รายงานวิจัยฉบับสมบูรณ์

โครงการ การสร้างรหัสปลอดรหัสเทียมและรหัสแลตทิซ

โดย ผศ. ดร. วิทวัชร์ โฆษิตวัฒนฤกษ์

30 มิถุนายน 2560

สัญญาเลขที่ TRG 5880116

รายงานวิจัยฉบับสมบูรณ์

โครงการ การสร้างรหัสปลอดรหัสเทียมและรหัสแลตทิซ

ผศ. ดร. วิทวัชร์ โฆษิตวัฒนฤกษ์ มหาวิทยาลัยมหิดล

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย
และ มหาวิทยาลัยมหิดล

Abstract

Information is everywhere around us. Precipitated by the digital age, the way we process, store, and transmit information has become increasingly complex. This project aims to design error-resistant codes to be used in the digital mediums. We study the effects of cycles on the pseudocodeword error performance of a low-density parity-check (LDPC) code under iterative decoding and linear programming decoding. In addition, we introduce a novel binary coding scheme for multidimensional message spaces. To save transmission energy, a shaping map of lattice code constellations is studied and is numerically demonstrated to reduce average energy consumption by up to 30%.

Keywords: Coding theory, Fibonacci codes, lattice codes, LDPC codes, pseudocodewords

บทคัดย่อ

มีข้อมูลอยู่มากมายรอบตัวเรา ขั้นตอนการจัดการ เก็บรักษา และส่งข้อมูลนั้นซับซ้อนขึ้นมาก ตั้งแต่สังคมมนุษย์เริ่มก้าวเข้าสู้ยุคดิจิตอล โครงการนี้มีเป้าหมายเพื่อออกแบบรหัสที่ทนต่อความผิดพลาดจากสัญญาณรบกวนและเหมาะสมกับสื่อดิจิตอล เราได้ทำการศึกษาผลกระทบของวัฏจักร (cycle) ต่อประสิทธิภาพในการตรวจและแก้ไขข้อผิดพลาดของรหัสเทียมของรหัสตรวจสอบคู่-คี่ความหนาแน่นต่ำ (low-density parity-check, LDPC) ภายใต้การถอดรหัสด้วยวิธีทำซ้ำ (iterative decoding) และการถอดรหัสด้วยกำหนดการเชิงเส้น (linear programming decoding) นอกจากนี้เราได้ค้นพบรูปแบบการเข้ารหัสฐานสองแบบใหม่สำหรับปริภูมิข้อความหลายมิติ เพื่อประหยัดพลังงานในการส่งสัญญาณ การส่งแปลงรูปร่าง (shaping map) สำหรับเซตสัญญาณ (constellation) ของรหัสแลตทิซได้ถูกศึกษาและคำนวณพบว่าสามารถประหยัดพลังงานที่ใช้เฉลี่ยได้ถึง 30%

Keywords: Coding theory, Fibonacci codes, lattice codes, LDPC codes, pseudocodewords

รายละเอียดโครงการ

รหัสโครงการ: TRG 5880116

ชื่อโครงการ (ภาษาไทย): การสร้างรหัสปลอดรหัสเทียมและรหัสแลตทิซ

(ภาษาอังกฤษ): Construction of Pseudocodeword-Free Codes and Lattice Codes

ชื่อหัวหน้าโครงการ (ภาษาไทย): นาย วิทวัชร์ โฆษิตวัฒนฤกษ์

(ภาษาอังกฤษ): Wittawat Kositwattanarerk

(ตำแหน่งวิชาการ): ผู้ช่วยศาสตราจารย์

ระยะเวลาดำเนินงาน: 2 **ปี** 0 **เดือน**

เวลาทำงานวิจัยในโครงการประมาณสัปดาห์ละ 20 **ชั่วโมง (ไม่ต่ำกว่า 17.5 ชั่วโมง/สัปดาห์)**

สถานที่ติดต่อ:

ที่ทำงาน ภาควิชาคณิตศาสตร์
มหาวิทยาลัยมหิดล
แขวงทุ่งพญาไท เขตราชเทวี กรุงเทพ 10400

โทรศัพท์ 0 2201 5446

โทรสาร 0 2201 5343

e-mail wittawat.kos@mahidol.edu

ชื่อนักวิจัยที่ปรึกษา (ภาษาไทย): นางสาว พัฒนี อุดมกะวาณิช

(ภาษาอังกฤษ): Patanee Udomkavanich

(ตำแหน่งวิชาการ): ศาสตราจารย์

สถานที่ติดต่อ:

ที่ทำงาน ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย
กรุงเทพ 10330

โทรศัพท์ 0 2218 5141-2

โทรสาร 0 2255 2287

e-mail pattanee.u@chula.ac.th

เนื้อหางานวิจัย

Objectives

1. Identify LDPC codes that permit cycle-free Tanner graph.
2. Characterize pseudocodeword-free representations of LDPC codes under iterative decoding and linear programming decoding.
3. Generalize the Construction A of lattices from p-ary codes to lattices over number fields.
4. Analyze, test, and compare known constructions of codes with our new constructions.
5. Present and publish our results in international conferences and journals.

Methodology

1. Review necessary background from the literatures.
2. Examine and generalize constructions of lattices from codes.
3. Use computer program to simulate iterative decoding and linear programming decoding.
4. Categorize representations of LDPC codes that do not permit pseudocodewords.
5. State and prove theorems concerning pseudocodeword-free codes.
6. State and prove relevant results on the constructions of lattice codes. Apply our findings to practical real-world applications such as communications over a fading wireless channel.

Results

Coding theory allows information to be stored and transmitted without suffering from loss and alteration of data. This project investigates several facets of this area, particularly in the construction and analysis of low-density parity-check (LDPC) codes, Fibonacci codes, and lattice codes.

Pseudocodeword-free LDPC Codes

Modern decoding algorithms such as message-passing iterative decoding and linear programming decoding are extremely efficient and are shown to enable communications at rates near the channel capacity under several circumstances. However, they may not converge to the maximum-likelihood codeword when the underlying Tanner graph contains cycle. One can explain this divergence via the pseudocodeword. Since a pseudocodewords satisfy every condition set by the decoder, they are legitimate from the perspective of the algorithm. This is because the algorithms search for a codeword that satisfies each parity condition iteratively rather than one that satisfies every parity condition collectively. We study pseudocodeword given by the graph cover. For example, a code with a parity-check matrix

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

has a pseudocodeword $(1,2,1,0)$ as shown in Figure 1.

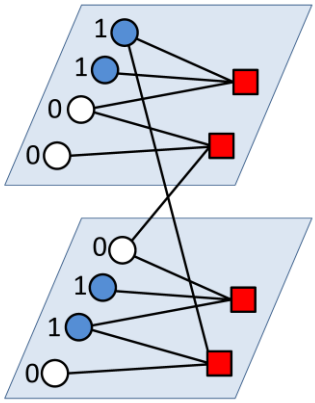

Figure 1: Pseudocodeword $(1,2,1,0) = (1,1,0,0) + (0,1,1,0)$ *on the Tanner graph*

To study the pseudocodewords, we introduce the notion of p-satisfy which localizes the conditions for pseudocodewords. This notion makes the Tanner graph sufficient to verify whether an integer vector is a pseudocodeword. We classify pseudocodeword-free parity-check matrices of a code as follows: if C has a cycle-free representation, then a parity-check matrix of this code is pseudocodeword-free if and only if (possibly empty) redundant rows of this matrix can be removed so as to obtain a representation of C that is cycle-free. The impact of this result is that it allows efficient characterization of an efficient characterization of code.

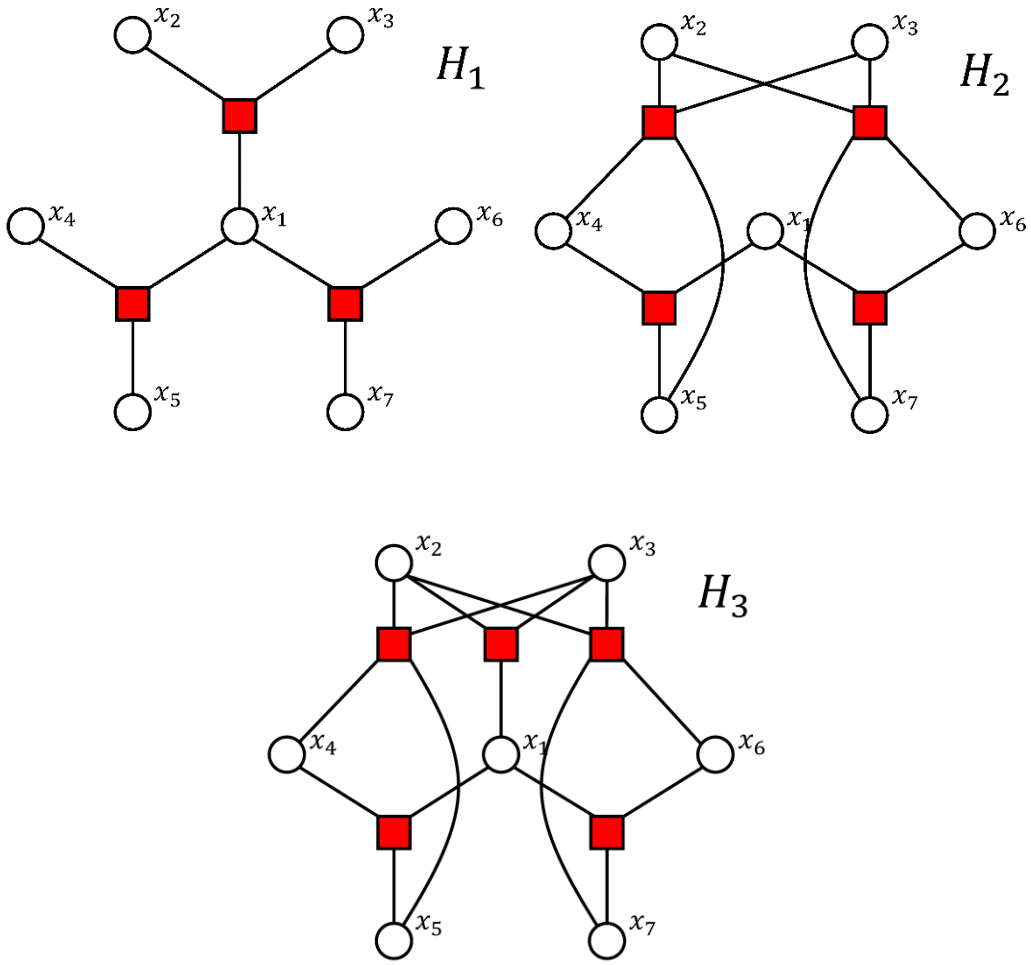

Figure 2: H_1, H_2, and H_3 are three representations of the same code. Here, H_1 is cycle-free and is automatically pseudocodeword-free. The second representation H_2, on the other hand, fail our criteria and has pseudocodewords $\{(2,0,0,1,1,1,1), (0,2,0,1,1,1,1), (0,0,2,1,1,1,1), (2,2,2,1,1,1,1)\}$ *Although the third representation H_3 has all the cycles presented in H_2, it contains the tree structure from H_1 and is demonstrated to be pseudocodeword-free.*

Our results agree with the experiment performed in MATLAB using several codes such as Hamming codes, cycle codes, and cycle-free codes. Randomized linear combination of rows are added in order to obtain another representation of the same code. New constraints may be added so as to make the code pseudocodeword-free. For example, pseudocodewords of the Hamming code of length 7 are computed using representations

$$H_1 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

and

$$H_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Here, both representations produce preudocodewords unless linearly independent parity conditions are introduced.

Fibonacci Codes

Suppose that arbitrary positive integers are to be sent sequentially over a binary channel. One simple way to do this is to encode each integer using its base-2 representation and send the corresponding string of 0 and 1 over the channel. An obvious problem with this approach is that the receiver needs to know the length of each message a priori. A popular alternative approach is to use variable-length prefix codes. Here, the number of bits in each codeword varies, and a receiver tell each codeword apart using some special properties from the design of the code. Fibonacci code is a prefix code that makes use of the fact that the Zeckendorf representation of an integer involves no 2 consecutive terms from the Fibonacci sequence. In addition, it is resistant to insertion and deletion errors. However, Zeckendorf representation had been limited to the integers.

Thus, we study generalizations of Zeckendorf's theorem and study any sequence that satisfies the generalized Fibonacci recurrence relation and examine elements that can be written as a sum of terms from the sequence. Since every term in the sequence is practically a linear combination of the sequence's initial conditions, one can at best hope to generate a module using sums of elements from this sequence. Hence, by design we adopt free $\mathbb{Z}$-modules as a mathematical space of interest.

To prove our results, we introduce the notion of k-equivalent sequences. This approach technically treats elements that can be generated from a Fibonacci sequence as a number system using sequence elements as a basis. These so-called coefficients can be manipulated, and we show that our coefficient manipulations operate independently of the underlying Fibonacci sequence. This approach frees us from having to worry about the choice of the initial conditions and makes our results hold in a broad sense.

We are able to show, theoretically and experimentally, that every element of a free $\mathbb{Z}$-module can be represented as a sum of elements from a Fibonacci sequence of higher order. As a result, our work unifies several variations of the classical theorem while also providing new grounds for its legitimacy. A sufficient condition that makes the representation unique is also given. Perhaps more importantly, this enables Fibonacci coding for modules. Not only that our work substantially enlarges the message spaces for Fibonacci code, the resulting codes inherit robustness property from the standard Fibonacci code. Naturally, this also gives binary numeration systems for spaces such as Gaussian integers, Eisenstein integers, quarternions, the ring of integers of any algebraic number field, and lattices.

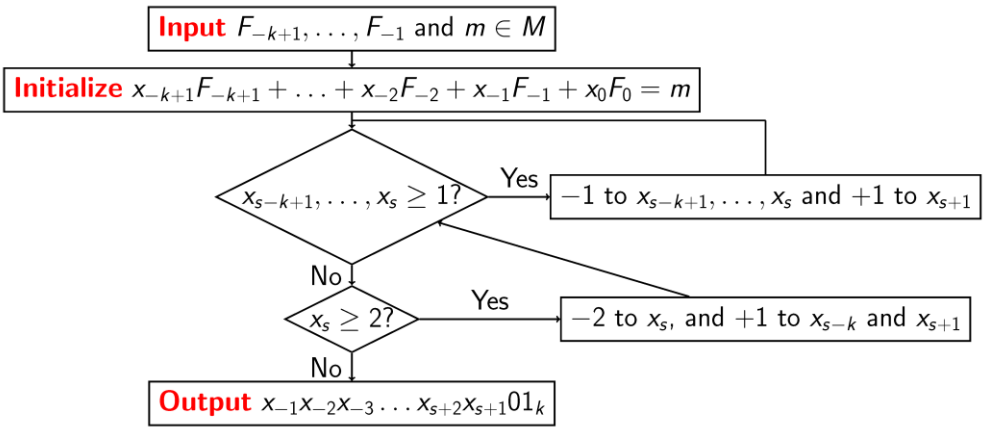

Figure 3: Encoding algorithm for Fibonacci coding

Since the classical Zeckendorf representation contains no two consecutive terms, each codeword of the Fibonacci code contains exactly one instance of "11" at the end. Each codeword in our generalized scheme contains exactly one instance k consecutive 1's at the end. When codewords are sent sequentially over a channel, the receiver only needs to look for the occurrence of k consecutive 1's to separate each codeword.

As an example, consider a Fibonacci sequence

$$\ldots, -4+4i, 5+i, -2-3i, -1+2i, 2, -1-i, i, 1$$

Every Gaussian integer can be written as a sum of elements from this sequence with no 3 consecutive terms. For instance,

$$-2 = (-2-3i) + (-1+2i) + (i) + (1).$$

and

$$-2+3i = (-4+4i) + (2) + (-1-i) + (1).$$

We illustrate Fibonacci code for several Gaussian integers in the table below.

	$a=-2$	$a=-1$	$a=0$	$a=1$	$a=2$
$b=2i$	01100111	00000111	10000111	00010111	1001011
$b=i$	00100111	10100111	00111	10111	0100111
$b=0$	110010111	010111	111	0111	0000111
$b=-i$	100010111	000111	100111	0010111	1010111
$b=-2i$	010000111	110000111	010100111	110000111	0110010111

Table 1: Fibonacci code for $a+b$ are given. Note that each code contains exactly one instance of 111.

Lattice Codes

Construction of lattices from linear codes has facilitated a powerful way of designing efficient Euclidean codes for communications over Additive White Gaussian Noise (AWGN) channels. In multi-level constructions, increasing powers of the base, which is typically 2 or $1+i$, are multiplied to the binary code at each level in the chain, thus creating artificial "levels,"' which has useful applications in the design of low-complexity lattice decoders as it allows multi-stage decoding. A classical example of such processes is the construction of Barnes-Wall lattices from Reed-Muller codes. In particular, let $RM(r,m)$ denote the Reed-Muller code of order r and length $n=2^m$. The complex BW lattice of dimension 2^m can be obtained from Construction C using Reed-Muller codes as

$$BW_{2^m} = \sum_{r=0}^{m-1} (1+i)^r RM(r,m) + (1+i)^m \mathbb{Z}[i]^{2^m}.$$

Although the above encoding procedure involves embedding Reed-Miller codewords onto different levels of multi-level construction, the resulting codewords need to be suitably shaped in order to optimize the performance of the lattice codes in terms of average energy consumption. Here, the set of complex points in each dimension is of the form

$$a_0 + a_1(1+i) \dots + a_{m-1}(1+i)^{m-1}$$

where $a_j \in \{0,1\}$. This set forms a fractal-like shape whose boundary is known as a twin-dragon curve. While it is possible to center the constructed region at the origin and reduce transmission energy somewhat, there exists a simple map that equipped the above region with a cubic shaping.

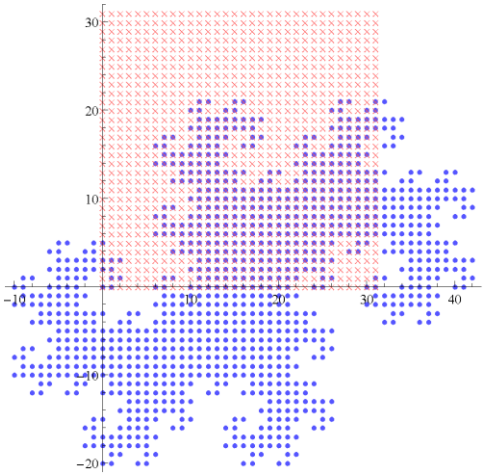

Figure 4: The set of complex points of the form $a_0 + a_1(1+i) \ldots + a_9(1+i)^9$ *where* $a_j \in \{0,1\}$ *(marked in blue) takes a cubic-shaped constellation (marked in red) upon applying the mapping.*

We proposed a generalized shaping map $t_d: S_{\alpha,m} \to \mathbb{Z}_{d,d^*}[\theta]$ given by

$$t_d(a + b\theta) = (a \bmod d) + (b \bmod d^*)\theta$$

where

$$S_{\alpha,m} = \left\{ a_0 + a_1\alpha \ldots + a_{m-1}\alpha^{m-1} \middle| a_j \in \{0,1,2, \ldots, N(\alpha) - 1\} \right\}$$

and

$$\mathbb{Z}_{d,d^*}[\theta] = \{x + y\theta | x \in \mathbb{Z}_d \; and \; y \in \mathbb{Z}_{d^*}\}$$

For several values of α, we prove that the map t_d is a bijection. As a result, the proposed mapping is applicable for shaping lattice codes obtained via multi-level constructions from linear codes over $\mathbb{F}_q$, for $q > 2$.

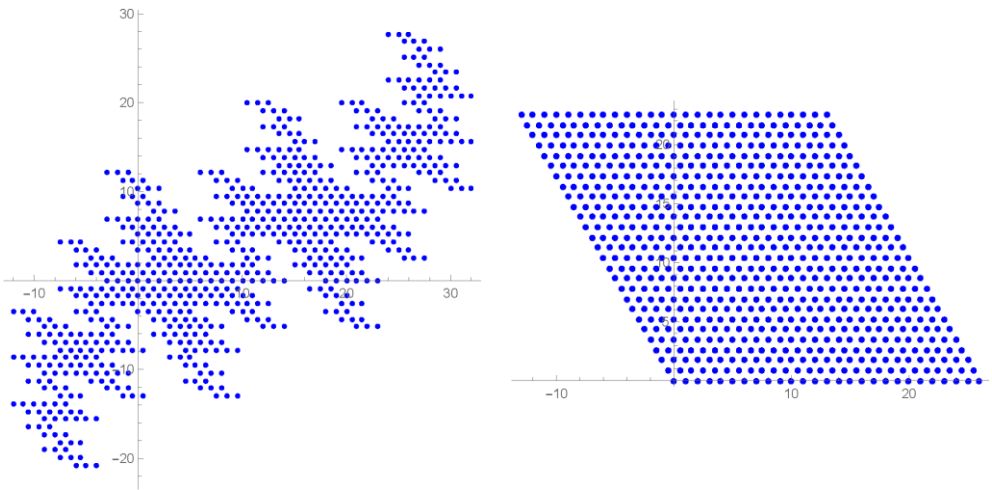

Figure 5: The constellation $S_{-1+\omega}$ *before (left) and after (right) shaping*

The mapping presented here satisfies the objectives of the shaping map: average energy of the image is reduced, and the lattice structure of the points is retained. In addition, since the information bits are embedded until level $m-1$, translation of the codewords by a multiple of α^m still makes them amenable to low-complexity multi-level decoding. This work finds applications in using multi-level constructions of lattice codes over $\mathbb{Z}[\theta]$ for communication over AWGN channels. Similar to Barnes-Wall lattices, multi-level constructions in $\mathbb{Z}[\theta]$ allow the potential use of low-complexity successive cancellation decoders at the receiver, which instead of directly decoding the lattice codewords decode linear code at each level of the construction.

