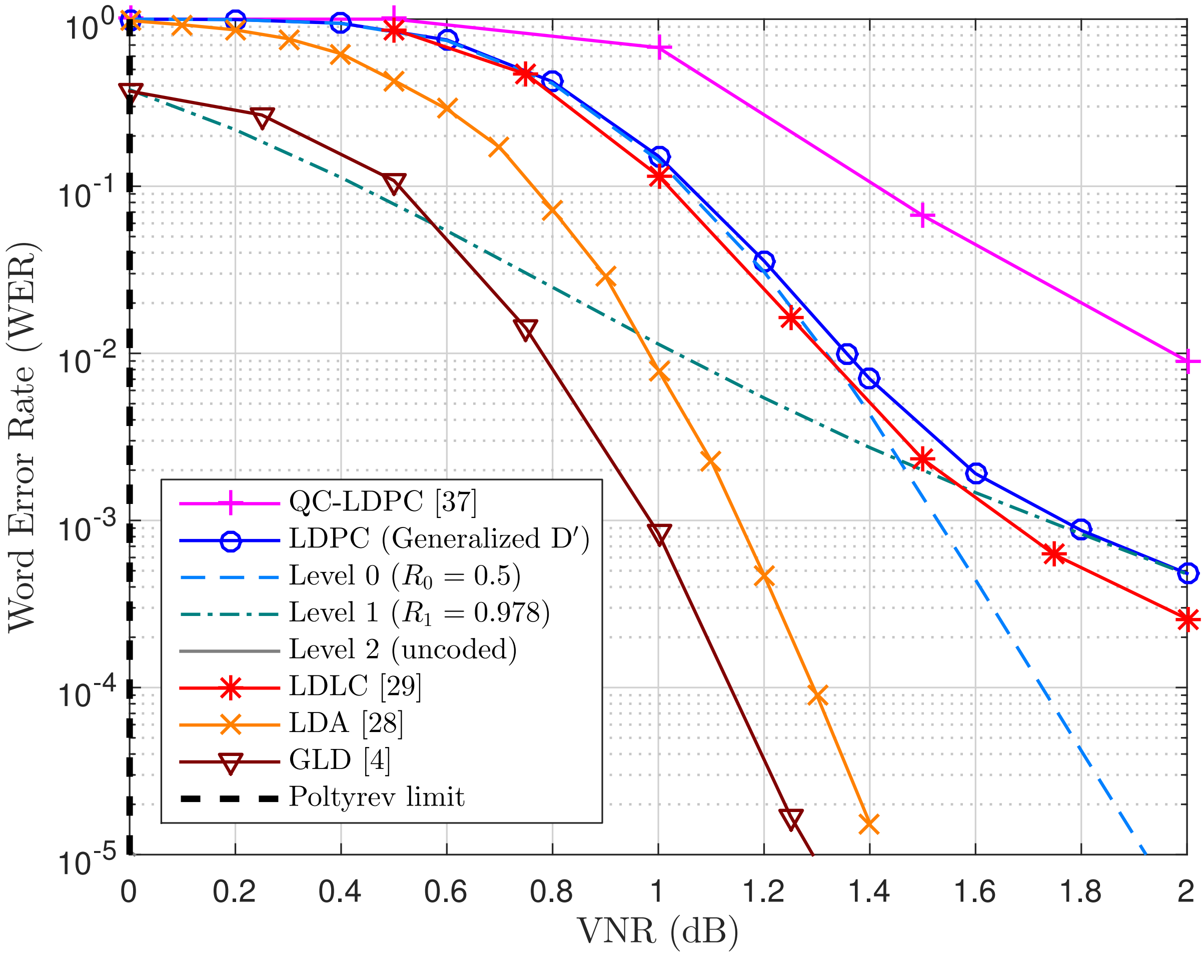

Figure 4.2: Performance of a Generalized Construction D′ 2-level LDPC lattice of dimension 1000, designed to achieve $P_e \leq 10^{-2}$ under multistage decoding. For comparison, the performance of a 1-level QC-LDPC lattice with $n = 1190$ [38], an LDLC lattice with $n = 1000$ [29], an LDA lattice with $n = 1000$ [28], a GLD lattice with $n = 1000$ [4], and the Poltyrev limit are also shown.

