arXiv:2108.01990v1 [math.CO] 4 Aug 2021

The k-Centre Selection Problem for Multidimensional Necklaces

Duncan Adamson, Argyrios Deligkas, Vladimir V. Gusev, Igor Potapov

August 5, 2021

Abstract

This paper introduces the natural generalisation of necklaces to the multidimensional setting – multidimensional necklaces. One-dimensional necklaces are known as cyclic words, two-dimensional necklaces correspond to toroidal codes, and necklaces of dimension three can represent periodic motives in crystals. Our central results are two approximation algorithms for the k-Centre selection problem, where the task is to find k uniformly spaced objects within a set of necklaces. We show that it is NP-hard to verify a solution to this problem even in the one dimensional setting. This strong negative result is complimented with two polynomial-time approximation algorithms. In one dimension we provide a $1 + f(k, N)$ approximation algorithm where $f(k, N) = \frac{\log_q(kN)}{N-\log_q(kN)} - \frac{\log_q^2(kN)}{2N(N-\log_q(kN))}$. For two dimensions we give a $1 + g(k, N)$ approximation algorithm where $g(k, N) = \frac{\log_q(kN)}{N-\log_q(kN)} - \frac{\log_q^2(k)}{2N(N-\log_q(kN))}$. In both cases N is the size of the necklaces and q the size of the alphabet. Alongside our results for these new problems, we also provide the first polynomial time algorithms for counting, generating, ranking and unranking multidimensional necklaces.

1 Introduction

The problem of finding k uniformly spaced points (centres) in a metric space is well known as the **k-centre selection problem**. Finding centres is important in many contexts: facility location and distribution, representative samples for state space exploration or identification of cluster centres. So far, the problem has been intensively studied for finite and explicitly given inputs like the k-centre problem for graphs [14], grids [35] or strings [24, 28].

In graph theory, the objective of the **k-centre problem** is to find a set C of k vertices, in a given undirected (weighted) graph $G = (V, E)$, for which the maximal distance $d(v, c)$ from any vertex v to its nearest centre c in C is minimised $\min_{|C|=k} \max_{v \in V} \min_{c \in C} d(c, v)$.
In the area of stringology finding k-centres within a set of words can be seen as a problem in a complete weighted graph. Thus, vertices are words and the distance between words depends on their closeness such as the Hamming distance or overlap/Jaccard coefficients for contextual similarity. However, the configuration space of many algebraic and combinatorial structures cannot be explicitly given due to the exponential growth and infeasibility of listing/enumerating the space. So the solutions for centre-selection problem on graphs, or explicitly given finite set of strings, is impractical to apply directly on these objects.

In this paper we introduce a fundamental class of combinatorial objects, *multidimensional necklaces*, generalising the classical combinatorial necklaces, and we study k-centre selection problem for these objects. Multidimensional words in automata theory literature are known as picture-languages and they are a well-studied generalisation of one-dimensional languages to two dimensions [3, 8, 18, 27, 29, 34]. The level of complexity to deal with such objects moves even higher if we consider natural classes of words which are equivalent under translation symmetry, known as *necklaces* [6, 19].

One-dimensional necklaces are known as cyclic words, i.e. strings over a finite alphabet, which are equivalent under the cyclic shift operation. One-dimensional necklaces also closely related to *Lyndon words*, i.e., aperiodic necklaces. For both one-dimensional necklaces and Lyndon words efficient algorithms for generation, ranking and unranking have been discovered only recently

[25, 26, 33]. Two-dimensional necklaces correspond to toroidal codes, which have recently attracted attention in the combinatorics on words community in the context of bioinformatic applications [4].

Periodic motives in crystals is another example to illustrate the application of necklaces up to dimension three, e.g. see representation of $SrTiO_3$ in Figure 1. The methods for effective exploration of a configuration space of crystal structures and a search for potentially stable materials, see EMMA [12, 13], FUSE [11], AIRSS [31], require procedures for selecting equally spaced seeding configurations in contrast to purely random initial positions. The solution to *k-centre problem* on combinatorial necklaces can be used to build representative sample in discrete configuration space of crystalline materials [1, 5] and speed up in silico predictions of novel materials, known to be one of the major scientific challenges of our time. The substantial gap of knowledge in solving k-centre problem for implicitly represented objects and applications in computational chemistry motivate the study of k-centre selection problem for multidimensional necklaces.

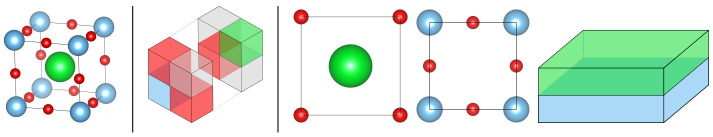

Figure 1: The crystal of $SrTiO_3$ (left) and its 3D (middle) and 1D (right) necklace representations.

Main Contributions: Our contribution is twofold. Firstly, we introduce the k-centre problem for necklaces and develop approximation algorithms for it. Secondly, we derive efficient procedures for foundational operations on multidimensional necklaces. These operations are used by our algorithms, and are of independent interest.

For the k-centre problem, we introduce the overlap distance for necklaces. Using this, we provide both negative and positive results for the problem. On the negative side we prove that it is NP-hard to evaluate the quality of a solution for the problem even for 1-dimensional necklaces. Although this is does not resolve the complexity of k-centre problem, it is a strong indication that the problem is hard. On the positive side, we design two polynomial-time algorithms that achieve $1+f(N,k)$ approximation. In the 1-dimensional case this is $1+\frac{\log_q(kn)}{n-\log_q(kn)}-\frac{\log_q^2(kn)}{2n(n-\log_q(kn))}$, and for the multidimensional case it is $1+\frac{\log_q(kN)}{N-\log_q(kN)}-\frac{\log_q^2(k)}{2N(N-\log_q(kN))}$ where N is the size of a necklace and q the size of the alphabet. Our analysis for both algorithms relies on technical Lemma 2, providing an upper bound on the distance between necklaces and the nearest sample based on the number of subwords which we cover with at least one sample. Our first algorithm works when the input includes a de Bruijn sequences of logarithmic size relative to the number of necklaces; these can be efficiently computed for one-dimensional necklaces, while no algorithm is known for higher dimensions. Our second algorithm bypasses this limitation using new operations on multidimensional necklaces like counting and ranking that we establish in Section 5.

Our second set of results contains the generalisation of several fundamental results from one-dimensional necklaces to the multidimensional setting. This theme focuses on d-dimensional necklaces defined over a given set of dimensions $\overline{\mathbf{n}} = (n_1, n_2, \ldots, n_d)$ and an alphabet Σ. Throughout the paper we use N to denote $n_1 \cdot n_2 \cdot \ldots \cdot n_d$. Our results include the first formal definition of multidimensional necklaces, along with the algorithms to:

- *Count* the number of necklaces of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ in polynomial time.
- *Generate* the set of all necklaces of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ in at most $O(N)$ time per necklace.
- *Rank* a necklace $\widehat{w}$ in the set of all necklaces of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ in $O(N^5)$ time.
- *Unrank* the i^{th} necklace of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ in $O(N^{6(d+1)})$ time.

The remainder of this paper is organised as follows. Section 2 provides definitions and notation used throughout the paper. Section 3 gives the general results about the k-centre problem on necklaces, including both hardness results and bounds on the optimal solutions. Section 4 provides

two approximation algorithms for this problem in one and d dimensional cases. Finally, Section 5 is devoted to the foundational results for multidimensional necklaces.

2 Preliminaries

We denote by $[n]$ the set of integers from 1 to n inclusive and by $[m, n]$ the set of integers from m to n inclusive. Let Σ be a linearly ordered finite alphabet such that $|\Sigma| = q$. We denote by Σ^* the set of all words over Σ and by Σ^n the set of all words in Σ^* of length n. The notation $\bar{w}$ is used to clearly denote that the variable w is a word. The *length* of a word $\bar{u} \in \Sigma^*$ is denoted by $|\bar{u}|$. We use $\bar{u}_i$ to denote the i^{th} symbol of $\bar{u}$, where $i \in [|\bar{u}|]$. The *concatenation* of words $\bar{w}$ and $\bar{u}$, denoted $\bar{w} : \bar{u}$, returns the word $\bar{w}_1\bar{w}_2 \dots \bar{w}_{|\bar{w}|}\bar{u}_1\bar{u}_2 \dots \bar{u}_{|\bar{u}|}$. We extend the ordering from Σ to Σ^* in the usual lexicographic manner. Formally, let $\bar{u}, \bar{v} \in \Sigma^*$ be a pair of words, where $|\bar{u}| \leq |\bar{v}|$. We say $\bar{u} < \bar{v}$ if and only if there exists an $i \in [\min(|\bar{u}|, |\bar{v}|) - 1]$ where $\bar{u}_1\bar{u}_2 \dots \bar{u}_{i-1} = \bar{v}_1\bar{v}_2 \dots \bar{v}_{i-1}$ and $\bar{u}_i < \bar{v}_i$. For a given set of words $\mathbf{S}$ the *rank* of $\bar{v}$ with respect to $\mathbf{S}$ is the number of words in $\mathbf{S}$ that are smaller than $\bar{v}$.

The *translation* of a word $\bar{w} = \bar{w}_1\bar{w}_2 \dots \bar{w}_n$ by $r \in [n-1]$, denoted $\langle \bar{w} \rangle_r$, returns the word $\bar{w}_{r+1} \dots \bar{w}_n\bar{w}_1 \dots \bar{w}_r$. A word $\bar{u}$ is equivalent to $\bar{v}$ under translation if $\bar{v} = \langle \bar{u} \rangle_r$ for some r. The t^{th} power of a word $\bar{w}$, denoted $\bar{w}^t$, is equal to t concatenations of $\bar{w}$. A word $\bar{w}$ is *periodic* if there is some word $\bar{u}$ and an integer $t \geq 2$ such that $\bar{w} = \bar{u}^t$. The smallest such t is called the *period* of $\bar{w}$. A word is *aperiodic* if it is not periodic.

A *necklace*, also called a *cyclic word*, is the equivalence class of words under the translation operation. For notation, a word $\bar{w}$ is written as $\widehat{w}$ when treated as a necklace. Given a necklace $\widehat{w}$, the *canonical representative* is the lexicographically smallest element of the set of words in the equivalence class $\widehat{w}$. A *cyclic subword* of the word $\bar{w}$, denoted $\bar{w}_{[i,j]} \sqsubseteq \bar{w}$, is the word $\bar{u}$ such that $\bar{u}_p = \bar{w}_{i+p \bmod |\bar{w}|}$ for all $p \in [0, n + j - i \bmod n]$. Here and in the future we tacitly assume that $\bar{w}_0$ is equivalent to $\bar{w}_n$. Since we consider only cyclic subwords in the paper, we omit "cyclic" in the future. If $\bar{w} = \bar{u} : \bar{v}$, then $\bar{u}$ is a prefix and $\bar{v}$ is a suffix. A prefix or suffix of $\bar{u}$ is *proper* if its length is smaller than $|\bar{u}|$.

Multidimensional Words and Necklaces A *d-dimensional word* over Σ is an array of d-dimensions given by a vector $\overline{\mathbf{n}} = (n_1, n_2, \dots, n_d)$ of elements from Σ. For notation, given a vector $\overline{\mathbf{n}} = (n_1, n_2, \dots, n_d)$ where every $n_i \geq 0$, $[\overline{\mathbf{n}}]$ is used to denote the set $\{(x_1, x_2, \dots, x_d) \in \mathbb{N}^d | \forall i \in [d], x_i \leq n_i\}$. Similarly $[\overline{\mathbf{m}}, \overline{\mathbf{n}}]$ is used to denote the set $\{(x_1, x_2, \dots, x_d) \in \mathbb{N}^d | \forall i \in [d], m_i \leq x_i \leq n_i\}$. Let $|\bar{w}|$ be the dimensions of $\bar{w}$. Given a vector of dimensions $\overline{\mathbf{n}} = (n_1, n_2, \dots, n_d)$, $\Sigma^{\overline{\mathbf{n}}}$ is used to denote the set of all words of dimensions $\overline{\mathbf{n}}$ over Σ. Let $N = n_1 \cdot n_2 \cdot \ldots \cdot n_d$ for a dimension vector $\overline{\mathbf{n}}$. For a d-dimensional word $\bar{w}$, the notation $\bar{w}_{(p_1,p_2,\dots,p_d)}$ is used to refer to the symbol at position $(p_1, p_2, \dots, p_d)$ in the array. Given 2 d-dimensional words $\bar{w}, \bar{u}$ such that $|\bar{w}| = (n_1, n_2, \dots, n_{d-1}, a)$ and $|\bar{u}| = (n_1, n_2, \dots, n_{d-1}, b)$, the concatenation $\bar{w} : \bar{u}$ is performed along the last coordinate, returning the word $\bar{v}$ of dimensions $(n_1, n_2, \dots, n_{d-1}, a + b)$ such that $\bar{v}_{\overline{\mathbf{p}}} = \bar{w}_{\overline{\mathbf{p}}}$ if $p_d \leq a$ and $\bar{v}_{\overline{\mathbf{p}}} = \bar{u}_{(p_1,p_2,\dots,p_{d-1},p_d-a)}$ if $p_d > a$.

A *multidimensional cyclic subword* of $\bar{w}$ of dimensions $\overline{\mathbf{m}}$ is denoted $\bar{v} \sqsubseteq_{\overline{\mathbf{m}}} \bar{w}$. As in the one-dimensional case, a subword is defined by a starting position in the original word and set of dimensions defining the size of the subword. The subword $\bar{v} \sqsubseteq_{\overline{\mathbf{m}}} \bar{w}$ starting at position $\overline{\mathbf{p}}$ with dimensions $\overline{\mathbf{m}}$ is the word $\bar{v}$ such that $\bar{v}_{\bar{\mathbf{i}}} = \bar{w}_{\bar{\mathbf{j}}}$ for all $\bar{\mathbf{j}}$ of the form $(p_1 + i_1 \bmod n_1, p_2 + i_2 \bmod n_2, \dots, p_d + i_d \bmod n_d)$, where $i_j \in [n_j]$. Such a subword $\bar{v}$ we denote by $\bar{w}_{\overline{\mathbf{p}},\overline{\mathbf{m}}}$. One important class of subwords are what we call *slices*, an example of which is given in Figure 2. The i^{th} slice of $\bar{w}$, denoted by $\bar{w}_i$, is the subword of dimensions $(n_1, n_2, \dots, n_{d-1}, 1)$ starting at position $(i, 1, \dots, 1, 1)$ of $\bar{w}$. In the 2D case, the i^{th} slice corresponds to the i^{th} row of a word. We use $\bar{w}_{[i,j]}$ to denote $\bar{w}_i : \bar{w}_{i+1} : \dots : \bar{w}_j$. A *prefix* of length l for a multidimensional word $\bar{w}$ is the first l slices of $\bar{w}$ in order. A *suffix* of length l for a multidimensional word $\bar{w}$ is the last l slices of $\bar{w}$ in order. In the two-dimensional case, the prefix and suffix of length l corresponds to the first and last l rows respectively.

