

Properties of Multidimensional Vector Zeckendorf Representations

Ivan Bortnovskiy^a, Julian Duvivier^b, Pedro Espinosa^c, Michael Lucas^a, Steven J. Miller^c, Tiancheng Pan^a, Arman Rysmakhanov^c, Iana Vranesko^c, Ren Watson^d, and Steven Zanetti^e

^aDepartment of Pure Mathematics and Mathematical Statistics, University of Cambridge, Cambridge, United Kingdom, CB3 0WA; ^bDepartment of Mathematics, Reed College, Portland, OR, 97202; ^cDepartment of Mathematics, Williams College, Williamstown, MA, 01267; ^dDepartment of Mathematics, University of Texas at Austin, Austin, TX 78712; ^eDepartment of Mathematics, University of Michigan, Ann Arbor, MI 48109

ARTICLE HISTORY

Compiled October 23, 2025

Article type: research

ABSTRACT

Zeckendorf's Theorem says that for all $k \geq 3$, every nonnegative integer has a unique k -Zeckendorf representation as a sum of distinct k -bonacci numbers, where no k consecutive k -bonacci numbers are present in the representation. Anderson and Bicknell-Johnson [1] extend this result to the multidimensional context: letting the k -bonacci vectors $\vec{X}_i \in \mathbb{Z}^{k-1}$ be given by $\vec{X}_0 = \vec{0}$, $\vec{X}_{-i} = \vec{e}_i$ for $1 \leq i \leq k-1$, and $\vec{X}_n = \sum_{i=1}^k \vec{X}_{n-i}$ for all $n \in \mathbb{Z}$, they show that for all $k \geq 3$, every $\vec{v} \in \mathbb{Z}^{k-1}$ has a unique k -bonacci vector Zeckendorf representation, a sum of distinct k -bonacci vectors where no k consecutive k -bonacci vectors are present in the representation. Their proof provides an inductive algorithm for finding such representations. We present two improved algorithms for finding the k -bonacci vector Zeckendorf representation of $\vec{v}$ and analyze their relative efficiency. We utilize a projection map $S_n : \mathbb{Z}^{k-1} \rightarrow \mathbb{Z}_{\geq 0}$, introduced in [1], that reduces the study of k -bonacci vector representations to the setting of k -bonacci number representations, provided a lower bound is established for the most negatively indexed k -bonacci vector present in the k -bonacci vector Zeckendorf representation of $\vec{v}$. Using this map and a bijection between $\mathbb{Z}^{k-1}$ and $\mathbb{Z}_{\geq 0}$, we further show that the number of and gaps between summands in k -bonacci vector Zeckendorf representations exhibit the same properties as those in k -Zeckendorf representations and that k -bonacci vector Zeckendorf representations exhibit summand minimality.

KEYWORDS

Fibonacci numbers; Zeckendorf's Theorem; Lekkerkerker's Theorem

Ren Watson: renwatson@utexas.edu

The first listed author was supported by The Winston Churchill Foundation of the United States. The NSF grant DMS2242623 supported the second, third, seventh, eighth, ninth, and tenth listed authors, and the third, seventh, and eighth authors were supported by Williams College and the Finnerty fund. Finally, the fourth and sixth listed authors were supported by the Dr. Herchel Smith Fellowship Fund.

1. Introduction

1.1. History and Motivation

A beautiful theorem of Zeckendorf [11] states that every nonnegative integer n can be written uniquely as a sum of non-consecutive Fibonacci numbers.¹ We refer to this representation as the **Zeckendorf decomposition** of n . Zeckendorf decompositions and their generalizations have been the subject of extensive previous study; for instance, see [3, 4, 5, 7, 8, 9, 10]. One natural extension of the Fibonacci sequence is as follows, where the Fibonacci numbers are given by taking $k = 2$.

Definition 1.1 (k -bonacci Sequence). *For a fixed choice of k , the k -bonacci sequence $\{x_n\}$ ² is given by $x_n = 0$ for $-k + 2 \leq n \leq 0$, $x_1 = 1$, and $x_n = \sum_{i=1}^k x_{n-i}$ for all $n \in \mathbb{Z}$.*

The original proof of Zeckendorf's Theorem, via the greedy algorithm for $k = 2$, naturally extends to $k \geq 3$, giving the following result. Note that the restriction $i \geq 2$ is imposed to guarantee $x_1 = x_2 = 1$ are not both allowed in the decomposition, as this would result in loss of uniqueness of decomposition.

Theorem 1.2 (Zeckendorf). *Every nonnegative integer n can be written uniquely as a sum of distinct k -bonacci numbers $n = \sum_{i \geq 2} c_i x_i$ such that $c_i \in \{0, 1\}$ for all i and no k consecutive c_i 's are equal to 1.*

For a fixed k , we refer to the unique representation of n given by Theorem 1.2 as the **k -Zeckendorf representation of n** . Formally, the greedy algorithm used to find the k -Zeckendorf representation of a positive integer n is as follows.

Definition 1.3 (k -bonacci Number Greedy Algorithm). *For a fixed $k > 1$ and positive integer n , the k -bonacci number greedy algorithm finds the unique k -Zeckendorf representation of n as follows.*

- (1) Initialize $R := n$.
- (2) Find $\ell_1 \geq 2$ maximal such that $x_{\ell_1} \leq R$.
- (3) Reset $R := R - x_{\ell_1}$.
- (4) If $R = 0$, the algorithm terminates. Else, repeat step (2) to find $\ell_2 \geq 2$ maximal such that $x_{\ell_2} \leq R$ and reset $R := R - x_{\ell_2}$. Repeat this process until $R = 0$. The finite sum $\sum_{i \geq 1} x_{\ell_i}$ forms the k -Zeckendorf representation of n .

This paper extends the study of multidimensional Zeckendorf representations initiated by Anderson and Bicknell-Johnson [1]. Before stating our results, we introduce relevant notation and prior results.

Definition 1.4 (k -bonacci Vectors). *The k -bonacci vectors $\vec{X}_i \in \mathbb{Z}^{k-1}$ are given by $\vec{X}_0 = \vec{0}$, $\vec{X}_{-i} = \vec{e}_i$ for $1 \leq i \leq k-1$, and $\vec{X}_n = \sum_{i=1}^k \vec{X}_{n-i}$ for all $n \in \mathbb{Z}$.*

The notion of k -Zeckendorf representations for nonnegative integers extends naturally to representations of vectors $\vec{v} \in \mathbb{Z}^{k-1}$ as sums of k -bonacci vectors.

Theorem 1.5. [1, Theorem 2] *Every $\vec{v} \in \mathbb{Z}^{k-1}$ has a unique representation $\vec{v} = \sum_{i \geq 1} c_i \vec{X}_{-i}$ such that $c_i \in \{0, 1\}$ for all i and no string of k consecutive c_i 's is equal*

¹We define these by $F_1 = 1, F_2 = 2$ and $F_{n+1} = F_n + F_{n-1}$; otherwise, we lose uniqueness.

²Because k is fixed, we suppress it in the notation for ease of reading.

to 1.

For instance, taking $k = 3$, the first several 3-bonacci vectors are given by:

$$\begin{aligned}\vec{\mathbf{X}}_{-1} &= (1, 0), \\ \vec{\mathbf{X}}_{-2} &= (0, 1), \\ \vec{\mathbf{X}}_{-3} &= (-1, -1), \\ \vec{\mathbf{X}}_{-4} &= (2, 0), \\ \vec{\mathbf{X}}_{-5} &= (-1, 2), \\ \vec{\mathbf{X}}_{-6} &= (-2, -3), \\ \vec{\mathbf{X}}_{-7} &= (5, 1).\end{aligned}$$

Then the unique representation of $(7, 0)$ satisfying the criteria of Theorem 1.5 is given by

$$\begin{aligned}(7, 0) &= (5, 1) + (2, 0) + (-1, -1) + (1, 0) \\ &= \vec{\mathbf{X}}_{-7} + \vec{\mathbf{X}}_{-4} + \vec{\mathbf{X}}_{-3} + \vec{\mathbf{X}}_{-1}.\end{aligned}$$

Definition 1.6 (Satisfying Representation). For $\vec{\mathbf{v}} \in \mathbb{Z}^{k-1}$, a representation $\vec{\mathbf{v}} = \sum_{i \geq 1} c_i \vec{\mathbf{X}}_{-i}$ such that all $c_i \in \{0, 1\}$ and no string of k consecutive c_i 's are equal to 1 is called a **satisfying representation (SR)** of $\vec{\mathbf{v}}$, and the unique such representation of $\vec{\mathbf{v}}$ may be denoted $SR(\vec{\mathbf{v}})$. A representation $\vec{\mathbf{v}} = \sum_{i \geq 1} c_i \vec{\mathbf{X}}_{-i}$ such that all $c_i \in \{0, 1, 2\}$, no string of k consecutive c_i 's are nonzero, and only one string of consecutive nonzero c_i 's contains any 2's is called a **nearly satisfying representation (NSR)** of $\vec{\mathbf{v}}$.

Anderson and Bicknell-Johnson [1] introduced the following scalar product, which will allow us to utilize known results regarding the k -bonacci numbers in our analysis of the k -bonacci vectors.

Definition 1.7 (k -bonacci Projection Map). For $n \geq k - 2$, let $S_n : \mathbb{Z}^{k-1} \rightarrow [0, x_n)$ be the linear map

$$S_n(\vec{\mathbf{v}}) = \vec{\mathbf{v}} \cdot (x_{n-1}, \dots, x_{n-(k-1)}) \pmod{x_n}.$$

Lemma 1.8. [1, Lemma 3] We have $S_n(\sum_{i=1}^p c_i \vec{\mathbf{X}}_{-i}) \equiv \sum_{i=1}^p c_i x_{n-i} \pmod{x_n}$.