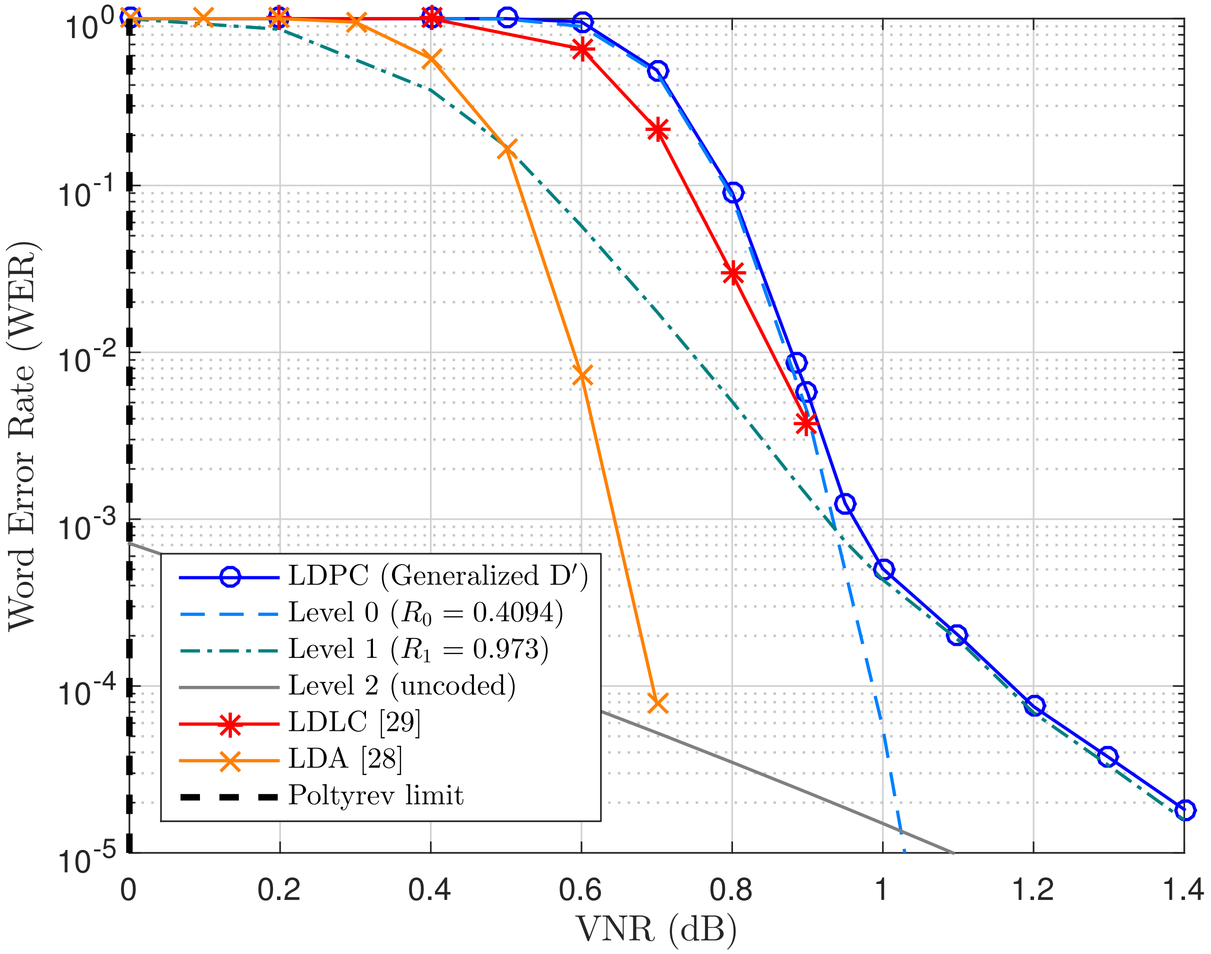

Figure 4.3: Performance of a Generalized Construction D′ 2-level LDPC lattice of dimension 10000, designed to achieve $P_e \leq 10^{-2}$ under multistage decoding. For comparison, the performance of an LDLC lattice with $n = 10000$ [29], an LDA lattice with $n = 10000$ [28], and the Poltyrev limit are also shown.

the low performance of C_1. This may be partly explained by the high value of R_1 for this block length and by the presence of a substantial amount of 4-cycles in the parity-check matrix $\mathbf{H}_1$.