A d-dimensional translation r is defined by a vector $(r_1, r_2, \dots, r_d)$. The translation of the word $\bar{w}$ of dimensions $\overline{\mathbf{n}}$ by r, denoted $\langle \bar{w} \rangle_r$ returns the word $\bar{v}$ such that $|\bar{v}| = \overline{\mathbf{n}}$ and $\bar{v}_{\overline{\mathbf{p}}} = \bar{w}_{\bar{\mathbf{j}}}$ for all $\bar{\mathbf{j}}$ of the form $(p_1 + r_1 \bmod n_1, p_2 + r_2 \bmod n_2, \dots, p_d + r_d \bmod n_d)$. We can assume that

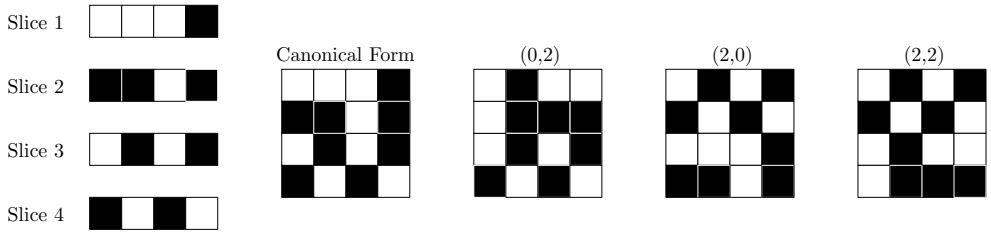

Figure 2: Example of a 2-dimensional word $\bar{w}$ of size $(4, 4)$ over a binary alphabet: the 4 slices of $\bar{w}$; the canonical form of $\bar{w}$; and three translations of $\bar{w}$.

$r_i \in [0, n_i - 1]$, so the set of translations is equivalent to the direct product of the cyclic groups $Z_{n_1} \times Z_{n_2} \times \ldots \times Z_{n_d}$. For notation let $Z_{\overline{\mathbf{n}}} = Z_{n_1} \times Z_{n_2} \times \ldots \times Z_{n_d}$. Given two translations $r = (r_1, r_2, \ldots, r_d)$ and $t = (t_1, t_2, \ldots, t_d)$ in $Z_{\overline{\mathbf{n}}}$, $t + r$ is used to denote the translation $(r_1 + t_1 \bmod n_1, r_2 + t_2 \bmod n_2, \ldots, r_d + t_d \bmod n_d)$.

Definition 1. *A multidimensional necklace (multidimensional cyclic word) $\widehat{w}$ is an equivalence class of all multidimensional words under the translation operation.*

Informally, given a necklace $\widehat{w}$ containing the word $\bar{v}$, $\widehat{w}$ contains every word $\bar{u}$ where there exists some translation r such that $\langle \bar{v} \rangle_r = \bar{u}$. Let $\mathcal{N}_q^{\overline{\mathbf{n}}}$ denote the set of necklaces of dimensions $\overline{\mathbf{n}}$ over an alphabet of size q. As in the 1D case, a *canonical representation* of a multidimensional necklace is defined as the smallest element in the equivalence class, denoted $\langle \widehat{w} \rangle$. Similarly, given a word $\bar{v} \in \widehat{w}$, $\langle \bar{v} \rangle$ denotes the canonical representation of the necklace $\widehat{w}$, i.e. $\langle \bar{v} \rangle = \langle \widehat{w} \rangle$. To determine the smallest element in the equivalence class, an ordering needs to be defined. First, we introduce an ordering over translations.

Definition 2. *Let $Z_{\overline{\mathbf{n}}}$ be the direct product of the cyclic groups $Z_{n_1} \times Z_{n_2} \times \ldots \times Z_{n_d}$, i.e. the set of all translations of words of dimensions $\overline{\mathbf{n}}$. The translation $g \in Z_{\overline{\mathbf{n}}}$ is indexed by the injective function* $\text{index}(g) \rightarrow \sum_{i=1}^{d} \left(g_i \cdot \prod_{j=1}^{i-1} n_j \right)$

The translation $g \in Z_{\overline{\mathbf{n}}}$ is smaller than $t \in Z_{\overline{\mathbf{n}}}$ if $\text{index}(g) < \text{index}(t)$. Note that $(0, 0, \ldots, 0)$ is the smallest translation and $(n_1 - 1, n_2 - 1, \ldots, n_d - 1)$ is the largest. Using this definition an ordering on multidimensional words is defined recursively. The key idea is to compare each slice based on the canonical representations. For notation, given two words $\bar{u}, \bar{s} \in \widehat{w}$, let $G(\bar{u}, \bar{s})$ return the smallest translation g where $\langle \bar{u} \rangle_g = \bar{s}$. Note that G can be computed in $O(N^2)$ time by simply checking each translation in $Z_{|\bar{u}|}$. In one dimension, the smallest such translation can be found in $O(n)$ time [7].

Definition 3. *Let $\bar{w}, \bar{u} \in \Sigma^{\overline{\mathbf{n}}}$ and let i be the smallest integer such that $\bar{w}_i \neq \bar{u}_i$. Then $\bar{w} < \bar{u}$ if either $\langle \bar{w}_i \rangle < \langle \bar{u}_i \rangle$, or $\langle \bar{w}_i \rangle = \langle \bar{u}_i \rangle$ and* $\text{index}(G(\bar{w}_i, \langle \bar{w}_i \rangle)) < \text{index}(G(\bar{u}_i, \langle \bar{u}_i \rangle))$. *Further, given necklaces $\widehat{w}$ and $\widehat{u}$, we have $\widehat{w} < \widehat{u}$ if and only if $\langle \widehat{w} \rangle < \langle \widehat{u} \rangle$.*

An example of the ordering is given in Figure 4. In what follows, $\mathcal{N}_q^{\overline{\mathbf{n}}}$ is assumed to be ordered as in Definition 3. The ***rank*** of a necklace $\widehat{w} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ is defined as the number of necklaces smaller than $\widehat{w}$ in $\mathcal{N}_q^{\overline{\mathbf{n}}}$. In the other direction, the i^{th} necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ is the necklace $\widehat{w} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ with the rank i, i.e. the necklace $\widehat{w}$ for which there are i smaller necklaces.

In order to answer some of the key questions regarding multidimensional necklaces, there are two further concepts that need to be defined for multidimensional necklaces. The first is the *period* of a word. Informally the period of $\bar{w}$ of dimensions $\overline{\mathbf{n}}$ can be thought of as the smallest subword that can tile d-dimensional space equivalently to $\bar{w}$. In order to define the period of a word, it is easiest to first define the concept of *aperiodicity*.

Definition 4. *A word $\bar{w}$ of dimensions $\overline{\mathbf{n}}$ is aperiodic if there exists no subword $\bar{v} \sqsubseteq \bar{w}$ of dimensions $\overline{\mathbf{m}} \neq \overline{\mathbf{n}}$ such that $m_i \leq n_i$ for every $i \in [1, d]$, and $\bar{w}_{\bar{\mathbf{j}}} = \bar{v}_{\bar{\mathbf{j}}'}$ where $\bar{\mathbf{j}}' = (j_1 \bmod m_1, j_2 \bmod m_2, \ldots, j_d \bmod m_d)$ for every position $\bar{\mathbf{j}} \in n_1 \times n_2 \times \ldots \times n_d$ in $\bar{w}$.*

Definition 5. *The* period *of a word $\bar{a}$ of dimensions $\overline{\mathbf{n}}$, denoted* Period($\bar{a}$)*, is the length of the aperiodic subword $\bar{b} \sqsubseteq \bar{a}$ of dimensions $\overline{\mathbf{m}}$ such that $\bar{a}_{\bar{\mathbf{i}}} = \bar{b}_{\bar{\mathbf{i}}'}$ for every position $\bar{\mathbf{i}} \in n_1 \times n_2 \times \ldots \times n_d$ and $\bar{\mathbf{i}}' = (i_1 \bmod m_1, i_2 \bmod m_2, \ldots, i_d \bmod m_d)$.*

$$period\left(\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}\right) = (3,1)$$

$$period\left(\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}\right) = (2,2)$$

$$period\left(\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}\right) = (3,2)$$

Figure 3: Example of necklaces with the smallest period representing them.

See Figure 3 as an example. By Definition 5 every word, including aperiodic ones, has a unique period [16]. In the case of an aperiodic word $\bar{w}$, the period is equal to the dimensions of $\bar{w}$. It is easy to see that a multidimensional necklace $\widehat{w}$ is aperiodic if every word $\bar{v} \in \widehat{w}$ is aperiodic. Further, note that if some word in $\widehat{w}$ is aperiodic, then every word is. An aperiodic necklace is called a *Lyndon word*. A related but distinct concept is an *atranslational* word. A word $\bar{w}$ is *atranslational* if there exists no translation $g \neq (n_1, n_2, \ldots, n_d)$ such that $\bar{w} = \langle \bar{w} \rangle_g$.

Definition 6. *A necklace $\widehat{w}$ if dimensions $\overline{\mathbf{n}}$ is* atranslational *if there exists no pair of translations $g, h \in Z_{\overline{\mathbf{n}}}$ where $g \neq h$ and $\langle \widehat{w} \rangle_g = \langle w \rangle_h$.*

In one dimension every aperiodic necklace is atranslational, while in any higher dimension every atranslational word is aperiodic, although not every aperiodic word is atranslational. For example $\begin{bmatrix} a & b \\ b & a \end{bmatrix}$ is aperiodic but not atranslational, as there are only two unique representations of the cyclic word. On the other hand $\begin{bmatrix} a & a \\ a & b \end{bmatrix}$ is both atranslational and aperiodic. For notation, $TR(\bar{w})$ is used to denote the index of the smallest translation $g \in Z_{\overline{\mathbf{n}}}$ where $\langle \bar{w} \rangle_g = \bar{w}$. Similarly $TP(\bar{w})$ is used to denote the index of the smallest translation $g \in Z_{\overline{\mathbf{n}}}$ where $\langle\langle \bar{w} \rangle\rangle_g = \bar{w}$, i.e. the smallest rotation of the canonical representative to get $\bar{w}$.

$$\bar{w} = \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \bar{w}_3 \\ \bar{w}_4 \end{bmatrix} = \begin{bmatrix} a & a & a & b \\ a & a & b & a \\ b & a & a & a \\ b & a & a & a \end{bmatrix}, \bar{u} = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \\ \bar{u}_4 \end{bmatrix} = \begin{bmatrix} a & a & a & b \\ a & a & b & a \\ a & b & a & a \\ b & a & a & a \end{bmatrix}, \bar{v} = \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \bar{v}_3 \\ \bar{v}_4 \end{bmatrix} = \begin{bmatrix} a & a & a & b \\ a & a & b & a \\ a & a & b & b \\ b & a & a & a \end{bmatrix}$$

Figure 4: An example of three words, $\bar{w}, \bar{u}$, and $\bar{v}$, ordered as follows $\bar{w} < \bar{u} < \bar{v}$. Note that $\bar{w}_1 : \bar{w}_2 = \bar{v}_1 : \bar{v}_2 = \bar{u}_1 : \bar{u}_2$. However, $\langle \bar{w}_3 \rangle = \langle \bar{u}_3 \rangle = aaab$, which is smaller than $\langle \bar{v}_3 \rangle = aabb$. Further, $\bar{w}_3 < \bar{u}_3$ as $G(\bar{w}_3, \langle \bar{w}_3 \rangle) = 1$ and $G(\bar{u}_3, \langle \bar{u}_3 \rangle) = 2$, which is larger than 1.

3 The Overlap Distance and the k-Centre problem

In this section we formally define the k-centre problem for necklaces. The input to our problem is an alphabet of size q, a vector of dimensions $\overline{\mathbf{n}}$ that defines the size of the multidimensional words, and a positive integer k. The goal is to choose a set $\mathbf{S}$ of k necklaces from the set $\mathcal{N}_q^{\overline{\mathbf{n}}}$ such that

the maximum "distance" between any necklace $\widehat{w} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ and the set $\mathbf{S}$ is minimised. Since there is no standard notion of distance between necklaces, our first task is to define one. We introduce the *overlap distance*, which aims to capture similarity between crystalline materials as an extension of the overlap metric between words. This can be seen as a natural distance based "bag-of-words" techniques used in machine learning [17].

The Overlap coefficient for Necklaces. Our definition of the overlap distance depends of the well studied *overlap coefficient*, defined for a pair of set A and B as $\frac{|A \cap B|}{\min(|A|,|B|)}$. For notation let $\mathfrak{C}(A, B)$ return the overlap coefficient between two sets A and B. Observe that $\mathfrak{C}(A, B)$ returns a rational value between 0 and 1, with 0 indicating no common elements and 1 indicating that either $A \subseteq B$ or $B \subseteq A$. In the context of necklaces the overlap coefficient $\mathfrak{C}(\widehat{w}, \widehat{v})$ is defined as the overlap coefficient between the multisets of all subwords of $\widehat{w}$ and $\widehat{v}$. For some necklace $\widehat{w}$ of dimensions $\overline{\mathbf{n}}$, the multiset of subwords of dimensions $\overline{\ell}$ contains all $\bar{u} \sqsubseteq_{\overline{\ell}} \bar{w}$. For each subword $\bar{u}$ appearing m times in $\widehat{w}$, m copies of $\bar{u}$ are added to the multiset. This gives a total of N subwords of dimensions $\overline{\ell}$ for any $\overline{\ell}$, where $N = n_1 \cdot n_2 \cdot \ldots \cdot n_d$. For example, given the necklace represented by $aaab$, the multiset of subwords of length 2 are $\{aa, aa, ab, ba\} = \{aa \times 2, ab, ba\}$. The multiset of all subwords is the union of the multisets of the subwords for every set of dimensions, having a total size of N^2; see Figures 5 and 6.

A $aaaa$ B $aaab$ C $aabb$
D $abab$ E $abbb$ F $bbbb$

$\widehat{w} \backslash \widehat{v}$	A	B	C	D	E	F
A	0	$\frac{10}{16}$	$\frac{13}{16}$	$\frac{14}{16}$	$\frac{15}{16}$	1
B	$\frac{10}{16}$	0	$\frac{9}{16}$	$\frac{10}{16}$	$\frac{12}{16}$	$\frac{15}{16}$
C	$\frac{13}{16}$	$\frac{9}{16}$	0	$\frac{10}{16}$	$\frac{8}{16}$	$\frac{13}{16}$
D	$\frac{14}{16}$	$\frac{10}{16}$	$\frac{10}{16}$	0	$\frac{6}{16}$	$\frac{14}{16}$
E	$\frac{15}{16}$	$\frac{12}{16}$	$\frac{8}{16}$	$\frac{10}{16}$	0	$\frac{10}{16}$
F	1	$\frac{15}{16}$	$\frac{13}{16}$	$\frac{14}{16}$	$\frac{8}{16}$	0

Figure 5: Example of the overlap distance for binary cyclic words of length 4.

	word *ababab*	word *abbabb*	Intersection
1	$a \times 3, b \times 3$	$a \times 2, b \times 4$	5
2	$ab \times 3, ba \times 3$	$ab \times 2, bb \times 2, ba \times 2$	4
3	$aba \times 3, bab \times 3$	$abb \times 2, bba \times 2, bab \times 2$	2
4	$abab \times 3, baba \times 3$	$abba \times 2, bbab \times 2, babb \times 2$	0
5	$ababa \times 3, babab \times 3$	$abbab \times 2, bbabb \times 2, babba \times 2$	0
6	$ababab \times 3, bababa \times 3$	$abbabb \times 2, bbabba \times 2, babbab \times 2$	0
Total			11

Figure 6: Example of the overlap coefficient calculation for a pair of words *ababab* and *abbabb*. There are 11 common subwords out of the total number of 36 subwords of length from 1 till 6, so $\mathfrak{C}(ababab, abbabb) = \frac{11}{36}$ and $\mathfrak{O}(ababab, abbabb) = \frac{25}{36}$.