Proof. For $0 \leq i \leq k - 1$,

$$S_n(\vec{\mathbf{X}}_{-i}) = \vec{\mathbf{e}}_i \cdot (x_{n-1}, \dots, x_{n-(k-1)}) = x_{n-i}.$$

For $i \geq k$, we have $S_n(\vec{\mathbf{X}}_{-i}) \equiv x_{n-i} \pmod{x_n}$ by the recursive definitions of $\vec{\mathbf{X}}_{-i}$ and x_{n-i} . Linearity completes the proof. $\square$

Lemma 1.9. [1, Corollary 5] Suppose $\vec{\mathbf{v}} = \sum_{i=1}^M c_i \vec{\mathbf{X}}_{-i}$ and $\vec{\mathbf{v}}' = \vec{\mathbf{v}} + \vec{\mathbf{X}}_{-p} = \sum_{i=1}^{M'} c'_i \vec{\mathbf{X}}_{-i}$ are SR's and that $p \leq M$. Then $M' \leq k + M$.

For a fixed choice of $k \geq 2$ and $\vec{\mathbf{v}} \in \mathbb{Z}^{k-1}$, Anderson and Bicknell-Johnson's [1] proof of Theorem 1.5 provides a recursive approach to finding the SR of $\vec{\mathbf{v}}$. A primary

focus of this paper is to provide an improved algorithmic approach for finding the SR of $\vec{v}$. Before stating our results, we recall the following definition.

Definition 1.10 (Big-O Notation). *Let $f(x)$ and $g(x)$ be real-valued functions. We say that $f(x) = O(g(x))$ if there exists some constant $c > 0$ and $x_0 \in \mathbb{R}$ such that $|f(x)| \leq c|g(x)|$ for all $x \geq x_0$.*

1.2. Main Results

In order to compare our algorithmic approaches to finding the SR of $\vec{v}$, we first provide in Appendix A a more formal definition and complexity analysis of the algorithm given by [1], yielding the following result.

Lemma 1.11. *Let L be the number of summands in the SR of $\vec{v} \in \mathbb{Z}^{k-1}$. The recursive algorithm for finding the SR of $\vec{v}$ in [1] runs in $O(k \cdot L \cdot \|\vec{v}\|_1)$ steps.*

In Section 2, we establish two improved algorithmic approaches for finding the SR of $\vec{v}$ and provide an analysis of their relative efficiency. Both algorithms rely on the following definition.

Definition 1.12 (Maximal k -bonacci index). *The **maximal k -bonacci index** of $\vec{v} \in \mathbb{Z}^{k-1} \setminus \{\vec{0}\}$ is*

$$J(\vec{v}) := \max\{i \in \mathbb{N} \mid \vec{X}_{-i} \text{ appears in the SR of } \vec{v}\}.$$

Both of our algorithms utilize the following modification of the k -bonacci number greedy algorithm.

Definition 1.13 (Vector Zeckendorf Greedy Algorithm). *Let $k > 1$ and $\vec{v} \in \mathbb{Z}^{k-1} \setminus \{\vec{0}\}$. Suppose that $j \geq J(\vec{v})$. Then the **vector Zeckendorf greedy algorithm** for $\vec{v}$ proceeds as follows.*

- (1) Apply $S_{j+1}(\vec{v}) = \vec{v} \cdot (x_j, \dots, x_{j+1-(k-1)}) \pmod{x_{j+1}}$.
- (2) Use the k -bonacci number greedy algorithm to write $S_{j+1}(\vec{v}) = \sum_{i=1}^j c_i x_{j+1-i} \pmod{x_{j+1}}$, where $c_i \in 0, 1$ and no k consecutive c_i 's are equal to 1.
- (3) Output the representation $\sum_{i=1}^j c_i \vec{X}_{-i}$.

Thus, our algorithmic approaches rely on finding efficient methods for obtaining a value of j . The following lemma allows us to compute a value for j from any k -bonacci vector decomposition of $\vec{v}$.

Lemma 1.14. *Suppose $\vec{v} = \sum_{i=1}^N \vec{X}_{-n_i}$ is any k -bonacci decomposition of $\vec{v}$ (not necessarily an SR). Then*

$$k(N-1) + \max_{1 \leq i \leq N} n_i \geq J(\vec{v}), \quad (1.1)$$

so we can use $j = k(N-1) + \max_i n_i$ in the vector Zeckendorf greedy algorithm.

Proof. The proof follows from inductive use of Lemma 1.9. □

The different values for j arise from different ways of constructing a k -bonacci decomposition of $\vec{v}$. The first relies on taking a series of small steps, consisting of

a combination of $\vec{\mathbf{X}}_{-k}$ and the standard basis vectors $\{\vec{\mathbf{X}}_i\}_{i=1}^{k-1}$, and obtains a larger upper bound on j more quickly.

Definition 1.15 (Small Steps Bound for j). *Fix $\vec{\mathbf{v}} = (v_1, v_2, \dots, v_{k-1}) \in \mathbb{Z}^{k-1} \setminus \{\vec{\mathbf{0}}\}$. Proceed according to the following cases.*

- (1) *If $\vec{\mathbf{v}}$ contains some entry $v_i < 0$, then let v_m be the maximally negative entry of $\vec{\mathbf{v}}$. First, take v_m steps in direction $\vec{\mathbf{X}}_{-k} = (-1, -1, \dots, -1)$; then, take $v_j - v_m$ steps in direction $\vec{\mathbf{X}}_{-j} = \vec{\mathbf{e}}_j$ for each $j \in \{1, \dots, k\}$.*
- (2) *If $\vec{\mathbf{v}}$ contains no negative entries, then take v_j steps in direction $\vec{\mathbf{X}}_{-j}$ for each $j \in \{1, \dots, k\}$.*

This algorithm gives rise to the following bound on j .

Theorem 1.16. *Let $\vec{\mathbf{v}} = (v_1, v_2, \dots, v_{k-1}) \in \mathbb{Z}^{k-1} \setminus \{\vec{\mathbf{0}}\}$. If $v_i < 0$ for some i , then let v_m be the negative entry of $\vec{\mathbf{v}}$ that is largest in absolute value. Let j_{ssb} be defined by*

$$j_{ssb} := \begin{cases} |v_m|k + \sum_{i=1}^{k-1} (v_i - v_m)k & \text{if } v_i < 0 \text{ for some } i \\ k \sum_{i=1}^{k-1} v_i - 1 & \text{if } v_i \geq 0 \text{ for all } i. \end{cases}$$

Then $j_{ssb} \geq J(\vec{\mathbf{v}})$.

We verify in Proposition 2.5 that this approach runs in $O(k^3 \|\vec{\mathbf{v}}\|_\infty)$ steps. We can make the bound on j given by the left-hand side in (1.1) smaller by taking fewer steps. This motivates a second approach, in which each step is chosen to bring us as close as possible to $\vec{\mathbf{v}}$. Formally, this is given below.

Definition 1.17 (Large Steps Bound for j). *Fix $\vec{\mathbf{v}} \in \mathbb{Z}^{k-1}$. Let $\vec{\mathbf{v}}_1 = \vec{\mathbf{v}}$, and given $\vec{\mathbf{v}}_i$, take $n_i \in \mathbb{N}$ such that $\|\vec{\mathbf{v}}_i - \vec{\mathbf{X}}_{-n_i}\|_2$ is minimal, and let $\vec{\mathbf{v}}_{i+1} = \vec{\mathbf{v}}_i - \vec{\mathbf{X}}_{-n_i}$. Stop when $i = M + 1$ and $\|\vec{\mathbf{v}}_{M+1}\|_2 \geq \|\vec{\mathbf{v}}_M\|_2$. Then, use the algorithm given in the small steps bound to write $\vec{\mathbf{v}}_M = \sum_{i=M}^N \vec{\mathbf{X}}_{-n_i}$ with $n_i \in \{1, \dots, k\}$. This gives a decomposition $\vec{\mathbf{v}} = \sum_{i=1}^N \vec{\mathbf{X}}_{-n_i}$.*

We use Lemma 1.14 to compute

$$j_{lsb} := k(N - 1) + \max_i n_i \geq J(\vec{\mathbf{v}}).$$

When $k = 3$, the large steps bound allows us to obtain a logarithmic bound for j .

We provide Python code carrying out these algorithms in the following Colab file:

<https://colab.research.google.com/drive/1MNQNCxUA83AshL2vjkcXCuvpVs8Kw1kH?usp=sharing>.

Theorem 1.18. *Fix $k = 3$. There exist $c_3, d_3 > 0$ such that for all $\vec{\mathbf{v}} \in \mathbb{Z}^2 \setminus \{\vec{\mathbf{0}}\}$,*

$$j_{lsb} \leq c_3 \log \|\vec{\mathbf{v}}\|_2 + d_3.$$

Hence, we have $j_{lsb} = O(\log \|\vec{\mathbf{v}}\|_2) = O(\log \|\vec{\mathbf{v}}\|_\infty)$ by the Lipschitz equivalence of L^p norms.

We conjecture based on strong computational evidence that this result may be extended to all $k \geq 3$.

Conjecture 1.19. *Fix $k \geq 3$. There exist $c_3, d_3 > 0$ such that for all $\vec{v} \in \mathbb{Z}^2 \setminus \{\vec{0}\}$,*

$$j_{lsb} \leq c_3 \log \|\vec{v}\|_2 + d_3.$$

Hence, we have $j_{lsb} = O(\log \|\vec{v}\|_2) = O(\log \|\vec{v}\|_\infty)$ by the Lipschitz equivalence of L^p norms.

In Section 3, we turn our attention to the number of and gaps between summands in k -bonacci vector Zeckendorf representations. We show that they exhibit the same properties as those in the one-dimensional case.

Definition 1.20. *For $n \geq 0$, let $D_n := \{\vec{v} \in \mathbb{Z}^{k-1} : J(\vec{v}) \leq n\}$.*

Definition 1.21. *For $n > 0$, consider the discrete outcome space $D_n \setminus D_{n-1}$ with probability measure*

$$\mathbb{P}_n(A) = \frac{|A|}{x_{n+2} - x_{n+1}}, \quad A \subset D_n \setminus D_{n-1}.$$

Note $|D_n \setminus D_{n-1}| = x_{n+2} - x_{n+1}$, so this is the uniform measure. We define the random variable K_n by setting $K_n(\vec{v})$ equal to the number of summands of $\vec{v} \in D_n \setminus D_{n-1}$ in its SR.