Summary and comment

Non-binary multi-level construction of lattices has many promising properties that has yet to be exploited. For example, in terms of density, kissing number, and coding gain, $\mathbb{Z}[\omega]$ is a better lattice than $\mathbb{Z}[i]$ and is the best lattice in two dimensions. Our work addresses the shaping of lattice constellations carved from lattices over $\mathbb{Z}[\theta]$, and we are able to demonstrate several tileable constellations, both theoretically and numerically.

Exponent	$\pm(2+\omega)$	$\pm(1+2\omega)$	$\pm(-1+\omega)$
$m \equiv 1 \pmod 2$	$3^{\frac{m}{2}}$	$3^{\frac{m}{2}}$	$3^{\frac{m}{2}}$
$m \equiv 1 \pmod 6$	$3^{\frac{m+1}{2}}$	$3^{\frac{m+1}{2}}$	$3^{\frac{m+1}{2}}$
$m \equiv 3 \pmod 6$	$3^{\frac{m+1}{2}}$		$3^{\frac{m-1}{2}}$
$m \equiv 5 \pmod 6$	$3^{\frac{m-1}{2}}$		$3^{\frac{m+1}{2}}$

Table 2: Tileable sets $S_{\alpha,m}$ for the ring $\mathbb{Z}[\omega]$ for all $m \geq 2$

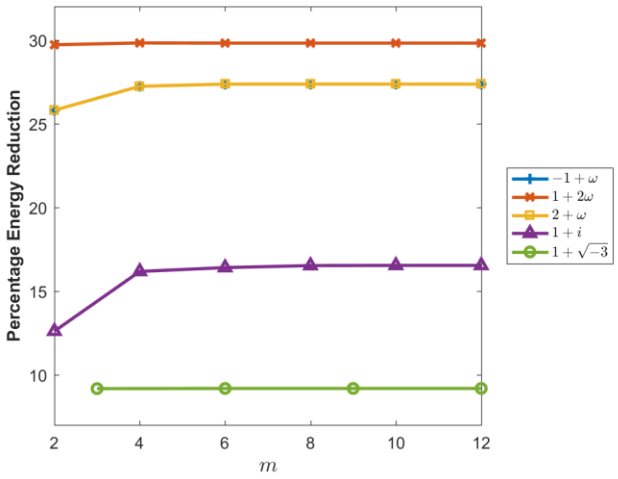

Figure 6: Average energy reduction (in percentage) for some values of α

We identify pseudocodeword-free codes as codes with the property that

$$PC(H) = \{\sum_{c \text{ is a codeword}} a_c c \,|a_c \in \mathbb{N}\}$$

This means that the set of pseudocodewords is kept as small as possible. Our work provides an exact condition for codes with cycle-free Tanner graph to be pseudocodeword-free. The results given here yield a surprising insight on the structure of the pseudocodewords: good representation for this class of codes has nothing to do with cycles as long as there exists a spanning tree of the Tanner graph that represents the same code. As a result, this work sheds light on empirical phenomenons where, under certain circumstances, iterative decoders perform well despite a number of small cycles in the Tanner graph and eliminating small cycles does not significantly improve decoding performance of LDPC codes.

Recommendation and Future works

Digital communication affects the lives of many people, and it remains an important research topic to improve the speed, performance, and security of communication over these mediums. This work improves upon and offers a solution to a communication problem over a certain situation. Nonetheless, several prominent relevant questions remain unanswered. For example, examining other conditions to which the Zeckendorf representation remains unique will give a complete solution to the problem of identifying mathematical spaces that permit Fibonacci code. Under such environments, one can view Zeckendorf representation as a number system and develop generalized Zeckendorf arithmetic. In addition, it

would also be interesting to study the proposed coding algorithm from the perspective of data compression and computational complexity.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

1.1 P. Pooksombat, J. Harshan, and **W. Kositwattanarerk**, On Shaping Complex Lattice Constellations from Multi-level Constructions, IEEE International Symposium on Information Theory (2017), 2598-2602.

1.2 P. Pooksombat, P. Udomkavanich, and **W. Kositwattanarerk**, Zeckendorf's Theorem and Fibonacci Coding for Modules, submitted to Discrete Applied Mathematics.

1.3 **W. Kositwattanarerk**, Pseudocodeword-Free Criterion for Codes with Cycle-Free Tanner Graph, submitted to Designs, Codes and Cryptography.

2. การนำผลงานวิจัยไปใช้ประโยชน์

2.1 **การสร้างเครือข่ายความร่วมมือ:** หัวหน้าโครงการได้รับเชิญไปร่วมงานวิจัยกับ Nanyang Technological University, Singapore ระหว่างวันที่ 2-6 พฤศจิกายน 2558 และได้มีผลงานวิจัยร่วมกับ Jagadeesh Harshan จาก Human-Centered Cyber-Physical Systems Programme, Advanced Digital Sciences Center, Agency for Science, Technology and Research (A*STAR), Singapore

ในโอกาสถัดมา หัวหน้าโครงการได้เชิญ Gary Greaves จาก Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore มาพูดสัมมนาหัวข้องานวิจัยที่ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย และ ภาควิชาคณิตศาสตร์ มหาวิทยาลัยมหิดล ส่งผลให้ มีนักศึกษาระดับปริญญาเอก 1 คน คือ ศรุต มลิลา จาก ภาควิชาคณิตศาสตร์ มหาวิทยาลัยมหิดล ได้ไปร่วมงานกับ Gary Greaves ที่ Nanyang Technological University, Singapore

2.2 **การสร้างนักวิจัยใหม่:** หัวหน้าโครงการได้รับนักศึกษาระดับปริญญาเอกที่ได้หัวข้องานวิจัยจากโครงการนี้ 1 คน คือ นาย พีรธร ผูกสมบัติ ปัจจุบันกำลังศึกษาอยู่ที่ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล

3. การเสนอผลงานในที่ประชุมวิชาการ

3.1 หัวหน้าโครงการได้รับเชิญไปพูดสัมมนางานวิจัยที่ Nanyang Technological University, Singapore ในวันที่ 4 พฤศจิกายน 2558 ในหัวข้อ Recent Results on Constructions of Lattices from Codes

3.2 หัวหน้าโครงการได้รับเชิญไปพูดสัมมนาที่มหาวิทยาลัยบูรพา ในวันที่ 16 กุมภาพันธ์ 2560 ในหัวข้อ Introduction to Coding Theory and Cryptography

3.3 หัวหน้าโครงการได้รับเชิญไปนำเสนองานวิจัยในงานประชุมวิชาการ The 22nd Annual Meeting in Mathematics (AMM 2017) ที่ จ.เชียงใหม่ ระหว่างวันที่ 2-4 มิถุนายน 2560 ในหัวข้อ Generalized Zeckendorf's Theorem and Fibonacci Coding for Modules

ภาคผนวก

On Shaping Complex Lattice Constellations from Multi-level Constructions

Perathorn Pooksombat
Department of Mathematics
Faculty of Science, Mahidol University
Bangkok, Thailand 10400
Email: perathorn.pok@student.mahidol.edu

J. Harshan
Advanced Digital Sciences Center
Singapore
Email: harshan.j@adsc.com.sg

Wittawat Kositwattanarerk
Department of Mathematics
Centre of Excellence in Mathematics
Faculty of Science, Mahidol University
Bangkok, Thailand 10400
Email: wittawat.kos@mahidol.edu

***Abstract*—Constructions of lattices are known to have a strong connection to the study of classical linear codes over $\mathbb{F}_2$; one such celebrated construction is that of Barnes-Wall (BW) lattices over $\mathbb{Z}[i]$, with $i = \sqrt{-1}$, wherein weighted sum of nested Reed-Muller codes over powers of the base $1 + i$ leads to the famous *multi-level construction* of BW lattices. Although these constructions facilitate simple encoding and decoding of information bits, the resulting codewords need to be mapped onto their representatives in order to reduce the average energy of the lattice code. Drawing inspirations from the case of BW lattices, in this work, we address a general question of how to arrive at representatives of the codewords of a lattice code over $\mathbb{Z}[\theta]$, where θ is a quadratic integer, that is generated via a multi-level construction over linear codes over $\mathbb{F}_q$, for $q > 2$. In particular, we introduce a novel shaping function $\tau : \mathbb{Z}[\theta] \longrightarrow \mathbb{Z}[\theta]$ on the components of lattice codewords, and prove that the natural constellation of lattices from multi-level constructions can be rearranged into a multi-dimensional cube or parallelepiped under such a map. We demonstrate numerically that our mapping results in a reduction of the average energy of lattice constellations. Our proposed mapping has applications in communications, particularly in encoding and decoding of lattice codes from multi-level constructions over q-ary linear codes.**

I. Introduction

Construction of lattices from linear codes has facilitated a powerful way of designing efficient Euclidean codes for communications over Additive White Gaussian Noise (AWGN) channels. For instance, Construction A produces lattices from a single linear code and equips the resulting lattice with practical encoding and decoding capabilities. Also, many important lattice properties such as volume and minimum distance can be easily derived from the base code, which makes such construction of lattices indeed convenient. Other examples include multi-level constructions of lattices such as Construction C, D, D', and Forney's code formula, which use a chain of linear codes as an ingredient, and these recipes are known to produce lattice codes with better parameters than the simpler constructions [2]. In multi-level constructions, increasing powers of the base, which is typically 2 or $1+i$, are multiplied to the binary code at each level in the chain, thus creating artificial "levels," which has useful applications in the design of low-complexity lattice decoders as it allows multi-stage decoding [4], [7], [8]. In this paper, we are interested in multi-level constructions of lattices from linear codes.

A classical example of such processes is the construction of Barnes-Wall lattices from Reed-Muller codes. In particular, let $\mathcal{RM}(r, m)$ denote the Reed-Muller code of order r and length $n = 2^m$. The complex BW lattice of dimension 2^m can be obtained from Construction C using Reed-Muller codes as

$$BW_{2^m} = \sum_{r=0}^{m-1} (1+i)^r \mathcal{RM}(r, m) + (1+i)^m \mathbb{Z}[i]^{2^m}. \quad (1)$$

Here and throughout the paper, we use $+$ to denote addition over $\mathbb{C}$. Note that we abuse the notation by treating elements of $\mathbb{F}_2^n$ as reals so that they can be added and multiplied by scalars like real (complex) vectors. Later, Forney [4] identified the coset codes of BW lattices to be Reed-Muller codes, i.e., one may decompose $\Lambda(0, m) := BW_{2^m}$ as

$$\Lambda(0, m)/\Lambda(1, m)/\ldots/\Lambda(m, m) \quad (2)$$

where

$$\Lambda(r, m) = \sum_{s=r}^{m-1} (1+i)^{s-r} \mathcal{RM}(s, m) + (1+i)^{m-r} \mathbb{Z}[i]^{2^m}. \quad (3)$$

This decomposition results in what is called a *code formula* for the BW lattices.

Recently, [3] has studied the performance of Barnes-Wall lattice codes in efficiently communicating information over AWGN channels. Specifically, the authors studied how to encode and decode complex BW lattice codes that are generated from a multi-level construction using RM codes. The emphasis on employing multi-level constructions was attributed to the existence of high-performance, low-complexity successive-cancellation decoders. Although the encoding procedure involves embedding RM codewords onto different levels of multi-level construction, the resulting codewords were suitably shaped in order to optimize the performance of the lattice codes in terms of average energy consumption. In their construction, the set of complex points in each dimension is of the form $a_0 + a_1(1+i) + \ldots + a_{m-1}(1+i)^{m-1}$ where $a_j \in \{0, 1\}$. This set forms a fractal-like shape whose boundary is known as a twin-dragon curve. While it is possible to center the constructed region at the origin and reduce transmission energy somewhat, [3] found a simple map that equipped the above region with a cubic shaping. For example, Fig. 1 illustrates

978-1-5090-4096-4/17/$31.00 ©2017 IEEE

the constellations before and after applying the mapping in [3]. In a nutshell, when m is even, $(1+i)^m$ takes one of the points in $\{\pm 2^{\frac{m}{2}}, \pm 2^{\frac{m}{2}} i\}$, and therefore applying modulo $2^{\frac{m}{2}}$ on the real and imaginary part of a Gaussian integer translates it to its own representative in $\mathbb{Z}[i]/(1+i)\mathbb{Z}[i]$ without meddling with lattice properties. We note here that, in theory, spherical shaping is optimal in terms of average energy. Nonetheless, doing so in practice would normally require an erratic codebook, possibly forgoing lattice property in the process. The mapping presented here satisfies the objectives of the shaping map: average energy of the image is reduced, and the lattice structure of the points is retained. In addition, since the information bits are embedded until level $m-1$, translation of the codewords by a multiple of $(1+i)^m$ still makes them amenable to low-complexity multi-level decoding.

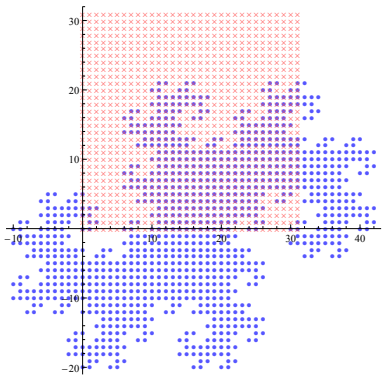

Fig. 1. The set of complex points of the form $a_0 + a_1(1+i) + \ldots + a_9(1+i)^9$, $a_j \in \{0,1\}$, (marked in blue) takes a cubic-shaped constellation (marked in red) upon applying the mapping in [3].

Inspired by the results of [3] and powerful multi-level constructions of lattices in recent literatures, in this work we address the shaping of lattice constellations carved from lattices over $\mathbb{Z}[\theta]$, where θ can take values other than $1+i$. Our specific contributions are summarized below:

- As a generalization of the work in [3], we first investigate shaping of complex lattice constellations from BW lattices over $\mathbb{Z}[i]$ using generic bases. Then, we map their elements to an image that provides cubic shaping. (See Section II.) Although the proposed method retains lattice structure only for even values of m, it lays down foundations to derive mappings on lattice constellations from generalized lattices over $\mathbb{Z}[\theta]$.
- We study shaping maps on lattice constellations carved from lattices over $\mathbb{Z}[\theta]$, where θ is a quadratic integer. The proposed mapping is applicable for shaping lattice codes obtained via multi-level constructions from linear codes over $\mathbb{F}_q$, for $q > 2$. In particular, we apply our procedures on lattices over $\mathbb{Z}[\omega]$, where $\omega = -\frac{1}{2} + \frac{\sqrt{-3}}{2}$, and with bases $\pm(2+\omega)$, $\pm(1+2\omega)$, and $\pm(-1+\omega)$, to show that the image of our mapping yields reduced average energy as well as retains the lattice structure for even values of m. To prove our results, we use the property that the base values are real radicals. Identifying suitable bases for our objectives is nontrivial, as providing cubic- or parallelepiped-shaping for a lattice constellation and at the same time preserving its structure is hardly achievable. (See Section III.)

Our work finds applications in using multi-level constructions of lattice codes over $\mathbb{Z}[\theta]$ for communication over AWGN channels. Similar to BW lattices, multi-level constructions in $\mathbb{Z}[\theta]$ allow the potential use of low-complexity successive cancellation decoders at the receiver, which instead of directly decoding the lattice codewords decode linear code at each level of the construction. This aspect of the work is also discussed in Section IV.

II. Shaping Complex Constellations from Barnes-Wall Lattices

In this section, we outline the shaping of complex BW lattices given in [3] and develop a generalization. Denote by $\mathcal{RM}(r,m)$ the Reed-Muller code of order r and length $n = 2^m$. One may write BW lattice as

$$BW_{2^m} = \mathcal{EC}_{2^m} + (1+i)^m \mathbb{Z}[i]^{2^m}$$

where

$$\mathcal{EC}_{2^m} := \sum_{r=0}^{m-1} (1+i)^r \mathcal{RM}(r,m).$$

In each dimension, the elements of $\mathcal{EC}_{2^m}$ have the form $a_0 + a_1(1+i) + \ldots + a_{m-1}(1+i)^{m-1}$ where $a_j \in \{0,1\}$. Depending on the parity of m, a shaping function ϕ is defined as

$$\phi(x) = \begin{cases} x \pmod{2^{\frac{m}{2}}}, & \text{when } m \text{ is even;} \\ \varphi\left(x \pmod{2^{\frac{m+1}{2}}}\right), & \text{when } m \text{ is odd,} \end{cases} \tag{4}$$

where $\varphi : \mathbb{Z}_{2^{\frac{m+1}{2}}}[i] \to \mathbb{Z}_{2^{\frac{m+1}{2}}}[i]$ is given by

$$\varphi(z) = \begin{cases} z, & \text{when } \Im(z) < 2^{\frac{m-1}{2}}; \\ z + \left(2^{\frac{m-1}{2}} - 2^{\frac{m-1}{2}} i\right), & \text{when } \Re(z) < 2^{\frac{m-1}{2}} \\ & \text{and } \Im(z) \geq 2^{\frac{m-1}{2}}; \\ z - \left(2^{\frac{m-1}{2}} + 2^{\frac{m-1}{2}} i\right), & \text{when } \Re(z) \geq 2^{\frac{m-1}{2}} \\ & \text{and } \Im(z) \geq 2^{\frac{m-1}{2}}. \end{cases} \tag{5}$$

It was shown in [3] that ϕ is one-to-one on $\mathcal{EC}_{2^m}$, $\phi(\mathcal{EC}_{2^m})$ is cubic-shaping when m is even, and

$$BW_{2^m} = \phi(\mathcal{EC}_{2^m}) + (1+i)^m \mathbb{Z}[i]^{2^m},$$

altogether meaning that $\phi(\mathcal{EC}_{2^m})$ is an alternative representation of the constellation $\mathcal{EC}_{2^m}$ for the BW lattice that is better in terms of shaping. As a step toward generalizing this contribution, we note that multi-level constructions of BW lattices rely on the fact that the quotient ring $\mathbb{Z}[i]/(1+i)\mathbb{Z}[i]$ is the binary field. In a search for bases other than $1+i$, the next lemma will lead us to the conclusion that $\pm 1 \pm i$ are the only alternatives fit for our purpose. Here, the squared norm of a complex number $x+yi$ is defined by $N(x+yi) = x^2 + y^2$.

Lemma 2.1: Let $\alpha = x + yi \in \mathbb{Z}[i]$. The quotient $\mathbb{Z}[i]/\alpha\mathbb{Z}[i]$ is a ring, and if $\gcd(x,y) = 1$ then $\mathbb{Z}[i]/\alpha\mathbb{Z}[i] \simeq \mathbb{Z}_{N(\alpha)}$.

Since the only Gaussian integers of norm 2 are $\alpha = \pm 1 \pm i$, it follows from the Lemma 2.1 that $\mathbb{Z}[i]/\alpha\mathbb{Z}[i] \simeq \mathbb{Z}_2$ for only these values of α. Now, one can alternatively define complex Barnes-Wall lattices as

$$\sum_{r=0}^{m-1} \alpha^r \mathcal{RM}(r,m) + \alpha^m \mathbb{Z}[i]^{2^m}$$

where $\alpha \in \{1+i, -1+i, 1-i, -1-i\}$. We can see this by noting that groups (and fields) of order 2 are trivially isomorphic, and that the decomposition (2) and (3) can be similarly done using $-1+i$, $1-i$, and $-1-i$ instead of $1+i$. Geometrically speaking, the lattices constructed using $-1+i$, $1-i$, or $-1-i$ as a basis are the mirror images of the lattice constructed using $1+i$ as a basis.

Next, we propose a cubic shaping map analogous to (4) using componentwise modulo which works for all $\alpha = \pm 1 \pm i$.

Definition 2.1: Let m be a positive integer, and let $\alpha \in \{1+i, -1+i, 1-i, -1-i\}$. Define a map $\phi : \mathbb{Z}[i]^{2^m} \to \mathbb{Z}[i]^{2^m}$ by

$$\phi(x+yi) = (x \bmod d) + (y \bmod d^*)i,$$

where $d = 2^{\lfloor \frac{m}{4} \rfloor + \lceil \frac{m}{4} \rceil}$ and $d^* = \frac{2^m}{d}$. We call this map ϕ a *tiling map*, and call d and d^* *width* and *height*, respectively. The values of d and d^* are as shown in the table below.