Overlap Distance for Necklaces. To use the overlap coefficient as a distance between $\widehat{w}$ and $\widehat{v}$, the overlap coefficient is inverted so that a value of 1 means $\widehat{w}$ and $\widehat{v}$ share no common subwords while a value of 0 means $\widehat{w} = \widehat{v}$. The overlap distance (see example in Figure 6) between two necklaces $\widehat{w}$ and $\widehat{v}$ is $\mathfrak{O}(\widehat{w}, \widehat{v}) = 1 - \mathfrak{C}(\widehat{w}, \widehat{v})$. Proposition 1 shows that this distance is a metric distance.

Proposition 1. *The overlap distance for necklaces is a metric distance.*

Proof. Let $\widehat{a}, \widehat{b}, \widehat{c} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$, for some arbitrary vector $\overline{\mathbf{n}} \in \mathbb{N}^d$ and $q \in \mathbb{N}$. In order for the overlap distance to satisfy the metric property, $\mathfrak{O}(\widehat{a}, \widehat{b})$ must be less than or equal to $\mathfrak{O}(\widehat{a}, \widehat{c}) + \mathfrak{O}(\widehat{b}, \widehat{c})$. Rewriting this gives $1 - \mathfrak{C}(\widehat{a}, \widehat{b}) \leq 2 + \mathfrak{C}(\widehat{a}, \widehat{b}) - \mathfrak{C}(\widehat{b}, \widehat{c})$ which can be rewritten in turn as $\mathfrak{C}(\widehat{a}, \widehat{b}) + \mathfrak{C}(\widehat{b}, \widehat{c}) \leq 1 + \mathfrak{C}(\widehat{a}, \widehat{b})$. Observe that if $\mathfrak{C}(\widehat{a}, \widehat{c}) + \mathfrak{C}(\widehat{b}, \widehat{c}) > 1$ then $\frac{|\widehat{a} \cup \widehat{c}|}{N^2} + \frac{|\widehat{b} \cup \widehat{c}|}{N^2} > 1$, meaning that $|\widehat{a} \cup \widehat{c}| + |\widehat{b} \cup \widehat{c}| > N^2$. This implies that $\widehat{a}$ and $\widehat{b}$ share at least $|\widehat{a} \cup \widehat{c}| + |\widehat{b} \cup \widehat{c}| - N^2$

subwords. Therefore $\mathfrak{C}(\widehat{a}, \widehat{n})$ must be at least $\mathfrak{C}(\widehat{a}, \widehat{n}) + \mathfrak{C}(\widehat{b}, \widehat{c}) - 1$. Hence $\mathfrak{O}(\widehat{a}, \widehat{b}) \leq \mathfrak{O}(\widehat{a}, \widehat{c}) + \mathfrak{O}(\widehat{b}, \widehat{c})$. □

The k-Centre Problem. The goal of the k-Centre problem for necklaces is to select a set of k necklaces of dimensions $\overline{\mathbf{n}}$ over an alphabet of size q that are "central" within the set of necklaces $\mathcal{N}_q^{\overline{\mathbf{n}}}$. Formally the goal is to choose a set $\mathbf{S}$ of k necklaces such that the maximum distance between any necklace $\widehat{w} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ and the nearest member of $\mathbf{S}$ is minimised. Given a set of necklaces $\mathbf{S} \subset \mathcal{N}_q^{\overline{\mathbf{n}}}$, we use $\mathfrak{D}(\mathbf{S}, \mathcal{N}_q^{\overline{\mathbf{n}}})$ to denote the maximum overlap distance between any necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ and its closest necklace in $\mathbf{S}$. Formally:

$$\mathfrak{D}(\mathbf{S}, \mathcal{N}_q^{\overline{\mathbf{n}}}) = \max_{\widehat{v} \in \mathcal{N}_q^{\overline{\mathbf{n}}}} \min_{\widehat{s} \in \mathbf{S}} \mathfrak{O}(\widehat{s}, \widehat{v}).$$

Problem 1. ***k-Centre problem for necklaces:*** *Given a set of dimensions $\overline{\mathbf{n}}$, alphabet Σ of size q, and an integer k, what is the set $\mathbf{S} \subseteq \mathcal{N}_q^{\overline{\mathbf{n}}}$ of size k minimising $\mathfrak{D}(\mathbf{S}, \mathcal{N}_q^{\overline{\mathbf{n}}})$?*

There are two major challenges we have to overcome in order to solve Problem 1: the exponential size of $\mathcal{N}_q^{\overline{\mathbf{n}}}$, and the lack of structural, algorithmic, and combinatorial results for multidimensional necklaces. We show that the conceptually simpler problem of verifying whether a set of necklaces is a solution for Problem 2 is NP-hard for any dimension d.

Problem 2. *Given a set of k necklaces $\mathbf{S}$ of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ and a distance ℓ, does there exist some necklace $\widehat{v} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ such that $\mathfrak{O}(\widehat{s}, \widehat{v}) > \ell$ for every $\widehat{s} \in \mathbf{S}$?*

Theorem 1. *Problem 2 is NP-hard for any dimension d.*

Proof. We prove the claim via a reduction from the Hamiltonian cycle problem on bipartite graphs to Problem 2 in one dimension. Note that if the problem is hard in the 1D case, then it is also hard in any dimension $d \geq 1$ by using the same reduction for necklaces of dimensions $(n_1, 1, 1, \ldots, 1)$. Let $G = (V, E)$ be a bipartite graph containing an even number $n \geq 6$ of vertices. The alphabet Σ is constructed with size n such that there is a one to one correspondence between each vertex in V and symbol in Σ. Using Σ a set $\mathbf{S}$ of necklaces is constructed as follows. For every pair of vertices $u, v \in V$ where $(u, v) \notin E$, the necklace corresponding to the word $(uv)^{n/2}$ is added to the set of centres $\mathbf{S}$. Further the word v^n, for every $v \in V$, is added to the set $\mathbf{S}$.

For the set $\mathbf{S}$, we ask if there exists any necklace in $\mathcal{N}_q^n$ that is further than a distance of $1 - \frac{3}{n^2}$. For the sake of contradiction, assume that there is no Hamiltonian cycle in G, and further that there exists a necklace $\widehat{w} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ such that the distance between $\widehat{w}$ and every necklace $\widehat{v} \in \mathbf{S}$ is greater than $1 - \frac{3}{n^2}$. If $\widehat{w}$ shares a subword of length 2 with any necklace in $\mathbf{S}$ then $\widehat{w}$ would be at a distance of no less than $1 - \frac{3}{n^2}$ from $\mathbf{S}$. Therefore, as every subword of length 2 in $\mathbf{S}$ corresponds to a edge that is not a member of E, every subword of length 2 in $\widehat{w}$ must correspond to a valid edge.

As $\widehat{w}$ can not correspond to a Hamiltonian cycle, there must be at least one vertex v for which the corresponding symbol appears at least 2 times in $\widehat{w}$. As G is bipartite, if any cycle represented by $\widehat{w}$ has length greater than 2, there must exist at least one vertex u such that $(v, u) \notin E$. Therefore, the necklace $(uv)^{n/2}$ is at a distance of no more than $\frac{n^2}{3}$ from $\widehat{w}$. Alternatively, if every cycle represented by $\widehat{w}$ has length 2, there must be some vertex v that is represented at least 3 times in $\widehat{w}$. Hence in this case $\widehat{w}$ is at a distance of no more than $1 - \frac{3}{n^2}$ from the word $v^n \in \mathbf{S}$. Therefore, there exists a necklace at a distance of greater than $1 - \frac{3}{n^2}$ if and only if there exists a Hamiltonian cycle in the graph G. Therefore, it is NP-hard to verify if there exists any necklace at a distance greater than l for some set $\mathbf{S}$. □

The combination of this negative result with the exponential size of $\mathcal{N}_q^{\overline{\mathbf{n}}}$ makes finding an optimal solution for Problem 1 in polynomial time relative to the values of q and $\overline{\mathbf{n}}$ exceedingly unlikely. As such the remainder of our work on the k-centre problem for necklaces focuses on approximation algorithms. Lemma 1 provides a lower bound on the optimal distance.

Lemma 1. *Let $\mathbf{S} \subseteq \mathcal{N}_q^{\overline{\mathbf{n}}}$ be an optimal set of k centres minimising $\mathfrak{D}(\mathbf{S}, \mathcal{N}_q^{\overline{\mathbf{n}}})$ then $\mathfrak{D}(\mathbf{S}, \mathcal{N}_q^{\overline{\mathbf{n}}}) \geq 1 - \frac{\log_q(k \cdot N)}{N}$.*

Proof. We first prove the lemma for the one-dimensional case, then extend the proof to the multi-dimensional setting. Recall that the distance between any pair of necklaces $\widehat{u}$ and $\widehat{v}$ is determined by the overlap coefficient and by extension the number of shared subwords between $\widehat{u}$ and $\widehat{v}$. Hence the distance between the furthest necklace $\widehat{w} \in \mathcal{N}_q^n$ and the optimal set $\mathbf{S}$ is bound from bellow by determining an upper bound on the number of shared subwords between $\widehat{w}$ and the words in $\mathbf{S}$. For the remainder of this proof let $\widehat{w}$ to be the necklace furthest from the optimal set $\mathbf{S}$. Further for the sake of determining an upper bound, the set $\mathbf{S}$ is treated as a single necklace $\widehat{S}$ of length $n \cdot k$. This may be thought of as the necklace corresponding to the concatenation of each necklace in $\mathbf{S}$. Note that the length of $\mathbf{S}$ is $k \cdot n$. As the distance between $\widehat{w}$ and $\widehat{S}$ is no more than the distance between $\widehat{w}$ and any $\widehat{v} \in \mathbf{S}$, the distance between $\widehat{w}$ and $\widehat{S}$ provides a lower bound on the distance between $\widehat{w}$ and $\mathbf{S}$.

In order to determine the number of subwords shared by $\widehat{w}$ and $\widehat{S}$, consider first the subwords of length 1. In order to guarantee that $\widehat{w}$ shares at least one subword of length 1, $\widehat{S}$ must contain each symbol in Σ, requiring the length of $\widehat{S}$ to be at least q. Similarly, in order to ensure that $\widehat{w}$ shares two subwords of length 1 with $\widehat{S}$, $\widehat{S}$ must contain 2 copies of every symbol on Σ, requiring the length of $\widehat{S}$ to be at least $2q$. More generally for $\widehat{S}$ to share i subwords of length 1 with $\widehat{w}$, $\widehat{S}$ must contain i copies of each symbol in Σ, requiring the length of $\widehat{S}$ to be at least $i \cdot q$. Hence the maximum number of subwords of length 1 that $\widehat{w}$ can share with $\widehat{S}$ is either $\lfloor \frac{n \cdot k}{q} \rfloor$, if $\lfloor \frac{n \cdot k}{q} \rfloor \leq n$, or n otherwise.

In the case of subwords of length 2, the problem becomes somewhat more complicated. Note that in order to share a single word of length 2, it is not necessary to to have every subword of length 2 appear as a subword of $\widehat{w}$. Instead, it is sufficient to use only the prefixes of the canonical representations of each necklace. For example, given the binary alphabet $\{a, b\}$, every necklace has either aa, ab or bb as the prefix of length 2. Note that any necklace of length 2 followed by the largest symbol q in the alphabet $n-2$ times belongs to the set N_q^n. As such, a simple lower bound on the number of prefixes of the canonical form of necklaces is the number of necklaces of length 2, which in turn is bounded by $\frac{q^2}{2}$. Noting that these prefixes in $\widehat{S}$ may overlap, in order to ensure that $\widehat{S}$ and $\widehat{w}$ share at least one subword of length 2, the length of $\widehat{S}$ must be at least $\frac{q^2}{2}$. Similarly, for $\widehat{S}$ and $\widehat{w}$ to share i subwords of length 2, the length of $\widehat{S}$ must be at least $\frac{i \cdot q^2}{2}$. Hence the maximum number of subwords of length 2 that $\widehat{S}$ and $\widehat{w}$ can share is either $\lfloor \frac{2n \cdot k}{q^2} \rfloor$, if $\lfloor \frac{2n \cdot k}{q^2} \rfloor \leq n$, or n otherwise. More generally, in order for $\widehat{S}$ to share at least one subword of length j with $\widehat{w}$, the length of $\widehat{S}$ must be at least $\frac{q^j}{j}$. Further the maximum number of subwords of length j that $\widehat{S}$ and $\widehat{w}$ can share is either $\lfloor \frac{j \cdot n \cdot k}{q^j} \rfloor$, if $\lfloor \frac{j \cdot n \cdot k}{q^j} \rfloor \leq n$ or n otherwise.

Using these observations, the maximum length of a common subword that $\widehat{w}$ can share with $\widehat{S}$ is the largest value l such that $\frac{q^l}{l} \leq n \cdot k$. By noting that $\frac{q^l}{l} \geq \frac{q^l}{n}$, a upper bound on l can be derived by rewriting the inequality $\frac{q^l}{n} \leq n \cdot k$ to $l = 2\log_q(n \cdot k)$. Note further that, for any value $l' > l$, there must be at least one necklace that does not share any subword of length l' with $\widehat{S}$ as $\widehat{S}$ can not contain enough subwords to ensure that this is the case. This bound allows an upper bound number of shared subwords between $\widehat{w}$ and $\widehat{S}$ to be given by the summation $\sum_{i=1}^{2\log_q(n \cdot k)} \min(\lfloor \frac{i \cdot n \cdot k}{q^i} \rfloor, n) \leq n \cdot \log_q(n \cdot k) + \frac{\log_q(k \cdot n)}{q-1} \approx \frac{q \cdot n \log_q(k \cdot n)}{q-1} \approx n \log_q(k \cdot n)$. Using this bound, the distance between $\widehat{w}$ and $\widehat{S}$ must be no less than $1 - \frac{\log_q(k \cdot n)}{n}$.

The same arguments can be applied to the multidimensional case. Let $\overline{\mathbf{m}} = (m_1, m_2, \ldots, m_d)$ be a vector of d-dimensions such that $M = m_1 \cdot m_2 \cdot \ldots \cdot m_d$. The largest value of M such that $\widehat{S}$ can contain every subword with M positions is $2\log_q(n \cdot k)$. The upper bound on the number of words of dimensions $\overline{\mathbf{m}}$ is $\frac{q^M}{M}$. Let $F(x, \overline{\mathbf{m}})$ return the size of the set $[\overline{\mathbf{m}}]$, i.e. the number of vectors with x positions that are less than or equal to $\overline{\mathbf{m}}$ in each dimension. Using this notation,

the maximum number of shared subwords between $\widehat{w}$ and $\widehat{S}$ is $\sum_{i=1}^{M} F(i, \overline{\mathbf{m}}) \cdot \frac{i \cdot N \cdot k}{q^i}$. Note that $\sum_{i=1}^{M} F(i, \overline{\mathbf{m}}) \cdot \frac{i \cdot N \cdot k}{q^i} \leq \sum_{i=1}^{M} \frac{i \cdot N \cdot k}{q^i}$. Therefore, the upper bound on the number of common subwords in the multidimensional setting is $N \log_q(k \cdot N)$, giving a bound on the distance of $1 - \frac{\log_q(k \cdot N)}{N}$. □

4 Two Approximation Algorithms for the k-Centre Problem

In this section we provide two approximation algorithms for the k-centre problem. The first is $1+(\frac{\log_q(kN)}{N-\log_q(kN)} - \frac{\log_q^2(kN)}{2N(N-\log_q(kN))})$-approximate with a running time $O(N \cdot k)$, but it requires access to the de Bruijn hypertori of the multidimensional necklaces; this is a generalisation of de Bruijn sequences. When $d = 1$, there exists an efficient algorithm for computing the de Bruijn sequence. However, for $d > 1$, no algorithm is known for computing a de Bruijn hypertori. Therefore, we develop a second algorithm that is $1+(\frac{\log_q(kN)}{N-\log_q(kN)} - \frac{\log_q^2(k)}{2N(N-\log_q(kN))})$-approximation with running time $O(N^6)$, requiring techniques presented in Section 5.2.