Definition 1.22. *Let $\vec{v} \in D_n \setminus D_{n-1}$, and let $\kappa(\vec{v})$ denote the number of summands in the SR of $\vec{v}$. Let $\{r_j\}_{j=1}^{\kappa(\vec{v})}$ be the set of indices present in the SR of $\vec{v}$.*

- Spacing gap measure: *We define the spacing gap measure of $\vec{v}$ by*

$$\nu_{\vec{v};n}(x) := \frac{1}{\kappa(\vec{v}) - 1} \sum_{j=2}^{\kappa(\vec{v})} \delta(x - (r_j - r_{j-1})), \quad (1.2)$$

where δ is the Dirac delta functional.³ We do not include the gap to the first summand, as this is not a gap between summands; for almost all $\vec{v}$, one extra gap is negligible in the limit.

- Average spacing gap measure: *The SR of $\vec{v}$ has $\kappa(\vec{v}) - 1$ gaps. Thus the total number of gaps for all $\vec{v} \in D_n \setminus D_{n-1}$ is*

$$N_{\text{gaps}}(n) := \sum_{\vec{v} \in D_n \setminus D_{n-1}} (\kappa(\vec{v}) - 1). \quad (1.3)$$

³Thus for any continuous function f we have $\int_{-\infty}^{\infty} f(x)\delta(x-a)dx = f(a)$; we may view $\delta(x-a)$ as representing a unit point mass concentrated at a .

We define the average spacing gap measure for all $\vec{v} \in D_n \setminus D_{n-1}$ by

$$\begin{aligned}\nu_n(x) &:= \frac{1}{N_{\text{gaps}}(n)} \sum_{\vec{v} \in D_n \setminus D_{n-1}} \sum_{j=2}^{\kappa(\vec{v})} \delta(x - (r_j - r_{j-1})) \\ &= \frac{1}{N_{\text{gaps}}(n)} \sum_{\vec{v} \in D_n \setminus D_{n-1}} (\kappa(\vec{v}) - 1) \nu_{\vec{v};n}(x).\end{aligned}\quad (1.4)$$

If $P_n(l)$ is the probability of getting a gap of length l among all gaps from the decompositions of all $\vec{v} \in D_n \setminus D_{n-1}$, then

$$\nu_n(x) = \sum_{l=0}^{n-1} P_n(l) \delta(x - l). \quad (1.5)$$

Theorem 1.23. Let K_n be the random variable of Definition 1.21 and denote its mean by μ_n . Then there exist constants $C_{\text{Lek}} > 0$ (the constant in the Generalized Lekkerkerker's Theorem for PLRS's [10, Theorem 1.2 (Generalized Lekkerkerker)]), d , and $\gamma_1 \in (0, 1)$ depending only on k such that

$$\mu_n = C_{\text{Lek}}n + d + o(\gamma_1^n).$$

The mean μ_n and variance σ_n^2 of K_n grow linearly in n , and $(K_n - \mu_n)/\sigma_n$ converges weakly to the standard normal $N(0, 1)$ as $n \rightarrow \infty$.

The following is a standard lemma which helps us state a result on the distribution of gaps.

Lemma 1.24 (Generalized Binet's Formula). Let $\lambda_1, \dots, \lambda_k$ be the roots of the characteristic polynomial for the k -bonacci sequence, which is

$$p(x) := x^k - x^{k-1} - \dots - x - 1 = 0. \quad (1.6)$$

Order the roots so that $|\lambda_1| \geq \dots \geq |\lambda_k|$. Then $\lambda_1 > 1$ is the unique positive real root, and there exist constants such that

$$x_{n+1} = a_1 \lambda_1^n + O(n^{k-2} \lambda_2^n). \quad (1.7)$$

Theorem 1.25. Let $\lambda_1 > 1$ denote the largest root (in absolute value) of the characteristic polynomial for the k -bonacci sequence, and let a_1 be the leading coefficient in the generalized Binet expansion. Let $P_n(l)$ be the probability of having a gap of length l among the decompositions of $\vec{v} \in D_n \setminus D_{n-1}$, and let $P(l) = \lim_{n \rightarrow \infty} P_n(l)$. Then

$$P(l) = \begin{cases} 0 & \text{if } l = 0 \\ \lambda_1^{-1} \left(\frac{1}{C_{\text{Lek}}} \right) (\lambda_1(1 - 2a_1) + a_1) & \text{if } l = 1 \\ (\lambda_1 - 1)^2 \left(\frac{a_1}{C_{\text{Lek}}} \right) \lambda_1^{-l} & \text{if } l \geq 2. \end{cases} \quad (1.8)$$

In particular, the probability of having a gap of length $l \geq 2$ decays geometrically, with decay constant the largest root of the characteristic polynomial.

We further extend the study of summand minimality to the multidimensional case, giving the following.

Theorem 1.26. *Let $\vec{v} \in \mathbb{Z}^{k-1}$. Then $SR(\vec{v}) = \sum_{i \geq 1} c_i \vec{X}_{-i}$ is summand minimal: that is, there is no way to write $\vec{v}$ as a linear combination of k -bonacci vectors with nonnegative integer coefficients using strictly fewer terms.*

2. Improved Algorithms for Finding Vector Representations

In this section, we fix a choice of $k > 1$ and $\vec{v} \in \mathbb{Z}^{k-1} \setminus \{\vec{0}\}$, noting that the SR of $\vec{0}$ is the empty representation. We provide two algorithms to compute the SR of $\vec{v}$ which improve the algorithmic approach of [1]. We first show how to use the greedy algorithm to find the vector decomposition of $\vec{v}$ for a specific j .

Lemma 2.1. *Let $k > 1$ and $\vec{v} \in \mathbb{Z}^{k-1} \setminus \{\vec{0}\}$. Let $S_{j+1}(\vec{v}) = \sum_{i=1}^j c_i x_{j+1-i} \pmod{x_{j+1}}$ be the output of the vector Zeckendorf greedy algorithm for $\vec{v}$. Then $\sum_{i=1}^j c_i \vec{X}_{-i}$ is the SR of $\vec{v}$.*

Proof. This follows because the map S_{j+1} is injective on the set of vectors with SRs whose indices are bounded by j . $\square$

2.1. Small Steps Algorithm

Let $k > 1$ and $\vec{v} \in \mathbb{Z}^{k-1} \setminus \{\vec{0}\}$. Lemma 2.1 requires us to have a value of j before using the vector Zeckendorf greedy algorithm to generate the SR of $\vec{v}$. We verify the small steps bound on j by writing $\vec{v}$ as a linear combination of $\{\vec{X}_{-i}\}_{i=1}^k$ and applying Lemma 1.14.

Proof of Theorem 1.16. First, suppose that $v_i < 0$ for some $i \in \{1, \dots, k-1\}$. Note that for any choice of $k > 1$, $\vec{X}_{-k} = (-1, -1, \dots, -1)$. Hence we can write

$$\vec{v} = |v_m| \vec{X}_{-k} + \sum_{i=1}^{k-1} (v_i - v_m) \vec{X}_{-i}.$$

By Lemma 1.14, we have

$$J(\vec{v}) \leq |v_m|k + \sum_{i=1}^{k-1} (v_i - v_m)k = j_{ssb},$$

Otherwise, $v_i \geq 0$ for every $i \in \{1, \dots, k-1\}$. In this case, we write

$$\vec{v} = \sum_{i=1}^{k-1} v_i \vec{X}_{-i}.$$

Again, by Lemma 1.14,

$$J(\vec{v}) \leq k \left(\sum_{i=1}^{k-1} v_i - 1 \right) + k - 1 = k \sum_{i=1}^{k-1} v_i - 1 = j_{ssb}.$$

□

Remark 2.2. The bound j_{ssb} generated by the small steps algorithm is larger than necessary even for small k and vectors with relatively small integer entries. The following examples illustrate this.

Example 2.3. Consider $\vec{v} = (2, -2)$. As $k = 3$, we get $j = 2(3) + 4(3) = 18$. Then

$$S_{19}(2, -2) \equiv 2(19513) - 2(10609) \pmod{35890} \equiv 17808 \pmod{35890}.$$

The k -bonacci number greedy algorithm gives the decomposition

$$\begin{aligned} 17808 &= 10609 + 5768 + 927 + 504 \\ &= x_{17} + x_{16} + x_{13} + x_9 \\ &= x_{19-2} + x_{19-3} + x_{19-6} + x_{19-7}. \end{aligned}$$

Then the SR of $(2, -2)$ is $\vec{X}_{-2} + \vec{X}_{-3} + \vec{X}_{-6} + \vec{X}_{-7} = (0, 1) + (-1, -1) + (-2, -3) + (5, 1)$.

Example 2.4. Consider $\vec{v} = (3, 0)$. As $k = 3$, we get $j = 3 \cdot 3 - 1 = 8$. Then $S_9(3, 0) = 3 \cdot 44 \pmod{81} = 51$. The k -bonacci number greedy algorithm gives the decomposition

$$\begin{aligned} 51 &= 44 + 7 \\ &= x_8 + x_5 \\ &= x_{9-1} + x_{9-4}. \end{aligned}$$

So the SR of $(3, 0)$ is $\vec{X}_{-1} + \vec{X}_{-4} = (1, 0) + (2, 0)$.

The following proposition is proved in Appendix A.

Proposition 2.5. The time complexity of the small steps algorithm is $O(k^3 \|\vec{v}\|_\infty)$.

The time complexity of the algorithm in [1] is $O(k \cdot L \cdot \|\vec{v}\|_1)$ (see Appendix A), where L is the number of summands in the decomposition of $\vec{v}$. For fixed k , this is $O(\|\vec{v}\|_\infty \log \|\vec{v}\|_\infty)$. As well as being slow, calculating the SR of $\vec{v}$ requires us to calculate all of the SRs along a path from $\vec{0}$ to $\vec{v}$.

The time complexity of the Small Steps Algorithm is $O(k^3 \|\vec{v}\|_\infty)$ [see Appendix A]. For fixed k , it is $O(\|\vec{v}\|_\infty)$. Although this is an improvement over the algorithm in [1], Example 2.3 demonstrates that we end up working with very large numbers, even for small k and $\vec{v}$ with small integer entries. This arises from the linear bound for $J(\vec{v})$. This motivates our second algorithm, which finds an improved bound on j .

2.2. Large Steps Algorithm

We first illustrate the Large Steps Algorithm with an example.