TABLE I
WIDTH AND HEIGHT OF THE TILING MAP ϕ IN TERMS OF m

	d	d^*
$m \equiv 0 \pmod 2$	$2^{\frac{m}{2}}$	$2^{\frac{m}{2}}$
$m \equiv 1 \pmod 4$	$2^{\frac{m+1}{2}}$	$2^{\frac{m-1}{2}}$
$m \equiv 3 \pmod 4$	$2^{\frac{m-1}{2}}$	$2^{\frac{m+1}{2}}$

We first note that the choice of width and height is independent of α but depends on m, the level of the construction. When m is even, our tiling map is identical to that of [3]. When m is odd, the auxiliary map (5) is incorporated into the modulo operation. Let $\mathcal{L}_{2^m} = \phi(\mathcal{EC}_{2^m})$ be the image under ϕ of the constellation $\mathcal{EC}_{2^m} = \sum_{r=0}^{m-1} \alpha^r \mathcal{RM}(r,m)$. Then, the inclusion $\mathcal{L}_{2^m} \subseteq (\mathbb{Z}_d[i] + \mathbb{Z}_{d^*}[i])^{2^m}$ follows directly from the definition of the tiling map. Indeed, we will argue next that they cannot be proper.

Proposition 2.2: The tiling map ϕ restricted to $\mathcal{EC}_{2^m}$ is a one-to-one correspondence.

Theorem 2.3: When m is even, the constellation $\mathcal{EC}_{2^m}$ of a Barnes-Wall lattice can be replaced by the tiled constellation $\mathcal{L}_{2^m}$; that is,

$$BW_{2^m} = \mathcal{EC}_{2^m} + \alpha^m \mathbb{Z}[i]^{2^m} = \mathcal{L}_{2^m} + \alpha^m \mathbb{Z}[i]^{2^m}.$$

The proofs of the above proposition and theorem are similar to those in [3] and are omitted for brevity. The key aspect of Proposition 2.2 is that the tiling map ϕ, when restricted to the set of all elements of the form

$$\mathbf{a}_0 + \mathbf{a}_1\alpha + \mathbf{a}_2\alpha^2 + \cdots + \mathbf{a}_{m-1}\alpha^{m-1}$$

where $\mathbf{a}_j \in \{0,1\}^{2^m}$, is a one-to-one correspondence. Therefore, the tiling map restricted to $\mathcal{EC}_{2^m}$, which is a subset of the aforementioned set, is also injective. Finally, Theorem 2.3 makes it possible to apply the map in application; the tiling map ϕ preserves the geometry of the lattice as both $\mathcal{EC}_{2^m}$ and $\mathcal{L}_{2^m}$ serve as a constellation of the BW lattice. In other words, the shaping map does not affect the underlying lattice structure while also reducing average transmission energy.

III. Shaping Complex Constellations from Generic Multi-level Constructions

From the discussion in the preceding section, it is clear that when $\alpha = \pm 1 \pm i$, there are 2^m elements of the form

$$a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1},$$

for $a_j \in \{0,1\}$. We call α the *base* of the expansion, and m an *exponent*. In this section, we develop a generalization of the shaping map given in Definition 2.1 to rings other than $\mathbb{Z}[i]$ and bases other than $\alpha = \pm 1 \pm i$. In particular, we are interested in understanding the multi-level decomposition of elements in the rings of the form $\mathbb{Z}[\theta]$, where θ is a quadratic integer defined as a root of the equation $x^2 + Ax + B = 0$, such that $A \in \{0,1\}$ and $B > 0$. In such a generalized structure of rings, the elements are of the form $x + y\theta$, for $x, y \in \mathbb{Z}$.

Before we present the main results, we develop necessary ingredients for Proposition 2.2 to work in the general setting of $\mathbb{Z}[\theta]$. Towards that direction, we first generalize the results of Lemma 2.1 to $\mathbb{Z}[\theta]$ in the following lemma, where $N(x + y\theta) := x^2 - Axy + By^2$ denotes the squared norm of $x + y\theta$.

Lemma 3.1: Let $\alpha = x + y\theta \in \mathbb{Z}[\theta]$. The quotient $\mathbb{Z}[\theta]/\alpha\mathbb{Z}[\theta]$ is a ring, and if $\gcd(x,y) = 1$ then $\mathbb{Z}[\theta]/\alpha\mathbb{Z}[\theta] \simeq \mathbb{Z}_{N(\alpha)}$.

In the next subsection, we describe a generalized decomposition of elements in $\mathbb{Z}[\theta]$ over a suitable base, and explain how to arrive at a cubic- or parallelepiped-shaped constellations in $\mathbb{Z}[\theta]$.

A. Generalized Tiling Map in $\mathbb{Z}[\theta]$

Let $\alpha = x + y\theta \in \mathbb{Z}[\theta]$ such that $\gcd(x,y) = 1$. Further, let $\mathcal{S}_{\alpha,m} \subset \mathbb{Z}[\theta]$ denote the set of elements of $\mathbb{Z}[\theta]$ of the form

$$a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{m-1}\alpha^{m-1}, \tag{6}$$

where $a_j \in \{0, 1, 2, \ldots, N(\alpha) - 1\}$, and $m \geq 1$ is an integer. Since $\mathbb{Z}[\theta]/\alpha\mathbb{Z}[\theta] \simeq \mathbb{Z}_{N(\alpha)} = \{0, 1, 2, \ldots, N(\alpha) - 1\}$, each element of $\mathcal{S}_{\alpha,m}$ has a unique representation in (6). Therefore, the number of elements in $\mathcal{S}_{\alpha,m}$ is exactly $N(\alpha)^m$.

Let us denote another subset of $\mathbb{Z}[\theta]$ of the form

$$\mathbb{Z}_{d,d^*}[\theta] = \{x + y\theta \mid x \in \mathbb{Z}_d \text{ and } y \in \mathbb{Z}_{d^*}\}.$$

It is clear that $\mathbb{Z}_{d,d^*}[\theta]$ has better structure than $\mathcal{S}_{\alpha,m}$ for arbitrary values of θ, α, and m. Identifying the shaping benefits of $\mathbb{Z}_{d,d^*}[\theta]$, we generalize the tiling map given in Definition 2.1 below.

Definition 3.1: A *generalized tiling map* $\tau_d : \mathcal{S}_{\alpha,m} \to \mathbb{Z}_{d,d^*}[\theta]$ is defined by

$$\tau_d(a + b\theta) = (a \bmod d) + (b \bmod d^*)\theta,$$

where $dd^* = N(\alpha)^m$.

Observe that the number of elements in $\mathbb{Z}_{d,d^*}[\theta]$ is dd^*. Thus, if the map τ_d is one-to-one, then it gives a cubic (parallelepiped) shaping to the set $\mathcal{S}_{\alpha,m}$. We make this precise in the following definition.

Definition 3.2: The set $\mathcal{S}_{\alpha,m}$ is said to be *tileable* if there exists a width d such that τ_d is a bijection. Such a width d is called *tiling width*.

In the next section we continue our quest to identify sets $\mathcal{S}_{\alpha,m}$ in $\mathbb{Z}[\theta]$ that are *tileable*.

B. Main Theoretical Results

In the rest of this section, we present the main findings of this work. We showcase certain values of θ and α for which the sets $\mathcal{S}_{\alpha,m}$ are tileable for any $m \geq 2$. The algebraic rings of interest are Eisenstein integers $\mathbb{Z}[\omega]$, where $\omega = -\frac{1}{2} + \frac{\sqrt{-3}}{2}$ is the primitive third root of unity, and $\mathbb{Z}[\sqrt{-3}]$. In the course of proving tileability of the above sets, we make use of the special property that $\pm(2+\omega)$, $\pm(1+2\omega)$, $\pm(-1+\omega)$, and $\pm 1 \pm \sqrt{-3}$ are real radicals, i.e., these numbers are radicals of a real number. We only provide the proof for the case $\theta = \omega$ and $\alpha = \pm(2+\omega)$ for a number of cases, and omit the proofs for the rest.

Theorem 3.2: Let α be among

$$\pm(2+\omega), \pm(1+2\omega), \pm(-1+\omega).$$

In the ring $\mathbb{Z}[\omega]$, the set $\mathcal{S}_{\alpha,m}$ is tileable for all $m \geq 2$ with the tiling width given in Table II.

TABLE II

Exponent	$\pm(2+\omega)$	$\pm(1+2\omega)$	$\pm(-1+\omega)$
$m \equiv 0 \pmod 2$	$3^{\frac{m}{2}}$	$3^{\frac{m}{2}}$	$3^{\frac{m}{2}}$
$m \equiv 1 \pmod 6$	$3^{\frac{m+1}{2}}$	$3^{\frac{m+1}{2}}$	$3^{\frac{m+1}{2}}$
$m \equiv 3 \pmod 6$	$3^{\frac{m\pm1}{2}}$		$3^{\frac{m-1}{2}}$
$m \equiv 5 \pmod 6$	$3^{\frac{m-1}{2}}$		$3^{\frac{m\pm1}{2}}$

Proof: We give a proof for the case $\alpha = 2+\omega$. Notice that $N(\alpha) = 3$, and so $|\mathcal{S}_{\alpha,m}| = 3^m$. For an exponent $m \equiv 0 \pmod 6$, we choose $d = d^* = 3^{\frac{m}{2}}$. Let $s \in \mathcal{S}_{2+\omega,m}$. It follows that $\tau_d(s) = s + 3^{\frac{m}{2}} r$ for some $r \in \mathbb{Z}[\omega]$. Notice that $(2+\omega)^6 = -27$, and so $(2+\omega)^m = (-27)^{\frac{m}{6}} = (-1)^{\frac{m}{6}} 3^{\frac{m}{2}}$. Suppose that $s = a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}$ where $a_j \in \{0,1,2\}$. Then,

$$\tau_d(s) = \sum_{j=0}^{m-1} a_j \alpha^j + (-1)^{\frac{m}{6}} \alpha^m r,$$

which is essentially a decomposition base α. Hence, τ_d is bijective.

For an exponent $m \equiv 1 \pmod 6$, we choose $d = 3^{\frac{m+1}{2}}$ and $d^* = 3^{\frac{m-1}{2}}$. Again, it follows that $\tau_d(s) = s + 3^{\frac{m-1}{2}} r$ where $r = 3x + y\omega$ for some $x, y \in \mathbb{Z}$. We can now apply τ_d straightforwardly and obtain

$$\begin{aligned}\tau_d(s) &= \sum_{j=0}^{m-1} a_j \alpha^j + 3^{\frac{m-1}{2}} r \\ &= \sum_{j=0}^{m-1} a_j \alpha^j + (-1)^{\frac{m-1}{6}} \alpha^{m-1} r \\ &= \sum_{j=0}^{m-2} a_j \alpha^j + \left(a_{m-1} + (-1)^{\frac{m-1}{6}} r\right) \alpha^{m-1}.\end{aligned}$$

It is left only to show that

$$a_{m-1} + (-1)^{\frac{m-1}{6}} r = a'_{m-1} + (-1)^{\frac{m-1}{6}} r'$$

implies $a_{m-1} = a'_{m-1}$ and $r = r'$, so that we can finish the argument using unique decomposition base α. Comparing the integer coefficients of 1 and ω in the above equation yields

$$a_{m-1} + (-1)^{\frac{m-1}{6}} 3x = a'_{m-1} + (-1)^{\frac{m-1}{6}} 3x'$$

and $y = y'$. Since $a_{m-1}, a'_{m-1} \in \{0,1,2\}$, we reduce the above equation modulo 3 and conclude that $a_{m-1} = a'_{m-1}$. This completes the proof that τ_d is bijective for the case $m \equiv 1 \pmod 6$. The bijectivity of τ_d for other cases of m and α can be done in a similar fashion and is omitted. ■

It can be seen from Table II that the tiling width for each values of α is not the same and is dependent of the exponent m, which is the level of the construction. The next pair of figures portrays the set $\mathcal{S}_{-1+\omega,6}$ before and after shaping. It should be clear that, after centering, the fractal consumes more transmission energy on average. Indeed, the shaping τ_{27} here saves energy by 27.36%.

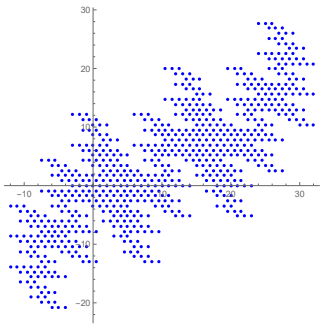

Fig. 2. $\mathcal{S}_{-1+\omega,6}$

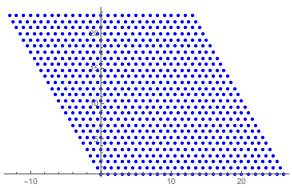

Fig. 3. $\tau_{27}(\mathcal{S}_{-1+\omega,6})$

The next set of results is for the ring $\mathbb{Z}[\sqrt{-3}]$ with base $\alpha = \pm 1 \pm \sqrt{-3}$. Figure 4 and 5 then depict that $\mathcal{S}_{1-\sqrt{-3},5}$ is tileable.

Theorem 3.3: In the ring $\mathbb{Z}[\sqrt{-3}]$, the set $\mathcal{S}_{\pm1\pm\sqrt{-3},m}$ is tileable for all $m \geq 2$ with the tiling width $2^{2\lceil \frac{m}{3} \rceil + \lfloor \frac{m}{3} \rfloor}$.

Figure 6 depicts the reduction in average transmission energy for the values of α given in [3], Theorems 3.2 and 3.3. Note that the diminution percentage converges quickly due to the self-repeating nature of fractals $\mathcal{S}_{\alpha,m}$. Here, average Euclidean norm for the entire constellation are used, and we

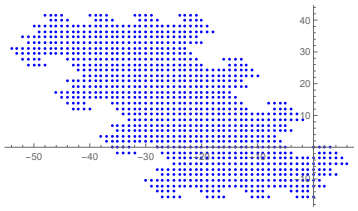

Fig. 4. $\mathcal{S}_{1-\sqrt{-3},5}$ (scaled)

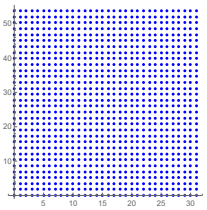

Fig. 5. $\tau_{32}\left(\mathcal{S}_{1-\sqrt{-3},5}\right)$ (scaled)

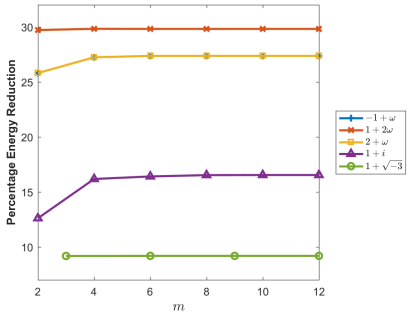

Fig. 6. Average energy reduction (in percentage) for some values of α.

remark that the results may differ when specific codes are applied.

IV. Applications to Lattice Construction and Encoding

Non-binary multi-level construction of lattices has many promising properties that is yet to be exploited. For example, in terms of density, kissing number, and coding gain, $\mathbb{Z}[\omega]$ is a better lattice than $\mathbb{Z}[i]$ and is the best lattice in two dimensions. Nonetheless, due to the constraint on the norm of its elements, $\mathbb{Z}[\omega]$ is not compatible with binary codes but supports ternary codes well. In addition, quaternary codes are practical for lattices constructed over $\mathbb{Z}[\sqrt{-3}]$. Our next theorem asserts that the tiling map given in the previous section preserves the arrangement of lattice points.

Theorem 4.1: Let $\alpha \in \mathbb{Z}[\theta]$, and let $\Lambda = \mathcal{EC} + \alpha^m\mathbb{Z}[\theta]^n$ be a lattice constructed from a multi-level construction using codes over $\mathbb{F}_q$ where $q = N(\alpha)$ with m levels. Then, $\Lambda = \mathcal{L} + \alpha^m\mathbb{Z}[\theta]^n$ where $\mathcal{L} = \tau_d(\mathcal{EC})$ for q, θ, α, and m as given in table III below.

TABLE III

q	θ	α	m
2	i	$\pm 1 \pm i$	$m \equiv 0 \pmod 2$
3	ω	$\pm(2+\omega)$	$m \equiv 0 \pmod 2$
		$\pm(1+2\omega)$	$m \equiv 0 \pmod 2$
		$\pm(-1+\omega)$	$m \equiv 0 \pmod 2$
4	$\sqrt{-3}$	$\pm 1 \pm \sqrt{-3}$	$m \equiv 0 \pmod 3$

Proof: Each value of m in Table III corresponds to the event that $d = d^*$ from Theorem 3.2-3.3. In addition, one can check that $\alpha^m = u|\alpha|^m$, where u is a unit in $\mathbb{Z}[\theta]$. Let $\mathbf{x} \in \Lambda$, and write $\mathbf{x}$ as $\mathbf{c} + \alpha^m\mathbf{z}$ where $\mathbf{c} \in \mathcal{EC}$ and $\mathbf{z} \in \mathbb{Z}[\theta]^n$. Since $d = d^*$, we have $d = |\alpha|^m$. Therefore, $\tau_d(\mathbf{c}) = \mathbf{c} + |\alpha|^m\mathbf{r}$ for some $\mathbf{r} \in \mathbb{Z}[\theta]^n$. Hence,

$$\mathbf{x} = \alpha^m\mathbf{z} + \tau_d(\mathbf{c}) - u^{-1}\alpha^m\mathbf{r},$$

implying that $\mathbf{x} = \tau_d(\mathbf{c}) + \alpha^m\mathbf{z}'$ for some $\mathbf{z}' \in \mathbb{Z}[\theta]^n$. Thus, $\Lambda \subseteq \mathcal{L} + \alpha^m\mathbb{Z}[\theta]^n$. The converse of this inclusion can be obtained similarly. ∎

We conclude with an explicit example showcasing the shaping of complex Coxeter-Todd lattice K_{12}.

Example 4.1: Construction D is used together with $\alpha = 1 + 2\omega$ and $m = 2$ in $\mathbb{Z}[\omega]$ to construct complex Coxeter-Todd lattice K_{12} in [1], [6]. Let $\mathbf{c}_1 = (1,1,1,1,1,1)$, $\mathbf{c}_2 = (0,1,1,0,0,0)$, $\mathbf{c}_3 = (0,0,1,1,0,0)$, $\mathbf{c}_4 = (0,0,0,1,1,0)$, $\mathbf{c}_5 = (0,0,0,0,1,1)$, $\mathbf{c}_6 = (0,0,0,0,0,1) \in \mathbb{F}_3^6$ so that $\mathbf{c}_1$ spans the trivial $[6,1,6]$ ternary codes and $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5$ span a $[6,5,2]$ ternary code. The lattice K_{12} can be obtained as an Eisenstein-integral combination of $\mathbf{c}_1, \alpha\mathbf{c}_2, \alpha\mathbf{c}_3, \alpha\mathbf{c}_4, \alpha\mathbf{c}_5$ and elements from $\alpha^2\mathbb{Z}[\omega]^6$. Here, we can write

$$K_{12} = \mathcal{EC} + \alpha^2\mathbb{Z}[\omega]^6.$$

where $\mathcal{EC} \subseteq \mathcal{S}^6_{1+2\omega,2}$. Theorem 4.1 now applies to give $\mathcal{EC}$ a cubic shaping.

Acknowledgement

We would like to thank Frédérique Oggier for her support during this collaboration. P. Pooksombat is supported by the Department of Mathematics and the Faculty of Graduate Studies, Mahidol University. J. Harshan is supported by the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). W. Kositwattanarerk is supported by the Thailand Research Fund under Research Grant TRG5880116.

References

[1] E. S. Barnes and N. J. A. Sloane, "New Lattice Packings of Spheres," *Can. J. Math.*, vol. 35, no. 1, 1983, pp. 117–130.
[2] J. H. Conway, and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups*, Third Edition, 1998, Springer-Verlag, New York.
[3] J. Harshan, E. Viterbo, and J. C. Belfiore, "Practical Encoders and Decoders for Euclidean Codes from Barnes-Wall Lattices," in *IEEE Transactions on Communications*, vol. 61, no. 11, Nov. 2013, pp. 4417–4427.
[4] G. D. Forney, "Coset Codes–Part I: Introduction and Geometrical Classification," in *IEEE Trans. Inform. Theory*, vol. 34, no. 5, Sept. 1988, pp. 1123–1151.
[5] W. Kositwattanarerk and F. Oggier, "Connections between Construction D and related constructions of lattices," *Designs, Codes and Cryptography, Special Issue on Coding and Cryptography*, vol. 73, no. 2, Feb. 2014, pp. 441–455.
[6] N. J. A. Sloane, *Self-dual codes and lattices, in Relations between combinatorics and other parts of mathematics*, Proc. Sympos. Pure Math. 34 (Amer. Math. Soc, Providence, RI, 1979), pp. 273–308.
[7] M.-R. Sadeghi, A. H. Banihashemi, and D. Panario, "Low-Density Parity-Check Lattices: Construction and Decoding Analysis," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, Oct. 2006, pp. 4481–4495.
[8] Y. Yan, C. Ling, and X. Wu, "Polar Lattices: Where Arikan Meets Forney," in the *Proc. IEEE Int. Symp. Inform. Theory 2013*, Istanbul, Turkey, pp. 1292–1296, July 7-12, 2013.

Zeckendorf's Theorem and Fibonacci Coding for Modules

Perathorn Pooksombat, Patanee Udomkavanich, and Wittawat Kositwattanarerk, *

June 7, 2017

Abstract

Zeckendorf's theorem states that every positive integer can be written uniquely as a sum of nonconsecutive Fibonacci numbers. This theorem induces a binary numeration system for the positive integers known as Fibonacci coding. Fibonacci code is a variable-length prefix code that is robust against insertion and deletion errors and is useful in data transmission and data compression. In this paper, we generalize the theorem of Zeckendorf and prove that every element of a free $\mathbb{Z}$-module can be represented as a sum of elements from a Fibonacci sequence of higher order. Immediate applications of these results include a Fibonacci coding for free $\mathbb{Z}$-modules, where encoding and decoding algorithms are obtained naturally from the approach of our theorems.

