Lattices from LDPC codes

・Name: Jongmin Lee

・Student Number: 23010402

・Advisor for Minor Research Project: KURKOSKI, Brian Michael

・Date and Year of Submission: 2024/08/26

・Japan Advanced Institute of Science and Technology

1. Introduction

In modern communication systems, particularly for high data-rate applications like WiFi, the need for efficient error-correcting codes is paramount. Low-Density Parity-Check (LDPC) codes have emerged as one of the most efficient error-correcting codes due to their ability to approach the Shannon limit and their practical decoding algorithms.[1] [2]

In this study, we explore LDPC codes in combination with a lattice construction technique known as **Construction A**. The aim is to evaluate the performance of this combination in terms of the **Word Error Rate (WER)** over a range of signal-to-noise ratios (SNRs). Specifically, we focus on the performance of these codes under different code rates to understand the impact on WER, ultimately identifying the best-performing code configuration.

2. LDPC Codes

Introduction to LDPC Codes

Low-Density Parity-Check (LDPC) codes are a class of linear block codes characterized by a sparse parity-check matrix $\boldsymbol{H}$. These codes were first introduced by Robert Gallager in the 1960s, [1] and later rediscovered in the 1990s due to their near-optimal performance under iterative decoding, particularly over noisy channels such as Additive White Gaussian Noise (AWGN) channels.[3]

LDPC codes are defined by the **parity-check matrix** $\boldsymbol{H}$, which relates the encoded message $\mathbf{x}$ to the transmitted codeword via the equation $\boldsymbol{H}\cdot\mathbf{x}^{\mathbf{T}} = 0$. The sparsity of the matrix $\boldsymbol{H}$ means that each row and column contains only a small number of non-zero entries, making both encoding

and decoding highly efficient.

Structure of LDPC Codes

The code length n of an LDPC code is the number of bits in the transmitted codeword, while the message length k is the number of information bits before encoding. The code rate R is defined as $R = k/n$, which directly impacts the amount of redundancy added to the message and the error-correction capability of the code.

The structure of the parity-check matrix H is often represented as a **Tanner graph**, where variable nodes correspond to the bits of the codeword and check nodes correspond to the parity checks imposed by H. The sparse nature of H ensures that each variable node connects to only a small number of check nodes, making the decoding process computationally feasible.

Decoding Algorithms for LDPC Codes

The decoding of LDPC codes is typically done using an iterative algorithm such as the **Belief Propagation (BP)** algorithm, also known as the **Message Passing Algorithm**.[3] This algorithm operates by passing probabilistic messages between variable nodes and check nodes in the Tanner graph.[4]

In each iteration, the messages from the check nodes update the likelihood of the bit values, and the variable nodes update their beliefs about the transmitted bits. This process is repeated for a predefined number of iterations or until a valid codeword is found. The iterative nature of this process allows the decoder to progressively refine its estimates, resulting in low error rates.

Advantages of LDPC Codes

LDPC codes offer several advantages, including:

- **Near-optimal performance:** LDPC codes can closely approach the Shannon limit, particularly when decoded using iterative algorithms like belief propagation.
- **Scalability:** The performance of LDPC codes can be fine-tuned by adjusting the sparsity of the parity-check matrix, making them suitable for a wide range of applications.
- **Efficient decoding:** The iterative decoding algorithms used for LDPC codes allow for efficient error correction, even for long block lengths, making them ideal for high-throughput applications.

3. Construction A Lattices

Introduction to Lattices

A **lattice** in n-dimensional Euclidean space is a discrete set of points that form a regular grid. Lattices are used in coding theory because they can efficiently map high-dimensional codewords into Euclidean space, where they can be used for signal transmission over noisy channels.[5]

In practical communication systems, lattices provide a framework for **coding and modulation** schemes that combine high code rates with efficient use of bandwidth. A lattice can be generated by a set of basis vectors, and the lattice points correspond to integer combinations of these vectors.

Construction A

Construction A is a method of building lattices from linear error-correcting codes, typically over finite fields.[1] It is one of the simplest ways to construct a lattice and is particularly useful in applications that involve both coding and modulation.