Example 2.6. Consider $\vec{v} = (2, -2)$. We let $\vec{v}_1 = (2, -2)$. The closest 3-bonacci vector to $\vec{v}$ is $\vec{X}_{-4} = (2, 0)$. Then we take $\vec{v}_2 = \vec{v}_1 - \vec{X}_{-4} = (0, -2)$. The closest 3-bonacci vector to $\vec{v}_2$ is $\vec{X}_{-3} = (-1, -1)$, so we set $\vec{v}_3 = \vec{v}_2 - \vec{X}_{-3} = (1, -1)$. The closest 3-bonacci vector to $\vec{v}_3$ is $\vec{x}_{-1} = (1, 0)$, so we set $\vec{v}_4 = \vec{v}_3 - \vec{x}_{-1} = (0, -1)$.

Now, subtracting off the closest 3-bonacci vector to $\vec{v}_4$ would not reduce the size, so we use the small steps algorithm to write $\vec{v}_4 = \vec{X}_{-3} + \vec{X}_{-2}$. This gives the decomposition

$$\vec{v} = \vec{X}_{-4} + 2\vec{X}_{-3} + 2\vec{X}_{-1},$$

and Lemma 1.14 tells us that

$$j_{lsb} = 4 + 3 \cdot 4 = 16.$$

We now proceed as before with the vector Zeckendorf greedy algorithm.

Example 2.7. Consider $\vec{v} = (3, 0)$. We let $\vec{v}_1 = (3, 0)$. The closest 3-bonacci vector to $\vec{v}$ is $\vec{X}_{-4} = (2, 0)$, so we set $\vec{v}_2 = \vec{v}_1 - \vec{X}_{-4} = (1, 0) = \vec{X}_{-1}$. This gives the decomposition

$$\vec{v} = \vec{X}_{-4} + \vec{X}_{-1},$$

which translates to

$$j_{lsb} = 4 + 3 \cdot 1 = 7$$

by Lemma 1.14. We then proceed with the vector Zeckendorf greedy algorithm.

2.3. Analysis of Large Steps Algorithm

We now utilize a geometric argument to prove that the value j_{lsb} provided by the large step algorithm is logarithmic in $\|\vec{v}\|_\infty$ for $k = 3$. We conjecture that this is true for all k .

Proof of Theorem 1.18 . We prove the following statements.

- (1) There exist natural numbers P and M such that for any sequence of P consecutive 3-bonacci vectors starting at $\vec{X}_{-n}$ for $n \geq M$, and any wedge in $\mathbb{R}^2$ centered about $\vec{0}$ with angle $2\pi/3$, one of the 3-bonacci vectors lies in the wedge.
- (2) There exist $\lambda < 1$ and $R > 0$ such that for any $\vec{v} \in \mathbb{Z}^2$ with $\|\vec{v}\|_2 \geq R$, there exists $\vec{X}_{-n}$ with

$$\left\| \vec{v} - \vec{X}_{-n} \right\|_2 \leq \lambda \|\vec{v}\|_2.$$

- (3) There exist constants c_3 and d_3 such that

$$j_{lsb} \leq c_3 \log \|\vec{v}\|_2 + d_3.$$

Proof of (1): We claim that $P = M = 9$ are sufficient for $k = 3$. We first rewrite the 3-bonacci vectors in $\mathbb{C}$, letting $Z_{-n} = \Phi(X_{-n})$, where $\Phi(v_1, v_2) = v_1 + iv_2$. Then Z_{-n} satisfies the recurrence relation $Z_{-n} = Z_{-(n-3)} - Z_{-(n-2)} - Z_{-(n-1)}$ for $n > 2$ with

$Z_0 = 0$, $Z_{-1} = 1$, and $Z_{-2} = i$. Then, we solve the recurrence relation to get

$$Z_{-n} = Ar^n e^{in\theta} + Br^n e^{-in\theta} + C\epsilon^n,$$

where

$$\begin{aligned} r &= 1.3562, \\ \theta &= 2.1762, \\ \epsilon &= 0.5437, \\ A &= -0.4578 - 0.3103i, \\ B &= -0.0612 - 0.0259i, \\ C &= 0.5190 + 0.3362i, \end{aligned}$$

each to 4 decimal places.

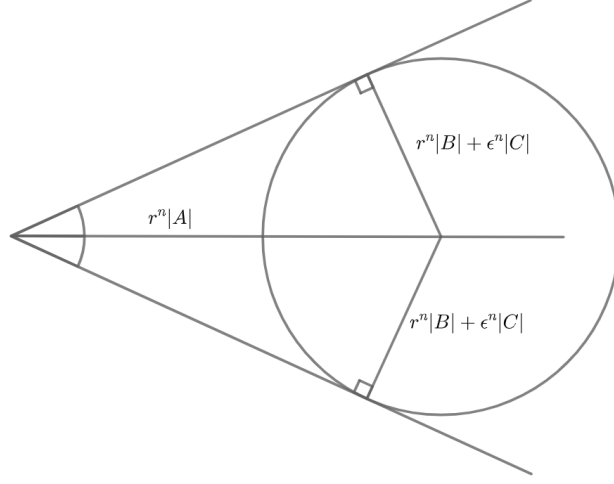


Figure 2.1. A figure displaying a circle in which Z_{-n} must fall. We use this to approximate the argument of Z_{-n} .

Figure 2.1 shows that the difference in angle between Z_{-n} and $Ae^{in\theta}$ is at most $\arcsin\left(\frac{|B|+|C|(\epsilon/r)^n}{|A|}\right)$, which decreases monotonically in n to

$$\arcsin \frac{|B|}{|A|} \approx 0.1204 \quad (\text{to 4 decimal places}).$$

For $n \geq 9$, we have

$$\arcsin\left(\frac{|B|+|C|(\epsilon/r)^n}{|A|}\right) \leq 0.121.$$

Hence, for $n, m \geq 9$, we have

$$\arg(Z_{-m}) - \arg(Z_{-n}) \in ((m-n)\theta - 0.242, (m-n)\theta + 0.242). \quad (2.1)$$

Consider for $n \geq 9$, the numbers $Z_{-n}, Z_{-(n+1)}, \dots, Z_{-(n+8)}$. To show that every cone with semiangle $\pi/3$ contains one of these numbers, we show that the angle between every two adjacent lines in Figure 2.2 is less than $2\pi/3$.

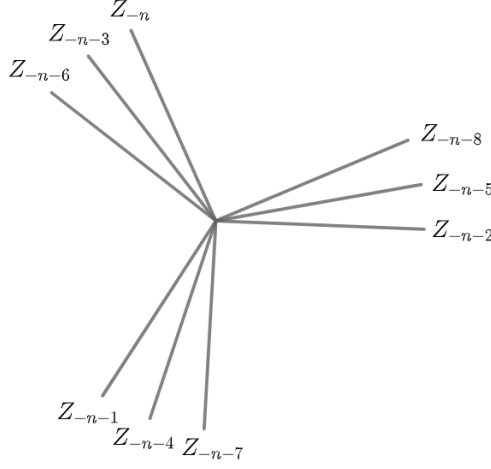


Figure 2.2. Nine consecutive Z_{-i} 's. We aim to show that the angle between each pair of adjacent lines is less than $2\pi/3$.

By (2.1), we have

$$\begin{aligned} \arg(Z_{-(n+6)}) - \arg(Z_{-n}) &\in (0.2490, 0.7331) \subseteq (0, 2\pi/3) \\ \arg(Z_{-(n+1)}) - \arg(Z_{-(n+6)}) &\in (1.4432, 1.9273) \subseteq (0, 2\pi/3) \\ \arg(Z_{-(n+7)}) - \arg(Z_{-(n+1)}) &\in (0.2490, 0.7331) \subseteq (0, 2\pi/3) \\ \arg(Z_{-(n+2)}) - \arg(Z_{-(n+7)}) &\in (1.4432, 1.9273) \subseteq (0, 2\pi/3) \\ \arg(Z_{-(n+8)}) - \arg(Z_{-(n+2)}) &\in (0.2490, 0.7331) \subseteq (0, 2\pi/3) \\ \arg(Z_{-n}) - \arg(Z_{-(n+8)}) &\in (1.1976, 1.6817) \subseteq (0, 2\pi/3). \end{aligned}$$

This completes the proof of (1).

Proof of (2): By the triangle inequality, we have

$$|A|r^n - |B|r^n - |C|\epsilon^n \leq \|\vec{\mathbf{X}}_{-n}\|_2 \leq |A|r^n + |B|r^n + |C|\epsilon^n.$$

This shows that $\|\vec{\mathbf{X}}_{-n}\|_2$ is increasing for $n > 1$.

For $\vec{\mathbf{v}}$ with $\|\vec{\mathbf{v}}\|_2 \geq 2(|A|r^{P+M-1} + |B|r^{P+M-1} + |C|\epsilon^{P+M-1}) = 682496$, let X_{-n} be the smallest 3-bonacci vector such that the upper bound of its norm is greater than $\frac{1}{2}\|\vec{\mathbf{v}}\|_2$. Consider some 3-bonacci vector $\vec{\mathbf{X}}_{-n'}$ (with $N \leq n - P \leq n' \leq n - 1$) that

lies in the cone with axis passing through $\vec{v}$ and semiangle $\pi/3$. We have

$$\begin{aligned}
\|\vec{X}_{-n'}\|_2 &\geq |A|r^{n'} - |B|r^{n'} - |C|\epsilon^{n'} \\
&\geq \frac{|A|r^{n'} - |B|r^{n'} - |C|\epsilon^{n'}}{|A|r^n + |B|r^n + |C|\epsilon^n} \frac{1}{2} \|\vec{v}\|_2 \\
&\geq \frac{|A|r^{n-P} - |B|r^{n-P} - |C|\epsilon^{n-P}}{|A|r^n + |B|r^n + |C|\epsilon^n} \frac{1}{2} \|\vec{v}\|_2 \\
&\geq \frac{|A|r^M - |B|r^M - |C|\epsilon^M}{|A|r^{M+P} + |B|r^{M+P} + |C|\epsilon^{M+P}} \frac{1}{2} \|\vec{v}\|_2,
\end{aligned}$$

since the fraction in the penultimate line is increasing in n and $n \geq M + P$. Plugging in our values of A, B, C, r, ϵ and rounding down, we have

$$\|\vec{X}_{-n'}\|_2 \geq \frac{1}{2} \gamma \|\vec{v}\|_2,$$

where $\gamma = 7.714 \times 10^{-4}$. This shows that the region E in Figure 2.3 contains a 3-bonacci vector. We therefore bound the distance from $\vec{v}$ to $\vec{X}_{-n'}$ by the distance from $\vec{v}$ to the furthest point from $\vec{v}$ in E . Hence, we have that

$$\begin{aligned}
\|\vec{v} - \vec{X}_{-n'}\|_2 &\leq \sqrt{\left(\frac{\sqrt{3}}{2} \|\vec{v}\|_2\right)^2 + \left(\frac{1}{2}(1 - \gamma) \|\vec{v}\|_2\right)^2} \\
&\leq \frac{1}{2} \sqrt{3 + (1 - \gamma)^2} \|\vec{v}\|_2 \\
&= \lambda \|\vec{v}\|_2,
\end{aligned}$$

where $\lambda = 0.9998073$.