Keywords: Zeckendorf's theorem, Fibonacci codes, $\mathbb{Z}$-module, Gaussian integers

1 Introduction

There are certainly many generalizations of the Fibonacci sequence. The classical one is defined recursively by

$$F_0 = 0, \quad F_1 = 1, \quad \text{and} \quad F_{n-2} + F_{n-1} = F_n \text{ for all } n \geq 2.$$

One may allow the index n to be negative, resulting in what is called negaFibonacci sequence. A generalized Fibonacci sequence of order k is a sequence in which each sequence element is the sum of the preceding k terms. In this view, the usual Fibonacci sequence is of order 2. Typically, the initial terms of the generalized Fibonacci sequence of order k are $F_{-k+2} = \ldots = F_0 = 0$ and $F_1 = 1$.

*P. Pooksombat and W. Kositwattanarerk are with the Department of Mathematics, Faculty of Science, Mahidol University, Bangkok 10400, Thailand and the Centre of Excellence in Mathematics, the Commission on Higher Education, Bangkok 10400, Thailand (e-mail: perathorn.pok@student.mahidol.edu and wittawat.kos@mahidol.edu). P. Udomkavanich is with the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand (email: pattanee.u@chula.ac.th). The research of W. Kositwattanarerk for this work is supported by the Thailand Research Fund under Research Grant TRG5880116.

The much-celebrated Zeckendorf's theorem states that every positive integer can be uniquely written as

$$F_{i_1} + F_{i_2} + \ldots + F_{i_j}$$

where $i_1, i_2 - i_1, \ldots, i_j - i_{j-1} \geq 2$. This result can be extended to all integers by using instead negaFibonacci sequence [5, 22]. In other words, every integer can be written uniquely as sum of nonconsecutive negaFibonacci numbers. Zeckendorf's theorem has been generalized and investigated in many directions; similar results can be stated for the generalized Fibonacci sequence of order k [6, 30] and linear recurrence sequences [9, 15]. The distribution of the number of terms in a Zeckendorf decomposition is studied in [4, 23, 25] where a probabilistic approach has recently been applied [2].

Zeckendorf's theorem also finds several applications in data transmission and compression. Since the theorem provides a binary numeration system for the integers, it can be used to design an error-resistant universal code called Fibonacci code. The scheme was introduced by [12] and generalized to higher order in [1]. Theoretically, Fibonacci code makes use of the fact that the Zeckendorf representation of an integer involves no two consecutive terms from the Fibonacci sequence. This coding scheme is optimal under some distributions and can be used as an alternative to Huffman codes [13, 27]. In addition, it comes with fast encoding and decoding algorithms [31, 32]. Other applications of Fibonacci sequence in data science include the study of Morse code as a monoid [7, 28] and Wavelet trees [21].

In this paper, we consider generalizations of Zeckendorf's theorem and Fibonacci code for free-$\mathbb{Z}$ modules. This includes, for example, the integers, Gaussian and Eisenstein integers, lattices, and the ring $\mathbb{Z}[\alpha]$ where α is algebraic. In fact, we will be studying generalizations of Zeckendorf's theorem in its purest form: we consider *any* sequence that satisfies the generalized Fibonacci recurrence relation and examine elements that can be written as a sum of terms from the sequence. Since every term in the sequence is practically a linear combination of the sequence's initial conditions, one can at best hope to generate a module using sums of elements from this sequence. Hence, by design we adopt free $\mathbb{Z}$-modules as a mathematical space of interest.

Noninteger Fibonacci sequences have been studied mostly on complex numbers [3, 16, 17, 14, 19, 26]. In particular, Jordan [19] and Harman [17] consider Fibonacci sequences whose initial terms and indices are Gaussian integers. Other generalizations usually involve an identity whose integer terms yield the classical Fibonacci numbers. For example, Binet's formula is exploited in [16, 26] where a Fibonacci function is defined over the real and complex numbers.

From number-theoretic point of view, Zeckendorf decomposition can be seen as a numeration system where binary digits are used with Fibonacci numbers as a base. As it turns out, binary representation for multidimensional number systems is not well studied. Kátai and Szabó [20] show that Gaussian integers can be written in base $-1 + i$, thus giving a binary representation for Gaussian integers where the real part and complex part are not dealt with separately. In [11], Forney uses $1 + i$ as a basis to construct complex lattice codes. While some ring of integers may have special characteristics that

permit similar binary representation, these systems at best provide fixed-length codes for communication over a channel.

This paper studies generalizations of Zeckendorf's theorem for modules and the subsequent Fibonacci coding. We summarize the contributions of this paper as follows.

- We introduce the notion of k-equivalent sequences. This approach technically treats elements that can be generated from a Fibonacci sequence as a number system using sequence elements as a basis. This key mechanism allows us to manipulate the "digits" without an explicit knowledge of the underlying Fibonacci sequence. We prove that an element has a Zeckendorf decomposition if and only if it can be written as a finite sum (with multiplicities) of sequence elements.

- We provide a sufficient and necessary condition for Zeckendorf's theorem to hold in a free $\mathbb{Z}$-module. Obviously, the classical Zeckendorf's theorem can be viewed as a case of our generalization. We also give a sufficient condition that makes the representation unique.

- We propose Fibonacci coding for any free $\mathbb{Z}$-module. This is the first work that makes possible Fibonacci coding for non-integers. Not only that our work substantially enlarges the message spaces for Fibonacci code, the resulting codes inherit robustness property from the standard Fibonacci code. Naturally, this also gives binary numeration systems for spaces such as Gaussian integers, Eisenstein integers, quarternions, the ring of integers of any algebraic number field, and lattices. In addition, explicit encoding and decoding algorithm are also given.

The remainder of this paper is organized as follows. Section 2 sets the definitions that will be used throughout the paper. In Section 3, we prove several useful results concerning k-equivalent sequences. Generalizations of Zeckendorf's theorem are given in Section 4. We establish Fibonacci coding in Section 5 and conclude with final discussion in Section 6.

2 Definitions

We first give a definition for free modules and Fibonacci sequences of higher order. A free $\mathbb{Z}$-module is a module over $\mathbb{Z}$ with a basis. Namely, M is a free $\mathbb{Z}$-module of rank l if it is a group under addition and

$$M = \alpha_1\mathbb{Z} \oplus \alpha_2\mathbb{Z} \oplus \ldots \oplus \alpha_l\mathbb{Z}$$

for some $\alpha_1, \alpha_2, \ldots, \alpha_l$ that are integrally independent, meaning that the only integer solution to the equation $n_1\alpha_1 + n_2\alpha_2 + \ldots + n_l\alpha_l = 0$ is $n_1 = n_2 = \ldots = n_l = 0$. Algebraically speaking, modules differ from rings in that multiplication may not be defined for module elements. Here, for $n \in \mathbb{Z}^+$ and $m \in M$, we write nm to mean $\underbrace{m + m + \ldots + m}_{n \text{ times}}$.

If $M = \alpha_1\mathbb{Z} \oplus \alpha_2\mathbb{Z} \oplus \ldots \oplus \alpha_l\mathbb{Z}$ for integrally independent $\alpha_1, \alpha_2, \ldots, \alpha_l$, every element of M can be written uniquely as $n_1\alpha_1 + n_2\alpha_2 + \ldots + n_l\alpha_l$ where $n_1, n_2, \ldots, n_l \in \mathbb{Z}$. The

elements $\alpha_1, \alpha_2, \ldots, \alpha_l$ are called a basis for M.

If $\alpha_1, \alpha_2, \ldots, \alpha_l$ are integrally independent, $\mathbb{Z}\langle\alpha_1, \alpha_2, \ldots, \alpha_l\rangle$ is the free $\mathbb{Z}$-module generated by $\alpha_1, \alpha_2, \ldots, \alpha_l$. For example, $\mathbb{Z}\langle 1\rangle$, $\mathbb{Z}\langle 1, i\rangle$, $\mathbb{Z}\langle 1, \omega\rangle$ are the set of integers, Gaussian integers, and Eisenstein integers respectively.

We are interested in a Fibonacci sequence whose entries are elements of a module, so we adopt a somewhat broad definition of a Fibonacci sequence and a corresponding generalization of a Zeckendorf representation.

Definition 2.1 *A Fibonacci sequence of order k is a doubly infinite sequence $(F_r)_{r\in\mathbb{Z}} = (\ldots, F_{-1}, F_0, F_1, \ldots)$ where*

$$F_{n-k} + \ldots + F_{n-2} + F_{n-1} = F_n$$

for all integer n.

Definition 2.2 *Let $(F_r)_{r\in\mathbb{Z}}$ be a Fibonacci sequence of order k. An element m is Zeckendorf in (F_r) if it can be written as a finite sum of elements in (F_r) with no k consecutive terms. A set M is Zeckendorf in (F_r) if every element of M is.*

Notice that we do not specify the initial conditions for a Fibonacci sequence. In fact, most of our results hold irrespective of the initial conditions of the sequence. A Fibonacci sequence of order k has degree of freedom k; knowing any k consecutive terms is sufficient to generate the entire sequence. For $k \geq 2$, the characteristic polynomial of a Fibonacci sequence of order k is $x^k - x^{k-1} - x^{k-2} - \ldots - x - 1$. It is known [24] that this polynomial has k distinct roots $\lambda_1, \lambda_2, \ldots, \lambda_k$ where $\lambda_1 \in (1, 2)$ and $\lambda_2, \ldots, \lambda_k$ lie in the unit circle. Note that the sequence $(\lambda_1^r)_{r\in\mathbb{Z}}$ is a Fibonacci sequence of order k. We call this sequence *primitive*.

Example 2.3 *In this example we consider several Fibonacci sequences of order 3.*

i.) The usual Tribonacci sequence is defined using $F_0 = F_1 = 1$, $F_2 = 2$, and $F_{n-3} + F_{n-2} + F_{n-1} = F_n$. The first few terms of this sequence are

$$1,\ \ 1,\ \ 2,\ \ 4,\ \ 7,\ \ 13,\ \ 24,\ \ 44,\ \ 81,\ \ 149,\ \ 274,\ \ \ldots.$$

The two-way generalization of this sequence is given by

$$\begin{array}{rrrrrrrrrrrr} \ldots, & 5, & -8, & 4, & 1, & -3, & 2 & 0, & -1, & 1, & 0, & 0, \\ & 1, & 1, & 2, & 4, & 7, & 13, & 24, & 44, & 81, & 149, & 274, & \ldots. \end{array}$$

ii.) The three roots of a polynomial $x^3 - x^2 - x - 1$ are $\lambda_1 \approx 1.839$, $\lambda_2 \approx -0.42 - 0.606i$, and $\lambda_3 \approx -0.42 + 0.606i$. The primitive Fibonacci sequence of order 3 is the sequence $(\lambda_1^r)_{r\in\mathbb{Z}}$, which is approximately

$$\ldots,\ \ 0.161,\ \ 0.296,\ \ 0.544,\ \ 1,\ \ 1.839,\ \ 3.383,\ \ 6.222,\ \ 11.445,\ \ 21.05,\ \ \ldots.$$

iii.) A Fibonacci sequence of order 3 where $F_0 = 0$, $F_1 = 1 + i$, and $F_2 = 2 + i$ is given by

$$\begin{array}{cccccccccc} \ldots, & 5+i, & -2-3i, & -1+2i, & & 2, & -1-i, & & i, & & 1, \\ & & 0, & & 1+i, & & 2+i, & 3+2i, & 6+4i, & 11+7i, & 20+13i, & \ldots. \end{array}$$

The element $9 + 5i$, for example, is Zeckendorf in this sequence since $9 + 5i = (5 + i) + (-1 + 2i) + (2) + (3 + 2i)$.

In order to study elements that are Zeckendorf in (F_r), it is natural to consider elements of the form $(x_r) \cdot (F_r)$ where (x_r) is a doubly infinite sequence of integers and $\cdot$ is an infinite dot product, i.e.

$$(x_r) \cdot (F_r) = \sum_{r \in \mathbb{Z}} x_r F_r.$$

Roughly speaking, a collection of all elements of the form $(x_r) \cdot (F_r)$ is a number system base (F_r) with the digit set $\mathbb{Z}$. Here, (x_r) acts like digits or coefficients for the basis (F_r). These so-called coefficients can be manipulated using the following methods.

Definition 2.4 *Doubly infinite sequences (x_r) and (y_r) are k-equivalent, denoted $(x_r) \sim_k (y_r)$, if (y_r) can be obtained from (x_r) using finite applications of the following two operations.*
Operation 1: For some $n \in \mathbb{Z}$, subtract 1 from x_n and add 1 to all of $x_{n-k} \ldots, x_{n-2}, x_{n-1}$.
Operation 2: For some $n \in \mathbb{Z}$, add 1 to x_n and subtract 1 from all of $x_{n-k} \ldots, x_{n-2}, x_{n-1}$.

It is not hard to see that *Operations 1* and *2* "cancel out", and $\sim_k$ is an equivalent relation. These operations also preserve the value of the dot product: if (F_r) is a Fibonacci sequence of order k and $(x_r) \sim_k (y_r)$, then $(x_r) \cdot (F_r) = (y_r) \cdot (F_r)$. As we will see later, this observation is an important ingredient of our results. Finally, we give a lexicographical order $\succ$ to the doubly infinite sequences that are zero almost everywhere, i.e., $(x_r)_{r \in \mathbb{Z}} \succ (y_r)_{r \in \mathbb{Z}}$ if the entry of (x_r) is larger than that of (y_r) at the rightmost index where they are not equal.

3 k-equivalent Sequences

In a way, our approach in proving a generalized theorem of Zeckendorf is in asking the converse question: given a Fibonacci sequence (F_r) of order k, which elements can be written as a finite sum of elements from (F_r) using no k consecutive terms? It is not hard to see that they are elements of the form $(y_r) \cdot (F_r)$ where (y_r) is binary, 0 almost everywhere, and has no k consecutive 1's. Now, instead of characterizing these elements directly, we will focus on manipulating the coefficients (y_r) using *Operations 1* and *2* from Definition 2.4. These coefficient manipulations operate independently of the underlying Fibonacci sequence (F_r), and so this approach frees us from having to worry about the choice of (F_r). As we will see, our results hold in a broad sense and are not limited to just any specific Fibonacci sequence. Propositions 3.1 and 3.2 given in this section will provide an effective mechanism for identifying sequences that are k-equivalent to a binary sequence with the required properties.

Proposition 3.1 *A sequence $(x_r)_{r\in\mathbb{Z}}$ of integers that is 0 almost everywhere is k-equivalent to a sequence $(y_r)_{r\in\mathbb{Z}}$ that is either*

- *zero everywhere,*
- *positive at some k consecutive terms and zero elsewhere, or*
- *negative at some k consecutive terms and zero elsewhere.*

In addition, if $(x_r)_{r\in\mathbb{Z}} \neq (0)_{r\in\mathbb{Z}}$ is nonnegative, then the k consecutive terms of $(y_r)_{r\in\mathbb{Z}}$ are positive.

Proof We keep applying either *Operation 1* or *2* from Definition 2.4 to eliminate the rightmost nonzero term of (x_r). Since (x_r) is 0 almost everywhere, we will eventually obtain a sequence $(y_r)_{r\in\mathbb{Z}}$ which is 0 everywhere except for some k consecutive terms. If the k consecutive terms of (y_r) are all positive, all negative, or all zero, then we are done. Otherwise, we can keep eliminating the rightmost nonzero term of (y_r) and obtain yet another sequence that is 0 everywhere except for some k consecutive terms. We will prove that this process will eventually yield a sequence with k consecutive terms that are either all positive or all negative.

We initialize $(y_{0,r}) = (y_r)$ and collect the k nonzero terms of $(y_{0,r})$ into a $1 \times k$ vector $\boldsymbol{\gamma}_0 = (\gamma_1, \gamma_2, \ldots, \gamma_k)$. The rightmost term of $(y_{0,r})$ is γ_k. Using either *Operation 1* or *2*, we eliminate this term an obtain $(y_{1,r})$ whose nonzero entries are given by $\boldsymbol{\gamma}_1 = (\gamma_k, \gamma_1 + \gamma_k, \gamma_2 + \gamma_k, \ldots, \gamma_{k-1} + \gamma_k)$. In other words, we have

$$\boldsymbol{\gamma}_1^\top = A\boldsymbol{\gamma}_0^\top$$

where

$$A = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix}.$$

Note that this is independent of the value of γ_k. We repeat this process, and it is not hard to see that $(y_{0,r})$ is k-equivalent to $(y_{n,r})$ for all positive integer n. In addition, the nonzero terms of $(y_{n,r})$ are given by the entries of $\boldsymbol{\gamma}_n^\top = A^n\boldsymbol{\gamma}_0^\top$. Thus, it remains to show that there is an n for which $A^n\boldsymbol{\gamma}_0^\top$ is either positive or negative.

It should not be surprising that A is the transpose of the usual Fibonacci matrix. Thus, A has characteristic equation $x^k - x^{k-1} - x^{k-2} - \ldots - x - 1$ and eigenvalues $\lambda_1 \in (1, 2)$ and $\lambda_2, \ldots, \lambda_k$ whose complex norms are less than 1 [24]. We denote by $\boldsymbol{v}_i$ eigenvector corresponding to the eigenvalue λ_i. Note that

$$\boldsymbol{v}_1 = \begin{pmatrix} \lambda_1^{k-2} \\ \lambda_1^{k-2} + \lambda_1^{k-3} \\ \lambda_1^{k-2} + \lambda_1^{k-3} + \lambda_1^{k-4} \\ \vdots \\ \lambda_1^{k-2} + \lambda_1^{k-3} + \cdots + \lambda_1 + 1 \\ \lambda_1^{k-1} \end{pmatrix}$$

is real and positive. Now, one may diagonalize A as CDC^{-1} where $D = \mathsf{diag}(\lambda_1, \ldots, \lambda_k)$ and $C = \begin{pmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_k \end{pmatrix}$. It follows that

$$\begin{aligned} \boldsymbol{\gamma}_n^\top &= A^n \boldsymbol{\gamma}_0^\top \\ &= CD^nC^{-1}\boldsymbol{\gamma}_0^\top \\ &= \begin{pmatrix} \lambda_1^n \boldsymbol{v}_1 & \cdots & \lambda_1^n \boldsymbol{v}_k \end{pmatrix} C^{-1}\boldsymbol{\gamma}_0^\top \\ &= \lambda_1^n c_1 \boldsymbol{v}_1 + \ldots + \lambda_k^n c_k \boldsymbol{v}_k \end{aligned}$$

where $(c_1, \ldots, c_k)^\top = C^{-1}\boldsymbol{\gamma}_0^\top$. Since $|\lambda_1| > 1$, $|\lambda_2|, \ldots, |\lambda_k| < 1$, and $\boldsymbol{v}_1$ is real and positive, for a sufficiently large n we will have $\boldsymbol{\gamma}_n^\top \approx \lambda_1^n c_1 \boldsymbol{v}_1$, and so the entries of $\boldsymbol{\gamma}_n^\top$ will either be all positive or all negative, depending on the sign of c_1. If $c_1 = 0$, then $\boldsymbol{\gamma}_n^\top = \mathbf{0}$, implying that $\boldsymbol{\gamma}_0^\top = \mathbf{0}$.

For the last part of the proposition, if $(x_r)_{r\in\mathbb{Z}} \neq (0)_{r\in\mathbb{Z}}$ is nonnegative, then we only need to use *Operation 1*, and so the k consecutive terms of $(y_r)_{r\in\mathbb{Z}}$ cannot be negative or zero. ∎

Proposition 3.2 *A sequence $(x_r)_{r\in\mathbb{Z}}$ of nonnegative integers that is 0 almost everywhere is k-equivalent to a sequence $(y_r)_{r\in\mathbb{Z}}$ of 0 and 1 with no k consecutive 1's.*

Proof Consider all sequences of nonnegative integers that are k-equivalent to $(x_r)_{r\in\mathbb{Z}}$. Recall that if $(x_r) \sim_k (y_r)$, then $(x_r) \cdot (\lambda_1^r) = (y_r) \cdot (\lambda_1^r)$ where (λ_1^r) is the primitive Fibonacci sequence. Let N be an integer such that $(x_r) \cdot (\lambda_1^r) < \lambda_1^N$. Now, if a sequence of nonnegative integers (y_r) is k-equivalent to (x_r), then we must have $(y_r) \cdot (\lambda_1^r) < \lambda_1^N$. Since (λ_1^r) is strictly positive, it follows that $y_r = 0$ for all $r \geq N$. This allows us to conclude that, among all the sequences of nonnegative integers that are k-equivalent to (x_r), there must be one with the highest lexicographical order. We call this sequence $(y_r)_{r\in\mathbb{Z}}$ and claim that it satisfies the conditions required.

Suppose that $y_s \geq 2$ for some $s \in \mathbb{Z}$. We perform *Operation 1* at $n = s$ and *Operation 2* at $n = s + 1$. This results in a sequence with higher lexicographical order than (y_r), contradicting our choice of (y_r). Suppose now that $y_{s-k} = \ldots = y_{s-2} = y_{s-1} = 1$ for some $s \in \mathbb{Z}$. Perform *Operation 2* at $n = s$ yields a sequence with higher lexicographical order, and once again that contradicts the choice of (y_r). We conclude that (y_r) consists only of 0 and 1 with no k consecutive 1's. ∎

We illustrate the transformations given in Propositions 3.1 and 3.2 in an example below.