The main idea behind both algorithms is to try to find the largest set of dimensions $\overline{\ell}$ such that every subword of length $\overline{\ell}$ may appear in some word within the set. In this setting $\overline{\mathbf{m}}$ is larger than $\overline{\ell}$ if $m_1 \cdot m_2 \cdot \ldots \cdot m_d > l_1 \cdot l_2 \cdot \ldots \cdot l_d$. This is motivated by observing that if two necklaces share a subword of length l, they must also share 2 subwords of length $l-1$, 3 of length $l-2$, and so on. Lemma 2 provides an upper bound for the overlap distance between any necklace in $|\mathcal{N}_q^n|$ and the set $\mathbf{S}$ containing all subwords of length l.

Lemma 2. *Given $\widehat{w}, \widehat{v} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ sharing a common subword $\bar{a}$ of dimensions $\overline{\mathbf{m}}$, the distance between $\widehat{w}$ and $\widehat{v}$ is no more than $\mathfrak{O}(\widehat{w}, \widehat{v}) \leq 1 - \frac{M^2}{2N^2}$ where $M = m_1 \cdot m_2 \cdot \ldots \cdot m_d$ and $N = n_1 \cdot n_2 \cdot \ldots \cdot n_d$.*

Proof. Note that the minimum intersection between $\widehat{w}$ and $\widehat{v}$ is the number of subwords of $\bar{a}$, including the word $\bar{a}$ itself. To compute the number of subwords of $\bar{a}$, consider the number of subwords starting at some position $\bar{\mathbf{j}} \in [|\bar{a}|]$. Assuming that $|\bar{a}|_i < n_i$ for every $i \in [d]$, the number of subwords starting at $\bar{\mathbf{j}}$ corresponds to the size of the set $[\bar{\mathbf{j}}, |\bar{a}|]$, equal to $\prod_{i=1}^{d} m_i - |\bar{a}|_i$. This gives the number of shared subwords as being at least $\sum_{\bar{\mathbf{j}} \in [|\bar{a}|]} \prod_{i \in [d]} m_i - |\bar{a}|_i \geq \sum_{j \in [M]} j \geq \frac{M^2}{2}$. Therefore, the distance between $\widehat{w}$ and $\widehat{v}$ is no more than $1 - \frac{M^2}{2N^2}$. □

4.1 Approximating the k-centre problem using de Bruijn sequences

In this section we provide our first approximation algorithm that requires access to de Bruijn sequences for the one-dimensional case and to de Bruijn hypertori for higher dimensions. The main idea is to determine the largest de Bruijn sequence that can "fit" into the set of k-centres. As the de Bruijn sequence of order l contains every word in Σ^l as a subword, by representing the de Bruijn sequence of order l in the set of centres we ensure that every necklace shares a subword of length l with the set of k-centres. Therefore, by determining the longest sequence that can be represented by k centres, an upper bound on the distance between the furthest necklace and the set of k-centres is derived.

Definition 7. *A de Bruijn hypertorus of order $\overline{\mathbf{n}}$ is a cyclic d-dimensional word T containing, as a subword, every word of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ of size q. Further, each such word of dimensions $\overline{\mathbf{n}}$ over the alphabet Σ appears exactly once as a subword of T.*

Lemma 3. *There exists an $O(nk)$ time algorithm for the k-Centre problem on $\mathcal{N}_q^n$ such that every word in $\mathcal{N}_q^n$ shares a common subword of length at least $\log_q(n \cdot k)$ with one or more centres. Further, no word in $\mathcal{N}_q^n$ is at a distance of more than $1 - \frac{\log_q^2(kn)}{2n^2}$ from the nearest centre.*

Sequence:	000000100001100010100011100100101100110100111101010111011011111
Centre	Word
1	000000100001100010100
2	101000111001001011001
3	110011010011111010111
4	010110110111111000000

Figure 7: Example of how to split the de Bruijn sequence of order 6 between 4 centres. Highlighted parts are the shared subwords between two centres.

Proof. The high level idea of this algorithm is to spilt a de Bruijn sequence of order λ between the k centres. The motivation behind this approach is to represent every word of length λ as a subword of at least one centre. Note that the length of the de Bruijn sequence of order λ is q^λ.

Given a de Bruijn sequence $\bar{s}$, naively splitting $\bar{s}$ into k words may lead to subwords being lost. For example, take the de Bruijn sequence of order 4 over the alphabet $\{a, b\}$ *aaaabaabbbababbbb*, dividing this between two words of length 8 results in the samples *aaaabaab* and *babaabbb*, missing the words *aabb*, *abba*, and *bbaa*. In order to account for this, the sequence is split into centres of size $n - \lambda + 1$, with the final $\lambda - 1$ symbols of the i^{th} centre being shared with the $(i+1)^{th}$ centre. In this manner, the first centre is generated by taking the first n symbols of the de Bruijn sequence. To ensure that every subword of length λ occurs, the first $\lambda - 1$ symbols of the second centre is the same as the last $\lambda - 1$ symbol of the first centre. Repeating this, the i^{th} centre is the subword of length n starting at position $i(n - \lambda + 1) + 1$ in the de Bruijn sequence. An example of this is given in Figure 7.

The leaves the problem of determining the largest value of λ such that $q^\lambda \leq k \cdot (n - \lambda + 1)$. The inequality $q^\lambda \leq k \cdot (n - \lambda + 1)$ can be rearranged in terms of λ as $\lambda \leq \log_q(k \cdot (n+1) - k \cdot \lambda)$. Noting that λ must be no more than $\log_q(k \cdot n)$, this upper bound on the value of λ can be rewritten as $\log_q(k \cdot (n + 1 - \log_q(k \cdot n))) \approx \log_q(k \cdot n)$. Using Lemma 2, along with $\log_q(k \cdot n)$ as an approximate value of λ gives an upper bound on the distance between between each necklace in $\mathcal{N}_q^n$ and the set of samples of $1 - \frac{\log_q^2(kn)}{2n^2}$.

As the corresponding de Bruijn sequence can be computed in no more than $O(k \cdot n)$ time [32] and the set of samples can be further derived from the sequence in at most $O(k \cdot n)$ time, the total complexity is at most $O(k \cdot n)$. Note that any algorithm that outputs such a set of centres must take at most $\Omega(k \cdot n)$ time. □

Theorem 2. *Problem 1 in 1D can be approximated in $O(nk)$ time with an approximation factor of $1 + f(n,k)$ where $f(n,k) = \frac{\log_q(kn)}{n - \log_q(kn)} - \frac{\log_q^2(kn)}{2n(n - \log_q(kn))}$ and $f(n,k) \to 0$ for $n \to \infty$.*

Proof. Recall from Lemma 1 that the overlap distance is bounded by $1 - \frac{\log_q(k \cdot n)}{n}$. Using the lower bound of $1 - \frac{\log_q^2(kn)}{2n^2}$ given by Lemma 3 gives an approximation ratio of $\frac{1 - \frac{\log_q^2(kn)}{2n^2}}{1 - \frac{\log_q(k \cdot n)}{n}} = \frac{2n^2 - \log_q^2(kn)}{2n^2 - 2n\log_q(kn)}$ $= 1 + \frac{2n\log_q(kn) - \log_q^2(kn)}{2n^2 - 2n\log_q(kn)} = 1 + \frac{\log_q(kn)}{n - \log_q(kn)} - \frac{\log_q^2(kn)}{2n(n - \log_q(kn))}$. Note that $f(n,k) = \frac{2n\log_q(kn) - \log_q^2(kn)}{2n^2 - 2n\log_q(kn)}$ converges to 0 when $n \to \infty$ for a constant $k < q^n/n$. □

Theorem 3. *Let T be a d-dimensional de Bruijn hyper torus of dimensions $(x, x, \ldots, x)$. There exist k subwords of T that form a solution to the k-centre problem for $\mathcal{N}_q^{(y,y,\ldots,y)}$ with an approximation factor of $1 + f(n,k)$ where $f(n,k) = \frac{\log_q(kN)}{N - \log_q(kN)} - \frac{\log_q^2(kN)}{2N(N - \log_q(kN))}$, $f(n,k) \to 0$, $N \to \infty$.*

Proof. Recall from Lemma 1 that the lower bound on the distance between the centre and every necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ is $1 - \frac{log_q(k \cdot N)}{N}$. As in Theorem 2, the goal is to find the largest de Bruijn torus that can "fit" into the centres. To simplify the reasoning, the de Bruijn hyper tori here is limited to those corresponding to the word where the length of each dimension is the same. Formally, the de Bruijn hypertori are restricted to be of the dimensions $m_1 = m_2 = \ldots = m_j = \sqrt[j]{N}$ for some $j \in [d]$, giving the total number of positions in the tori as M. Similarly, the centres is assumed to have dimensions $n_1 = n_2 = \ldots = n_d = \sqrt[d]{N}$, giving N total positions.

$k\backslash n$	1	2	3	4	5	6	7	8
1	1.0	1.75	1.8242	1.75	1.6657	1.59388	1.53532	1.4875
2	1.0	1.0	4.54496	2.875	2.322	2.04096	1.86822	1.75
3	1.0	1.0	1.0	5.76696	3.17774	2.48677	2.15592	1.95785
4	1.0	1.0	1.0	1.0	4.61912	3.00217	2.43963	2.14583
5	1.0	1.0	1.0	1.0	7.98402	3.65337	2.73732	2.32623
6	1.0	1.0	1.0	1.0	27.84082	4.54496	3.06221	2.50535
7	1.0	1.0	1.0	1.0	1.0	5.88615	3.4276	2.68724
8	1.0	1.0	1.0	1.0	1.0	8.19368	3.84946	2.875

$k\backslash n$	1	2	3	4	5	6	7	8
1	1.0	1.18333	1.19493	1.18333	1.16897	1.15565	1.144	1.13393
2	1.41667	1.41667	1.34509	1.29167	1.25296	1.22393	1.20138	1.18333
3	1.8242	1.59388	1.44797	1.36238	1.30633	1.26659	1.23682	1.2136
4	2.33333	1.75	1.53018	1.41667	1.34644	1.29825	1.2629	1.23575
5	3.09914	1.89704	1.6006	1.46153	1.379	1.32369	1.28372	1.25334
6	4.54496	2.04096	1.66333	1.50021	1.40664	1.34509	1.30113	1.26799
7	8.75423	2.18549	1.72065	1.53449	1.4308	1.36364	1.31615	1.28059
8	1.0	2.33333	1.77396	1.56548	1.45235	1.38007	1.32939	1.29167

Table 1: Table of approximation ratio for the algorithm given in Theorem 2 for different values of n and k for a binary alphabet (top) and an alphabet of size 8 (below). Note that when $k \geq q^n$ the approximation ratio is 1 as every necklace can be represented in the set.

Observe that the largest torus that can be represented in the set of centres has M positions such that $q^M \leq k \cdot N^{(d-j)/d}(\sqrt[d]{N} - \sqrt[j]{M}+1)^j$. This can be rewritten to give $M \leq \log_q(k \cdot N^{(d-j)/d}(\sqrt[d]{N} - \sqrt[j]{M}+1)^j)$. Noting that M is of logarithmic size relative to N, this is approximately equal to $M \leq \log_q(k \cdot N)$. Using Lemma 2, the minimum distance between any necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ is $1 - \frac{\log_q^2(kN)}{2N^2}$. This is compared to the optimal solution, following the arguments from Theorem 2 giving a ratio of $1 + f(N,k)$ where $f(N,k) = \frac{2 \cdot N \log_q(k \cdot N) - \log_q^2(k \cdot N)}{2 \cdot N^2 - 2 \cdot N \cdot \log_q(k \cdot N)} = \frac{\log_q(kN)}{N - \log_q(kN)} - \frac{\log_q^2(kN)}{2N(N - \log_q(kN))}$. □

For both cases table providing some explicit examples of the approximation ratio for different values of n and k is given in Table 1. While this provides a good starting point for solving the k-Centre problem for $\mathcal{N}_q^{\overline{\mathbf{n}}}$, results on generating de Bruijn tori are highly limited, focusing on the cases with small dimensions [10, 20, 21, 22, 23]. As such an alternate approach is needed.

4.2 Approximating the k-centre problem using Prefix Trees

In this section we present our second approximation algorithm. At a high level our algorithm works as follows. It recursively builds a tree of possible necklace prefixes, starting with the empty string, in a breadth first manner, continuing until there are k such prefixes. Once these prefixes have been generated, the centres are built as necklaces containing these prefixes. Our algorithm relies on the operations of efficiently counting and ranking multidimensional necklaces. However, there are no known algorithms for these operations for high-dimensional necklaces. For this reason Section 5 provides such algorithms.

Lemma 4. *The number of necklaces in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ sharing a given prefix $\bar{a}$ can be determined in $O(N^5)$ time.*

Proof. This is done by comparing the rank of the smallest and largest necklaces with the prefix $\bar{a}$. Let $\widehat{u} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ be the necklace with the smallest rank such that $\bar{a}$ is a prefix of $\langle \widehat{u} \rangle$. Note that the value of $\langle \widehat{u} \rangle$ can be found in $O(N)$ time using Lemma 19 starting with the word $\bar{A}$ of dimensions $\overline{\mathbf{n}}$ where $\bar{A}_{\overline{\mathbf{i}}} = \bar{a}_{(i_1 \bmod |\bar{a}|_1, i_2 \bmod |\bar{a}|_2, \ldots i_d \bmod |\bar{a}|_d)}$. Let $\widehat{v} \in \mathcal{N}_q^{\overline{\mathbf{n}}}$ be the necklace with the largest rank such that $\bar{a}$ is a prefix of $\langle \widehat{v} \rangle$. $\widehat{v}$ may be constructed from the prefix $\bar{a}$ by filling every position after the prefix with the symbol $q \in \Sigma$. The number of necklaces sharing $\bar{a}$ as a pref is given by $rank(\langle \widehat{v} \rangle) - rank(\langle \widehat{u} \rangle) + 1$. Following Theorem 5 the rank is computed in $O(N^5)$ time. Therefore the difference may also be computed with on more than $O(N^5)$ time. □

The k-Centres selection based on a tree of necklace prefixes: At a high level, this prefix algorithm works by finding a set of k-necklace prefixes, i.e. a set of k words corresponding to prefixes of the canonical forms of necklaces. The algorithm recursively builds the tree of possible necklace prefixes in a breadth first manner, starting with the empty string and continuing until there are k such prefixes. Once these prefixes have been generated, the centres are built as necklaces containing these prefixes.