4.4 Design for $P_e \leq 10^{-2}$ and $n = 10000$

As our last example, we design an LDPC lattice with dimension $n = 10000$ for $P_e \leq 10^{-2}$. We use the same design rule as in Section 4.3. However, we place an additional constraint on R_1, requiring it to be sufficiently small such that the PEG construction [15] does not generate any 4-cycles. Following this constraint, we designed for WER $= 10^{-2}$, arriving at rates $R_0 = 0.4094$ ($m_0 = 5906$) and $R_1 = 0.973$ ($m_1 = 270$) and VNR = 0.884 dB. The simulated result indicates the crossing of WER $= 10^{-2}$ at VNR = 0.8790 dB.

Fig. 4.3 shows the word error rate versus VNR curve for our LDPC lattice. As benchmarks, we have also plotted performance curves for other lattices with dimension $n = 10000$, including an LDLC lattice with degree 7 and an LDA lattice, once again based on a $(2,5)$-regular LDPC code over $\mathbb{F}_{11}$.

Similarly to Section 4.3, we see that our LDPC lattice almost matches the performance of the LDLC lattice, with the benefit of providing less complex decoding. We can also see that the gap to the LDA lattice performance has decreased.

4.5 Discussion

A limitation of the check splitting procedure is that the variable-node degree distributions for the component codes must be the same. Since the independent optimization of the component codes leads to different variable-node degree distributions, it is clear that using a single distribution for codes of significantly different rates cannot be optimal.

Note that this restriction on degree distributions is not necessarily imposed by the Generalized Construction D′, but is rather a consequence of check splitting. Finding a method of designing nested LDPC codes with a more flexible choice of variable-node degree distributions remains an open problem.

5

Conclusion

We achieved results for lattice codes in power-unconstrained channels. Two main contributions are provided to multilevel LDPC lattices: an alternative description and a generalization of Construction D′.

The alternative description of Construction D′ enables sequential encoding of the component codes. This in turn ensures the use of multistage decoding. We show that low-complexity off-the-shelf binary LDPC encoders and decoders can be used to produce multilevel encoding and multistage decoding algorithms for LDPC lattices with a complexity that is linear in the total number of coded bits. The linear complexity of these operations has not been attained by any other existing Poltyrev-limit approaching lattices.

Generalized Construction D′ relaxes the nesting constraints on the parity-check matrices of the component codes, significantly facilitating the design; specifically, under the new construction, only the component codes have to be nested, not their parity-check matrices. We showed that the codes are nested if their parity-check matrices respect the principle of linearity where the linear operator can be seen as a multiplier matrix with integer coefficients. Following this result, we have devised a general principle for constructing nested codes based on the partitioning of parity-check equations, as well as a practical method inspired by the PEG algorithm to construct check-split LDPC matrices of large girth that can be efficiently encoded.

Based on this new construction, low-complexity multilevel LDPC lattices are designed whose performance under multistage decoding is shown to be comparable to that of polar lattices, closing a long-standing gap in performance between Construction D and Construction D′ lattices. Our proposed LDPC lattices are also shown to achieve a performance close to that of LDLCs, albeit with a much lower decoding complexity.

While the achieved performance is still far from that of p-ary lattices such as GLD and LDA, it should be noted that only variable-regular lattices have been considered in this thesis. One can reasonably expect that a much better performance may be achieved by irregular LDPC lattices with carefully designed degree distributions. However, in that case, a new design procedure would be required in order to ensure that the resulting codes remain nested.

5.1 Future Work

As hinted previously, the use of irregular LDPC codes using the most optimized degree distributions for each level is an interesting problem with challenging requirements. The optimized distributions for low-rate and high-rate LDPC codes tend (as seen by EXIT Chart analysis) to have distributions with weights that cannot be attained by conventional parity-check matrix nesting procedures or by check splitting.

A possible research direction is to create a practical procedure that allows the use of irregular codes and improves/generalizes upon the check splitting framework. The idea is to tailor matrix $\mathbf{F}_\ell$ (the integer coefficient matrix of the linear operation) in (3.18) to the code rate design necessary for a particular channel and lattice application. In other words, it would be highly desirable to create at will generic linear combinations ($\mathbf{F}_\ell$ multiplier matrices), fully realizing the potential of Generalized Construction D′. This would further enable the use of irregular codes during lattice code construction. Accordingly, this would allow better optimized degree distributions for the codes used at each level, i.e., the distributions that optimize the EXIT Chart analysis for each component code.