Example 3.3 *We let $k = 3$ and consider a doubly infinite sequence (x_r) that is zero everywhere except $x_{-1} = -2$, $x_1 = -1$, $x_2 = -2$, and $x_4 = 1$, i.e.,*

$$(x_r) = (\ldots, 0, 0, 0, 0, -2, 0, -1, -2, 0, 1, \ldots).$$

We now transform (x_r) into a sequence that is positive at some 3 consecutive terms and

zero elsewhere.

$$\begin{aligned}(\ldots,0,0,0,0,-2,0,-1,-2,0,1,\ldots) &\sim_3 (\ldots,0,0,0,0,-2,0,0,-1,1,0,\ldots)\\ &\sim_3 (\ldots,0,0,0,0,-2,1,1,0,0,0,\ldots)\\ &\sim_3 (\ldots,0,0,0,1,-1,2,0,0,0,0,\ldots)\\ &\sim_3 (\ldots,0,0,2,3,1,0,0,0,0,0,\ldots).\end{aligned}$$

Next, we transform $(\ldots,0,0,2,3,1,0,0,0,0,0,\ldots)$ *into a binary sequence with no 3 consecutive 1's.*

$$\begin{aligned}(\ldots,0,0,2,3,1,0,0,0,0,0,\ldots) &\sim_3 (\ldots,0,0,1,2,0,1,0,0,0,0,\ldots)\\ &\sim_3 (\ldots,1,1,2,1,0,1,0,0,0,0,\ldots)\\ &\sim_3 (\ldots,1,0,1,0,1,1,0,0,0,0,\ldots).\end{aligned}$$

Consider now a Fibonacci sequence of Gaussian integers

$$(F_r) = (\ \ldots,\ \ -1+2i,\ \ 2,\ \ -1-i,\ \ i,\ \ 1,\ \ 0,\ \ 1+i,\ \ 2+i,\ \ 3+2i,\ \ 6+4i,\ \ \ldots.\ )$$

from Example 2.3. Note that all the above 3-equivalent sequences represent the same quantity when multiplied by (F_r)*. In particular, if* $(y_r) = (\ldots,0,0,2,3,1,0,0,0,0,0,\ldots)$ *and* $(z_r) = (\ldots,1,0,1,0,1,1,0,0,0,0,\ldots)$*, then*

$$(x_r)\cdot(F_r) = (y_r)\cdot(F_r) = (z_r)\cdot(F_r) = -1+i.$$

This, in fact, would hold for any other Fibonacci sequences (F_r) *of order 3.*

The previous example illustrates a major strength of our approach–each equivalency class represents the same quantity, given a Fibonacci sequence. If one can find a legitimate Zeckendorf representation in an equivalency class, then, under any Fibonacci sequence, the element represented by that class has a Zeckendorf decomposition. We finish this section with a powerful corollary to Proposition 3.2 and another example. The following result characterizes *all* elements that are Zeckendorf in a Fibonacci sequence (F_r).

Corollary 3.4 *Let* (F_r) *be a Fibonacci sequence of order* k*. An element* m *is Zeckendorf in* (F_r) *if and only if* m *can be written as a finite sum (with multiplicities) of elements from* (F_r)*.*

Proof An element m is Zeckendorf in (F_r) if it is a finite sum of elements from (F_r). The converse of this statement follows from Proposition 3.2. ∎

Example 3.5 *Consider the primitive Fibonacci sequence of order 2*

$$\ldots,\ \ \varphi^{-3},\ \ \varphi^{-2},\ \ \varphi^{-1},\ \ 1,\ \ \varphi,\ \ \varphi^2,\ \ \varphi^3,\ \ \ldots.$$

where $\varphi = \frac{1+\sqrt{5}}{2}$ *is the golden ratio. The elements that are Zeckendorf in this sequence are precisely positive elements in the ring* $\mathbb{Z}[\varphi]$*, and this numeration system is known as golden ratio base.*

4 Zeckendorf's Theorem for Free $\mathbb{Z}$-Modules

Intuitively, if an element m is Zeckendorf in a Fibonacci sequence (F_r) of order k, then it is an integer combination of the initial terms (or, rather, any k consecutive terms) of (F_r). Integer span of these forms a module, so it is natural to attempt to represent every element of a module using sequence elements. Subsection 4.1 deals with doubly infinite Fibonacci sequences using tools developed in the last section. One-way Fibonacci sequences are studied in Subsection 4.2 as a mean to achieve unique Zeckendorf decomposition.

4.1 Two-way Sequences

Corollary 3.4 lays a strong foundation for our work. While it characterizes elements that are Zeckendorf in a sequence, it gives insufficient information on the algebraic structure of the set of all elements that have a legitimate representation. The following theorem now reverses the process. It gives a sufficient and necessary condition for every element of a free $\mathbb{Z}$-module M to be Zeckendorf in a Fibonacci sequence (F_r).

Theorem 4.1 *Let M be a free $\mathbb{Z}$-module of rank l, and $(F_r)_{r\in\mathbb{Z}} = (\ldots, F_{-1}, F_0, F_1, \ldots)$ be a Fibonacci sequence of order k. Then, M is Zeckendorf in (F_r) if and only if all of the following conditions are satisfied.*

1. *$F_i, \ldots, F_{i+k-1}$ span M for all i.*
2. *$l + 1 \leq k$.*

Proof We first remark that the integral span of $F_i, \ldots, F_{i+k-1}$ is the same as the integral span of $F_{i+1}, \ldots, F_{i+k}$ since $F_i + \cdots + F_{i+k-1} = F_{i+k}$. Thus, $F_i, \ldots, F_{i+k-1}$ spans M for *all* i if and only if $F_i, \ldots, F_{i+k-1}$ spans M for *any* i.

It is easy to see that the first condition is necessary. Since the elements of $(F_r)_{r\in\mathbb{Z}}$ are integer combinations of $F_i, \ldots, F_{i+k-1}$, so must be any sums of the elements from this sequence. It follows that $F_i, \ldots, F_{i+k-1}$ span M. This readily implies $l \leq k$. We will now show that l cannot equal to k, thus establishing the necessity of the second condition.

Suppose that $l = k$, and let m be any nonzero element of M. Since both m and $-m$ are Zeckendorf in (F_r), we have

$$m = (x_r) \cdot (F_r) \quad \text{and} \quad -m = (y_r) \cdot (F_r)$$

for some sequences $(x_r), (y_r)$ of 0 and 1 with no k consecutive 1's. Hence, $0 = (x_r + y_r) \cdot (F_r)$. By Proposition 3.1, $(x_r + y_r)$ is k-equivalent to a sequence that is positive at some k consecutive terms and zero elsewhere. Thus, we have

$$0 = (x_r + y_r) \cdot (F_r) = (z_r) \cdot (F_r) = z_i F_i + z_{i+1} F_{i+1} + \ldots + z_{i+k-1} F_{i+k-1}$$

for some i in which $z_i, z_{i+1}, \ldots, z_{i+k-1} > 0$. This, however, means that $F_i, F_{i+1}, \ldots, F_{i+k-1}$ are integrally dependent, and so they cannot span a module of rank $l = k$.

We are left to show that the two conditions are sufficient. Since $F_0, \ldots, F_{k-1}$ span M and $l+1 \leq k$, $F_1, \ldots, F_{k-1}$ must be integrally dependent. That is, we may write

$$0 = x_0 F_0 + \cdots + x_{k-1} F_{k-1}$$

where $x_0, \ldots, x_{k-1}$ are not all zeros. In other words, we have

$$0 = (x_r) \cdot (F_1)$$

where (x_r) is zero everywhere but some k consecutive terms. We apply Proposition 3.1 and obtain a sequence (y_r) that is k-equivalent to (x_r) and contain k consecutive terms of the same sign and zero elsewhere. Note that (y_r) cannot be zero everywhere since that would imply that $x_0 = \ldots = x_{k-1} = 0$.

Suppose now that (y_r) is nonzero at the terms $y_i, \ldots, y_{i+k-1}$. This means

$$0 = y_i F_i + \cdots + y_{i+k-1} F_{i+k-1}$$

where either $y_i, \ldots, y_{i+k-1} > 0$ or $y_i, \ldots, y_{i+k-1} < 0$. Since $F_i, \ldots, F_{i+k-1}$ span M, we can write any element $m \in M$ as

$$m = z_i F_i + \cdots + z_{i+k-1} F_{i+k-1}$$

where $z_i, \ldots, z_{i+k-1}$ are integers. Now, we have

$$m = m + 0 \cdot N = (z_i + y_i N) F_i + \cdots + (z_{i+k-1} + y_{i+k-1} N) F_{i+k-1}$$

for all integer N. For a sufficiently large (and possibly negative) N, the terms $z_i + y_i N, \ldots, z_{i+k-1} + y_{i+k-1} N$ will all be positive, and it follows from Corollary 3.4 that m is Zeckendorf in (F_r). This completes the proof of the theorem. ∎

Corollary 4.2 *Let $(F_r)_{r \in \mathbb{Z}} = (\ldots, F_{-1}, F_0, F_1, \ldots)$ be a Fibonacci sequence of order k. If $F_1, \ldots, F_k$ are integrally dependent, then every element in the integral span of $F_1, \ldots, F_k$ is Zeckendorf in (F_r).*

In view of Theorem 4.1, every integer has a (classical) Zeckendorf representation since $\mathbb{Z}$ is a module over itself with rank 1, and the negaFibonacci sequence is of order 2 with $F_0 = 0$ and $F_1 = 1$. In fact, the same result would hold as long as F_0 and F_1 are relatively prime (and hence span $\mathbb{Z}$ integrally). This is technically the case for the Lucas sequence where $L_0 = 2$ and $L_1 = 1$. Next, we give an example for the case of Gaussian integers.

Example 4.3 *Consider a Fibonacci sequence*

$$\begin{array}{rrrrrrrrr} \ldots, & -4+4i, & 5+i, & -2-3i, & -1+2i, & 2, & -1-i, & i, & 1, \\ & 0, & 1+i, & 2+i, & 3+2i, & 6+4i, & 11+7i, & 20+13i, & 37+24i, \quad \ldots. \end{array}$$

from Example 2.3. Every Gaussian integer can be written as a sum of elements from this sequence with no 3 consecutive terms. For instance,

$$\begin{aligned} -2 &= (-2-3i)+(-1+2i)+(i)+(1) & -2i &= (-2-3i)+(2)+(i) \\ -1 &= (-1-i)+(i) & -i &= (-1-i)+(1) \\ 2 &= (2) & 2i &= (-1+2i)+(1) \\ 3 &= (2)+(1) & 3i &= (-1+2i)+(i)+(1) \\ 1+2i &= (-1+2i)+(2) & 1-2i &= (-2-3i)+(i)+(1) \\ -2+i &= (-1+2i)+(-1-i) & -2-i &= (-2-3i)+(-1+2i)+(1). \end{aligned}$$

One may observe that $(i)+(1)=(1+i)$*, and so Zeckendorf representation for a Gaussian integer in this sequence is not unique. Nonetheless, we will see in the next subsection how certain adjustments can make unique representation possible.*

On the contrary, no Fibonacci sequence of order 2 can generate all Gaussian integers. For example, $-1-i$ *cannot be written as a sum of elements from the sequence*

$$\begin{array}{cccccccc} \ldots, & -21+13i, & 13-8i, & -8+5i, & 5-3i, & -3+2i, & 2-i, & -1+i, \\ & 1, & i, & 1+i, & 1+2i, & 2+3i, & 3+5i, & 5+8i, \quad \ldots. \end{array}$$

Note that Corollary 4.2 does not apply here since 1 *and* i *are not integrally dependent. However, it follows from Corollary 3.4 that every element of the form* $a+bi$*,* $a,b\in\mathbb{Z}^+$*, is Zeckendorf in this sequence.*

4.2 One-way Sequences and Unique Representation

So far we have left out one very important aspect of Zeckendorf's Theorem: the uniqueness property. It is known that every positive integer can be written uniquely as a sum of nonconsecutive elements from the Fibonacci sequence

$$1,2,3,5,8,13,21,34,\ldots,$$

and every integer can be written uniquely as a sum of nonconsecutive elements from the negaFibonacci sequence

$$\ldots,34,-21,13,-8,5,-3,2,-1,1.$$

However, the uniqueness property obviously no longer holds for the two-way sequence

$$\ldots,34,-21,13,-8,5,-3,2,-1,1,0,1,1,2,3,5,8,13,21,34,\ldots$$

since this sequence contains elements that are equal. In this subsection, we establish analogous results for Fibonacci sequences of higher order. The key ingredient of this development is Corollary 4.5, which states that binary sequences with no k consecutive 1's cannot be k-equivalent.

Lemma 4.4 *Let* a *be an integer, and let* $(\lambda_1^r)_{r\in\mathbb{Z}}$ *be the primitive Fibonacci sequence of order* k*. Then,*

$$\sum_{\substack{r<a \\ k\nmid a-r}} \lambda_1^r = \lambda_1^a.$$

Proof We have

$$\begin{aligned}\sum_{\substack{r<a\\k\nmid a-r}} \lambda_1^r &= \sum_{r<a} \lambda_1^r - \sum_{\substack{r<a\\k\mid a-r}} \lambda_1^r \\ &= \frac{\lambda_1^a}{\lambda_1 - 1} - \frac{\lambda_1^a}{\lambda_1^k - 1} \\ &= \lambda_1^a \left(\frac{(\lambda_1^{k-1} + \lambda_1^{k-2} + \ldots + 1) - 1}{\lambda_1^k - 1} \right) \\ &= \lambda_1^a \left(\frac{\lambda_1^k - 1}{\lambda_1^k - 1} \right) \\ &= \lambda_1^a.\end{aligned}$$

∎

Corollary 4.5 *Let (x_r) and (y_r) be doubly infinite binary sequences with no k consecutive 1's that are 0 almost everywhere.*

1. *If $(x_r) \succ_k (y_r)$, then $(x_r) \cdot (\lambda_1^r) > (y_r) \cdot (\lambda_1^r)$.*
2. *If $(x_r) \sim_k (y_r)$, then $(x_r) = (y_r)$.*

Proof Suppose that $(x_r) \succ_k (y_r)$, and let a be the largest index where (x_r) and (y_r) are not equal. It follows that $(x_a) = 1$, $(y_a) = 0$, and

$$\begin{aligned}(x_r) \cdot (\lambda_1^r) &\geq \lambda_1^a + \sum_{r>a} x_r \lambda_1^r \\ &= \sum_{\substack{r<a\\k\nmid a-r}} \lambda_1^r + \sum_{r>a} x_r \lambda_1^r \\ &> \sum_{r\leq a} y_r \lambda_1^r + \sum_{r>a} y_r \lambda_1^r = (y_r) \cdot (\lambda_1^r).\end{aligned}$$

Now, if $(x_r) \neq (y_r)$, then we may assume without loss of generality that $(x_r) \succ_k (y_r)$. This means $(x_r) \cdot (\lambda_1^r) > (y_r) \cdot (\lambda_1^r)$, and so $(x_r) \nsim_k (y_r)$. This readily establishes part *ii*. ∎

Typically, $(x_r) \sim_k (y_r)$ means that $(x_r) \cdot (F_r) = (y_r) \cdot (F_r)$, and so (x_r) and (y_r) are two representations of the same element. Corollary 4.5 now allows us to identify Fibonacci sequences (or parts of) that permit unique Zeckendorf representation: if a sequence (F_r) has the property that $(x_r) \cdot (F_r) = (y_r) \cdot (F_r)$ implies $(x_r) \sim_k (y_r)$, then any element that is Zeckendorf in (F_r) will have a unique representation. Since we are interested in generating every element in a $\mathbb{Z}$-module, we look into generalizing negaFibonacci sequence, which is technically a one-sided Fibonacci sequence to the left whose first excluded term is 0. It turns out that this observation generalizes well to modules.

Theorem 4.6 *Let $k \geq 2$ be an integer, and M be a free $\mathbb{Z}$-module of rank $k-1$. Let $(F_r)_{r\in\mathbb{Z}} = (\ldots, F_{-1}, F_0, F_1, \ldots)$ be a Fibonacci sequence of order k where $F_0 = 0$ and*

$F_{-k+1}, \ldots, F_{-1}$ span M. Then, every element $m \in M$ can be uniquely written as a finite sum of elements from the one-way sequence

$$\ldots, F_{-3}, F_{-2}, F_{-1}$$

with no k consecutive terms.

Proof We first prove the existence. Let $m \in M$, and write

$$m = x_{-k+1}F_{-k+1} + \ldots + x_{-2}F_{-2} + x_{-1}F_{-1}$$

where $x_{-k+1}, \ldots, x_{-2}, x_{-1}$ are integers. We extend $x_{-k+1}, \ldots, x_{-2}, x_{-1}$ to a sequence $(x_r)_{r\in\mathbb{Z}}$ that is zero everywhere except $r = -k+1, \ldots, -2, -1$. Now, consider all sequences of integers that are k-equivalent to $(x_r)_{r\in\mathbb{Z}}$ such that the terms with positive index are zero and the terms with negative index are nonnegative. Such a sequence exists since one may apply *Operation 1* from Definition 2.4 to (x_r) at $n = 0$ for $\max\{|x_{-k+1}|, \ldots, |x_{-2}|, |x_{-1}|\}$ times. Denote by $(y_r)_{r\in\mathbb{Z}}$ the sequence that has the aforementioned properties with highest lexicographical order. We claim that (y_r) gives a Zeckendorf representation for m in $\ldots, F_{-3}, F_{-2}, F_{-1}$.

We first note that $y_r = 0$ for $r \in \mathbb{Z}^+$ and $F_0 = 0$, and so $(y_r)\cdot(F_r) = \sum_{r\in\mathbb{Z}^-} y_r F_r$. If $y_s \geq 2$ for some $s \in \mathbb{Z}^-$, then we perform *Operation 1* at $n = s$ and *Operation 2* at $n = s+1$ to obtain a sequence with higher lexicographical order than (y_r). If $y_{s-k+1} = \ldots = y_{s-1} = y_s = 1$ for some $s \in \mathbb{Z}^-$, then performing *Operation 2* at $n = s$ results in a sequence with higher lexicographical order. Thus, (y_r) is binary with no k consecutive 1's, and $\sum_{r\in\mathbb{Z}^-} y_r F_r$ is a Zeckendorf representation for m in $\ldots, F_{-3}, F_{-2}, F_{-1}$.

Suppose now for the sake of contradiction that there exists an element $m \in M$ that has two representations as a finite sum of elements from the sequence $\ldots, F_{-3}, F_{-2}, F_{-1}$ with no k consecutive terms. That is, we have $m = (x_r)\cdot(F_r) = (y_r)\cdot(F_r)$ where $(x_r) \neq (y_r)$ are doubly infinite binary sequences which are zero for $r \geq 0$. Now, we keep applying either *Operation 1* or *2* to eliminate the leftmost nonzero term of (x_r) and (y_r) and obtain sequences (x'_r) and (y'_r) which are zero everywhere except for $r = -k+1, \ldots, -2, -1, 0$. This means

$$(x'_r)\cdot(F_r) = x'_{-k+1}F_{-k+1} + \ldots + x'_{-2}F_{-2} + x'_{-1}F_{-1} + x'_0F_0$$

and

$$(y'_r)\cdot(F_r) = y'_{-k+1}F_{-k+1} + \ldots + y'_{-2}F_{-2} + y'_{-1}F_{-1} + y'_0F_0.$$

are both equal to m. Since $F_{-k+1}, \ldots, F_{-1}$ span M of rank $k-1$, we must have $x'_r = y'_r$ for $r = -k+1, \ldots, -2, -1$. If $x'_0 = y'_0$, then $(x_r) \sim_k (x'_r) = (y'_r) \sim_k (y_r)$, and so (x_r) and (y_r) must be the same from Corollary 4.5. Otherwise, we assume without loss of generality that $x'_0 > y'_0$. We now have

$$\begin{aligned}(x_r)\cdot(\lambda_1^r) - (y_r)\cdot(\lambda_1^r) &= (x'_r)\cdot(\lambda_1^r) - (y'_r)\cdot(\lambda_1^r)\\ &= (x'_0 - y'_0)\lambda_1^0\\ &\geq 1.\end{aligned}$$

Since (y_r) is binary, it follows that $(x_r) \cdot (\lambda_1^r) \geq 1$. However, we see from Lemma 4.4 that

$$(x_r) \cdot (\lambda_1^r) < \sum_{\substack{r<0 \\ k \nmid -r}} \lambda_1^r = \lambda_1^0 = 1,$$

which is a contradiction. This completes the proof of the theorem. ∎

Obviously, the usual negaFibonacci sequence serves as an instance of Theorem 4.6. We give another example using Gaussian integers.

Example 4.7 *Consider a Fibonacci sequence*

$$\begin{array}{rrrrrrrrrr} \ldots, & -4+4i, & 5+i, & -2-3i, & -1+2i, & 2, & -1-i, & i, & 1, & \\ & 0, & 1+i, & 2+i, & 3+2i, & 6+4i, & 11+7i, & 20+13i, & 37+24i, & \ldots. \end{array}$$

from Example 2.3 and 4.3. One can see that every representation given in Example 4.3 only involve terms from the sequence

$$\ldots, \quad -4+4i, \quad 5+i, \quad -2-3i, \quad -1+2i, \quad 2, \quad -1-i, \quad i, \quad 1.$$

In fact, every Gaussian integer can be uniquely written as a sum of elements from this sequence with no 3 consecutive terms. This property will be exploited when we develop Fibonacci coding for Gaussian integers in Example 5.1 in the next section.