This is achieved as follows. At each step there is the set of prefixes P with l symbols such that the number of prefixes is less than k. The set P' of prefixes of length $l+1$ is generated from the set P by observing that every prefix $p' \in P'$ can be written as $\bar{p} : \bar{x}$ for some prefix $\bar{p} \in P$ and word $\bar{x} \in \Sigma^{n_1,n_2,\ldots,n_{d-1}}$. Given $\bar{p} \in P$ and $\bar{x} \in \Sigma^{n_1,n_2,\ldots,n_{d-1}}$, $\bar{p} : \bar{x}$ is in P' if and only if it is the prefix of a necklace. The set P' is generated by determining the set of prefixes for each $\bar{p}$ repeating this process for every $p \in P$, $\sigma \in \Sigma$. Once the size of P' is greater than k, the algorithm terminates using the prefixes in P as a basis. For each $p \in P$, a centre is generated by appending an arbitrary subword following the prefix. Note that as the number of necklaces with a given prefix must be determined, this is only possible in the multidimensional case due to our novel ranking procedure.

Theorem 4. *There exists a polynomial-time algorithm to construct k centres of $\mathcal{N}_q^{\mathbf{n}}$ that is an approximation of the optimal solution by a factor of $1 + f(N,k)$ where $f(n,k) = \frac{\log_q(kN)}{N-\log_q(kN)} - \frac{\log_q^2(k)}{2N(N-\log_q(kN))}$ and $f(N,k) \to 0$ for $N \to \infty$.*

Proof. Let $\overline{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_d)$ be the dimensions of the prefixes at the termination of the algorithm. To bound $\overline{\lambda}$, observe that each centre corresponds to a prefix of length $\overline{\lambda}$. Therefore, this becomes the problem of determining the largest value of $\overline{\lambda}$ such that the number of prefixes over the alphabet Σ of size q is no more than l. Let $L = \lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_d$. Using L, an upper bound on the number of prefixes of dimensions $\overline{\lambda}$ can be derived as q^L. As the number of prefixes must be no more than k, $q^L \leq k$, giving $L \leq \log_q(k)$. Using Lemma 2, the distance between each word and the nearest centre is no more than $1 - \frac{\log_q(k)(\log_q(k)+1)}{2N^2}$, which is bounded by $\frac{\log_q^2(k)}{2N^2}$. Lemma 1 gives a lower bound on the distance between every necklace in $\mathcal{N}_q^{\mathbf{n}}$ and the nearest centre in the centre of $1 - \frac{\log_q(k \cdot N)}{N}$. Therefore, this algorithm gives an approximation of the optimal solution by a factor of $\frac{1-\frac{\log_q^2}{k}2N^2}{1-\frac{\log_q^2(k)}{2N^2}}$, which is simplified to a factor of $1 + \frac{2N\log_q(kN)-\log_q^2(k)}{2N^2-2N\log_q(kN)}$. Note that $f(N,k) = \frac{2N\log_q(kN)-\log_q^2(k)}{2N^2-2N\log_q(kN)} = \frac{\log_q(kN)}{N-\log_q(kN)} - \frac{\log_q^2(k)}{2N(N-\log_q(kN))}$ converges to 0 when $n \to \infty$ for any fixed k.

To show that the method terminates in polynomial time, we note that the time to compute the number of necklaces with a given prefix is $O(N^5)$. For every $i \leq \lambda + 1$, at most k centres are checked. To determine the longest λ, let there be p_i prefixes of length i. Observe that there are at least $p_i + q - 1$ words of length $i = 1$. Therefore the number of prefixes of length i is at least $(i+1)q - i$. Therefore the longest length is $\frac{k-q}{q-1}$. Thus the maximum number prefixes that need to be checked is $k \cdot \frac{k-q}{q-1}$ and the total complexity is $O\left(k \cdot \frac{k-q}{q-1} N^5\right)$, which is simplified to $O\left(k^2 N^5\right)$. □

5 Efficient Operations on Multidimensional Necklaces

This section provides polynomial time algorithms for counting, ranking, unranking, and generating multidimensional necklaces. These are well-studied problems in the 1D case, but to the best of our knowledge our work is the first that considers these natural generalisations.

5.1 Counting Multidimensional Necklaces

In this section we prove closed form formulas for the size of several different subsets of multidimensional words. In addition, the derived formulas allow us to provide bounds on the relationship between the cardinality of these sets.

For both necklaces and Lyndon words, explicit counting is done by application of the Pólya enumeration theorem to the group operations defined in Section 2. The equations below are classical formulas for counting the number of one-dimensional necklaces and one-dimensional Lyndon words respectively. A classical proof for the Necklace Equation is provided by Graham et. al. [19], while Perrin [30] provides a proof of the Lyndon word Equation.

$$|\mathcal{N}_q^n| = \sum_{d|n} \phi\left(\frac{n}{d}\right) q^d. \tag{1}$$

$$L_q^n = \sum_{d|n} \mu\left(\frac{n}{d}\right) q^d. \tag{2}$$

Where ϕ is Euler's totient function and μ is the Möbius function. Formally, $\phi(n)$ gives the number of natural numbers smaller than n which are co-prime to n, and $\mu(n)$ returns -1, 0, or 1 depending on the prime factorisation of n. These equations form the starting point for counting multidimensional necklaces. Recall from the preliminaries that multidimensional necklaces of dimensions $\overline{\mathbf{n}}$ are equivalence classes of words in $\Sigma^{\overline{\mathbf{n}}}$ under the group $Z_{\overline{\mathbf{n}}} = Z_{n_1} \times Z_{n_2} \times \ldots \times Z_{n_d}$ where $\times$ denotes the direct product and Z_x the cyclic group of order x. A straightforward way to compute the number of necklaces of dimensions $\overline{\mathbf{n}}$ is by using the *Pólya enumeration formula*, giving:

$$|\mathcal{N}_q^{\overline{\mathbf{n}}}| = \frac{1}{N} \sum_{g \in Z_{\overline{\mathbf{n}}}} q^{c(g)}.$$

Where $g = (g_1, g_2, \ldots, g_d)$ is some group action in $Z_{\overline{\mathbf{n}}}$ and $c(g)$ returns the number of cycles from the group action g. Since $Z_{\overline{\mathbf{n}}}$ is formed by the direct product of the cyclic groups, for each group action g we have that $g = (g_1, g_2, \ldots, g_d)$, where $1 \leq i_j \leq n_j$. Therefore, the number of necklaces, $|\mathcal{N}_q^{\overline{\mathbf{n}}}|$, is rewritten as:

$$|\mathcal{N}_q^{\overline{\mathbf{n}}}| = \frac{1}{N} \sum_{g_1=1}^{n_1} \sum_{g_2=1}^{n_2} \cdots \sum_{g_d=1}^{n_d} q^{c((g_1, g_2, \ldots i_d))}$$

In order to determine the value of $c(g)$, consider the permutation induced by g. Given some position $\mathbf{j} = (j_1, \ldots, j_d)$, let $\mathbf{j}'$ be the position following $\mathbf{j}$ in the cycle induced by g, i.e. $\mathbf{j}' = \mathbf{j} \cdot g$. The coordinate of $\mathbf{j}'$ in the i^{th} dimension is equal to the coordinate in the i^{th} dimension of $\mathbf{j}$ shifted by g_i. Since this is a cyclic operation, this shift is done modulo the length of dimension i, n_i. This gives $j_i' = (j_i + g_i) \bmod n_i$.

Let g^t denote the group action made by applying t times operation g to the identity operation I, i.e. $I \cdot g \cdot g \ldots \cdot g$. The length of the cycle induced by some cyclic shift g is the smallest value $t > 0$ such that $\mathbf{j} \cdot g^t = \mathbf{j}$. In other words, the length of the cycle equals the number of times g must be applied to itself to become the identity operation. The length of this cycle is therefore the smallest t such that for every i, $(\mathbf{j}_i + t \cdot g_i) \bmod n_i \equiv \mathbf{j}_i$. To compute this, note that t must be divisible by the smallest value l_i for each dimension such that $(\mathbf{j}_i + l_i \cdot g_i) \bmod n_i \equiv \mathbf{j}_i$. As such, the smallest value t may have is the least common multiple of every l_i. For any smaller non-zero value, there is some dimension i for which $(\mathbf{j}_i + t \cdot g_i) \bmod n_i \not\equiv \mathbf{j}_i$. By the properties of modular addition, it is clear that every cycle has the same length. Therefore, the number of cycles of length t is $\frac{N}{t}$.

This is rewritten as follows. Observe that the only possible values for l_i are divisors of n_i. For each divisor f_i of n_i, there are $\phi(\frac{n_i}{f_i})$ values for which $f_i = l_i$. As this is independent in each dimension, this is used to derive the following equation for the number of necklaces:

$$|\mathcal{N}_q^{\overline{\mathbf{n}}}| = \frac{1}{N} \sum_{f_1|n_1} \phi(f_1) \sum_{f_2|n_2} \phi(f_2) \ldots \sum_{f_d|n_d} \phi(f_d)\, q^{\frac{N}{\operatorname{lcm}(f_1, f_2, \ldots, f_d)}}.$$

The necklace counting formula is used to compute the number of Lyndon words through repeated application of the Möbius inversion formula, giving:

$$L_q^{\overline{\mathbf{n}}} = \sum_{f_1|n_1} \mu\left(\frac{n_1}{f_1}\right) \sum_{f_2|n_2} \mu\left(\frac{n_2}{f_2}\right) \dots \sum_{f_d|n_d} \mu\left(\frac{n_d}{f_d}\right) |\mathcal{N}_q^{f_1,f_2\dots f_d}|$$

Related to the concept of aperiodic necklaces are *atranslational necklaces*. A necklace $\widehat{w}$ is atranslational if there exists no cyclic shift $g \in Z_{\overline{\mathbf{n}}}$ such that $g \neq (n_1, n_2, \dots, n_d)$ and $\langle \widehat{w} \rangle_g = \langle \widehat{w} \rangle$. Note that while every atranslational word is aperiodic, not every aperiodic word is atranslational. Lemma 5 formally characterises the aperiodic words that are not atranslational.

Lemma 5. *Every word $\bar{w} \in \mathbf{L}_q^{\overline{\mathbf{n}}}$ is either in $\mathbf{A}_q^{\overline{\mathbf{n}}}$ or of the form $\bar{u}^p : \bar{u}^p \cdot g : \dots : \bar{u}^p \cdot g^{t-1}$ where:*

- *g is a translation where $g_d = p$ and there exists no translation $r < g$ where $\langle \bar{u}^p \rangle_r = \bar{u}^p$.*
- *$\bar{u} \in \mathbf{L}_q^{(r/p, n_d-1, \dots, n_1)}$. $t = \frac{n_d}{r}$ and is the smallest value greater than 0 such that $g^t = I$.*

Proof. For the sake of contradiction let $\bar{w} \in \mathbf{L}_q^{\overline{\mathbf{n}}}$ be an aperiodic word that is neither atranslational nor of the form $\bar{u}^p : \bar{u}^p \cdot g : \dots : \bar{u}^p \cdot g^{t-1}$ for $\bar{u} \in \mathbf{L}_q^{(r/p, n_d-1, \dots, n_1)}$. As $\bar{w}$ is not atranslational, let g be the translation such that $\bar{w} = \langle \bar{w} \rangle_g$. Further let $\bar{u}$ be the prefix of $\bar{w}$ corresponding to the first g_d slices. If $\bar{u} \notin \mathbf{L}_q^{(r/p, n_d-1, \dots, n_1)}$ then $\bar{u}$ has some period which is also a period of $\bar{w}$. Otherwise note that $\langle \bar{w} \rangle = \bar{w}$. Therefore as $\langle \bar{w} \rangle_g = \bar{w}$, $\langle \bar{w}_{[g_d+1, 2g_d]} \rangle_{(g_1, g_2, \dots, g_{d-1})} = \bar{u}$. More generally, $\langle \bar{w}_{[(l-1)\cdot g_d+1, l\cdot g_d]} \rangle_{(g_1, g_2, \dots, g_{d-1})^l} = \bar{u}$. This allows $\bar{w}$ to be written as $\bar{u} : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})} : \dots : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})^{t-1}}$. Note that if $t < \frac{n_d}{g_d}$ then $\langle \bar{w} \rangle_g = \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})} : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})^2} : \dots : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})^{t+1}}$, therefore $\langle \bar{w} \rangle_g = \bar{w}$ if and only if $\bar{u} = \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})}$. If $\bar{u} = \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})}$, then $\bar{w} = \bar{u} : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})} : \dots : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})^t} = \bar{u}^t$. Hence in this case $\bar{w}$ would be periodic. Therefore for $\bar{w}$ to be aperiodic and not a translational it must be of the form $\bar{u} : \langle \bar{u} \rangle_{(g_1, \dots, g_d)} : \dots : \langle \bar{u} \rangle_{(g_1, \dots, g_d)^{t-1}}$. In the other direction, if $\bar{w} = \bar{u} : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})} : \dots : \langle \bar{u} \rangle_{(g_1, g_2, \dots, g_{d-1})}$ and $\bar{u} \in \mathbf{A}_q^{(n_1, n_2, \dots, n_{d-1}, r)}$ then $\bar{w} \in \mathbf{A}_q^{\overline{\mathbf{n}}}$. Similarly if $\bar{u} \notin \mathbf{A}_q^{(n_1, n_2, \dots, n_{d-1}, r)}$ it must be in $\mathbf{L}_q^{(n_1, n_2, \dots, n_{d-1}, r)}$. □

Following the characterisation of translational Lyndon words given by Lemma 5, the next obvious question is how to count the number of atranslational words. To do so two further results are needed to reduce the complexity of the counting problem. Lemmas 6 and 7 provide an outline for how to reduce the number of atranslational words that need to be counted.