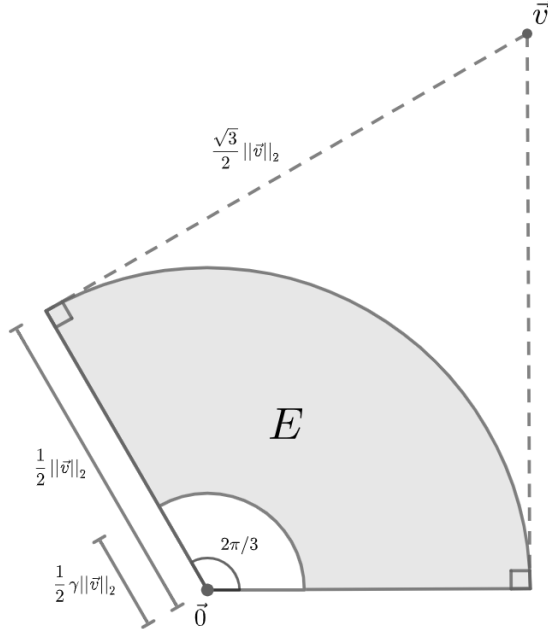


Figure 2.3. Figure showing a region E which is guaranteed to contain a 3-bonacci vector.

Proof of (3): The large step algorithm gives a decomposition

$$\vec{v} = \sum_{i=1}^N \vec{X}_{-n_i}. \quad (2.2)$$

To prove that N is finite, from the previous statement we obtain

$$\|\vec{v}_i\|_2 \leq \lambda \|\vec{v}_{i-1}\|_2 \leq \dots \leq \lambda^{i-1} \|\vec{v}\|_2. \quad (2.3)$$

Now, for $k = 3$, at most the last two terms are from the case where no largest step exists, so $\vec{v}_{N-1}$ has norm at least 1, and none of the $\vec{v}_i$'s for $i < N - 1$ are $(-1, 0)$ or $(0, -1)$. Therefore, $1 \leq \lambda^{N-2} \|\vec{v}\|_2$, so we obtain the bound

$$N \leq \frac{\log(\|\vec{v}\|_2)}{\log(1/\lambda)} + 2. \quad (2.4)$$

By Lemma 1.14, we have

$$j_{lsb} \leq 3(N - 1) + \max_{i \in \{0, 1, \dots, N+1\}} n_i. \quad (2.5)$$

Finally, we bound the possible values of n_i that can appear in the decomposition of $\vec{v}$. Statement (2) ensures that $\|\vec{X}_{-n_i}\|_2 \leq 2 \|\vec{v}\|_2$, and we know by the triangle inequality that $\|\vec{X}_{-n_i}\|_2 \geq (|A| - |B|)r^{n_i} - |C|\epsilon^{n_i} \geq (|A| - |B|)r^{n_i} - |C|\epsilon$. Hence, $(|A| - |B|)r^{n_i} \leq 2 \|\vec{v}\|_2 + |C|\epsilon$, so

$$n_i \leq \frac{\log\left(\frac{2\|\vec{v}\|_2 + |C|\epsilon}{|A| - |B|}\right)}{\log r}. \quad (2.6)$$

Putting this all together, we have

$$\begin{aligned} j_{lsb} &\leq 3 \left(\frac{\log(\|\vec{v}\|_2)}{\log(1/\lambda)} + 1 \right) + \frac{\log\left(\frac{2\|\vec{v}\|_2 + |C|\epsilon}{|A| - |B|}\right)}{\log r} \\ &\leq 3 \left(\frac{\log(\|\vec{v}\|_2)}{\log(1/\lambda)} + 1 \right) + \frac{\log\left(\frac{2 + |C|\epsilon}{|A| - |B|} \|\vec{v}\|_2\right)}{\log r} \\ &= \left(\frac{3}{\log(1/\lambda)} + \frac{1}{\log r} \right) \log(\|\vec{v}\|_2) + \left(3 + \frac{\log(2 + |C|\epsilon) - \log(|A| - |B|)}{\log r} \right). \end{aligned}$$

Plugging values in for λ, r, A, B , and C , we have $c_3 \approx 15570$ and $d_3 \approx 5.018$, rounded up to 4 significant figures. $\square$

This proof not only shows the existence of constants c_3 and d_3 , but also gives us their values. However, the value of c_3 exhibited in this proof is too large for practical use. A computation over all vectors $\vec{v}$ with $\|\vec{v}\|_\infty \leq 100$ suggests that $c_3 = 15, d = 10$ is sufficient. We also support this with a scatter plot for 1000 randomly generated vectors with L^∞ norm at most 10000.

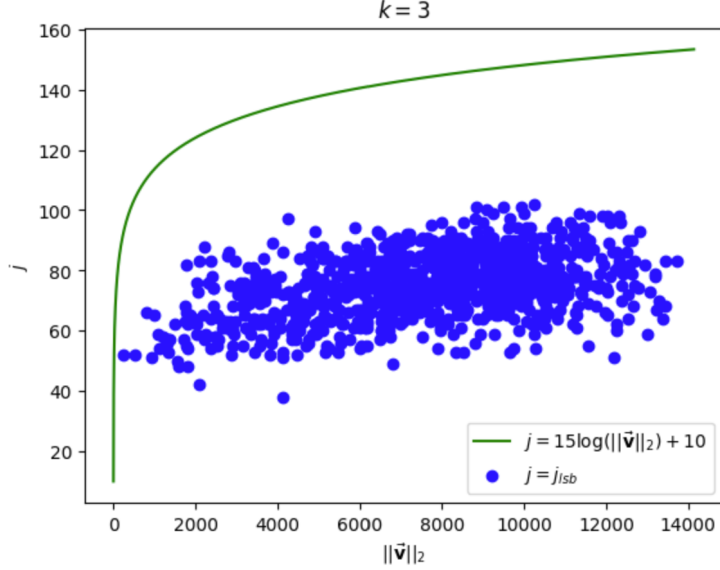


Figure 2.4. Scatter plot of j_{lsb} and the proposed upper bound with 1000 randomly generated vectors

We give the time-complexity of the algorithm in the next proposition.

Proposition 2.8. *For fixed k , the time complexity of the large step algorithm is $O(\|\vec{v}\|_\infty)$.*

We prove Proposition 2.8 in Appendix A.

2.4. Relative Efficiency of Algorithms for Finding SRs

In summary, we have two methods to find a value of j for the vector Zeckendorf greedy algorithm. For each fixed k , $j_{ssb} = O(\|\vec{v}\|_\infty)$, but assuming Conjecture 1.19, we have $j_{lsb} = O(\log \|\vec{v}\|_\infty)$. Considering k fixed, both methods have the same time-complexity, but the slowest part of each method is the greedy algorithm. We gain more insight into the running time by considering how long it takes to calculate j_{ssb} and j_{lsb} . The proofs in Appendix A show that the calculation of j_{ssb} takes $O(1)$ operations, and that the calculation of j_{lsb} takes $O((\log \|\vec{v}\|_\infty)^2)$ operations for fixed k .

3. Properties of Vector Representations

3.1. Overview

The one-dimensional k -Zeckendorf representation possesses many interesting properties. The number of summands in decompositions of integers in $[x_n, x_{n+1})$ converges to a Gaussian distribution, and the distribution of gaps between summands follows geometric decay. Furthermore, the decompositions exhibit summand minimality; that is, there is no way to express any nonnegative integer n as a linear combination of k -bonacci numbers with nonnegative integer coefficients using strictly fewer terms than the k -Zeckendorf representation.

We show that vector Zeckendorf representations have extremely similar properties

(Theorems 1.23, 1.25, and 1.26). A useful strategy is to reduce the problem to the one-dimensional case by considering appropriate bijections between subsets of $\mathbb{Z}^{k-1}$ and subsets of $\mathbb{Z}_{\geq 0}$. An example of such a map is S_n in Definition 1.7. The following map, first observed by Anderson and Bicknell-Johnson [1], also provides valuable insight.

Remark 3.1. *The satisfying sequences $\{c_i\}$ in Theorem 1.5 are essentially the same as k -Zeckendorf representations for non-negative integers (as in Theorem 1.2). This gives a bijection between $\mathbb{Z}^{k-1}$ and $\mathbb{Z}_{\geq 0}$.*

Recall that $D_n = \{\vec{v} \in \mathbb{Z}^{k-1} : J(\vec{v}) \leq n\}$, as given in Definition 1.20. Then, the following result is a more precise statement of Remark 3.1.

Lemma 3.2. *Define $f : \mathbb{Z}^{k-1} \rightarrow \mathbb{Z}_{\geq 0}$ as follows. For any $\vec{v} \in \mathbb{Z}^{k-1}$, let $\vec{v} = \sum_{i \geq 1} a_i \vec{X}_{-i}$ be its (unique) SR. Let*

$$f(\vec{v}) := \sum_{i \geq 2} a_{i-1} x_i. \quad (3.1)$$

Then f is a bijection. For $n > 0$, the image of D_n under f is $[0, x_{n+2}) \cap \mathbb{Z}$.