To construct a lattice using Construction A:

1. **Start with a linear code** $\boldsymbol{C}$ over a finite field $\mathbf{F_q}$. In this case, $\boldsymbol{C}$ is typically an LDPC code.
2. **Map codewords** from the finite field to points in $\mathbf{Z^n}$, the integer lattice.
3. A vector $\mathbf{x} \in \mathbf{Z^n}$ is a point in the lattice if and only if $\mathbf{x}$ mod $\boldsymbol{q}$ is a codeword in $\boldsymbol{C}$.

Thus, each codeword in $\boldsymbol{C}$ corresponds to a coset of the lattice in $\mathbf{Z^n}$.[5] This construction effectively embeds the LDPC code into a lattice, allowing us to leverage the structure of both the code and the lattice for efficient error correction.

Properties of Construction A Lattices

When LDPC codes are combined with Construction A, the resulting lattice inherits the properties of both the code and the lattice. The **minimum distance** of the lattice (the shortest distance between any two lattice points) depends on both the minimum Hamming distance of the LDPC code and the chosen field size $\boldsymbol{q}$.

In practical terms, Construction A lattices provide a robust framework for transmitting codewords over noisy channels, with the lattice structure helping to improve the error

correction performance of the underlying code.

Applications of Construction A Lattices

Construction A lattices are used in various communication applications, particularly those requiring high data rates and efficient use of bandwidth. They have been employed in systems like:

- **Wireless communication**: Lattices are well-suited for modulating signals in high-dimensional spaces, making them useful for multi-antenna systems and MIMO configurations.
- **Data storage**: Lattices provide an efficient way to map large codewords to discrete points, improving error correction in data storage systems.

4. Experimental Setup

Objective

The primary objective of this experiment was to evaluate the WER performance of LDPC codes combined with Construction A under various signal-to-noise ratio (SNR) conditions. The focus was on four different code configurations, each with a different code rate, to analyze how the choice of code rate impacts the WER.[1]

The experiment was carried out using MATLAB, where the LDPC codes were encoded, mapped into lattice points using Construction A, and transmitted over an Additive White Gaussian Noise (AWGN) channel. The decoded messages were then analyzed to compute the WER.

Code Configurations

These configurations correspond to different code rates, calculated as $\boldsymbol{R = k/n}$ where $\boldsymbol{k}$ is the number of information bits and $\boldsymbol{n}$ is the total number of transmitted bits. For instance, in our experiment, the code rates for each configuration are as follows.:

- For $\boldsymbol{k = 324}$, the code rate $\boldsymbol{R = 324/648 = 0.5}$.
- For $\boldsymbol{k = 432}$, the code rate $\boldsymbol{R = 432/648 = 0.6667}$.
- For $\boldsymbol{k = 486}$, the code rate $\boldsymbol{R = 486/648 = 0.75}$.
- For $\boldsymbol{k = 540}$, the code rate $\boldsymbol{R = 540/648 = 0.8333}$.

Higher code rates reduce redundancy but also decrease the error-correcting capability of the LDPC code. Each of these configurations represents a different code rate, which directly affects

the density of the lattice and the redundancy added during encoding.

Channel and Noise Model

The AWGN channel was chosen for the experiment due to its simplicity and prevalence in communication theory. The channel introduces Gaussian noise to the transmitted lattice points, and the decoder must then map the noisy points back to the nearest lattice points.

The **Variance-to-Noise Ratio (VNR)** was varied across a wide range to simulate different noise levels, and the WER was computed at each VNR value.

Decoding Process

The decoding algorithm used in this study was an **iterative belief propagation decoder** specifically optimized for **lattice decoding**. This method combines the principles of **LDPC codes** and **lattice theory**, where the algorithm attempts to map noisy lattice points back to the nearest valid lattice points. The decoding process is influenced by both the **LDPC code structure** and the lattice construction (**Construction A**).

In this context, the **log-likelihood ratio (LLR)** plays a crucial role. The received signal, which is affected by noise, is transformed into LLR values that indicate the probability of a bit being a '0' or '1'. The belief propagation algorithm then iteratively processes these LLR values across the LDPC Tanner graph, updating the belief estimates for each bit and converging towards the most likely codeword. In the case of lattice decoding with **Construction A**, the algorithm not only decodes the LDPC code but also ensures that the decoded points fall within the valid lattice points defined by the **Construction A** framework.