5 Fibonacci Coding for Free $\mathbb{Z}$-Modules

We first give a quick overview of applications of Zeckendorf's theorem in data storage and transmission. Suppose that arbitrary positive integers $n_1, n_2, \ldots, n_l$ are to be sent over a binary channel. One simple way to do this is to encode each integer using its base-2 representation and send the corresponding string of 0 and 1 over the channel. An obvious problem with this approach is that the receiver needs to know the length of each message *a priori*. It is possible to also send the length of each message over the channel, resulting in what is technically called logarithmic ramp (see, for example, [10]). On the other hand, if codes of fixed length are used, then it is not possible to send large integers, and at the same time it is wasteful if a lot of small integers are to be sent. Finally, all the above schemes are vulnerable to insertion and deletion errors; if a bit fails to reach the receiver or an extraneous bit is picked up, then all of the subsequent messages may not be decoded correctly. A popular alternative approach is to use variable-length prefix codes. Here, the number of bits in each codeword varies, and a receiver tell each codeword apart using some special properties from the design of the code. We refer the readers to [27] for more details and discuss next the case of Fibonacci coding.

Recall that we initialize the classical Fibonacci sequence with $F_0 = 0$ and $F_1 = 1$. To encode a positive integer n, we appeal to Zeckendorf's theorem and write n as $\sum_{r=1}^{t} a_r F_{r+1}$. Then, the Fibonacci coding of n is $a_1 a_2 \ldots a_r 1$.

Message	Representation	Codeword
1	1	11
2	2	011
3	3	0011
4	$1+3$	1011
5	5	00011
6	$1+5$	10011
7	$2+5$	01011
8	8	000011
9	$1+8$	100011
10	$2+8$	010011
11	$3+8$	001011
12	$1+3+8$	101011

Table 1: Fibonacci code for $n = 1, 2, \ldots, 12$

Since the classical Zeckendorf representation contains no two consecutive terms, each codeword of the Fibonacci code contains exactly one instance of "11" at the end. Thus, if several codewords from Fibonacci code are to be sent sequentially over a channel, one can look for "11" and mark that as the end of a codeword. For example, 01011111011 can be uniquely decoded as 01011 11 1011. Fibonacci code is robust against insertion and deletion errors in that the separator "11" will make the error local. In addition, it is not hard to see that an extra bit may be added so that it is possible to represent negative integers. Alternatively, one may instead use negaFibonacci numbers to begin with.

We now wish to develop an analogous scheme for messages that are from a module. Modules encompass composite message spaces such as complex numbers and lattices, which find use in Quadrature Amplitude Modulation (QAM) and wireless communications. See, for example, [29, 33]. We see from Theorem 4.6 that it is possible to write any element of a module as a sum of entries from a sequence using no k consecutive terms. This suggests Fibonacci coding for modules with the use of k consecutive 1's as a separator. The intuition formalizes into Algorithm 1. Here, we denote k consecutive 1's, $\underbrace{11\ldots1}_{k}$, by 1_k.

Note that the coefficients are encoded in reverse order, and each codeword contains exactly one instance of 1_k at the end. Thus, when codewords are sent sequentially over a channel, the receiver only needs to look for the occurrence of 1_k to separate each codeword. In addition, note that the algorithm simply mimics the arguments given in the proof of Theorem 4.6. This guarantees that the algorithm terminates since the operations performed in Step 4 and 5 increases the lexicographical order of the sequence. At every step, the value $\sum_{r\in\mathbb{Z}^-} x_r F_r$ remains unchanged. This fact makes the decoding, which is outlined as Algorithm 2, easy.

To illustrate, we consider once again the Gaussian Fibonacci sequence given in Examples 2.3, 4.3, and 4.7.

Example 5.1 *Consider the Fibonacci sequence*

$$\ldots, \quad -4+4i, \quad 5+i, \quad -2-3i, \quad -1+2i, \quad 2, \quad -1-i, \quad i, \quad 1,$$

Algorithm 1 Fibonacci Encoding

Input: A Fibonacci sequence $\ldots, F_{-3}, F_{-2}, F_{-1}$ of order k where $F_{-k} + \ldots + F_{-1} = 0$ and $F_{-k+1}, \ldots, F_{-1}$ span a free $\mathbb{Z}$-module M of rank $k-1$, and an element $m \in M$.

Output: Fibonacci code for m.

1: If $m = 0$, output 1_k. END.

2: Write m as

$$x_{-k+1}F_{-k+1} + \ldots + x_{-2}F_{-2} + x_{-1}F_{-1}.$$

Set $x_0 = 0$ and $x_r = 0$ for $r \leq -k$.

3: If $\check{x} = \min\{x_{-k+1}, \ldots, x_{-2}, x_{-1}\} < 0$, subtract $\check{x}$ from $x_{-k}, x_{-k+1}, \ldots, x_{-2}, x_{-1}$.

4: Find the largest index $s < 0$ such that $x_{s-k+1}, \ldots, x_s \geq 1$. If there is none, go to Step 5. Otherwise, subtract 1 from $x_{s-k+1}, \ldots, x_s$, add 1 to x_{s-1}, and repeat this step.

5: Find the largest index $s < 0$ such that $x_s \geq 2$. If there is none, go to Step 6. Otherwise, subtract 2 from x_s, add 1 to x_{s-k} and x_{s+1}, and go to Step 4.

6: Find the smallest index $s < 0$ such that $x_s = 1$. Output $x_{-1}x_{-2}x_{-3}\ldots x_{s+2}x_{s+1}01_k$. END.

Algorithm 2 Fibonacci Decoding

Input: A Fibonacci sequence $\ldots, F_{-3}, F_{-2}, F_{-1}$ of order k, and a binary string $x_1x_2x_3\ldots x_{s-1}x_s1_k$.

Output: An element m.

1: If $x_1x_2x_3\ldots x_{s-1}x_s$ is empty, output 0. END.

2: Output $x_1F_{-1} + x_2F_{-2} + \ldots + x_{s-1}F_{-s+1} + F_{-s}$. END.

and let $m = -2 + 3i$. To encode, we initialize $x_{-2} = 3$, $x_{-1} = -2$ and obtain $x_{-3} = 2$, $x_{-2} = 5$, $x_{-1} = 0$ after Step 2. We shorthand this sequence as $2, 5, 0$. Iterations of Step 3 and 4 now yield

$$2, 5, 0 \to 1, 0, 2, 3, 1 \to 1, 0, 1, 2, 0 \to 2, 0, 1, 0, 1 \to 1, 0, 0, 0, 1, 1, 0, 1.$$

Thus, $m = -2 + 3i$ is encoded as 10110000111*. This string can then be decoded as $F_{-1} + F_{-3} + F_{-4} + F_{-8} = 1 + (-1 - i) + 2 + (-4 + 4i) = -2 + 3i$. The table below gives Fibonacci code under this sequence for some Gaussian integers.*

	$a = -2$	$a = -1$	$a = 0$	$a = 1$	$a = 2$
$b = 2i$	01100111	00000111	10000111	00010111	1001011
$b = i$	00100111	10100111	00111	10111	0100111
$b = 0$	110010111	010111	111	0111	0000111
$b = -i$	100010111	000111	100111	0010111	1010111
$b = -2i$	010000111	110000111	010100111	110000111	0110010111

Table 2: Fibonacci code for $a + b$ using the sequence $\ldots, -4 + 4i, 5 + i, -2 - 3i, -1 + 2i, 2, -1 - i, i, 1$.

Clearly, one may replace i in Example 5.1 by any other quadratic integer and obtain a Fibonacci coding for the corresponding ring of quadratic integers. Next, we give an example of Fibonacci coding for a lattice.

Example 5.2 *Consider the lattice E_8 given by*

$$E_8 = \left\{ \mathbf{x} \in \mathbb{Z}^8 \cup (\mathbb{Z} + \tfrac{1}{2})^8 \mid \sum x_i \equiv 0 \pmod 2 \right\}.$$

This lattice provides optimal sphere packing and kissing number in 8 dimensions and has many other interesting properties [8]. We set the basis for E_8 as

$$\begin{aligned}
\mathbf{v}_1 &= (2,0,0,0,0,0,0,0) & \mathbf{v}_2 &= (-1,1,0,0,0,0,0,0) \\
\mathbf{v}_3 &= (0,-1,1,0,0,0,0,0) & \mathbf{v}_4 &= (0,0,-1,1,0,0,0,0) \\
\mathbf{v}_5 &= (0,0,0,-1,1,0,0,0) & \mathbf{v}_6 &= (0,0,0,0,-1,1,0,0) \\
\mathbf{v}_7 &= (0,0,0,0,0,-1,1,0) & \mathbf{v}_8 &= (\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2},\tfrac{1}{2})
\end{aligned}$$

and consider a Fibonacci sequence of order 9 given by

$$\ldots, 2\mathbf{v}_2 - \mathbf{v}_1, 2\mathbf{v}_1, -\mathbf{v}_1 - \mathbf{v}_2 - \mathbf{v}_3 - \mathbf{v}_4 - \mathbf{v}_5 - \mathbf{v}_6 - \mathbf{v}_7 - \mathbf{v}_8, \mathbf{v}_8, \mathbf{v}_7, \mathbf{v}_6, \mathbf{v}_5, \mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2, \mathbf{v}_1.$$

It follows from Theorem 4.6 that every element of E_8 can be written as a sum of elements from this sequence with no 9 consecutive terms. Let $\mathbf{m} = (\frac{1}{2}, 2\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \in E_8$. Then, $\mathbf{m} = \mathbf{v}_1 + 2\mathbf{v}_2 + 2\mathbf{v}_3 + \mathbf{v}_8$ and can be encoded as 010000000110111111111.

6 Conclusion

In this paper, we generalize Zeckendorf's theorem to modules. The notion of equivalent sequences allows us to identify elements that can be represented as a sum of elements from a Fibonacci sequence of order k with no k consecutive terms. This results in necessary and sufficient conditions for a Fibonacci sequence of higher order to generate a module. In addition, under certain circumstances the representation is unique, allowing us to establish Fibonacci coding for modules. Future work involves identifying other conditions to which the representation remains unique. Under such environments, one can view Zeckendorf representation as a number system and develop generalized Zeckendorf arithmetic. It would also be interesting to study the proposed coding algorithm from the perspective of data compression and computational complexity.

References

[1] A. Apostolico, A. Fraenkel, Robust Transmission of Unbounded Strings Using Fibonacci Representations, IEEE Transactions on Information Theory **33** (1987), 238–245.

[2] I. Ben-Ari and S. J. Miller, A Probabilistic Approach to Generalized Zeckendorf Decompositions, SIAM Journal on Discrete Mathematics **30** (2016), no. 2, 1302–1332.

[3] Berzsenyi, Gaussian Fibonacci Numbers, Fibonacci Quarterly **15** (1977), no. 3, 233–236.

[4] A. Best, P. Dynes, X. Edelsbrunner, B. McDonald, S. J. Miller, C. Turnage-Butterbaugh, and M. Weinstein, Gaussian Behavior of the Number of Summands in Zeckendorf Decompositions in Small Intervals, Fibonacci Quarterly **52** (2014), no. 5, 47–53.

[5] M. W. Bunder, Zeckendorf Representations Using Negative Fibonacci Numbers, Fibonacci Quarterly **30** (1992), 111–115.

[6] L. Carlitz, V. E. Hoggatt, Jr., and R. Scoville, Fibonacci Representations of Higher Order, Fibonacci Quarterly **10** (1972), no. 1, 43–69.

[7] J. Cigler, Some Algebraic Aspects of Morse Code Sequences, Discrete Mathematics and Theoretical Computer Science **6** (2003), 55–68.

[8] J. Conway, and N.J.A. Sloane, Sphere Packings, Lattices and Groups, Springer.

[9] P. Demontigny, T. Do, A. Kulkarni, S. J. Miller, D. Moon, and U. Varma, Generalizing Zeckendorf's Theorem to f-decompositions, Journal of Number Theory **141** (2014), 136–158.

[10] S. Even and M. Rodeh, Economical Encoding of Commas Between Strings, Communications of the ACM **21** (1978), 315–317.

[11] G. D. Forney, Coset Codes–Part I: Introduction and Geometrical Classification, IEEE Trans. Inform. Theory **34** (1988), no. 5, 1123–1151.

[12] A. Fraenkel and S. Klein, Robust Universal Complete Codes As Alternatives to Huffman Codes, Technical Report CS85-16, The Weizmann Institute of Science, Rehovot, 1985.

[13] A. Fraenkel and S. Klein, Robust Universal Complete Codes for Transmission and Compression, Discrete Applied Mathematics **64** (1996), 31–55.

[14] I. J. Good, Complex Fibonacci and Lucas Numbers, Continued Fractions, and the Square Root of the Golden Ratio, Journal of the Operational Research Society **43** (1992), no. 8, 837–842.

[15] P. J. Grabner, R. F. Tichy, I. Nemes, and A. Pethő, Generalized Zeckendorf Expansions, Applied Mathematics Letters **7** (1994), no. 2, 25–28.

[16] E. Halsey, The Fibonacci Number F_u where u is not an Integer, Fibonacci Quarterly **3** (1965), no. 2, 147–152.

[17] C. J. Harman, Complex Fibonacci Numbers, Fibonacci Quarterly **19** (1981), no. 1, 82–86.

[18] D. Huffman, A Method for the Construction of Minimum Redundancy Codes, Proceedings of the IRE (1952), 1098–1101.

[19] J. H. Jordan, Gaussian Fibonacci and Lucas Numbers, Fibonacci Quarterly **3** (1965), no. 4, 315–318.

[20] I. Kátai, J. Szabó, Canonical Number Systems for Complex Integers, Acta Sci. Math., **37** (1975), 255–260.

[21] S. T. Klein and D. Shapira, Random Access to Fibonacci Encoded Files, Discrete Applied Mathematics **212** (2016), 115–128.

[22] D. E. Knuth, Fibonacci Multiplication, Applied Mathematics Letters **1** (1988), no. 1, 57–60.

[23] C. G. Lekkerkerker, Voorstelling van natuurlyke getallen door een som van getallen van Fibonacci, Simon Stevin **29** (1951-1952), 190–195.

[24] E. P. Miles, Jr., Generalized Fibonacci Numbers and Associated Matrices, American Math. Monthly, **67** (1960), 745–57.

[25] S. J. Miller and Y. Wang, From Fibonacci Numbers to Central Limit Type Theorems, Journal of Combinatorial Theory, Series A **119** (2012), no. 7, 1398–1413.

[26] F. D. Parker, A Fibonacci function, Fibonacci Quarterly **6** (1968), no. 1, 1–2.

[27] D. Salomon, Data Compression The Complete Reference, Third Edition, Springer–Verlag, New York, 2004.

[28] S. Sato, Fibonacci Sequence and its Generalizations Hidden in Algorithms for Generating Morse Codes, Applications of Fibonacci Numbers: Volume 5 Proceedings of 'The Fifth International Conference on Fibonacci Numbers and Their Applications' (1993), 481–486.

[29] M. Schwartz, Mobile Wireless Communications, Cambridge University Press, 2005.

[30] N. Utgoff, A generalization of Zeckendorf's Theorem, preprint.

[31] J. Walder, M. Kratky, and J. Platos, Fast Fibonacci Encoding Algorithm, DATESO (2010).

[32] J. Walder, M. Kratky, R. Baca, J. Platos, and V. Snasel, Fast Decoding Algorithms for Variable-Lengths Codes, Information Sciences **183** (2012), 66–91.

[33] R. Zamir, Lattice Coding for Signals and Networks, Cambridge University Press, 2014.

Pseudocodeword-Free Criterion for Codes with Cycle-Free Tanner Graph

Wittawat Kositwattanarerk *

March 6, 2017

Abstract

Iterative decoding and linear programming decoding are guaranteed to converge to the maximum-likelihood codeword when the underlying Tanner graph is cycle-free. Therefore, cycles are usually seen as the culprit of low-density parity-check (LDPC) codes. In this paper, we argue in the context of graph cover pseudocodeword that, for a code that permits a cycle-free Tanner graph, cycles have no effect on error performance as long as they are a part of redundant rows. Specifically, we characterize all parity-check matrices that are pseudocodeword-free for such class of codes.

Keywords: Iterative decoding, linear programming decoding, low-density parity-check (LDPC) code, Tanner graphs, pseudocodewords

Mathematics Subject Classification: 94B05

1 Introduction

Modern decoding algorithms such as message-passing iterative decoding and linear programming decoding are extremely efficient and are shown to enable communications at rates near the channel capacity under several circumstances. These decoders are known to converge when the Tanner graph is cycle-free, and so one of the design criteria for the constructions of LDPC codes is the number and size of cycles in the underlying graphs. Thus, regular and irregular LDPC codes are usually constructed semi-randomly with procedures that avoid cycles with small girth [13, 16, 19]. There are also investigations on the effects of cycles on the performance of iterative decoders [11].

Another explanation for when iterative decoders fail to converge is by means of the pseudocodewords. Since the pseudocodewords satisfy every condition set by the decoder, they are legitimate from the perspective of the algorithm. The so-called pseudoweight acts as Hamming weight for LDPC codes, and the pseudocodewords provide a tangible

*W. Kositwattanarerk is with the Department of Mathematics, Faculty of Science, Mahidol University, Bangkok 10400, Thailand and the Centre of Excellence in Mathematics, the Commission on Higher Education, Bangkok 10400, Thailand (e-mail: wittawat.kos@mahidol.edu).

framework for the study of the error performance of LDPC codes. For this reason, these noncodeword outputs and their properties have been extensively studied in the literature [1, 6, 7, 8, 18, 20].

Wiberg [17] is among the first to study noncodeword outputs from iterative decoders where computations from the algorithms are retracted and laid out as a tree. Since iterative decoders perform calculations locally, it is probable that the algorithms try to make an estimate on a graph that behaves locally like the original Tanner graph. In [7], the pseudocodewords are characterized using a finite degree lift of the Tanner graph called a graph cover, and this description of the pseudocodeword relates well to noncodeword outputs from linear programming decoding [4]. Excellent overviews of the pseudocodewords can be found, for example, in [1, 6].

In this paper, we focus on the pseudocodewords arising from graph covers of the Tanner graph. We provide an exact condition for codes with cycle-free Tanner graph to be pseudocodeword-free. Although codes in this class are known to have limited capabilities [3], the results given here yield a surprising insight on the structure of the pseudocodewords: good representation for this class of codes has *nothing* to do with cycles as long as there exists a spanning tree of the Tanner graph that represents the same code. As a result, this work sheds light on empirical phenomenons where, under certain circumstances, iterative decoders perform well despite a number of small cycles in the Tanner graph [9] and eliminating small cycles does not significantly improve decoding performance of LDPC codes [12].

The remainder of this paper is organized as follows. Section 2 provides some background on parity-check codes and the pseudocodewords. We introduce the notion of p-satisfy in Section 3 to ease our searches for the pseudocodewords. The main result of this work is stated as Theorem 3.5. Several examples are given in Section 4.

2 Preliminaries

A code can be represented by many parity-check matrices, but a parity-check matrix uniquely determines a code. This choice of representation can be especially problematic in the context of decoding algorithms that operate on a parity-check matrix. To avoid unnecessary confusions, we define a code based on its parity-check matrix.

Definition 2.1 *Let $H \in \mathbb{F}_2^{r\times n}$. The binary linear code with parity-check matrix H, denoted $C(H)$, is the null space of H. In other words,*

$$C(H) = \{\mathbf{y} \in \mathbb{F}_2^n \mid H\mathbf{y}^T = \mathbf{0} \in \mathbb{F}_2^{r\times 1}\}.$$

Here, we do not impose that rows of H are linearly independent. The above definition coincides with the usual terminology–if $H \in \mathbb{F}_2^{r\times n}$, then $C(H)$ is a subspace of $\mathbb{F}_2^n$ of dimension at least $n-r$. We shall write C as a code to mean a subspace of $\mathbb{F}_2^n$ without a specific choice of parity-check matrix and say that H represents C if $C = C(H)$. Throughout, our discussions will focus on the parity-check matrix H and not just the code C, since iterative decoders and linear programming decoder depend rather on the choice of parity-check matrix of the code. The Tanner graph from a binary matrix is defined next.

Definition 2.2 *Let $H \in \mathbb{F}_2^{r\times n}$. The Tanner graph of H, denoted $T(H)$, is a bipartite graph with biadjacency matrix H. Specifically,*

$$T(H) = (X \cup F, E)$$

where $X = \{x_1, \ldots, x_n\}$ represents the columns of H and is called the set of bit nodes, $F = \{f_1, \ldots, f_r\}$ represents the rows of H and is called the set of check nodes, and

$$E = \{\{x_i, f_j\} \mid h_{ji} = 1\}.$$

Given a parity-check matrix H, it is clear that the Tanner graph is bipartite. The converse of this statement is also true: given a bipartite graph, it can be viewed as a Tanner graph of some parity-check matrix. In a way, the Tanner graph gives a graphical representation of the parity-check matrix. Suppose that binary values $c_1, c_2, \ldots, c_n$ are assigned to the bit nodes of the Tanner graph $T(H)$. Then, $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ is a codeword of $C(H)$ if and only if the binary sum of the values at the neighbors of every check node is zero. Next, we give the definition of a graph cover.