Proof. Bijectivity follows from Theorem 1.2 and Theorem 1.5. For the final statement, note that the non-negative integers with a k -Zeckendorf decomposition of the form $\sum_{i=2}^{n+1} c_i x_i$ are exactly the ones in $[0, x_{n+2})$. $\square$

The k -bonacci numbers belong to the class of **Positive Linear Recurrence Sequences**, which have been studied extensively (see [3, 4, 7, 9]). For these sequences, the properties of decompositions are well-understood.

Definition 3.3. [4] *We say a sequence $\{H_n\}_{n=1}^{\infty}$ of positive integers is a **Positive Linear Recurrence Sequence (PLRS)** if the following properties hold.*

(1) Recurrence relation: *There are non-negative integers $L, c_1, \dots, c_L$ such that*

$$H_{n+1} = c_1 H_n + \dots + c_L H_{n+1-L},$$

with L, c_1 and c_L positive.

(2) Initial conditions: $H_1 = 1$, *and for $1 \leq n < L$ we have*

$$H_{n+1} = c_1 H_n + c_2 H_{n-1} + \dots + c_n H_1 + 1.$$

*We call a decomposition $\sum_{i=1}^m a_i H_{m+1-i}$ of a positive integer N (and the sequence $\{a_i\}_{i=1}^m$) **legal** if $a_1 > 0$, the other $a_i \geq 0$, and one of the following two conditions holds:*

- *We have $m < L$ and $a_i = c_i$ for $1 \leq i \leq m$.*
- *There exists $s \in \{0, \dots, L\}$ such that*

$$a_1 = c_1, \quad a_2 = c_2, \quad \dots, \quad a_{s-1} = c_{s-1} \quad \text{and} \quad a_s < c_s, \quad (3.2)$$

$a_{s+1}, \dots, a_{s+\ell} = 0$ for some $\ell \geq 0$, and $\{b_i\}_{i=1}^{m-s-\ell}$ (with $b_i = a_{s+\ell+i}$) is legal.

*If $\sum_{i=1}^m a_i H_{m+1-i}$ is a legal decomposition of N , we define the **number of summands** (of this decomposition of N) to be $a_1 + \dots + a_m$.*

Informally, a legal decomposition is one where we cannot use the recurrence relation to replace a linear combination of summands with another summand and the coefficient of each summand is appropriately bounded.

Remark 3.4. *The k -bonacci sequence $\{x_{n+1}\}_{n=1}^{\infty}$ is a PLRS with $L = k$ and $c_1 = \dots = c_L = 1$.*

3.2. Distribution of the Number of Summands

Lemma 3.2 relates the number of summands in vector Zeckendorf representations to those in k -Zeckendorf representations. Then, it is no surprise that the distribution of the number of summands converges to a Gaussian.

Proof of Theorem 1.23. Follows from [10, Theorems 2 and 3]. $\square$

See [9, Section 4] for expressions characterizing the mean of the Gaussian for a general PLRS. For the case of vector Zeckendorf representations, we also propose an alternative way to find both the mean and the variance of the Gaussian by working with binary words.

Definition 3.5. *Let $W_{n,k}$ be the set of binary words $w = w_1 \dots w_n \in \{0,1\}^n$ such that*

- (1) $w_1 = 1$ (the first bit is fixed to 1);
- (2) w contains no substring of k consecutive 1's.

Lemma 3.6. *For each $n \geq 1$ there exists a bijection*

$$\Phi_n : D_n \setminus D_{n-1} \longrightarrow W_{n,k}$$

such that, for every $\vec{v} \in D_n \setminus D_{n-1}$ with the SR coefficient vector

$$\varepsilon(\vec{v}) = (\varepsilon_n, \varepsilon_{n-1}, \dots, \varepsilon_1) \in \{0,1\}^n,$$

$$\varepsilon_j = \begin{cases} 1, & \text{if the } j^{\text{th}} \text{ summand appears in the SR of } \vec{v}, \\ 0, & \text{otherwise,} \end{cases}$$

we have $\Phi_n(\vec{v}) = w$, where $w_j = \varepsilon_j$ for $1 \leq j \leq n$. The rules for the SR imply $\varepsilon_n = 1$ and no k consecutive 1's appear in $(\varepsilon_n, \dots, \varepsilon_1)$, so $\Phi_n(\vec{v})$ is well-defined and is in $W_{n,k}$.

Proof. By definition of $D_n \setminus D_{n-1}$ and SR, each $\vec{v}$ has a unique coefficient vector $\varepsilon(\vec{v}) \in \{0,1\}^n$ indicating the presence or absence of each summand up to index n , with $\varepsilon_n = 1$.

The rules for the SR correspond to forbidding k consecutive 1's among the coefficients. Mapping coefficients to bits produces a well-defined $w \in W_{n,k}$.

The map is injective by the uniqueness of the SR, and surjective by reversing the construction: read a word $w \in W_{n,k}$ as a valid coefficient vector, then form the SR. $\square$

Definition 3.7. For $n \geq 1$, we define the uniform probability measure on $W_{n,k}$:

$$\mathbb{P}_n^{(W)}(A) = \frac{|A|}{|W_{n,k}|}, \quad A \subseteq W_{n,k}.$$

Define the random variable $X_n : W_{n,k} \rightarrow \mathbb{N}$ as

$$X_n(w) = \text{the number of 1's in } w, \quad w \in W_{n,k}.$$

Using the bijection Φ_n from Lemma 3.6 and the uniform measure $\mathbb{P}_n$ on $D_n \setminus D_{n-1}$ from Definition 1.20, the pushforward measure $(\Phi_n)_* \mathbb{P}_n$ equals $\mathbb{P}_n^{(W)}$. In particular, X_n and K_n have the same distribution:

$$\mathbb{P}_n^{(W)}(X_n = t) = \mathbb{P}_n(K_n = t).$$

Counting binary words that avoid a pattern of k consecutive 1's (also known as the substring 1^k) is a classical problem in combinatorics (see, for example, [6, Ch. I]); our case only differs slightly in that the first bit is fixed to be 1.

We state a generating function for binary words that avoid 1^k , in which x marks the length and y marks the number of 1's. We write

$$F_k(x, y) = \sum_{n \geq 0} \sum_{m \geq 0} a_{n,m} x^n y^m,$$

where $a_{n,m}$ counts binary words of length n with exactly m total 1's and with no substring 1^k .

Proposition 3.8 (From [2, Prop. 1]). *For every fixed $k \geq 2$, the generating function for binary words avoiding 1^k is*

$$F_k(x, y) = \frac{y(1 - (xy)^k)}{y - xy^2 - xy + (xy)^{k+1}}.$$

Here x marks each letter and y marks each occurrence of the letter 1.

We now pass from the setting of Proposition 3.8 to our model $W_{n,k}$ in Definition 3.5, where the first bit is fixed to 1. Define

$$F_k^{\text{fix}}(x, y) = \sum_{n \geq 1} \sum_{m \geq 1} b_{n,m} x^n y^m,$$

where $b_{n,m}$ counts words in $W_{n,k}$ of length n and with exactly m ones. Thus $F_k^{\text{fix}}(x, y)$ is the generating function for our fixed-first-bit model, while $F_k(x, y)$ is the generating function for the unrestricted-first-bit model (which also includes the empty word).

Proposition 3.9. *For every fixed $k \geq 2$, the generating function for binary words that avoid 1^k and have 1 fixed as the first bit is*

$$F_k^{\text{fix}}(x, y) = (1 - x) F_k(x, y) - 1,$$

where $F_k(x, y)$ is the same as in Proposition 3.8. Equivalently,

$$F_k^{\text{fix}}(x, y) = \frac{(1 - x)y(1 - (xy)^k)}{y - xy^2 - xy + (xy)^{k+1}} - 1.$$

Proof. Every word counted by $F_k(x, y)$ either starts with 0 or 1, or is the empty word, which has weight 1. Let $F_{k,0}(x, y)$ be the generating function for words starting with 0, and $F_{k,1}(x, y)$ for words starting with 1. Then $F_k(x, y)$ can be decomposed as

$$F_k(x, y) = 1 + F_{k,0}(x, y) + F_{k,1}(x, y).$$

Adding a leading 0 to any word that avoids 1^k results in a word that still avoids 1^k and contributes weight x while contributing no factor of y . Similarly, removing the first bit from any word that starts with 0 results in a word that avoids 1^k . Therefore

$$F_{k,0}(x, y) = x F_k(x, y).$$

By definition, $F_k^{\text{fix}}(x, y)$ is the generating function of words that avoid 1^k and start with 1, so

$$F_k^{\text{start}=1}(x, y) = F_k^{\text{fix}}(x, y).$$

Hence

$$F_k(x, y) = 1 + x F_k(x, y) + F_k^{\text{fix}}(x, y),$$

leading to

$$F_k^{\text{fix}}(x, y) = (1 - x) F_k(x, y) - 1 = \frac{(1 - x)y(1 - (xy)^k)}{y - xy^2 - xy + (xy)^{k+1}} - 1.$$

□

Having found the generating function, we may derive the exact mean and variance of K_n . Throughout this derivation, let $[x^n]G(x)$ denote the coefficient of x^n in a series $G(x)$.

Remark 3.10. *It is standard in analytic combinatorics (see, for example, [6, Section III.2]) to identify combinatorial parameters with the random variables they induce under uniform sampling: a parameter χ counting a statistic in objects of length n becomes a random variable X_n when these objects are sampled uniformly. We cite relevant results involving parameters as if they were expressed in terms of random variables.*

The following result is a consequence of [6, Proposition III.2].

Proposition 3.11. *Let $F(x, y) = \sum_{n \geq 0} \sum_{m \geq 0} f_{n,m} x^n y^m$ be a generating function where $f_{n,m}$ counts objects of size n with parameter value m . For the random variable*

Y_n obtained from uniform sampling of size- n objects, with distribution

$$\mathbb{P}(Y_n = m) = \frac{f_{n,m}}{\sum_{j \geq 0} f_{n,j}},$$

The expectation and variance of Y_n are:

$$\mathbb{E}[Y_n] = \frac{[x^n] \frac{\partial F}{\partial y}(x, y)|_{y=1}}{[x^n] F(x, 1)}, \quad \text{Var}(Y_n) = \frac{[x^n] \left(\frac{\partial^2 F}{\partial y^2} + \frac{\partial F}{\partial y} \right)(x, y)|_{y=1}}{[x^n] F(x, 1)} - (\mathbb{E}[Y_n])^2.$$

From here we apply Proposition 3.11 for $Y_n = X_n$ and $F = F_k^{\text{fix}}$ from Proposition 3.9.