Lemma 6. *Let $\bar{a}, \bar{b} \in \mathbf{L}_q^{\overline{\mathbf{n}}}$. Given any integer r and translation $g \in Z_{(n_1, n_2, \dots, n_{d-1})}$ such that $g^t = I$, if $\bar{a}^r : \langle \bar{a}^r \rangle_g : \dots : \langle \bar{a}^r \rangle_{g^{t-1}} \in \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, m}$ then either $\bar{b}^r : \langle \bar{b}^r \rangle_g : \dots : \langle \bar{b}^r \rangle_{g^{t-1}} \in \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, m}$ or $\bar{b} = \bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1}}$ and $g'_i + g_i \bmod n_i \equiv 0$ for all $i \in [d]$.*

Proof. For the sake of contradiction assume that $\bar{b}^r : \langle \bar{b}^r \rangle_g : \dots : \langle \bar{b}^r \rangle_{g^{t-1}} \in \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, m}$ while $\bar{a}^r : \langle \bar{a}^r \rangle_g : \dots : \langle \bar{a}^r \rangle_{g^{t-1}} \notin \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, m}$. As $\bar{b}^r : \langle \bar{b}^r \rangle_g : \dots : \langle \bar{b}^r \rangle_{g^{t-1}} \in \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, m}$, g must be some operation such that $g^l \neq I$ for any $l < t$, as otherwise either $\bar{b}^r : \langle \bar{b}^r \rangle_g : \dots : \langle \bar{b}^r \rangle_{g^{t-1}}$ would be periodic, or there would exist some translation smaller than $\bar{b}^r : \langle \bar{b}^r \rangle_g : \dots : \langle \bar{b}^r \rangle_{g^{t-1}}$. Note that $\bar{a}^r : \langle \bar{a}^r \rangle_g : \dots : \langle \bar{a}^r \rangle_{g^{t-1}}$ must be periodic, as otherwise it would belong to $\mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, m}$. As $\bar{a}$ is in $\mathbf{L}_q^{\overline{\mathbf{n}}}$, either $\bar{a}$ is atranslational or $\bar{a} = \bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1}}$ for some atranslational word $\bar{c}$. If $\bar{a}$ is atranslational, then there is no translation g such that $\bar{a}^r : \langle \bar{a}^r \rangle_g : \dots : \langle \bar{a}^r \rangle_{g^{t-1}}$ is periodic without $\bar{b}^r : \langle \bar{b}^r \rangle_g : \dots : \langle \bar{b}^r \rangle_{g^{t-1}}$ being periodic. On the other hand, if $\bar{a} = \bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'}$ then $(\bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1})^r} : (\bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1})^r + g} : \dots : (\bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1})^r + g^{t-1}}$ must be periodic. For any value of g' where $g' + g \neq I$, $(\bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1})^r} : (\bar{c}^{r'} : \langle \bar{c}^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1})^r + g} : \dots : (\bar{c}^{r'} : \langle c^{r'} \rangle_{g'} : \dots : \langle \bar{c}^{r'} \rangle_{g'^{t'-1})^r + g^{t-1}}$ must be aperiodic. □

Lemma 7. *Let $\overline{\mathbf{n}}$ be a vector of dimensions. Given some value f which is a factor of n_d, and value c which is a factor of f, for any word $\bar{a} \in \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, c}$ such that $\bar{a}^r : \langle \bar{a}^r \rangle_g : \dots : \langle \bar{a}^r \rangle_{g^{t-1}} \in \mathbf{L}_q^{\overline{\mathbf{n}}}$ there exists some word $\bar{b} \in \mathbf{L}_q^{n_1, n_2, \dots, n_{d-1}, f}$ such that $\bar{a}^r : \langle \bar{a}^r \rangle_g : \dots : \langle \bar{a}^r \rangle_{g^{t-1}} = \bar{b} : \langle \bar{b} \rangle_{g'} : \dots : \langle \bar{b} \rangle_{g'^{t'-1}}$ where $r \cdot c \leq f$.*

Proof. This claim is shown by considering two cases based on the value of r relative to f. The first case is when $r = \frac{f}{c}$. In this case let $g' = g$ and $\bar{b} = \bar{a}^{r-1} : \langle \bar{a} \rangle_g$. Clearly the Lyndon word $\bar{a}^r : \langle \bar{a}^r \rangle_g : \ldots : \langle \bar{a}^r \rangle_{g^{t-1}}$ is equivalent to $\bar{b} : \langle \bar{b} \rangle_g : \ldots : \langle \bar{b} \rangle_{g^{t-1}}$. In the second case $r < \frac{f}{c}$. If $c \cdot r$ is a factor of f, then either the word $\bar{a}^r : \langle \bar{a}^r \rangle_g : \ldots : \langle \bar{a} \rangle_{g^{f/(r \cdot c)}} \in \mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},f}$ or $\bar{a}^r : \langle \bar{a}^r \rangle_g : \ldots : \langle \bar{a}^r \rangle_{g^{t-1}}$ is periodic, contradicting the initial assumption. If $c \cdot r$ is not a factor of f, then let $r' = \frac{f}{c} \bmod r$ and $t' = \lfloor \frac{f}{cr} \rfloor$. If $\bar{a}^r : \langle \bar{a}^r \rangle_g : \langle \bar{a}^{r'} \rangle_{g^{t'}}$ is not atranslational then $\bar{a}^r : \langle \bar{a}^r \rangle_g : \ldots : \langle \bar{a}^r \rangle_{g^{t-1}}$ must be periodic with a period in dimension d of at least f. Hence $\bar{a}^r : \langle \bar{a}^r \rangle_g : \langle \bar{a}^{r'} \rangle_{g^{t'}} \in A_q^{n_1,n_2,\ldots,n_{d-1},f}$. □

In order to use these characterisations to relate the number of Lyndon words to the number of atranslational words it is important to count the number of possible translations. To this end the set $\mathbf{G}(l, \overline{\mathbf{n}}) = \{(x_1, x_2, \ldots, x_{d-1}) \in [\overline{\mathbf{n}}] : x_i^{n_d/l} \bmod n_i \equiv 0$, and for some dimension i, there exists no value of $j \in [\frac{n_d}{l} - 1]$ such that $x_i^j \bmod n_i \equiv 0\}$ is introduced. This set counts the number of possible translations of a d-dimensional atranslational word that may be used to build a d-dimensional Lyndon word. The following Lemma provides an important step in the computation of the number of $d-1$-dimensional atranslational words that can be used to build a d-dimensional Lyndon word.

Lemma 8. *Let* $\mathbf{G}(l, \overline{\mathbf{n}}) = \{(x_1, x_2, \ldots, x_{d-1}) \in [\overline{\mathbf{n}}] : x_i^{n_d/l} \bmod n_i \equiv 0$, *and for some dimension* i, *there exists no value of* $j \in [\frac{n_d}{l} - 1]$ *such that* $x_i^j \bmod n_i \equiv 0\}$. *Given some pair of translations* $t, s \in \mathbf{G}(l, (n_1, n_2, \ldots, n_{d-1}))$, $(t_1, t_2, \ldots, t_{d-2}, \frac{n_{d-1}}{l}) \in \mathbf{G}(l, \overline{\mathbf{n}})$ *if and only if* $(s_1, s_2, \ldots, s_{d-2}, \frac{n_{d-1}}{l}) \in \mathbf{G}(l, \overline{\mathbf{n}})$.

Proof. For the sake of contradiction, assume that $(t_1, t_2, \ldots, t_{i-1}\frac{n_i}{l}) \in G(1, (n_1, n_2, \ldots, n_{i+1}))$ and $(s_1, s_2, \ldots, s_{i-1}\frac{n_i}{l}) \notin \mathbf{G}(1, (n_1, n_2, \ldots, n_{i+1}))$. There are two possible cases to consider. Either, for every $a \in [i-1]$, there exists some $j < n_{i+1}$ such that $s_a^j \bmod n_a \equiv 0$ or, for some $a \in [i-1]$ $s_a^{n_{i+1}} \bmod n_a \not\equiv 0$.

In the first case, observe that as $n_{i+1} \geq n$, either $n_{i+1} = n_i$ and $l = 1$ or $\left(\frac{n_i}{l}\right)^{n_{i+1}} \bmod n_i \equiv 0$ and n_{i+1} is co-prime to n_{i+1}. In either case, there will exist at least dimension a for which there does not exist any t_a value of j where $j < n_{i+1}$ such that $s_a^j \bmod n_a \equiv 0$.

This leaves the second case, that there must be some dimension a where $t_a^{n_{i+1}} \bmod n_a \equiv 0$ while $s_a^{n_{i+1}} \bmod n_a \not\equiv 0$. For this to be true, it must be the case that n_{i+1} is co-prime to $\frac{n_i}{l}$, as otherwise $n_{i+1} = \frac{n_i}{l}$. If n_{i+1} is co-prime to n_i then note that for $t_a^{n_{i+1}} \bmod n_a \equiv t_a^{n_i/l}$, $t_a = 0$ for every dimension a. However, this leads to a contradiction, as $(0, 0, \ldots, 0) \notin \mathbf{G}(1, (n_1, n_2, \ldots, n_{i+1}))$. Therefore if $(t_1, t_2, \ldots, t_{i-1}\frac{n_i}{l}) \in G(1, (n_1, n_2, \ldots, n_{i+1}))$ then $(s_1, s_2, \ldots, s_{i-1}\frac{n_i}{l}) \in G(1, (n_1, n_2, \ldots, n_{i+1}))$. □

Lemma 8 provides the basis for generalising the set $\mathbf{G}(l, \overline{\mathbf{n}})$ to count the number of ways a $d-i$-dimensional atranslational word can be used to form a d-dimensional Lyndon word. More explicitly, consider the i-dimensional atranslational word $\bar{w}$. To use $\bar{w}$ as the translational base of some d-dimensional Lyndon word, note that there must be some translation applied to $\bar{w}$ at every dimension from i to d. Let $\bar{u} = (\bar{w} : \langle \bar{w} \rangle_g : \ldots : \langle \bar{w} \rangle_{g^t}) : \langle (\bar{w} : \langle \bar{w} \rangle_g : \ldots : \langle \bar{w} \rangle_{g^t}) \rangle_h : \ldots : \langle (\bar{w} : \langle \bar{w} \rangle_g : \ldots : \langle \bar{w} \rangle_{g^t}) \rangle_{h^s}$. For $\bar{u}$ to be a Lyndon word, h must not be $(g_1, g_2, \ldots, g_i, n_d/l)$ as $(\bar{w} : \langle \bar{w} \rangle_g : \ldots : \langle \bar{w} \rangle_{g^t}) = \langle (\bar{w} : \langle \bar{w} \rangle_g : \ldots : \langle \bar{w} \rangle_{g^t}) \rangle_{(g_1,g_2,\ldots,g_i,n_d/l)}$.

Using this observation, the following two functions are needed to count the number possible ways an i-dimensional atranslational word can be used to build a d-dimensional word. Let $I(i, l, \overline{\mathbf{n}})$ return the number of dimensions $j \in [i, d]$ where there exists some translation $g \in \mathbf{G}(l_j, (n_1, n_2, \ldots, n_j))$ such that $(g_1, g_2, \ldots, g_{j-1}, \frac{n_j}{l}, 1, 1, \ldots, 1) \in \mathbf{G}(1, \overline{\mathbf{n}})$, where l_j equals 1 if $j > i$ and l otherwise.

The function $H(i, l, \overline{\mathbf{n}}, d)$ is used to return the number of possible sets of translations that can be used to build a d-dimensional Lyndon word from $\bar{w}$. Note that each such set requires $d - i$ translations if $l = n_i$, or $d - i + 1$ translations if $l < n_i$. If $i = d$ then the value of $H(i, l, \overline{\mathbf{n}}, d)$ is either 1, if $l = n_d$, or $|\mathbf{G}(l, \overline{\mathbf{n}})|$ otherwise. If $i < d$, the number of possible translations of dimensions d equals the size of $\mathbf{G}(1, \overline{\mathbf{n}})$ minus the number of dimensions where the translation in the lower dimension can be cancelled out by some translation in a higher dimension. Note that if any translation in dimension i can be cancelled out by some translation in dimensions $j > i$, then

following Lemma 8 every translation can be. Therefore the value of $H(i,l,\overline{\mathbf{n}},d)$ is given by the equation

$$H(i,l,\overline{\mathbf{n}},d) = \Big\{(|\mathbf{G}(1,\overline{\mathbf{n}})| - (I(i,l,\overline{\mathbf{n}}))) \cdot (H(i,l,(n_1,n_2,\ldots,n_{d-1}),d-1))$$

Using these results, the number atranslational words of dimensions $\overline{\mathbf{n}}$ are counted in terms of atranslational words of smaller dimensions and Lyndon words of dimensions $\overline{\mathbf{n}}$. Lemma 9 shows how to express the number of Lyndon words in terms of atranslational words. Lemma 1 builds on this to show how to count the number of atranslational words using Lemma 9.

Lemma 9. *The number of d-dimensional Lyndon words is given in terms of atranslational words as:*

$$\mathbf{L}_q^{\overline{\mathbf{n}}} = |A_q^{\overline{\mathbf{n}}}| + \sum_{i\in[d]}\sum_{l|n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)|\mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l}| \cdot H(i,l,\overline{\mathbf{n}},d) & 1 < l < n_d \end{cases}$$

Proof. Note that every Lyndon word is either atranslational itself, or of the form $\bar{a}^r : \langle\bar{a}^r\rangle_g : \ldots : \langle\bar{a}^r\rangle_{g^{t-1}}$ for some $\bar{a} \in \mathbf{A}^{n_1,n_2,\ldots,n_{d-1},f}$. Following Lemma 7, every Lyndon word of the form $\bar{a}^r : \langle\bar{a}^r\rangle_g : \ldots : \langle\bar{a}^r\rangle_{g^{t-1}}$ is rewritten as $\bar{b} : \langle\bar{b}\rangle_g : \ldots : \langle\bar{v}\rangle_{g^{t-1}}$ for some $\bar{b} \in \mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l\cdot r}$. Let $\bar{a}$ be an atranslational word of dimensions $(n_1,n_2,\ldots,n_{d-1},l)$. For Lyndon words with a d-dimensional translational period there are three cases to consider. If $l = n_d$, then $\bar{a} \in \mathbf{A}_q^{n_1,n_2,\ldots,n_d}$. If $\frac{n_d}{l}$ is prime then for every cyclic shift of $X = (x_1,x_2,\ldots,x_{d-1})$ where $x_i \in 1\ldots n_i - 1$ such that $x_i^{n_d/l} \bmod n_i \equiv 0$ and for some $i \nexists j \in 1\ldots\frac{n_d}{l}-1$, the word $\bar{a} : \langle\bar{a}\rangle_X : \ldots : \langle\bar{a}^r\rangle_{X^{(n_2/l)-1}} \in \mathbf{L}_q^{\overline{\mathbf{n}}}$. The number of words of the form $\bar{a} : \langle\bar{a}\rangle_g : \ldots : \langle\bar{a}\rangle_{g^{(n_d/l)-1}} \in \mathbf{L}_q^{\overline{\mathbf{n}}}$ is $|\mathbf{G}(l,\overline{\mathbf{n}})| \cdot |\mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l}|$.

In the case that $\frac{n_d}{l}$ is not prime, following Lemma 7 there exists some d' such that $\bar{b} = \bar{a} : \langle\bar{a}\rangle_g : \ldots : \langle\bar{a}\rangle_{g^{t'}}$ where $\bar{b}$ has dimensions $n_1 \times n_2 \times \ldots \times l'$. If there are at least two distinct prime factors of $\frac{n_d}{l}$, then note that $\bar{a} : \langle\bar{a}\rangle_g : \ldots : \langle\bar{a}\rangle_{g^t}$ is counted for each prime factor. Let p be the number of distinct prime factors. To avoid over counting, every word of size $n_1 \times n_2 \times \ldots \times n_{d-1} \times l$ needs to be subtracted $p-1$ times. To this end, a new function $P(t)$ is introduced to act as a correction factor.

If $p = 2$ then by setting $P(2) = -1$ the over counting is avoided. If $p = 3$, then as these words were counted three times for each prime factor, then subtracted three times $\frac{n_2}{d\cdot i}$ for each i in the set of prime factors, to avoid under counting these words $P(3)$ must return 1. One special case is when $\frac{n_d}{l}$ has a square prime factor, i^2. In this case as $\frac{n_d}{l\cdot i}$ has the same number of distinct primes, $P(\frac{n_d}{l})$ must return 0. Repeating this argument, $P(s)$ is -1 if s has an even number of prime factors, 1 if s has an odd number of prime factors, and 0 otherwise. Note that this corresponds to $-1(\mu\left(\frac{n_d}{l}\right))$ where $\mu\left(\frac{n_d}{l}\right)$ is the möbius function. Further, as $P(1) = 1$, both the prime and non-prime cases can be combined into one case.