The **code rate** used in this experiment was **R = k/n**, where **k** is the number of information bits, and **n** is the length of the codeword. By adjusting the **code rate**, the experiment aimed to optimize the trade-off between error correction performance and redundancy. Lower code rates introduce more redundancy, improving error correction capability, while higher code rates reduce redundancy, making the code more efficient but possibly less robust against noise.

Furthermore, the **LLR calculation** is critical in the decoding process. The LLR is computed as:

$$LLR = \log((\boldsymbol{P}(y|x=1)/(\boldsymbol{P}(y|x=0)))$$

where ***P***(y|x) is the probability of receiving the signal yyy given that the transmitted bit was xxx. This probabilistic approach allows the algorithm to effectively weigh the likelihood of each bit and refine its estimates iteratively.

In summary, this decoding approach leverages the **iterative belief propagation** algorithm's strength in LDPC codes, while ensuring the decoded points conform to the lattice structure defined by **Construction A**. This method provides robust performance, particularly in environments with noise, by maximizing the error correction capabilities inherent in both the lattice and LDPC coding schemes.

Results

Performance Comparison

The figure 1 showed that the WER decreased as the code rate increased, with $k = 540$ providing the lowest WER at most VNR levels. This is consistent with the theoretical expectation that higher-rate codes lead to denser lattices with smaller decision regions, thus reducing the probability of decoding errors.

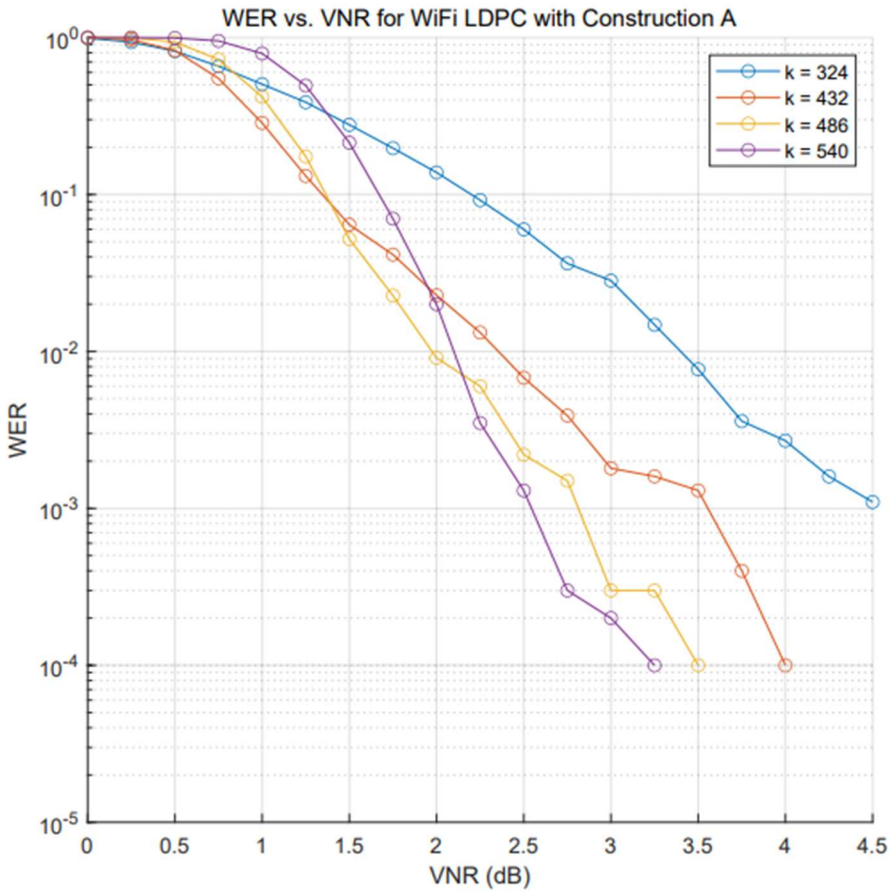

Figure 1. Result Graph

A plot of the WER versus VNR for each code configuration clearly demonstrated that $k = 540$ offers superior performance, making it the best choice for high-rate communication systems where minimizing errors is critical.