Definition 2.3 *Let m be a positive integer. An m-cover of $T(H)$ is a graph $\widehat{T}(H)$ with the property that there exists an m-to-1 mapping φ from the vertices of $\widehat{T}(H)$ to the vertices of $T(H)$ that preserves degree and the set of neighbors. Namely, if v is a vertex of $\widehat{T}(H)$ of degree t and with neighbors $u_1, \ldots, u_t$, then $\varphi(v)$ is a vertex of $T(H)$ of degree t and with neighbors $\varphi(u_1), \ldots \varphi(u_t)$.*

Although we only study a graph cover of the Tanner graph $T(H)$, we remark here that the above definition can be applied to give a graph cover of any graph. Since a graph cover $\widehat{T}(H)$ is bipartite, it can be viewed as a Tanner graph of some parity-check matrix. Similar to how $T(H)$ can be used to determine codewords, the graph cover $\widehat{T}(H)$ can be used to determine what is called graph cover pseudocodeword.

Before we proceed to the definition of graph cover pseudocodeword, we briefly discuss iterative decodings and linear programming decoding and refer readers to [4, 10] for a more precise description. Suppose that information are sent over a memoryless binary-input symmetric-output channel. Given a received word $\mathbf{w}$, maximum likelihood (ML) decoder finds $\mathbf{y} \in C$ that maximizes $P(\mathbf{w} \mid \mathbf{y})$. This is equivalent to choosing $\mathbf{y} \in C$ that minimizes

$$\gamma_1 y_1 + \gamma_2 y_2 + \ldots + \gamma_n y_n \tag{1}$$

where

$$\gamma_i = \log\left(\frac{P(w_i \mid y_i = 0)}{P(w_i \mid y_i = 1)}\right)$$

is the log-likelihood ratio at the i^{th} coordinate. Iterative decodings (also known as message-passing iterative decoding or belief propagation) perform inference on the Tanner graph. Roughly speaking, once a word $\mathbf{w}$ is received, the log-likelihood ratios γ_i are assigned to the bit nodes. Each bit node then broadcasts its value to the neighboring check nodes. Once a check node receives the likelihood from all its neighboring bit nodes, it uses this information to make new estimates and send them back. Each bit node updates its likelihood, and then the process iterates.

Linear programming decoding aims to minimize (1) over the convex hull of $C(H)$ when implicitly embedded in $\mathbb{R}^n$. Since the number of constraints needed to describe this convex hull is exponential in block length, linear programming decoding is typically done over a relaxed polytope

$$\bigcap_{j=1}^{r} conv(C(Row_j(H))),$$

which is the intersection of the convex hull of codewords from the r simple parity-check codes given by the rows of H.

The hallmark of iterative decoding is that the entire process is local, meaning that each vertex has its own agenda and makes decision independently. Rather than searching for a codeword that satisfies every parity condition collectively, the algorithms look for a codeword that satisfies each parity condition iteratively. Thus, the algorithms cannot distinguish between the Tanner graph and its cover. This results in the algorithms trying to converge to a legitimate binary value assignment on the graph cover. We call these graph cover pseudocodewords. In a similar fashion, linear programming decoding breaks parity-check matrix into a collection of parity conditions determined by each row of the matrix. While the relaxed polytope has a more tractable representation than the original codeword polytope, it could be the case that the relaxed polytope has a vertex that is not in the convex hull of $C(H)$. Those noncodeword vertex are a scale of a graph cover pseudocodeword [4]. From now on, we use the term pseudocodeword to refer to graph cover pseudocodeword and give its definition next.

Definition 2.4 *Let $\widehat{T}(H)$ be an m-cover of $T(H)$ with an m-to-1 mapping φ. Suppose that binary values $c_{i1}, c_{i2}, \ldots, c_{im}$ are assigned to the preimage of the bit node x_i of $T(H)$ under φ in such a way that the binary sum of the values at the neighbors of every check node of $\widehat{T}(H)$ is zero. (In other words, $c_{11}, c_{12}, \ldots, c_{1m}, \ldots, c_{n1}, c_{n2}, \ldots, c_{nm}$ is a legitimate codeword of $\widehat{T}(H)$.) Then, the integer vector*

$$\left(\sum_{k=1}^{m} c_{1k}, \ldots, \sum_{k=1}^{m} c_{nk} \right)$$

is a pseudocodeword of H. The set of all pseudocodewords of H is denoted $PC(H)$.

Note that $PC(H)$ collects pseudocodewords from an m-cover $\widehat{T}(H)$ for all values of m. In particular, a 1-cover of $T(H)$ is $T(H)$ itself, so each codeword of $C(H)$ is considered a pseudocodeword.

Example 2.5 *Consider a parity-check matrix*

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

It follows that

$$C(H) = \{(0,0,0,0), (0,1,1,1), (1,0,1,0), (1,1,0,1)\}.$$

The Tanner graph $T(H)$ and a codeword $(1,1,0,1)$ on $T(H)$ is shown in Figure 1. Figure 2 portrays a 2-cover of $T(H)$ along with a pseudocodeword $(1,2,1,0)$. Here, note that $(1,2,1,0)$ is not an integral combination of elements from $C(H)$.

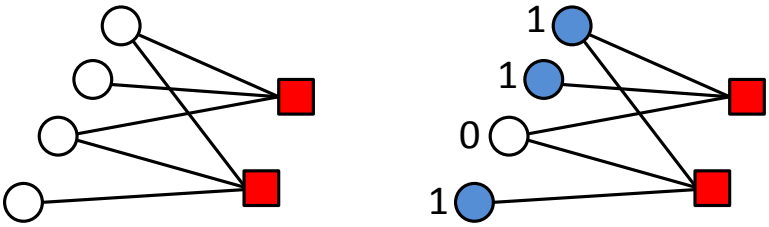

Figure 1: On the left is the Tanner graph $T(H)$ from Example 2.5. White circles are bit nodes and represent columns of H. Red squares are check nodes and represent rows of H. An assignment $1, 1, 0, 1$ to the bit nodes corresponds to the codeword $(1, 1, 0, 1)$ of $C(H)$ and is shown on the right.

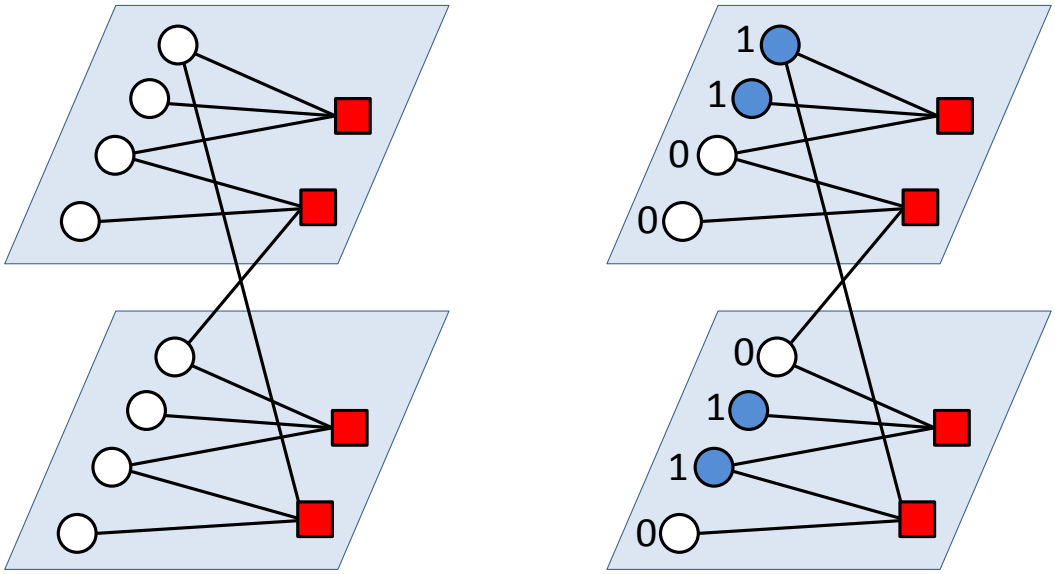

Figure 2: On the left is a 2-cover $\widehat{T}(H)$ of $T(H)$ from Example 2.5. A legitimate binary value assignment on the right corresponds to the pseudocodeword $(1, 2, 1, 0)$.

Given a code C, it is desirable to represent C with a parity-matrix H so that the set of pseudocodewords $PC(H)$ is as small as possible. Toward this goal, we implicitly embed $C(H)$ in $\mathbb{R}^n$ and enumerate some elements of $PC(H)$. We have already seen that

$$C(H) \subseteq PC(H).$$

In fact, since a graph consisting of m copies of the Tanner is consider an m-cover, any integral combinations of the elements of $C(H)$ whose coefficients are nonnegative are considered a pseudocodeword; that is,

$$\left\{ \sum_{\mathbf{c} \in C(H)} a_{\mathbf{c}} \mathbf{c} \;\middle|\; a_{\mathbf{c}} \in \mathbb{N} \right\} \subseteq PC(H).$$

Again, addition and multiplication here are done over $\mathbb{R}$. The case when the above inclusion becomes an equality deserves a special consideration.

Definition 2.6 *A parity-check matrix H is geometrically perfect if the pseudocodewords of H are precisely integer combinations of the codewords of H with nonnegative coefficients. In other words,*

$$PC(H) = \left\{ \sum_{\mathbf{c} \in C(H)} a_{\mathbf{c}} \mathbf{c} \;\middle|\; a_{\mathbf{c}} \in \mathbb{N} \right\}.$$

Being geometrically perfect means that the set of pseudocodewords is kept as small as possible. We see that the matrix H from Example 2.5 is not geometrically perfect since $PC(H)$ contains $(1, 2, 1, 0)$. A well-known class of geometrically perfect parity-check matrices is the collection of matrices whose Tanner graph is cycle-free. Here, if $\widehat{T}(H)$ is a cover of $T(H)$ that is cycle-free, then $\widehat{T}(H)$ simply consists of disconnected copies of $T(H)$, and so any pseudocodeword of H is an integral combination of the codewords.

In this work, we tie the property of being geometrically perfect to a *parity-check matrix* and not a *code*. A code can be represented by many parity-check matrices, and for one to be geometrically perfect does not guarantee that all others are. It is shown in [2, 5] that a code permits a geometrically perfect parity-check matrix if and only if it does not contain any code equivalent to $\mathcal{H}_7^{\perp}$, R_{10}, or $C(K_5)^{\perp}$ as a minor. However, it is not clear which parity-check matrix makes that so. Our work is a step toward understanding this choice of representation.

An algebraic characterization of the pseudocodewords is given in [7]. Since $PC(H)$ is closed under addition, its elements must form a cone in $\mathbb{R}^n$. Koetter et al. [7] identify this cone and precisely characterize the elements that are pseudocodewords.

Definition 2.7 *Let $H \in \mathbb{F}_2^{r \times n}$. The fundamental cone of H, denoted $K(H)$, is given by*

$$K(H) = \left\{ \mathbf{v} \in \mathbb{R}^n \;\middle|\; v_i \geq 0 \text{ and } \sum_{l=1, l \neq i}^{n} h_{jl} v_l \geq h_{ji} v_i \text{ for all } 1 \leq i \leq n \text{ and } 1 \leq j \leq r \right\}.$$

Theorem 2.8 *[7, Theorem 4.4] Let $H \in \mathbb{F}_2^{r\times n}$. An integer vector $\mathbf{p}$ is a pseudocodeword of H if and only if*

$$\mathbf{p} \in K(H) \quad \textit{and} \quad H\mathbf{p}^T = \mathbf{0} \pmod 2.$$

The inequalities of the fundamental cone demand, roughly speaking, that no single element of a pseudocodeword gets too large. Theorem 2.8 gives a convenient method to enumerate $P(H)$. Instead of going through all covers of the Tanner graph $T(H)$, one may verify whether $\mathbf{p}$ is a pseudocodeword by checking its entries against a set of inequalities and parity conditions determined by the rows of H. It is also easy to see that $C(H) \subset K(H)$, and so $K(H)$ contains every integral combination of the codewords.

3 Pseudocodeword-Free Representation of a Code

We have seen that the Tanner graph is a convenient graphical representation of a parity-check matrix. Each check node of the Tanner graph literally serves as a "parity-check" for a legitimate codeword. On the other hand, determining whether an integer vector is a pseudocodeword requires one to either find a suitable graph cover or appeal to Theorem 2.8 and test the vector algebraically. To assist our study of the pseudocodewords, we first give a new use to the Tanner graph.

Recall that $(c_1, c_2, \ldots, c_n)$ is a codeword of $C(H)$ if and only if the assignment $c_1, c_2, \ldots, c_n$ to the bit nodes of the Tanner graph $T(H)$ makes the binary sum of the neighbors of every check node zero. In other words, a check node is "satisfied" if the binary sum of its neighbors is zero, and a codeword is what makes every check node satisfied. Motivated by this observation, we generalize the property of being "satisfied" in the following definition.

Definition 3.1 *Let v be a vertex of a graph G with the set of neighbors $\{u_1, u_2, \ldots, u_t\}$. Suppose that integer values $a_1, \ldots, a_t$ are assigned to these vertices. We say that v is p-satisfied if all of the the following conditions hold:*

i) $a_i \geq 0$ *for all* i,

ii) $\sum_{l=1}^{t} a_l = 0 \pmod 2$, *and*

iii) $\sum_{l=1, l\neq i}^{t} a_l \geq a_i$ *for all* $1 \leq i \leq t$.

Technically, the property of being p-satisfy localizes the conditions for pseudocodewords given in Theorem 2.8. It makes the Tanner graph sufficient to verify whether an integer vector is a pseudocodeword, and we state this fact as the following proposition.

Proposition 3.2 *Let $T(H)$ be the Tanner graph of $H \in \mathbb{F}_2^{r\times n}$. An integer vector $(p_1, \ldots, p_n)$ is a pseudocodeword if and only if the assignment $p_1, \ldots, p_n$ to the corresponding bit nodes of $T(H)$ makes every check node p-satisfied.*

Proof Combining Definition 2.7 and Theorem 2.8, an integer vector $(p_1, \ldots, p_n)$ is a pseudocodeword if and only if, for all $1 \leq j \leq r$,

i) $p_i \geq 0$ for all i,

ii) $\sum_{l=1,l\neq i}^{n} h_{jl}p_l \geq h_{ji}p_i$ for all $1 \leq i \leq n$, and

iii) $\sum_{l=1}^{n} h_{jl}p_l \equiv 0 \pmod 2$.

Since $h_{jl} = 1$ precisely when the bit node x_l is adjacent to the check node f_j, the above three conditions are satisfied for all $1 \leq j \leq r$ if and only if every check node is p-satisfied with $p_1, \ldots, p_n$. ∎

We provide several useful results before stating the main finding of this work. The proof of Lemma 3.3 is trivial and is omitted. Here, $w(\cdot)$ denotes the Hamming weight of a binary vector.

Lemma 3.3 *Let H be a parity-check matrix. Suppose that the check node f_{j_1} corresponding to $Row_{j_1}(H)$ and the check node f_{j_2} corresponding to $Row_{j_2}(H)$ have no common neighbor. If $\mathbf{r}$ is the binary sum of $Row_{j_1}(H)$ and $Row_{j_2}(H)$, then*

$$w(\mathbf{r}) = w(Row_{j_1}(H)) + w(Row_{j_2}(H)).$$

In particular,

$$w(\mathbf{r}) \geq w(Row_{j_1}(H)) \quad \textit{and} \quad w(\mathbf{r}) \geq w(Row_{j_2}(H)).$$

Proposition 3.4 *Let $H \in \mathbb{F}_2^{r\times n}$. If $T(H)$ is a tree where every check node has degree at least 2, then*

i) the number of 1's in H is precisely $r + n - 1$,

ii) if

$$\mathbf{r} = Row_{j_1}(H) + Row_{j_2}(H) + \ldots + Row_{j_s}(H) \pmod 2,$$

then $w(\mathbf{r}) \geq w(Row_{j_t}(H))$ for $t = 1, 2, \ldots, s$, and

iii) the rows of H are linearly independent over $\mathbb{F}_2$.

Proof Since $T(H)$ is a tree with $r+n$ vertices, $T(H)$ has $r+n-1$ edges, and *i)* follows.

Let $\mathbf{r} = Row_{j_1}(H) + Row_{j_2}(H) + \ldots + Row_{j_s}(H) \pmod 2$. Without loss of generality, we shall prove that $w(\mathbf{r}) \geq w(Row_{j_1}(H))$, and *ii)* will readily follow. Consider the tree $T(H)$ with the check node f_{j_1} corresponding to $Row_{j_1}(H)$ as a root node. Clearly, $T(H) \setminus Row_{j_1}(H)$, the subgraph of $T(H)$ with vertex $Row_{j_1}(H)$ removed, is a graph with $\deg(f_{j_1}) = w(Row_{j_1}(H))$ connected components. Construct a check node $f_{\mathbf{r}}$ corresponding to $\mathbf{r}$ (that is, x_i is adjacent to $f_{\mathbf{r}}$ if and only if the i^{th} element of $\mathbf{r}$ is 1). If we can show that $f_{\mathbf{r}}$ is connected to each component of $T(H) \setminus Row_{j_1}(H)$, then $\deg(f_{\mathbf{r}}) \geq \deg(f_{j_1})$, and we can conclude that $w(\mathbf{r}) \geq w(Row_{j_1}(H))$ as required.

Let U be one of the connected components of $T(H) \setminus Row_{j_1}(H)$, and consider the check nodes among $f_{j_2}, \ldots, f_{j_s}$ that belong to this component. If there is none, then $f_{\mathbf{r}}$ is adjacent to the bit node in this component that is adjacent to f_{j_1}. Assume without loss of generality now that check nodes $f_{j_2}, \ldots, f_{j_k}$ belong to this component. Consider the subgraph of U consisting of check nodes $f_{j_2}, \ldots, f_{j_k}$ and their neighbors. This subgraph must be cycle-free and hence has at least two leaves. However, no check node can be a leaf since they have degree at least 2. The check node $f_{\mathbf{r}}$ must now be adjacent to one of the leaves and therefore connected to this component. This finishes the proof of *ii)*.

Finally, *iii)* follows from *ii)* since we cannot have

$$\mathbf{0} = Row_{j_1}(H) + Row_{j_2}(H) + \ldots + Row_{j_s}(H) \pmod 2$$

for any rows $j_1, \ldots, j_s$ of H.

∎

Proposition 3.4 will be helpful since we will be dealing with a parity-check matrix whose Tanner graph does not have a cycle. We know that a representation H of a code is geometrically perfect if $T(H)$ is cycle-free. The following theorem now classifies all geometrically perfect representations of such code.

Theorem 3.5 *Let C be a code that has a cycle-free representation. A parity-check matrix H of this code is geometrically perfect if and only if (possibly empty) redundant rows of H can be removed so as to obtain a representation of C that is cycle-free.*

Proof Suppose that C is a concatenation of e codes, and let $H' \in \mathbb{F}_2^{r\times n}$ be any cycle-free representation of C. It is not hard to see that $T(H')$ is a forest, i.e., a collection of disconnected e trees. We assume that $T(H')$ has no check node of degree 1, for if $T(H')$ has a check node of degree 1, then the coordinate corresponding to the neighboring bit node can be punctured or ignored.

Assume that (possibly empty) rows of H can be removed so as to obtain a representation $\widetilde{H}$ of C that is cycle-free. That is, $C(\widetilde{H}) = C(H)$, and $T(\widetilde{H})$ is cycle-free. Then, it follows from [6, Theorem 6.1] that

$$PC(H) \subseteq PC(\widetilde{H}).$$

Now, since $C(\widetilde{H}) = C(H)$, integer combinations of codewords of $\widetilde{H}$ and codewords of H are the same. As $\widetilde{H}$ is geometrically perfect, we have

$$P(H) \subseteq P(\widetilde{H}) = \left\{ \sum_{c\in C(\widetilde{H})} a_c c \mid a_c \in \mathbb{N} \right\} = \left\{ \sum_{c\in C(H)} a_c c \mid a_c \in \mathbb{N} \right\} \subseteq P(H).$$

Therefore, H is geometrically perfect.

Suppose now that it is not possible to remove redundant rows of H so that a cycle-free representation of C can be obtained. Let $j'_1, j'_2, \ldots, j'_s$ be the rows of H' that are not in H. This list is not empty since we cannot remove rows of H to obtain H'. Since H and H' represent the same null space (i.e., the code C), for each $k = 1, 2, \ldots, s$, some rows of H must be binary sum of j'_k and a linear combination of some other rows of H'; we denote such rows $j_{k,1}, \ldots, j_{k,a_k}$. It follows from Lemma 3.3 and Proposition 3.4 that

$$w(Row_{j_{k,l}}(H)) \geq w(Row_{j'_k}(H')) \tag{2}$$

for all $1 \leq l \leq a_k$. Suppose for the sake of contradiction that, for each $k = 1, 2, \ldots, s$, there is $1 \leq b_k \leq a_k$ such that $w(Row_{j_{k,b_k}}(H)) = w(Row_{j'_k}(H'))$. This means we can remove rows of H so that only rows $j_{1,b_1}, j_{2,b_2}, \ldots, j_{s,b_s}$ and rows of H that are in H' remain. This representation of C is cycle-free, and hence is a contradiction. Therefore,

there exists a row j'_p of H' such that the equality of (2) does not hold for any $1 \leq l \leq a_p$. This row is essential to our proof, and we call it *pivotal*.