Another direction for future projects is to use quasi-cyclic LDPC codes in the construction of the multilevel lattice codes, allowing the code length to increase and, consequently, the error correction performance to improve.

All the work described in this thesis was conducted over power-unconstrained channels. However, we studied the implementation of shaping for power-constrained channels (the conventional AWGN channel, for instance) using the framework for trellis shaping [27] which we adapted in [26] for any linear code as the shaping code (as opposed to convolutional codes only). In [26] we were able to enlarge the selection of possible data rates by constructing convolutional codes via the Smith Normal Form. It would be interesting to test more powerful codes as the shaping code, expanding on the results of [26] by using sparse or polar codes. Sparse codes figure as a possibility given the application of LDGM (low-density generator matrix) codes for the quantization problem, as seen in [44–46]. Based on [47–49], we also see that polar codes can be successfully used for the quantization problem and lossy compression and in [13] for the shaping problem itself.

Finally, one further objective would be to implement the multilevel lattices of this thesis in cooperative applications requiring lattice structures, such as distributed channel coding (see [50] and references therein), binning for the wiretap channel [51] (which could exploit the coset codes of Construction D$'$), and Compute-and-Forward Multiple Access (CFMA) [52].

References

[1] Ram Zamir. *Lattice Coding for Signals and Networks*. Cambridge, UK: Cambridge University Press, 2014.

[2] Naftali Sommer, Meir Feder, and Ofir Shalvi. “Low-Density Lattice Codes”. In: *IEEE Transactions on Information Theory* 54.4 (Apr. 2008). Conference Name: IEEE Transactions on Information Theory, pp. 1561–1585. DOI: 10.1109/TIT.2008.917684.

[3] N. di Pietro, J. J. Boutros, G. Zémor, and L. Brunel. “Integer Low-Density Lattices Based on Construction A”. In: *2012 IEEE Information Theory Workshop*. Sept. 2012, pp. 422–426. DOI: 10.1109/ITW.2012.6404707.

[4] J.J. Boutros, N. di Pietro, and N. Basha. “Generalized Low-Density (GLD) Lattices”. In: *2014 IEEE Information Theory Workshop (ITW)*. Nov. 2014, pp. 15–19. DOI: 10.1109/ITW.2014.6970783.

[5] E. S. Barnes and N. J. A. Sloane. “New Lattice Packings of Spheres”. en. In: *Canadian Journal of Mathematics* 35.1 (Feb. 1983). Publisher: Cambridge University Press, pp. 117–130. DOI: 10.4153/CJM-1983-008-1.

[6] John Conway and Neil J. A. Sloane. *Sphere Packings, Lattices and Groups*. en. 3rd ed. Grundlehren der mathematischen Wissenschaften. New York: Springer-Verlag, 1999. DOI: 10.1007/978-1-4757-6568-7.

[7] U. Wachsmann, R. F. H. Fischer, and J. B. Huber. “Multilevel Codes: Theoretical Concepts and Practical Design Rules”. In: *IEEE Trans. Inf. Theory* 45.5 (July 1999), pp. 1361–1391. DOI: 10.1109/18.771140.

[8] G. D. Forney, M. D. Trott, and Sae-Young Chung. “Sphere-Bound-Achieving Coset Codes and Multilevel Coset Codes”. In: *IEEE Trans. Inf. Theory* 46.3 (May 2000), pp. 820–850. DOI: 10.1109/18.841165.

[9] M.-R. Sadeghi, A.H. Banihashemi, and D. Panario. “Low-Density Parity-Check Lattices: Construction and Decoding Analysis”. In: *IEEE Trans. Inf. Theory* 52.10 (Oct. 2006), pp. 4481–4495. DOI: 10.1109/TIT.2006.881720.

[10] Young-Seob Choi, Ihn-Jung Baik, and Sae-Young Chung. “Iterative Decoding for Low-Density Parity-Check Lattices”. In: *2008 10th International Conference on Advanced Communication Technology*. Vol. 1. ISSN: 1738-9445. Feb. 2008, pp. 358–361. DOI: 10.1109/ICACT.2008.4493778.