Theorem 3.12 (Exact mean and variance). *For every $k \geq 2$ and $n \geq 1$, let*

$$A_k(x) = F_k^{\text{fix}}(x, 1),$$

$$B_k(x) = \partial_y F_k^{\text{fix}}(x, y)|_{y=1}, \text{ and}$$

$$C_k(x) = (\partial_y^2 F_k^{\text{fix}} + \partial_y F_k^{\text{fix}})(x, y)|_{y=1}.$$

Then the mean and variance of X_n under the uniform distribution on $W_{n,k}$ can be calculated using the expressions we have just defined as follows:

$$\mathbb{E}[X_n] = \frac{[x^n] B_k(x)}{[x^n] A_k(x)}, \quad \text{Var}(X_n) = \frac{[x^n] C_k(x)}{[x^n] A_k(x)} - \left(\frac{[x^n] B_k(x)}{[x^n] A_k(x)} \right)^2.$$

The explicit forms of A_k, B_k, C_k are listed below:

$$A_k(x) = \frac{x - x^k}{(1 - x) \Delta_k(x)} = \frac{x - x^k}{1 - 2x + x^{k+1}},$$

$$B_k(x) = \frac{1}{((1 - x) \Delta_k(x))^2} \left[(x - kx^k)(1 - x) \Delta_k(x) - (x - x^k) \left(-x \Delta_k(x) - (1 - x) \sum_{r=1}^{k-1} r x^{r+1} \right) \right],$$

$$\begin{aligned}
C_k(x) = & \frac{1}{((1-x)\Delta_k(x))^3} \left[(-k(k-1)x^k) ((1-x)\Delta_k(x))^2 \right. \\
& - (x-x^k) \left((1-x) \left(-\sum_{r=2}^{k-1} r(r-1)x^{r+1} \right) + 2x \sum_{r=1}^{k-1} r x^{r+1} \right) (1-x)\Delta_k(x) \\
& - 2(x-kx^k) (1-x)\Delta_k(x) \left(-x\Delta_k(x) - (1-x) \sum_{r=1}^{k-1} r x^{r+1} \right) \\
& \left. + 2(x-x^k) \left(-x\Delta_k(x) - (1-x) \sum_{r=1}^{k-1} r x^{r+1} \right)^2 \right] + B_k(x),
\end{aligned}$$

where

$$\Delta_k(x) = 1 - \sum_{j=1}^k x^j.$$

Proof. By Proposition 3.9,

$$F_k^{\text{fix}}(x, y) = \frac{xy(1-(xy)^{k-1})}{(1-xy)\Delta_k(x, y)}, \quad \Delta_k(x, y) = 1 - x - x^2y - \dots - x^ky^{k-1}.$$

To make this simpler, set

$$N(x, y) = xy - x^ky^k, \quad V(x, y) = (1-xy)\Delta_k(x, y).$$

From here we can write F_k as $F_k^{\text{fix}} = N/V$.

Now we evaluate the derivatives at $y = 1$:

$$N(x, 1) = x - x^k, \quad \frac{\partial N}{\partial y}(x, y) = x - kx^ky^{k-1} \Rightarrow \frac{\partial N}{\partial y}(x, 1) = x - kx^k,$$

$$\frac{\partial^2 N}{\partial y^2}(x, y) = -k(k-1)x^ky^{k-2} \Rightarrow \frac{\partial^2 N}{\partial y^2}(x, 1) = -k(k-1)x^k.$$

For the denominator,

$$V(x, 1) = (1-x)\Delta_k(x),$$

$$\frac{\partial V}{\partial y}(x, y) = -x\Delta_k(x, y) + (1-xy)\frac{\partial \Delta_k}{\partial y}(x, y),$$

so that

$$\frac{\partial \Delta_k}{\partial y}(x, y) = -x^2 - 2x^3y - \dots - (k-1)x^ky^{k-2},$$

$$\frac{\partial V}{\partial y}(x, 1) = -x \Delta_k(x) - (1 - x) \sum_{r=1}^{k-1} r x^{r+1}.$$

Differentiating again,

$$\frac{\partial^2 V}{\partial y^2}(x, y) = (1 - xy) \frac{\partial^2 \Delta_k}{\partial y^2}(x, y) - 2x \frac{\partial \Delta_k}{\partial y}(x, y),$$

with

$$\frac{\partial^2 \Delta_k}{\partial y^2}(x, y) = -2x^3 - 3 \cdot 2x^4y - \dots - (k-1)(k-2)x^k y^{k-3},$$

so that

$$\frac{\partial^2 V}{\partial y^2}(x, 1) = (1 - x) \left(- \sum_{r=2}^{k-1} r(r-1)x^{r+1} \right) + 2x \sum_{r=1}^{k-1} r x^{r+1}.$$

Now we compute A_k , B_k , C_k . By definition,

$$A_k(x) = \frac{N(x, 1)}{V(x, 1)} = \frac{x - x^k}{(1 - x) \Delta_k(x)},$$

which also equals $(x - x^k)/(1 - 2x + x^{k+1})$ since $(1 - x) \Delta_k(x) = 1 - 2x + x^{k+1}$.

For the first derivative, the quotient rule gives

$$\frac{\partial}{\partial y} \left(\frac{N}{V} \right) = \frac{\frac{\partial N}{\partial y} \cdot V - N \cdot \frac{\partial V}{\partial y}}{V^2}.$$

Evaluating this at $y = 1$ and substituting in the four expressions $N(x, 1)$, $\frac{\partial N}{\partial y}(x, 1)$, $V(x, 1)$, $\frac{\partial V}{\partial y}(x, 1)$ results in

$$\begin{aligned} B_k(x) &= \frac{1}{((1 - x) \Delta_k(x))^2} \left[(x - kx^k) (1 - x) \Delta_k(x) \right. \\ &\quad \left. - (x - x^k) \left(-x \Delta_k(x) - (1 - x) \sum_{r=1}^{k-1} r x^{r+1} \right) \right]. \end{aligned}$$

For the second derivative, we differentiate again:

$$\frac{\partial^2}{\partial y^2} \left(\frac{N}{V} \right) = \frac{\frac{\partial^2 N}{\partial y^2} \cdot V^2 - N \cdot \frac{\partial^2 V}{\partial y^2} \cdot V - 2 \frac{\partial N}{\partial y} \cdot V \cdot \frac{\partial V}{\partial y} + 2N \cdot \left(\frac{\partial V}{\partial y} \right)^2}{V^3}.$$

Evaluating at $y = 1$ and substituting $\frac{\partial^2 N}{\partial y^2}(x, 1)$, $\frac{\partial^2 V}{\partial y^2}(x, 1)$ together with the derived

expression gives:

$$\begin{aligned}
\frac{\partial^2 F_k^{\text{fix}}}{\partial y^2}(x, 1) &= \frac{1}{((1-x)\Delta_k(x))^3} \left[(-k(k-1)x^k) ((1-x)\Delta_k(x))^2 \right. \\
&\quad - (x-x^k) \left((1-x) \left(-\sum_{r=2}^{k-1} r(r-1)x^{r+1} \right) + 2x \sum_{r=1}^{k-1} r x^{r+1} \right) (1-x)\Delta_k(x) \\
&\quad - 2(x-kx^k) (1-x)\Delta_k(x) \left(-x\Delta_k(x) - (1-x) \sum_{r=1}^{k-1} r x^{r+1} \right) \\
&\quad \left. + 2(x-x^k) \left(-x\Delta_k(x) - (1-x) \sum_{r=1}^{k-1} r x^{r+1} \right)^2 \right].
\end{aligned}$$

By definition,

$$C_k(x) = \frac{\partial^2 F_k^{\text{fix}}}{\partial y^2}(x, 1) + B_k(x).$$

Applying Proposition 3.11 to $F = F_k^{\text{fix}}$ gives:

$$\mathbb{E}[X_n] = \frac{[x^n] B_k(x)}{[x^n] A_k(x)}, \quad \text{Var}(X_n) = \frac{[x^n] C_k(x)}{[x^n] A_k(x)} - \left(\frac{[x^n] B_k(x)}{[x^n] A_k(x)} \right)^2,$$

as was to be shown. $\square$

3.3. Distribution of Gaps Between Summands

Again, Lemma 3.2 allows us to reduce our analysis to the one-dimensional case. There are results on limiting gap probabilities, longest gaps, and much more (see [3, 4]).

The Proof of Theorem 1.25. follows from [3, Theorem 1.5] and [4, Theorem 1.5]. $\square$

3.4. Summand Minimality

We utilize the following definitions and result from [5].

Definition 3.13. For a PLRS given by the recurrence relation $H_{n+1} = c_1 H_n + \dots + c_L H_{n+1-L}$, we call $\sigma = (c_1, c_2, \dots, c_L)$ the **signature** of the PLRS.

As we observed in Remark 3.4, $\{x_{n+1}\}_{n=1}^\infty$ is a PLRS with $L = k$ and $c_1 = \dots = c_L = 1$. Kologlu et al. [7, Theorem 1.3] proved that every positive integer n has a unique legal decomposition associated to a given PLRS, called the **generalized Zeckendorf decomposition** of n . In the case of the k -bonacci sequence, this is the unique decomposition given by Theorem 1.2. We will say that a PLRS $\{H_n\}_{n=1}^\infty$ is **summand minimal** if no representation of any positive integer n as a linear combination with nonnegative coefficients of terms in $\{H_n\}_{n=1}^\infty$ uses fewer summands than the generalized Zeckendorf decomposition of n .

Theorem 3.14. [5, Theorem 1.1] A PLRS with signature $\sigma = (c_1, c_2, \dots, c_L)$ is summand minimal if and only if $c_1 \geq c_2 \geq \dots \geq c_L$.

Theorem 3.14 gives the following as an immediate corollary.