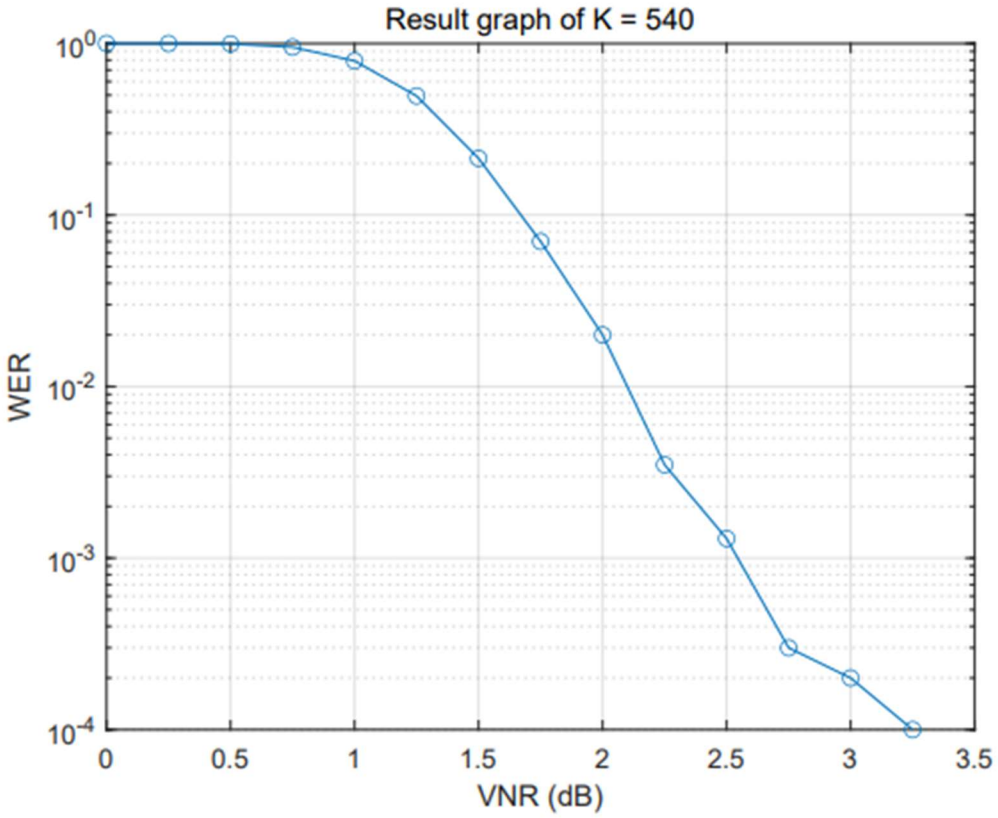

Figure2. Result graph of $K = 540$

5. Conclusion

The experiment successfully demonstrated that LDPC codes combined with Construction A provide robust error correction performance, especially for high code rates. [1] The configuration with $k = 540$ achieved the best overall WER performance, making it the most suitable for applications requiring low-latency and high-throughput communication.

Future research could explore more advanced decoding algorithms, further optimize the code configurations, or investigate other lattice constructions to enhance performance further.

6. References

[1] Gallager, R. G. (1963). Low-Density Parity-Check Codes. MIT Press, Cambridge, MA.

[2] Richardson, T., & Urbanke, R. (2001). "Design of capacity-approaching irregular low-density parity-check codes." *IEEE Transactions on Information Theory*, 47(2), 619-637.

[3] Fossorier, M., Paolini, E., Chiani, M. (2004). "Quasi-cyclic low-density parity-check codes from circulant permutation matrices." *IEEE Transactions on Information Theory*, 50(8), 1788–1793.

[4] Liva, G., & Chiani, M. (2007). "Protograph LDPC codes design based on EXIT analysis." In *Proceedings of the 2007 IEEE Global Telecommunications Conference*, pp. 3250–3254.

[5] Kurkoski, B. (2023). "Lattice Coding Theory Lecture Notes"