The same arguments may be applied to the lower dimensional case. Note that the number of possible translations in this case is given by $H(i,l,\overline{\mathbf{n}},d)$. In order to account for over counting, the number of possible Lyndon words is multiplied by $\left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)$. Therefore the total number of Lyndon words of dimensions $\overline{\mathbf{n}}$ is equal to:

$$\mathbf{L}_q^{\overline{\mathbf{n}}} = |A_q^{\overline{\mathbf{n}}}| + \sum_{i\in[d]}\sum_{l|n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)|\mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l}| \cdot H(i,l,\overline{\mathbf{n}},d) & 1 < l < n_d \end{cases}$$

□

Corollary 1. *The number of atranslational words is given by:*

$$|\mathbf{A}_q^{\overline{\mathbf{n}}}| = |L_q^{\overline{\mathbf{n}}}| - \sum_{i\in[d]}\sum_{l|n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)|\mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l}| \cdot H(i,l,\overline{\mathbf{n}},d) & 1 < l < n_d \end{cases}$$

Proof. It follows from Lemma 9 that the number of translational words in

$$|\mathbf{L}_q^{\overline{\mathbf{n}}}| = \sum_{i\in[d]}\sum_{l|n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)|\mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l}| \cdot H(i,l,\overline{\mathbf{n}},d) & 1 < l < n_d \end{cases}$$

Hence the number of atranslational words is

$$|\mathbf{A}_q^{\overline{\mathbf{n}}}| = |\mathbf{L}_q^{\overline{\mathbf{n}}}| - \sum_{i\in[d]}\sum_{l|n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)|\mathbf{A}_q^{n_1,n_2,\ldots,n_{d-1},l}| \cdot H(i,l,\overline{\mathbf{n}},d) & 1 < l < n_d \end{cases}$$

□

From these equations, an upper and lower bound on the number of necklaces is derived.

Lemma 10. *The number of necklaces is bounded by* $\frac{q^N}{N} \leq |\mathcal{N}_q^{\overline{\mathbf{n}}}| \leq q^N$ *where* $\overline{\mathbf{n}}$ *is the dimension vector and* q *is the size of the alphabet.*

Proof. The upper bound comes directly as the number of possible words. Using the above equations, observe that for every word n_i, 1 is a factor. As $\phi(1) = 1$, this gives the number of necklaces as at least $\frac{q^N}{N}$. □

Lemma 11. *For two sets of necklaces* $\mathcal{N}_q^{\overline{\mathbf{n}}}$*, and* $\mathcal{N}_q^{\overline{\mathbf{m}}}$*, such that* $m_i \geq n_i$ *for every dimension* i *and* $\overline{\mathbf{n}} \neq \overline{\mathbf{m}}$*,* $|\mathcal{N}_q^{\overline{\mathbf{n}}}| < |\mathcal{N}_q^{\overline{\mathbf{m}}}|$*.*

Proof. For every necklace $x \in \mathcal{N}_q^{\overline{\mathbf{n}}}$, a new necklace x' of size $\overline{\mathbf{m}}$ such that the symbol at position $\mathbf{i}$ is:

- The symbol in x at $\mathbf{i}$ if $\mathbf{i}_j \leq n_j$ for every dimension j.
- q Otherwise.

In addition to this, observe the necklace containing only the first symbol also belongs to $\mathcal{N}_q^{\overline{\mathbf{m}}}$. Therefore $|\mathcal{N}_q^{\overline{\mathbf{n}}}| < |\mathcal{N}_q^{\overline{\mathbf{m}}}|$. □

Lemma 12. *The number of aperiodic words is bounded by* $\frac{q^N}{N} - q^{N/2} \leq L_q^{\overline{\mathbf{n}}} \leq \frac{q^N}{N}$ *where* $\overline{\mathbf{n}}$ *is the dimension vector and* q *is the size of the alphabet.*

Proof. The upper bound comes from the observation that every atranslational word must have exactly N representations of it. As such, there is no more than $\frac{q^N}{N}$ Transnational words. The lower bound is derived using the lower bound on the number of necklaces as a starting point. By its definition, $\mu(\frac{n_i}{f_i})$ is 1 for $n_i = f_i$. To reduce the bound, note that given two values f_i and g_i such that $f_i > g_i$ where $\mu(\frac{n_i}{f_i}) = \mu(\frac{n_i}{g_i}) = -1$, there must exist some value h_i between f_i and g_i such that $\mu(\frac{n_i}{h_i}) = 1$. As the number of necklaces increases monotonically, the number of necklaces with a length of h_i in dimension i is more than those with a length of g_i. Therefore the largest negative value is $q^{N/2}$. This gives a lower bound on the number of Atranslational words of $\frac{q^N}{N} - q^{N/2}$. □

5.2 Ranking Multidimensional Necklaces

In this section we provide a polynomial-time algorithm for *ranking* multidimensional necklaces. Ranking classes of one-dimensional cyclic words such as Lyndon words, necklaces and bracelets has received a lot of attention in the past [2, 25, 26, 33]. Recall that the rank of a necklace $\widehat{w}$ in the set $|\mathcal{N}_q^{\overline{\mathbf{n}}}|$ is the number of necklaces smaller than or equal to $\widehat{w}$ under some ordering, in this case the ordering given in Definition 3. More broadly, we can take any word $\bar{v}$ and determine the number of necklaces that are represented by a word smaller than $\bar{v}$ using the same ordering. In this case, the smallest necklace greater than or equal to $\bar{v}$ is determined using the $NextNecklace$ algorithm given in Theorem 6. For the remainder of this section we assume that we are finding

the rank of some word that is the canonical representation of a necklace. Before we provide a high level overview of how this problem is tackled, we need to define a method of comparing two words of different sizes. In this section, two words $\bar{w} \in \Sigma^{\overline{\mathbf{n}}}$ and $\bar{u} \in \Sigma^{\overline{\mathbf{f}}}$ are compared if and only if $n_i \bmod f_i \equiv 0$ for every $i \in [d]$. As such, given such a pair of words $\bar{u}^{\overline{\mathbf{n}}/\overline{\mathbf{f}}}$ is used to denote the word $\bar{u}'$ where $\bar{u}'_{(i_1,i_2,\ldots,i_d)} = \bar{u}_{(i_1 \bmod n_1, i_2 \bmod n_2, \ldots, i_d \bmod n_d)}$. Using this notation, a comparison between word $\bar{w}$ and $\bar{u}$ is given as:

Definition 8. *Let $\bar{u} \in \Sigma^{(f_1,f_2,\ldots,f_d)}$, and $\bar{v} \in \Sigma^{(n_1,n_2,\ldots,n_d)}$ where $n_i \bmod f_i \equiv 0$. $\bar{u} < \bar{v}$ if and only if $\bar{u}^{\overline{\mathbf{n}}/\overline{\mathbf{f}}} < \bar{v}$ following Definition 3. Similarly, $\bar{u} > \bar{v}$ if and only if $\bar{u}^{\overline{\mathbf{n}}/\overline{\mathbf{f}}} > \bar{v}$.*

At a high level, the ranking algorithm for a word $\bar{w}$ works by first determining the number of words of size $f_1 \times f_2 \times \ldots \times f_d$ smaller than $\bar{w}$, denoted $T(\bar{w}, f_1, f_2, \ldots, f_d)$, for every f_i that is factor of n_i. This value is transformed, first from $T(\bar{w}, f_1, f_2, \ldots, f_d)$ to the number of aperiodic words smaller than $\bar{w}$, denoted $L(\bar{w}, f_1, f_2, \ldots, f_d)$, and finally to the number of atranslational words smaller than $\bar{w}$, $A(\bar{w}, f_1, f_2, \ldots, f_d)$. The set $A(\bar{w}, f_1, f_2, \ldots, f_d)$ is then translated into the rank of $\bar{w}$ within the set of atranslational necklaces $A_q^{(f_1,f_2,\ldots,f_d)}$, denoted $RA(\bar{w}, f_1, f_2, \ldots, f_d)$. This rank is than used to calculate the rank within the set of Lyndon words $RL(\bar{w}, f_1, f_2, \ldots, f_d)$. Finally, this rank is translated to the necklace rank $RN(\bar{w}, f_1, f_2, \ldots, f_d)$. Lemmas 13, and 14 show how to transform the size of the sets $T_{\bar{w},f_1,f_2,\ldots,f_d}$ into the size of $A(\bar{w}, n_1, n_2, \ldots, n_d)$. Lemmas 15, 16 and 17 show how to transform the size of the sets $A(\bar{w}, f_1, f_2, \ldots, f_d)$ into the value $RN(\bar{w}, n_1, n_2, \ldots, n_d)$.

In order to compute the size of $T(\bar{w}, f_1, f_2, \ldots, f_d)$, the set is partitioned into the subsets $B(\bar{w}, g, j, f_1, f_2, \ldots, f_d) \subseteq T(\bar{w}, f_1, f_2, \ldots, f_d)$. Here $B(\bar{w}, g, j, f_1, f_2, \ldots, f_d)$ contains the set of words $\bar{v} \in T(\bar{w}, f_1, f_2, \ldots, f_d)$ where: (1) g is the smallest translation such that $\langle \bar{v} \rangle_g < \bar{w}$ and (2) j is the length of the longest shared prefix between $\langle \bar{v}^{\overline{\mathbf{n}}/\overline{\mathbf{f}}} \rangle_g$ and $\bar{w}'$, i.e. the largest value such that $\left(\langle \bar{v}^{\overline{\mathbf{n}}/\overline{\mathbf{f}}} \rangle_g\right)_{[1:j]} = \bar{w}_{[1:j]}$. The size of each set $B(\bar{w}, g, j, f_1, f_2, \ldots, f_d)$ is computed by considering the structure of the words in $B(\bar{w}, g, j, f_1, f_2, \ldots, f_d)$. This requires the size of two further sets to be computed, the number of non-cyclic words where every suffix is greater than $\bar{w}$, and the number of words of dimensions $(f_1, f_2, \ldots, f_{d-1})$ that are smaller than $\bar{w}_{j+1}$. The first of these sets is the more technical, requiring a new recursive technique to be built which is provided in Subsection 5.2.1.

The remainder of this section is structured as follows. Lemmas 13 to 17 provide the theoretical tools needed to rank necklaces. Following these Lemmas, an overview of the method to compute the size of $T(\bar{w}, f_1, f_2, \ldots, f_d)$ is provided. Subsection 5.2.1 covers the main sub method used in the ranking process. Finally Theorem 5 is restated and formally proven.

Lemma 13. *The size of $L(\bar{w}, n_1, n_2, \ldots, n_d)$ is computed in terms of $T(\bar{w}, f_1, f_2, \ldots, f_d)$ using the equation:*

$$L(\bar{w}) = \sum_{f_1|n_1} \mu\left(\frac{n_1}{f_1}\right) \sum_{f_2|n_2} \mu\left(\frac{n_2}{f_2}\right) \cdots \sum_{f_d|n_d} \mu\left(\frac{n_d}{f_d}\right) T(\bar{w}, f_1, f_2, \ldots, f_d)$$

Proof. Observe that every word in $T(\bar{w}, n_1, n_2, \ldots, n_d)$ is either aperiodic, in which case it is in $L(\bar{w}, n_1, n_2, \ldots, n_d)$, or periodic, in which case it is in $L(\bar{w}, f_1, f_2, \ldots, f_d)$ where f_i is a factor of n_i. Following the same arguments as given in Section 5.1, the size of $T(\bar{w}, n_1, n_2, \ldots, n_d)$ is equal to $\sum_{f_1|n_1} \sum_{f_2|n_2} \cdots \sum_{f_d|n_d} |L(\bar{w}, f_1, f_2, \ldots, f_d)|$. By repeated application of the Möbius inversion formula, the size of $L(\bar{w}, n_1, n_2, \ldots, n_d)$ is computed as:

$$L(\bar{w}, n_1, n_2, \ldots, n_d) = \sum_{f_1|n_1} \mu\left(\frac{n_1}{f_1}\right) \sum_{f_2|n_2} \mu\left(\frac{n_2}{f_2}\right) \cdots \sum_{f_d|n_d} \mu\left(\frac{n_d}{f_d}\right) T(\bar{w}, f_1, f_2, \ldots, f_d)$$

□

Lemma 14. *The size of $A(\bar{w}, n_1, n_2, \ldots, n_d)$ equals*

$$|L(\bar{w}, n_1, n_2, \ldots, n_d)| - \sum_{i \in [d]} \sum_{l|n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t)\right) \left(-\mu\left(\frac{n_i}{l}\right)\right) |A_q^{n_1,n_2,\ldots,n_{d-1},l}| \cdot H(i, l, \overline{\mathbf{n}}, d) & 1 < l < n_d \end{cases}$$

Proof. Following the arguments given in Lemma 9, observe that any Lyndon word in $L(\bar{w}, n_1, n_2, \ldots, n_d)$ is either be atranslational, or of the form $\bar{a} : \langle \bar{a} \rangle_g : \ldots : \langle \bar{a} \rangle_{g^{t-1}}$. In the latter case, let $l = |\bar{a}|_d$. Note that $\bar{a}$ must be either in $A(\bar{w}_{[1,l]}, n_1, n_2, \ldots, n_{d-1}, l)$, if $l > 1$ or $L(\bar{w}_1)$ if $l = 1$. Repeating the same arguments as in Lemma 9 allows the size of $A(\bar{w}, n_1, n_2, \ldots, n_d)$ to be written as:

$$|L(\bar{w}, n_1, n_2, \ldots, n_d)| - \sum_{i \in [d]} \sum_{l | n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t) \right) \left(-\mu\left(\frac{n_i}{l}\right) \right) |A_q^{n_1, n_2, \ldots, n_{d-1}, l}| \cdot H(i, l, \overline{\mathbf{n}}, d) & 1 < l < n_d \end{cases}$$

□

Lemma 15. *The rank* $RA(\bar{w}, n_1, n_2, \ldots, n_d) = \frac{1}{N} |A(\bar{w}, n_1, n_2, \ldots, n_d)|$.