Corollary 3.15. The k -bonacci number decomposition of n given by Theorem 1.2 is summand minimal.

We are now able to prove the following.

Proof of Theorem 1.26. Consider any other representation $\vec{v} = \sum_{i \geq 1} c'_i \vec{X}_{-i}$ of $\vec{v}$ as a linear combination of k -bonacci vectors with nonnegative integer coefficients. Suppose the SR of $\vec{v}$ contains d summands and assume, for the sake of contradiction, that this representation $\vec{v} = \sum_{i \geq 1} c'_i \vec{X}_{-i}$ contains $\sum_{i \geq 1} c'_i = c < d$ summands. Let m be the maximal integer such that $\vec{X}_{-m}$ is present in one of the representations. By Lemma 1.8, we have

$$S_{m+1}\left(\sum_{i=1}^p c_i \vec{X}_{-i}\right) = \sum_{i=1}^p c_i x_{m+1-i} \pmod{x_{m+1}}$$

and

$$S_{m+1}\left(\sum_{i=1}^p c'_i \vec{X}_{-i}\right) = \sum_{i=1}^p c'_i x_{m+1-i} \pmod{x_{m+1}}.$$

Observe that $\sum_{i=1}^p c_i x_{m+1-i} \pmod{x_{m+1}}$ and $\sum_{i=1}^p c'_i x_{m+1-i} \pmod{x_{m+1}}$ contain, respectively, d and c summands. Now, by definition of the SR, $\sum_{i=1}^p c_i x_{m+1-i} \pmod{x_{m+1}}$ has no k consecutive terms and $c_i \in \{0, 1\}$ for every i , so $\sum_{i=1}^p c_i x_{m+1-i} \pmod{x_{m+1}}$ must be the k -bonacci number decomposition of $S_{m+1}(\vec{v})$. Then Lemma 3.15 implies $d \leq c$. $\square$

4. Future Work

A natural direction for further research concerns the behavior of the vector representation algorithm for general k . We proved that for $k = 3$, the number of steps required is logarithmic in the size of the vector, but for larger k , this remains conjectural (see Conjecture 1.19). Formally, it is conjectured that for all $k \geq 3$, there exist constants $c_k, d_k > 0$ such that the algorithm completes in at most $c_k \log |\mathbf{v}| + d_k$ steps. Proving this conjecture for higher k and understanding the underlying combinatorial mechanisms presents an important avenue for future research.

During our work, we also observed that the sets D_n increasingly resemble Rauzy fractals as n grows, displaying intricate self-similar structures. Despite this clear numerical and visual evidence, we were not able to provide a rigorous proof of their convergence. Establishing such a result, as well as understanding the precise geometric and combinatorial mechanisms behind this behavior, remains an interesting and challenging direction for future research.

Appendix A. Algorithm Complexity

In order to compare to the efficiency of our algorithms, we first provide the following complexity analysis of the recursive algorithm given in [1].

Proof of Lemma 1.11. The algorithm described in [1] follows a recursive approach to compute the Satisfying Representation (SR) of a vector $\vec{v}$. We distinguish two cases based on the signs of the coordinates of $\vec{v}$.

Recursive Algorithm:

- **Case 1:** If $\vec{v}$ has any positive coordinate, say $v_i > 0$, define $\vec{w} = \vec{v} - \vec{e}_i$. Then by definition,

$$SR(\vec{v}) = SR(\vec{w}) + \vec{e}_i.$$

Since $\vec{e}_i$ is a standard basis vector, this expression is either:

- a Nearly Satisfying Representation (NSR),
- a valid SR, or
- an “almost SR,” where the only violation is a single block of 1’s of length at most $2k - 1$.
- **Case 2:** If all coordinates of $\vec{v}$ are non-positive, define

$$\vec{w} = \vec{v} + \vec{X}_{-k} = \vec{v} - (\vec{e}_1 + \vec{e}_2 + \cdots + \vec{e}_{k-1}).$$

Then we have:

$$SR(\vec{v}) = SR(\vec{w}) + \vec{X}_{-k}.$$

The vector $\vec{w}$ has strictly larger entries than $\vec{v}$, so we recurse on $\vec{w}$. The resulting expression $SR(\vec{w}) + \vec{X}_{-k}$ is either a valid SR, or one of the following:

- an NSR, or
- an “almost SR” with a single block of 1’s of length at most $2k - 1$.

In either of these two cases, we proceed with a normalization step using the “borrow-carry” operation as described in Lemma 4.

When $\vec{v}$ has positive coordinates, each operation $\vec{v} \rightarrow \vec{v} - \vec{e}_i$ reduces the ℓ_1 norm of $\vec{v}$ by 1. Therefore, in the worst case, we perform this operation at most $O(\|\vec{v}\|_1)$ times. Appending $\vec{e}_i$ to the representation costs $O(1)$ per step. When we apply Lemma 4 to reduce blocks of 1’s (of length at most $2k - 1$), we know from the Lemma that a constant number (at most two) of “borrow-carry” operations suffices to restore the SR condition. Each such operation affects a block of length $O(k)$, so each normalization step takes $O(k)$ time. If the resulting expression is an NSR of length L (i.e., involving L summands), and we need to normalize it into an SR, then the total cost of normalization is $O(L \cdot k)$.

Assuming we begin with a vector $\vec{v} \in \mathbb{Z}^{k-1}$:

- We perform at most $O(\|\vec{v}\|_1)$ recursive steps.
- Converting each NSR to an SR costs $O(L \cdot k)$ time.

Therefore, the total worst-case time complexity is:

$$O(k \cdot L_{v_1} \cdot \|\vec{v}\|_1).$$

□

For comparison, we calculate the complexity of the small steps algorithm and the large steps algorithm.

Proof of Proposition 2.5. We list the steps of the algorithm.

- (i) Calculate $j = j_{ssb}$.
- (ii) Calculate the first j k -bonacci numbers.
- (iii) Apply the map S_{j+1} .
- (iv) Apply the greedy algorithm to $S_{j+1}(\vec{v})$.

Now, we calculate the complexity of each step.

- (i) Finding $j = j_{ssb}$ takes $O(k)$ operations.
- (ii) Calculating one k -bonacci number takes $O(k)$ operations, so calculating j k -bonacci vectors takes $O(kj)$ operations.
- (iii) Evaluating S_{j+1} requires a dot product modulo x_{j+1} . The dot product takes $O(k)$ operations, and since x_{j+1} is approximately $\|\vec{v}\|_\infty$ times smaller than the dot product, using repeated subtractions gives us that the modulo calculation takes $O(\|\vec{v}\|_\infty)$ operations.
- (iv) Applying the greedy algorithm requires $O(j)$ checks and $O(j)$ computations, so it takes $O(j)$ operations.

Since $j = O(k^2 \|\vec{v}\|_\infty)$, the whole algorithm has time complexity $O(k^3 \|\vec{v}\|_\infty)$. □

Proof of Proposition 2.8. As before, we list the steps of the algorithm.

- (i) Calculate all of the k -bonacci vectors $\vec{X}_{-n}$ with $\|\vec{X}_{-n}\|_2 \leq 2 \|\vec{v}\|_2$. This takes $O(\log \|\vec{v}\|_\infty)$ steps.
- (ii) Inductively:
 - (a) Find the closest k -bonacci vector to $\vec{v}_n$. This takes $O(\log \|\vec{v}\|_2)$ steps.
 - (b) Set $\vec{v}_{n+1}$. This takes $O(1)$ steps.
 - (c) Check if $\|\vec{v}_{n+1}\|_2 < \|\vec{v}_n\|_2$. This also takes $O(1)$ steps.
- (iii) Apply the greedy algorithm. We know that this takes $O(j + \|\vec{v}\|_\infty)$ operations.

Since $j = O(\log \|\vec{v}\|_\infty)$, the greedy algorithm takes $O(\|\vec{v}\|_\infty)$ operations. The large step algorithm for finding j_{lsb} is $O((\log \|\vec{v}\|_\infty)^2)$, so the whole algorithm for each k is linear in $\|\vec{v}\|_\infty$. □

Disclosure statement

No conflict of interest has been reported by the author(s).

References

- [1] Anderson, P., Bicknell-Johnson, M. (2011). Multidimensional Zeckendorf representations. *Fibonacci Quart.* 49(1):4–9.

- [2] Báril, J. P., Kirgizov, O., Vajnovszki, V. (2022). Asymptotic bit frequency in Fibonacci words and generalized Fibonacci words. *Pure Math Appl.* 30(1):23–30.
- [3] Beckwith, O., Bower, A., Gaudet, L., Insoft, R., Li, S., Miller, S. J., Tosteson, P. (2013). The average gap distribution for generalized Zeckendorf decompositions. *Fibonacci Quart.* 51(1):13–27.
- [4] Bower, A., Insoft, R., Li, S., Miller, S. J., Tosteson, P. (2015). The distribution of gaps between summands in generalized Zeckendorf decompositions. *Fibonacci Quart.* 135:130–160.
- [5] Cordwell, K., Hlavacek, M., Huynh, C., and Miller, S. J., Peterson, C., Vu, Y., Nhi T. (2018). Summand minimality and asymptotic convergence of generalized Zeckendorf decompositions. *Res. Number Theory* 4(4):1–27.
- [6] Flajolet, P., Sedgewick, R. (2009). *Analytic Combinatorics*. Cambridge University Press.
- [7] Kologlu, M., Kopp, G., Miller, S. J. Wang, Y. (2011). On the number of summands in Zeckendorf decompositions. *Fibonacci Quart.* 49(1):116–130.
- [8] Li, R., Miller, S. J. (2017). A collection of central limit type results in generalized Zeckendorf decompositions. *Fibonacci Quart.* 55(5):105–114.
- [9] Miller, S. J., Wang, Y. (2012). From Fibonacci numbers to central limit type theorems. *J. Combin. Theory Ser. A* 119(7):1398–1413.
- [10] Miller, S. J., Wang, Y. (2012). Gaussian behavior in generalized Zeckendorf decompositions. *Combinatorial and Additive Number Theory: CANT 2011 and 2012* 159–173.
- [11] Zeckendorf, E. (1972). Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas. *Bulletin de la Société Royale des Sciences de Liège* 41:179–182.

MSC2020: 11B39