The pivotal row j'_p of H' is represented by a check node $f_{j'_p}$ in the Tanner graph $T(H')$. Let U' be the connected component of $T(H')$ that j'_p belongs to (i.e., U' is one of e trees in the forest $T(H')$). Denote $d := w(Row_{j'_p}(H'))$ so that the check node $f_{j'_p}$ has degree d. It is clear that $U' \setminus f_{j'_p}$, the subgraph of U' with vertex $f_{j'_p}$ removed, is a cycle-free graph with d connected components. We assign values $2d$ to all bit notes in one component and 2 to all other bit nodes of $T(H')$. We will prove that this assignment does not p-satisfy $f_{j'_p}$ but p-satisfies every check node of $T(H)$. Hence, we can appeal to Proposition 3.4 and conclude that H is not geometrically perfect as $PC(H)$ contains an extraneous pseudocodeword.

Our arguments will center around condition *iii)* of Definition 3.1. Roughly speaking, this condition requires that the value at one neighboring bit node cannot be greater than the values at all other neighboring bit nodes combined. The check node $f_{j'_p}$ of H' is adjacent to 1 bit node whose value is $2d$ and $d-1$ bit nodes whose value is 2 (see Figure 3), and so this check node is not p-satisfied as $\underbrace{2+2+\ldots+2}_{d-1 \text{ terms}} < 2d$.

We are left to show that the same assignment makes a pseudocodeword for H. Conditions *i)* and *ii)* from Definition 3.1 are trivially satisfied for every check node of $T(H)$, and now only condition *iii)* is to be verified. Rows of H can be identified as either

- $j_{p,1}, \ldots, j_{p,a_p}$, or
- linear combination of rows $1, 2, \ldots, j'_p - 1, j'_p + 1, \ldots, r$ from H'.

Since

$$w(Row_{j_{p,l}}(H)) > w(Row_{j'_p}(H')) = d$$

for all $1 \leq l \leq a_p$, the required condition is satisfied for check nodes $j_{p,1}, \ldots, j_{p,a_p}$ of H (see Figure 3).

Consider now a check node that is a linear combination of rows $1, 2, \ldots, j'_p - 1, j'_p + 1, \ldots, r$ of H'. If this check node is only adjacent to bit nodes whose value is 2, then clearly condition *iii)* is satisfied. Finally, if this check node is adjacent to a bit node whose value is $2d$, it must be adjacent to at least 2 such bit nodes since each check node has degree at least 2. Thus, conditions *iii)* is satisfied, and this finishes the proof of the theorem. ∎

4 Examples

We present two examples in this section. The first one illustrates the key step used in the proof of Theorem 3.5. The second one showcases the code considered in [17]. As we will see, it is possible that a parity-check matrix is geometrically perfect despite the presence of several small cycles.

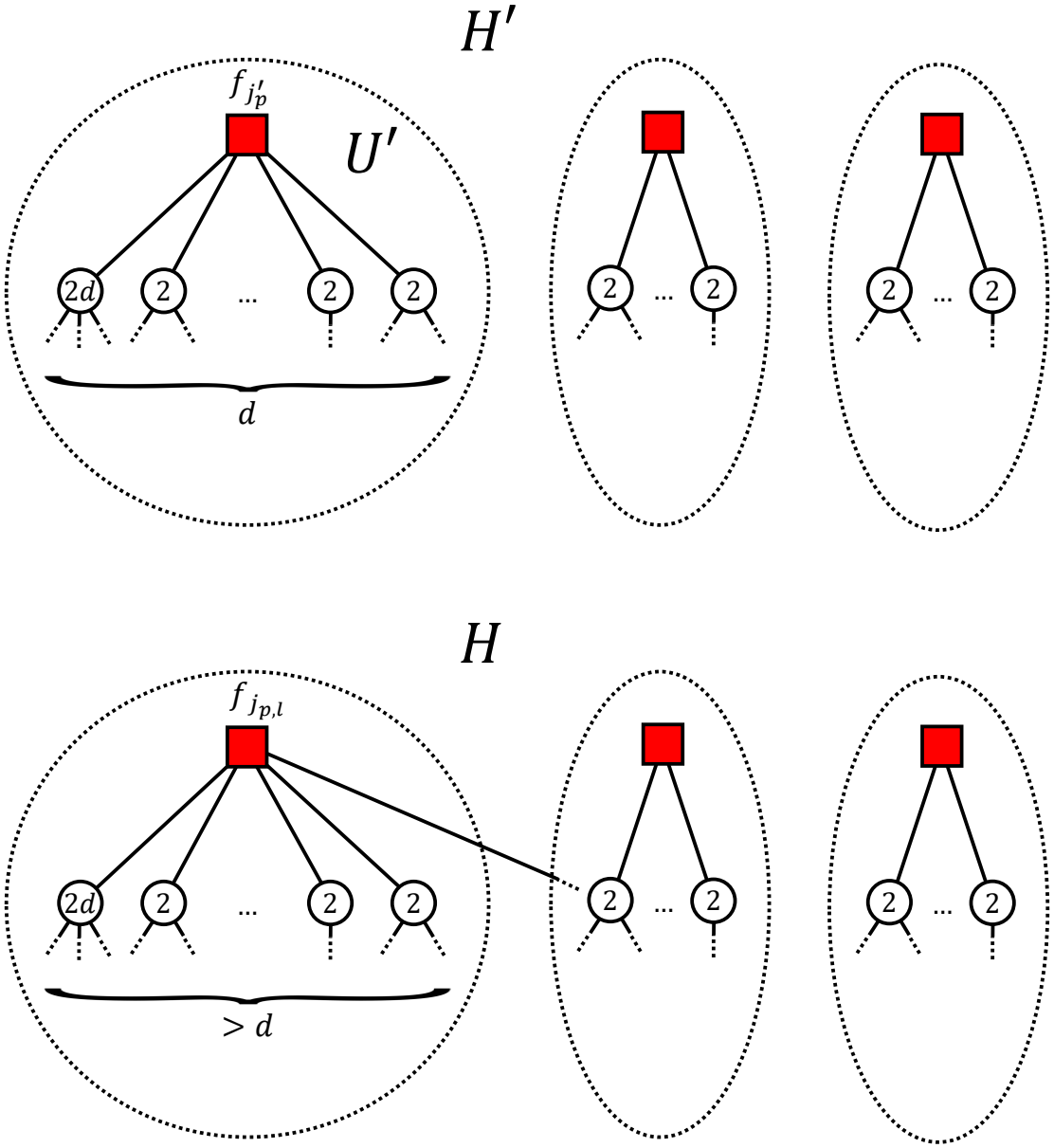

Figure 3: Check node $f_{j'_p}$ is not p-satisfied (top) while check node $f_{j_{p,l}}$ is (bottom).

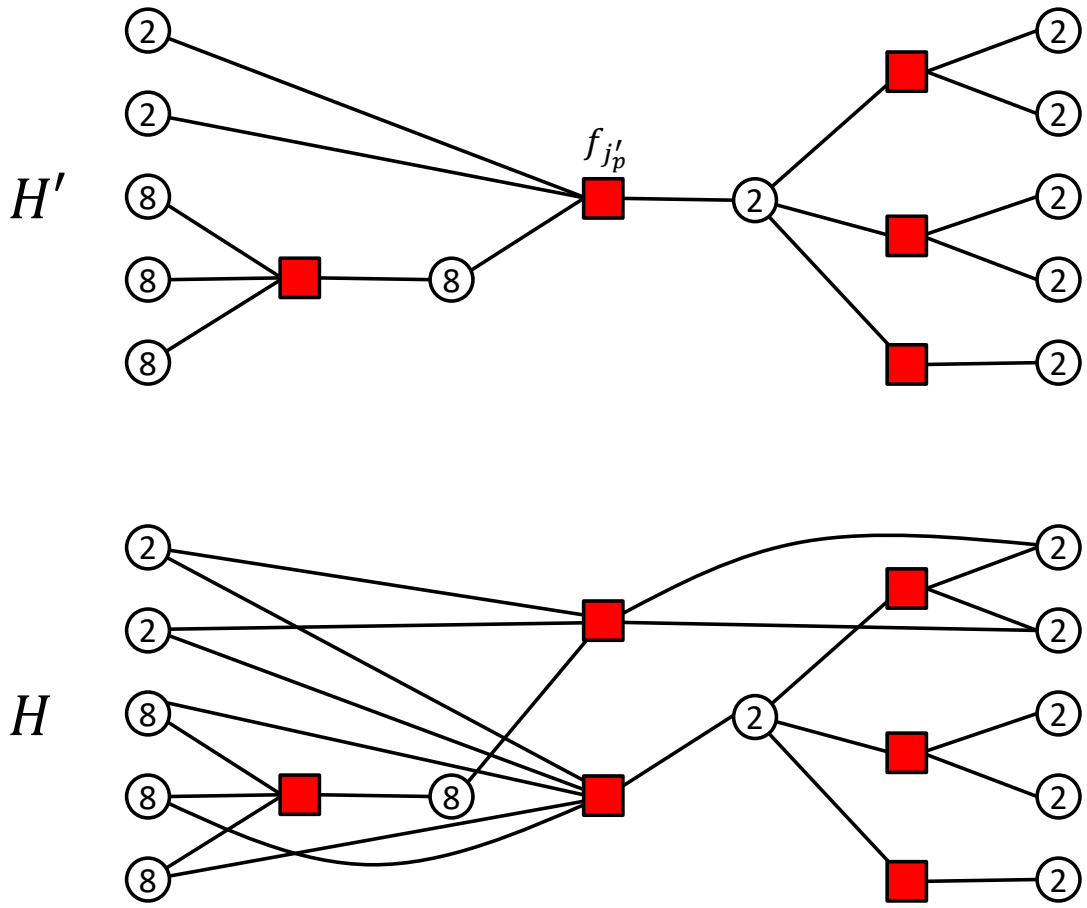

Figure 4: On the top presents the Tanner graph $T(H')$ and the pivotal check node $f_{j'_p}$ from Example 4.1. The check node in consideration is not p-satisfied. At the bottom is the Tanner graph of H. One readily sees that every check node is p-satisfied, and so $(2, 2, 8, 8, 8, 8, 2, 2, 2, 2, 2, 2)$ is a pseudocodeword for H.

Example 4.1 *Suppose that*

$$H' = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$H = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Let $C = C(H') = C(H)$. It is not possible to remove rows of H to obtain H' or any cycle-free representation of C. Here, the second row of H' is pivotal and is labeled $f_{j'_p}$ in Figure 4. We have $d = w(Row_2(H')) = 4$ and $T(H') \setminus f_{j'_p}$ has 4 connected components. Value 8 is assigned to all bit notes in one component, and value 2 is assigned to all other bit nodes. The check node $f_{j'_p}$ of $T(H')$ is not p-satisfied with this assignment. However, every check node of $T(H)$ is p-satisfied. This makes

$$PC(H) \subset PC(H'),$$

and so H' is not geometrically perfect.

Example 4.2 *This example considers three representations of the code of length 7 and dimension 4 used to demonstrate the min-sum algorithm in [17, Section 3.1]. The first representation H_1 yields a tree, the second representation H_2 is a cycle code (i.e., every bit node has degree 2), and the third representation combines check conditions of the two. See Figure 5.*

It is not hard to see that

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

Now, H_1 is geometrically perfect, and so

$$PC(H_1) = \left\{ \sum_{\mathbf{c} \in C} a_{\mathbf{c}} \mathbf{c} \;\middle|\; a_{\mathbf{c}} \in \mathbb{N} \right\}.$$

On the other hand, there is no redundant row of H_2 whose removal yields a cycle-free representation of C. This representation is not geometrically perfect as

$$PC(H_2) = \left\{ \sum_{\mathbf{d} \in D} a_{\mathbf{d}} \mathbf{d} \;\middle|\; a_{\mathbf{d}} \in \mathbb{N} \right\}$$

where

$$D = C \cup \{(2,0,0,1,1,1,1), (0,2,0,1,1,1,1), (0,0,2,1,1,1,1), (2,2,2,1,1,1,1)\}.$$

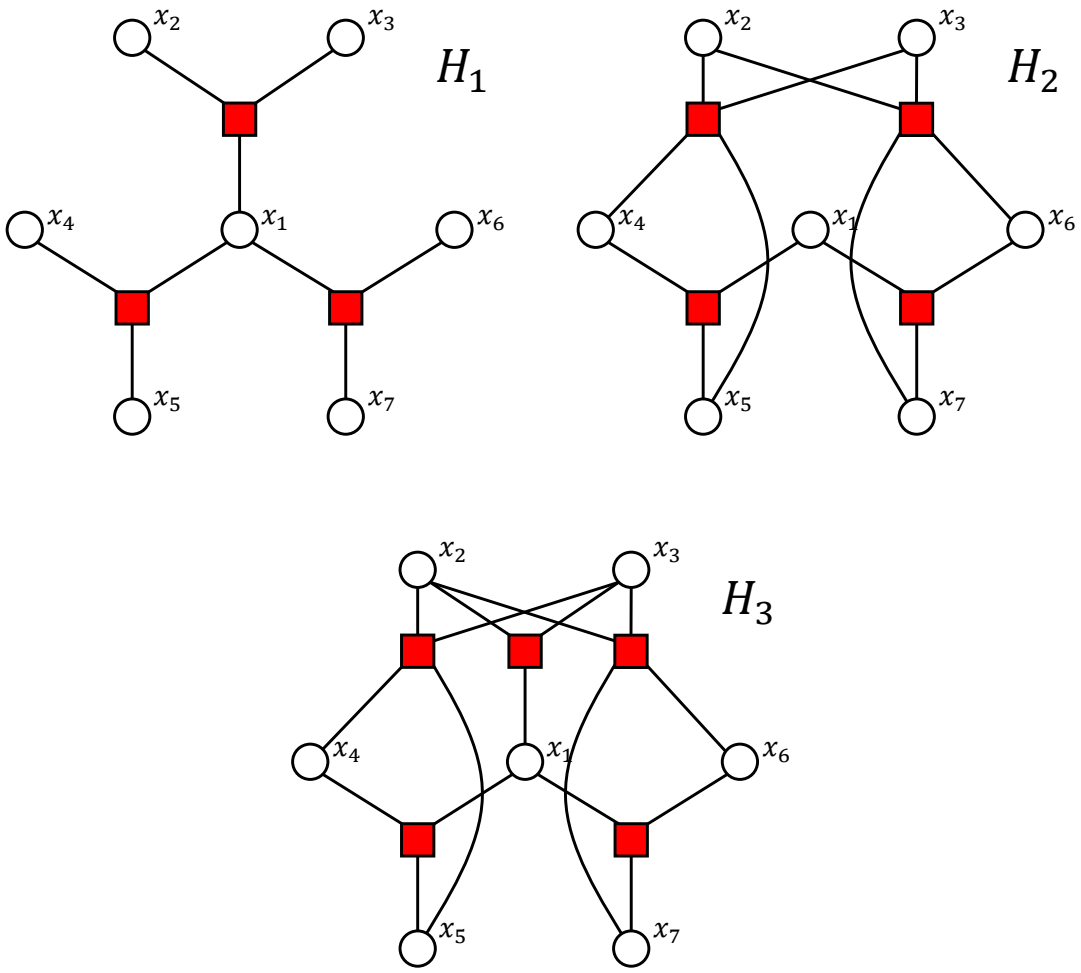

Figure 5: Three representations of the the code from Example 4.2

The third representation combines the rows of H_1 and H_2, and is quite interesting since it contains both a subgraph that is a tree and several 4-cycles. In this case, the tree dominates as the representation H_3 is geometrically perfect. In other words, there are no pseudocodewords besides the integral combinations of the codewords. One plausible explanation here in terms of graph cover is that, although small cycles bolster the pseudocodewords, the cycle-free subgraph $T(H_1)$ forbids one from coming up.

Acknowledgments

The author wishes to thank Gretchen L. Matthews for her support while the author is at Clemson University and Patanee Udomkavanich for her advice. This work is supported by the Thailand Research Fund under Grant TRG5880116 and the Centre of Excellence in Mathematics, the Commission on Higher Education, Thailand.

References

[1] N. Axvig, D. Dreher, K. Morrison, E. Psota, L. C. Perez, and J. L. Walker, Analysis of connections between pseudocodewords, IEEE Trans. Inform. Theory **55** (2009), no. 9, 4099–4107.

[2] F. Barahona and M. Grötschel, On the cycle polytope of a binary matroid, J. Comb. Theory, Ser. B **40** (1986), 40–62.

[3] T. Etzion, A. Trachtenberg, A. Vardy, Which codes have cycle-free Tanner graphs?, IEEE Trans. Inform. Theory **45** (1999), no. 6, 2173–2181.

[4] J. Feldman, M. J. Wainwright, and D. R. Karger, Using linear programming to decode binary linear codes, IEEE Trans. Inform. Theory **51** (2005), no. 3, 954–972.

[5] N. Kashyap, A decomposition theory for binary linear codes, IEEE Trans. Inform. Theory **54** (2008), no. 7, 3035–3058.

[6] C. Kelley and D. Sridhara, Pseudocodewords of Tanner graphs, IEEE Trans. Inform. Theory **53** (2007), no. 11, 4013–4038.

[7] R. Koetter, W.-C. W. Li, P. O. Vontobel, and J. Walker, Characterizations of pseudocodewords of (low-density) parity-check codes, Adv. Math. **213** (2007), no. 1, 205–229.

[8] W. Kositwattanarerk and G. L. Matthews, Lifting the fundamental cone and enumerating the pseudocodewords of a parity-check code, IEEE Trans. Inform. Theory **57** (2011), no. 2, 898–909.

[9] Y. Kou, S. Lin, and M. P. C. Fossorier, Low-density parity-check codes based on finite geometries: a rediscovery and new results, IEEE Trans. Inform. Theory **47** (2001), no. 7, 2711–2736.

[10] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Trans. Inform. Theory **47** (2001), no. 2, 498–519.

[11] G. Lechner, The effect of cycles on binary message-passing decoding of LDPC codes, Proc. Comm. Theory Workshop, Australia, IEEE (2010).

[12] D. J. C. MacKay and R. M. Neal, Near Shannon limit performance of low density parity check codes, Electronics Letters **32** (1996), 1645–1646.

[13] T. Richardson and A. Shokrollahi and R. Urbanke, Design of capacity-approaching irregular low-density parity-check codes, IEEE Trans. Inform. Theory **47** (2001), no. 2, 619–637.

[14] R. Smarandache and P. O. Vontobel, Pseudo-codeword analysis of Tanner graphs from projective and Euclidean planes, IEEE Trans. Inform. Theory **53** (2007), no. 7, 2376–2393.

[15] R. M. Tanner, A recursive approach to low-complexity codes, IEEE Trans. Inform. Theory **27** (1981), 533–547.

[16] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, Selective avoidance of cycles in irregular LDPC code construction, IEEE Trans. Comm. **52** (2004), 1242–1247.

[17] N. Wiberg, Codes and decoding on general graphs, Ph.D. thesis, Linköping University, Linköping, Sweden, 1996.

[18] S.-T. Xia and F.-W. Fu, Minimum pseudoweight and minimum pseudocodewords of LDPC codes, IEEE Trans. Inform. Theory **54** (2008), 480–185.

[19] J. Xu, L. Chen, I. Djurdjevic, S. Lin, K. Abdel-Ghaffar, Construction of regular and irregular LDPC codes: geometry decomposition and masking, IEEE Trans. Inform. Theory **53** (2007), 121–134.

[20] J. Zumbragel, V. Skachek, and M. F. Flanagan, On the pseudocodeword redundancy of binary linear codes, IEEE Trans. Inform. Theory **58** (2012), 4848–4861.

The 22nd Annual Meeting in Mathematics (AMM 2017)
Department of Mathematics, Faculty of Science
Chiang Mai University, Chaing Mai, Thailand

Generalized Zeckendorf's Theorem and Fibonacci Coding for Modules

Wittawat Kositwattanarerk[a,b,†,‡], and Perathorn Pooksombat[a,b]

[a]Department of Mathematics, Faculty of Science
Mahidol University, Bangkok 10400, Thailand

[b]Centre of Excellence in Mathematics
CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand

Abstract

Zeckendorf's theorem states that every positive integer can be written uniquely as a sum of nonconsecutive Fibonacci numbers. In this talk, we generalize this classical result and prove that every element of a free $\mathbb{Z}$-module can be represented as a sum of elements from a Fibonacci sequence of higher order. An immediate application of our result is a Fibonacci coding for multidimensional messages over binary channels. This coding scheme is a robust countermeasure against insertion and deletion errors.

Keywords: Fibonacci sequence, Fibonacci coding, Zeckendorf's theorem, Zeckendorf's representation, $\mathbb{Z}$-module.

2010 MSC: Primary 11B39 ; Secondary 68P30.

*This research was financially supported by the Thailand Research Fund under Research Grant TRG5880116.

†Corresponding author.

‡Speaker.

E-mail address: wittawat.kos@mahidol.edu (W. Kositwattanarerk), perathorn.pok@student.mahidol.edu (P. Pooksombat).