[11] Ihn-Jung Baik and Sae-Young Chung. "Irregular low-density parity-check lattices". In: *2008 IEEE International Symposium on Information Theory*. ISSN: 2157-8117. July 2008, pp. 2479–2483. DOI: 10.1109/ISIT.2008.4595437.

[12] Lida Safarnejad and Mohammad-Reza Sadeghi. "FFT Based Sum-Product Algorithm for Decoding LDPC Lattices". In: *IEEE Communications Letters* 16.9 (Sept. 2012). Conference Name: IEEE Communications Letters, pp. 1504–1507. DOI: 10.1109/LCOMM.2012.073112.120996.

[13] Ling Liu, Yanfei Yan, Cong Ling, and Xiaofu Wu. "Construction of Capacity-Achieving Lattice Codes: Polar Lattices". In: *IEEE Transactions on Communications* 67.2 (Feb. 2019). Conference Name: IEEE Transactions on Communications, pp. 915–928. DOI: 10.1109/TCOMM.2018.2876113.

[14] G. Poltyrev. "On Coding without Restrictions for the AWGN Channel". In: *IEEE Trans. Inf. Theory* 40.2 (Mar. 1994), pp. 409–417. DOI: 10.1109/18.312163.

[15] Xiao-Yu Hu, E. Eleftheriou, and D. M. Arnold. "Regular and Irregular Progressive Edge-Growth Tanner Graphs". In: *IEEE Trans. Inf. Theory* 51.1 (Jan. 2005), pp. 386–398. DOI: 10.1109/TIT.2004.839541.

[16] G.D. Forney. "Coset codes. I. Introduction and geometrical classification". In: *IEEE Transactions on Information Theory* 34.5 (Sept. 1988). Conference Name: IEEE Transactions on Information Theory, pp. 1123–1151. DOI: 10.1109/18.21245.

[17] Chen Feng, Danilo Silva, and Frank R. Kschischang. "Lattice network coding over finite rings". In: *2011 12th Canadian Workshop on Information Theory*. May 2011, pp. 78–81. DOI: 10.1109/CWIT.2011.5872128.

[18] R. Urbanke and B. Rimoldi. "Lattice Codes Can Achieve Capacity on the AWGN Channel". In: *IEEE Trans. Inf. Theory* 44.1 (Jan. 1998), pp. 273–278. DOI: 10.1109/18.651040.

[19] U. Erez and R. Zamir. "Achieving 1/2 Log (1+SNR) on the AWGN Channel with Lattice Encoding and Decoding". In: *IEEE Trans. Inf. Theory* 50.10 (Oct. 2004), pp. 2293–2314. DOI: 10.1109/TIT.2004.834787.

[20] D. Krithivasan and S. S. Pradhan. "Lattices for Distributed Source Coding: Jointly Gaussian Sources and Reconstruction of a Linear Function". In: *IEEE Trans. Inf. Theory* 55.12 (Dec. 2009), pp. 5628–5651. DOI: 10.1109/TIT.2009.2032853.

[21] C. Ling, L. Luzzi, J. C. Belfiore, and D. Stehlé. "Semantically Secure Lattice Codes for the Gaussian Wiretap Channel". In: *IEEE Trans. Inf. Theory* 60.10 (Oct. 2014), pp. 6399–6416. DOI: 10.1109/TIT.2014.2343226.

[22] B. Nazer and M. Gastpar. "Compute-and-Forward: Harnessing Interference through Structured Codes". In: *IEEE Trans. Inf. Theory* 57.10 (Oct. 2011), pp. 6463–6486. DOI: 10.1109/TIT.2011.2165816.

[23] Jiening Zhan, B. Nazer, U. Erez, and M. Gastpar. "Integer-Forcing Linear Receivers". In: *IEEE Trans. Inf. Theory* 60.12 (Dec. 2014), pp. 7661–7685. DOI: 10.1109/TIT.2014.2361782.

[24] C. Ling and J. C. Belfiore. "Achieving AWGN Channel Capacity with Lattice Gaussian Coding". In: *IEEE Trans. Inf. Theory* 60.10 (Oct. 2014), pp. 5918–5929. DOI: 10.1109/TIT.2014.2332343.

[25] Antonio Campello, Daniel Dadush, and Cong Ling. “AWGN-Goodness Is Enough: Capacity-Achieving Lattice Codes Based on Dithered Probabilistic Shaping”. In: *IEEE Transactions on Information Theory* 65.3 (Mar. 2019). Conference Name: IEEE Transactions on Information Theory, pp. 1961–1971. DOI: 10.1109/TIT.2018.2875004.

[26] Paulo Ricardo Branco da Silva, Roberto Augusto Philippi Martins, and Danilo Silva. “Projeto de Codigos de Reticulado LDPC Multinivel Irregulares e com o Uso de Shaping”. In: *XXXVI Simpósio Brasileiro de Telecomunicacoes e Processamento de Sinais - SBrT2018*. Campina Grande, PB, Sept. 2018, pp. 1–5.

[27] G.D. Forney. “Trellis shaping”. In: *IEEE Transactions on Information Theory* 38.2 (Mar. 1992). Conference Name: IEEE Transactions on Information Theory, pp. 281–300. DOI: 10.1109/18.119687.

[28] Nicola di Pietro, Gilles Zémor, and Joseph J. Boutros. “LDA Lattices Without Dithering Achieve Capacity on the Gaussian Channel”. In: *IEEE Transactions on Information Theory* 64.3 (Mar. 2018). Conference Name: IEEE Transactions on Information Theory, pp. 1561–1594. DOI: 10.1109/TIT.2017.2778158.

[29] R. A. Parrao Hernandez and B. M. Kurkoski. “The Three/Two Gaussian Parametric LDLC Lattice Decoding Algorithm and Its Analysis”. In: *IEEE Trans. Commun.* 64.9 (Sept. 2016), pp. 3624–3633. DOI: 10.1109/TCOMM.2016.2594286.

[30] Nicola di Pietro, Nour Basha, and Joseph J. Boutros. “Non-binary GLD codes and their lattices”. In: *2015 IEEE Information Theory Workshop (ITW)*. Apr. 2015, pp. 1–5. DOI: 10.1109/ITW.2015.7133127.

[31] N. di Pietro, G. Zémor, and J. J. Boutros. “New Results on Construction A Lattices Based on Very Sparse Parity-Check Matrices”. In: *2013 IEEE International Symposium on Information Theory*. July 2013, pp. 1675–1679. DOI: 10.1109/ISIT.2013.6620512.

[32] A. Sakzad, M. R. Sadeghi, and D. Panario. “Construction of Turbo Lattices”. In: *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Sept. 2010, pp. 14–21. DOI: 10.1109/ALLERTON.2010.5706882.

[33] A. Vem, Yu-Chih Huang, K.R. Narayanan, and H.D. Pfister. “Multilevel Lattices Based on Spatially-Coupled LDPC Codes with Applications”. In: *2014 IEEE International Symposium on Information Theory (ISIT)*. June 2014, pp. 2336–2340. DOI: 10.1109/ISIT.2014.6875251.

[34] Wittawat Kositwattanarerk and Frédérique Oggier. “Connections between Construction D and Related Constructions of Lattices”. en. In: *Des. Codes Cryptogr.* 73.2 (Nov. 2014), pp. 441–455. DOI: 10.1007/s10623-014-9939-3.

[35] Mohammad-Reza Sadeghi and Amin Sakzad. “On the performance of 1-level LDPC lattices”. In: *2013 Iran Workshop on Communication and Information Theory*. May 2013, pp. 1–5. DOI: 10.1109/IWCIT.2013.6555759.

[36] Hassan Khodaiemehr, Dariush Kiani, and Mohammad-Reza Sadeghi. “One-level LDPC lattice codes for the relay channels”. In: *2015 Iran Workshop on Communication and Information Theory (IWCIT)*. May 2015, pp. 1–6. DOI: 10.1109/IWCIT.2015.7140213.

[37] Hassan Khodaiemehr, Mohammad-Reza Sadeghi, and Amin Sakzad. “Practical Encoder and Decoder for Power Constrained QC LDPC-Lattice Codes”. In: *IEEE Transactions on Communications* 65.2 (Feb. 2017). Conference Name: IEEE Transactions on Communications, pp. 486–500. DOI: 10.1109/TCOMM.2016.2633343.

[38] Hassan Khodaiemehr, Dariush Kiani, and Mohammad-Reza Sadeghi. “LDPC Lattice Codes for Full-Duplex Relay Channels”. In: *IEEE Transactions on Communications* 65.2 (Feb. 2017). Conference Name: IEEE Transactions on Communications, pp. 536–548. DOI: 10.1109/TCOMM.2016.2638839.

[39] T. J. Richardson and R. L. Urbanke. “Efficient Encoding of Low-Density Parity-Check Codes”. In: *IEEE Trans. Inf. Theory* 47.2 (Feb. 2001), pp. 638–656. DOI: 10.1109/18.910579.

[40] Thomas J. Richardson and Rüdiger Leo Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

[41] M. Good and F. R. Kschischang. “Incremental Redundancy via Check Splitting”. In: *23rd Biennial Symposium on Communications, 2006*. 2006, pp. 55–58. DOI: 10.1109/BSC.2006.1644569.

[42] A. Subramanian, A. Thangaraj, M. Bloch, and S. W. McLaughlin. “Strong Secrecy on the Binary Erasure Wiretap Channel Using Large-Girth LDPC Codes”. In: *IEEE Trans. Inf. Forensics Secur.* 6.3 (Sept. 2011), pp. 585–594. DOI: 10.1109/TIFS.2011.2148715.

[43] Hassan Khodaiemehr, Mohammad-Reza Sadeghi, and Daniel Panario. “Construction of full-diversity 1-level LDPC lattices for block-fading channels”. In: *2016 IEEE International Symposium on Information Theory (ISIT)*. ISSN: 2157-8117. July 2016, pp. 2714–2718. DOI: 10.1109/ISIT.2016.7541792.

[44] Qingchuan Wang and Chen He. “Approaching 1.53-dB Shaping Gain with LDGM Quantization Codes”. In: *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*. ISSN: 1930-529X. Nov. 2007, pp. 1571–1576. DOI: 10.1109/GLOCOM.2007.302.

[45] Qingchuan Wang and Chen He. “Design and Analysis of LDGM-Based Codes for MSE Quantization”. In: *arXiv:0801.2423 [cs, math]* (Jan. 2008). arXiv: 0801.2423.

[46] Qingchuan Wang, Chen He, and Lingge Jiang. “Near-Ideal M-ary LDGM Quantization with Recovery”. In: *IEEE Transactions on Communications* 59.7 (July 2011). Conference Name: IEEE Transactions on Communications, pp. 1830–1839. DOI: 10.1109/TCOMM.2011.061511.100462.

[47] Satish Babu Korada and Rüdiger L. Urbanke. “Polar Codes are Optimal for Lossy Source Coding”. In: *IEEE Transactions on Information Theory* 56.4 (Apr. 2010). Conference Name: IEEE Transactions on Information Theory, pp. 1751–1768. DOI: 10.1109/TIT.2010.2040961.

[48] Junya Honda and Hirosuke Yamamoto. “Polar Coding Without Alphabet Extension for Asymmetric Models”. In: *IEEE Transactions on Information Theory* 59.12 (Dec. 2013). Conference Name: IEEE Transactions on Information Theory, pp. 7829–7838. DOI: 10.1109/TIT.2013.2282305.

[49] Marco Mondelli, S. Hamed Hassani, and Rüdiger L. Urbanke. “How to Achieve the Capacity of Asymmetric Channels”. In: *IEEE Transactions on Information Theory* 64.5 (May 2018). Conference Name: IEEE Transactions on Information Theory, pp. 3371–3393. DOI: 10.1109/TIT.2018.2789885.

[50] David Declercq, Marc Fossorier, and Ezio Biglieri, eds. *Channel Coding: Theory, Algorithms, and Applications: Academic Press Library in Mobile and Wireless Communications*. Inglês. Edição: 1. Academic Press, Sept. 2016.

[51] A. D. Wyner. “The wire-tap channel”. In: *The Bell System Technical Journal* 54.8 (Oct. 1975), pp. 1355–1387. DOI: 10.1002/j.1538-7305.1975.tb02040.x.

[52] Erixhen Sula, Jingge Zhu, Adriano Pastore, Sung Hoon Lim, and Michael Gastpar. “Compute–Forward Multiple Access (CFMA): Practical Code Design”. In: *arXiv:1712.10293 [cs, math]* (Dec. 2017). arXiv: 1712.10293.