Proof. Observe that any atranslational necklace of dimensions $\overline{\mathbf{n}}$ has exactly N representations. Therefore the number of atranslational necklaces smaller than $\bar{w}$ is $\frac{1}{N} A(\bar{w})$. Hence

$$RA(\bar{w}, n_1, n_2, \ldots, n_d) = \frac{1}{N} A(\bar{w}, n_1, n_2, \ldots, n_d).$$

□

In order to used the rank $RA(\bar{w}, n_1, n_2, \ldots, n_d)$ to get the rank $RL(\bar{w}, n_1, n_2, \ldots, n_d)$, one additional observation is needed. Let $\bar{w}$ be a translational, aperiodic word, with a translational period of $(g_1, g_2, \ldots, g_{d-1}, \frac{n_i}{l}, 1, 1, \ldots, 1)$ where $g \in \mathbf{G}(l, n_1, n_2, \ldots, n_i)$ for some $i \in [d]$. Let $\bar{u}$ be the word of dimensions g such that $\bar{u}_{\bar{\mathbf{i}}} = \bar{w}_{\bar{\mathbf{i}}}$. Further, let $\bar{u}[j]$ be the Lyndon word of dimensions $(g_1, g_2, \ldots, g_{i-1}, \frac{n_i}{l}, n_{i+1}, \ldots, n_j)$ such that $\bar{u}[j]_{\bar{\mathbf{i}}} = \bar{w}_{\bar{\mathbf{i}}}$. Note that $\bar{u}[j]$ can be written as $\bar{u}[j]\bar{u}[j-1] : \langle \bar{u}[j-1] \rangle_{r_j} : \ldots : \langle \bar{u}[j-1] \rangle_{r_j^{n_j - 1}}$, for some $r_j \in \mathbf{G}(l_j, (n_1, n_2, \ldots, n_j))$ where $l_j = 1$ if $j > i$ and 0 otherwise. Observe that the number of Lyndon word made from $\bar{\bar{u}}$ of dimensions $\overline{\mathbf{n}}$ that are smaller than $\bar{w}$ is equal to the sum of the number of translations in $\mathbf{G}(l_j, n_1, n_2, \ldots, n_j)$ multiplied by $H(i, l_i, \overline{\mathbf{n}})$. Let $S(g, l, (n_1, n_2, \ldots, n_j))$ return the number of translations in $\mathbf{G}(l, (n_1, n_2, \ldots, n_j))$ smaller than g. To this end let $U(\bar{w})$ return either:

- 0 if $\bar{w}$ is either atranslational or periodic.
- $\sum_{j=i}^{d} \begin{cases} S(r_j, l, (n_1, n_2, \ldots, n_j)) & j = i \\ S(r_j, 1, (n_1, n_2, \ldots, n_j)) & otherwise. \end{cases}$ if $\bar{w}$ is a Lyndon word with a translational period of g.

Using $U(\bar{w})$, the number of Lyndon words can be computed from $RA(\bar{w}, n_1, n_2, \ldots, n_d)$ as follows.

Lemma 16. *The rank*

$$RL(\bar{w}, n_1, n_2, \ldots, n_d) = RA(\bar{w}, n_1, n_2, \ldots, n_d) + U(\bar{w}) +$$

$$\sum_{i \in [d]} \sum_{l | n_i} \begin{cases} 0 & l = n_i \\ \left(\prod_{t=i+1}^{d-1} -\mu(n_t) \right) \left(-\mu\left(\frac{n_i}{l}\right) \right) |RA(\bar{w}_{[1,l]}, n_1, n_2, \ldots, n_{i-1})| \cdot H(i, l, \overline{\mathbf{n}}, d) & 1 < l < n_d \end{cases}$$

Proof. Note that every necklace smaller than $\bar{w}$ is either atranslational, in which case it is counted by $RA(\bar{w}, n_1, n_2, \ldots, n_d)$, or is translational. In the latter case following Lemma 9 for each necklace counted by $RA(\bar{w}_{[1,l]}, n_1, n_2, \ldots, n_{d-1}, l)$, there are $H(i, l, \overline{\mathbf{n}})$ translational necklace counted by $RL(\bar{w}, n_1, n_2, \ldots, n_d)$. Further, if $\bar{w}$ is a translational Lyndon word of the form $\bar{v} : \langle \bar{v} \rangle_g : \ldots : \langle \bar{v} \rangle_g$, then there are there are $U(\bar{w})$ Lyndon words of the form $\bar{v} : \langle \bar{v} \rangle_g : \ldots : \langle \bar{v} \rangle_g$ where $\bar{v}_{\bar{\mathbf{i}}} = \bar{w}_{\bar{\mathbf{i}}}$ for every $\bar{\mathbf{i}} \in [|\bar{v}|]$. Following Lemma 9 $RL(\bar{w}, n_1, n_2, \ldots, n_d)$ is counted in terms of $RA(\bar{w}, n_1, n_2, \ldots, n_{d-1}, l)$ as:

$$RL(\bar{w}, n_1, n_2, \ldots, n_d) = RA(\bar{w}, n_1, n_2, \ldots, n_d) + U(\bar{w}) +$$

$$\sum_{i\in[d]}\sum_{l|n_i}\begin{cases}0 & l=n_i\\ \left(\prod_{t=i+1}^{d-1}-\mu(n_t)\right)\left(-\mu\left(\frac{n_i}{l}\right)\right)|RA(\bar{w}_{[1,l]},n_1,n_2,\ldots,n_{i-1})|\cdot H(i,l,\overline{\mathbf{n}},d) & 1<l<n_d\end{cases}$$

□

Lemma 17. *The rank* $RN(\bar{w},n_1,n_2,\ldots,n_d)=\sum_{f_1|n_1}\sum_{f_2|n_2}\ldots\sum_{f_d|n_d}RL(\bar{w},f_1,f_2,\ldots,f_d)$.

Proof. Observe that every necklace counted by $RN(\bar{w},n_1,n_2,\ldots,n_d)$ has a period of $\overline{\mathbf{m}}$ where m_i is a factor of $|\bar{w}|_i$ for every $i\in 1\ldots d$. As $RL(\bar{w},f_1,f_2,\ldots,f_d)$ counts the rank among aperiodic necklaces of size $f_1\times f_2\times\ldots\times f_d$, the rank among necklaces is given by:

$$RN(\bar{w},n_1,n_2,\ldots,n_d)=\sum_{f_1|n_1}\sum_{f_2|n_2}\ldots\sum_{f_d|n_d}RL(\bar{w},f_1,f_2,\ldots,f_d)$$

□

This leaves the challenge of computing the size of $T(\bar{w},f_1,f_2,\ldots,f_d)$. To this end, $T(\bar{w},f_1,f_2,\ldots,f_d)$ is partitioned into the sets $B(\bar{w},g_d,j,f_1,f_2,\ldots,f_d)$ such that $B(\bar{w},g_d,j,f_1,f_2,\ldots,f_d)$ contains every word $\bar{v}\in T(\bar{w},f_1,f_2,\ldots,f_d)$ where:

- g_d is the smallest translation in dimension d of $\bar{v}$ such that $\langle\bar{v}\rangle_{(n_1,n_2,\ldots,n_{d-1},g_d)}<\bar{w}$.
- j is the largest value such that $(\langle\bar{v}'\rangle_{(n_1,n_2,\ldots,n_{d-1},g_d)})_{[1:j]}=\bar{w}'_{[1:j]}$.

For notation let $g=(n_1,n_2,\ldots,n_{d-1},g_d)$. To compute the size of $B(\bar{w},g_d,j,f_1,f_2,\ldots,f_d)$, there are two cases to consider based on the values of g_d and j.
Case 1: $g_d+j\leq n_d$. In this case every word $\bar{v}\in B(\bar{w},g_d,j,f_1,f_2,\ldots,f_d)$ is written as $\bar{a}:(\langle\bar{w}_{[1,j]}:\bar{b}\rangle_\theta):\bar{c}$ where:

- $\bar{a}$ is a word of dimensions $(f_1,f_2,\ldots,f_{d-1},g_d)$ for which there exists no translation $r\in Z_{(f_1,f_2,\ldots,g_d)}$ such that $(\langle\bar{a}\rangle_r)_{[1:g_d-r_d]}<\bar{w}_{[1,g_d-r_d]}$.
- $\bar{b}$ is some word of dimensions $(f_1,f_2,\ldots,f_{d-1})$ that is smaller than $\bar{w}_{j+1}$.
- θ is some translation in $Z_{(f_1,f_2,\ldots,f_{d-1})}$.
- $\bar{c}$ is an unrestricted word of dimensions $(f_1,f_2,\ldots,f_{d-1},f_d-(g_d+j+1))$.

To count the number of words of this form, it is necessary to compute the number of non-cyclic words of dimensions $(f_1,f_2,\ldots,f_{d-1},i)$ where every suffix of length i is greater than $\bar{w}_{[1,i]}$. To this end the set $\beta(\bar{w},i,j,f_1,f_2,\ldots,f_{d-1})$ is introduced containing every word $\bar{u}$ where:

- The dimensions of $\bar{u}$ are $(f_1,f_2,\ldots,f_{d-1},i)$.
- There exists no translation $g\in Z_{(f_1,f_2,\ldots,f_{d-1})}$ where $\langle\bar{u}_{[i-l,i]}\rangle_g\leq\bar{w}_{[1,l]}$.
- The first j slices of $\bar{u}$ are equal to the first j slices of $\bar{w}$, i.e. $\bar{u}_{[1,j]}=\bar{w}_{[1,j]}$.

When it is clear from context $\beta(\bar{w},i,j,f_1,f_2,\ldots,f_{d-1})$ is denoted $\beta(\bar{w},i,j,\overline{\mathbf{f}})$. A method to compute the size of $\beta(\bar{w},i,j,\overline{\mathbf{f}})$ is given in subsection 5.2.1. Using $|\beta(\bar{w},i,j,\overline{\mathbf{f}})|$ as a black box, the number of possible values of $\bar{a}$ is $|\beta(\bar{w},i,j,\overline{\mathbf{f}})|$. Similarly, the number of possible values of $\bar{b}$ is given by $q^{n_1\cdot n_2\cdot\ldots\cdot n_{d-1}}-|\beta(\bar{w}_{j+1},1,0,\overline{\mathbf{f}})|-1$. The number of possible values of θ is equal to the size of the set $\mathbf{\Theta}=\{r\in Z_{\overline{\mathbf{f}}}:\nexists s\in Z_{\overline{\mathbf{f}}}$ where $s<r$ and $\langle\bar{w}\rangle_r=\langle\bar{w}\rangle_s\}$. Finally, the number of values of $\bar{c}$ is given by $q^{n_1\cdot n_2\cdot\ldots\cdot n_{d-1}\cdot(n_d-(g_d+j+1))}$. Therefore the size of $B(\bar{w},g,j,f_1,f_2\ldots,f_d)$ when $g_d+j<n_d$ is given by:

$$|\beta(\bar{w},g_d,0,\overline{\mathbf{f}})|\cdot(q^{n_1\cdot n_2\cdot\ldots\cdot n_{d-1}}-|\beta(\bar{w}_{j+1},1,0,\overline{\mathbf{f}})|-1)\cdot|\mathbf{\Theta}|\cdot q^{n_1\cdot n_2\cdot\ldots\cdot n_{d-1}\cdot(n_d-(g_d+j+1))}$$

Case 2: $g_d+j>n_d$. In this case every word $\bar{v}\in B(\bar{w},g_d,j,f_1,f_2\ldots,f_d)$ is written as $\langle\bar{w}_{[j+g_d-n_d,j]}:\bar{b}\rangle_\theta:\bar{a}:\langle\bar{w}_{[1,j+g_d-n_d]}\rangle_\theta$ where:

- $\bar{a}$ is a $f_1 \times f_2 \times \ldots \times f_{d-1}, f_d - (j+1)$ dimensional word for which there exists no translation $r \in Z_{(f_1,f_2,\ldots,f_{d-1},g_d)}$ such that $\langle \bar{a} \rangle_r < \bar{w}_{[1,g_d]}$.
- $\bar{b}$ is some word of dimensions $(n_1, n_2, \ldots, n_{d-1})$ that is smaller than $\bar{w}_{j+1}$.
- θ is a translation in the set $\mathbf{\Theta} = \{r \in Z_{(f_1,f_2,\ldots f_{d-1})} : \nexists s \in Z_{(f_1,f_2,\ldots f_{d-1})}$ where $s < r$ and $\langle \bar{w}_{[1:j]} \rangle_r = \langle \bar{w}_{[1:j]} \rangle_s\}$.

The number of possible values of θ is equal to the size of the set $\mathbf{\Theta}$ as in Case 1. The number of possible values of $\bar{b}$ in this case is somewhat more complicated than in Case 1. Let t be the length of the longest suffix of $\bar{w}_{[j+g_d-n_d,j]}$ such that $\bar{w}_{[j-t,j]} = \bar{w}_{[1,t]}$. To avoid $\langle \bar{v} \rangle_\psi$, for some $\psi \in Z_{(f_1,f_2,\ldots,f_{d-1},n_d-g_d)}$, being smaller than $\bar{w}$, $\bar{b}$ must be greater than or equal to $\bar{w}_{t+1}$. Note that the number of words greater than $\bar{w}_{t+1}$ is given by $\beta(\bar{w}_{t+1}, 1, 0, \bar{\mathbf{f}})$. Therefore the number of possible values of $\bar{b}$ as $(q^{n_1 \cdot n_2 \cdot \ldots \cdot n_{d-1} \cdot (n_d - (g_d+j+1))} - \beta(\bar{w}_{j+1}, 1, 0) - 1) - (q^{n_1 \cdot n_2 \cdot \ldots \cdot n_{d-1} \cdot (n_d - (g_d+j+1))} - \beta(\bar{w}_{t+1}, 1, 0, \bar{\mathbf{f}})) = \beta(\bar{w}_{t+1}, 1, 0, \bar{\mathbf{f}}) - \beta(\bar{w}_{j+1}, 1, 0, \bar{\mathbf{f}}) + 1$. If $\bar{b} = \bar{w}_{t+1}$, the number of possible values of $\bar{a}$ is given by $|\beta(\bar{w}, n_d + t - j, t+1, \bar{\mathbf{f}})|$. Otherwise the number of possible values of $\bar{a}$ is given by $|\beta(\bar{w}, n_d - j - 1, 0, \bar{\mathbf{f}})|$. Therefore the total number of words of the form $\langle \bar{w}_{[j+g_d-n_d,j]} : \bar{b} \rangle_\theta : \bar{a} : \langle \bar{w}_{[1,j+g_d-n_d]} \rangle_\theta$ is:

$$|\beta(\bar{w}, n_d + t - j, t+1, \bar{\mathbf{f}})| + (\beta(\bar{w}_{t+1}, 1, 0, \bar{\mathbf{f}}) - \beta(\bar{w}_{j+1}, 1, 0, \bar{\mathbf{f}})) \cdot |\beta(\bar{w}, n_d - j - 1, 0, \bar{\mathbf{f}})| \cdot |\mathbf{\Theta}|$$

5.2.1 Computing $|\beta(\bar{w}, i, j, \bar{\mathbf{f}})|$

In the method outline in Section 5.2, in order to compute the size of $T(\bar{w})$ it is necessary to compute the size of the set $\beta(\bar{w}, i, j, \bar{\mathbf{f}})$. Let $\bar{v} \in \beta(\bar{w}, i, j, \bar{\mathbf{f}})$. Observe that if $v_{j+1} > \bar{w}_{j+1}$, then for any translation $g \in Z((f_1, f_2, \ldots, f_{d-1}, j+1))$, $\bar{v}_{[1,j+1]} > \bar{w}_{[1,j+1]}$. Therefore the number of possible values of $\bar{v}_{[j+2,i]} = |\beta(\bar{w}, i-j-1, 0, \bar{\mathbf{f}})|$. Similarly the number of values of $\bar{v}$ where $\bar{v}_{j+1} = \bar{w}_{j+1}$ is $|\beta(\bar{w}, i, j+1, \bar{\mathbf{f}})|$. This allows the size of $\beta(\bar{w}, i, j, \bar{\mathbf{f}})$ to be computed in a recursive manner. In the special case where $j = i$, there is either one word in $\beta(\bar{w}, i, j, \bar{\mathbf{f}})$, if $j = 0$, or none if $j > 0$. Let $NS(\bar{w}, j, \bar{\mathbf{f}})$ return the number of possible slices of dimensions $f_1 \times f_2 \times \ldots \times f_{d-1}$ that are greater than $\bar{w}_{j+1}$. Using $NS(\bar{w}, j, \bar{\mathbf{f}})$ as a black box, the size of $\beta(\bar{w}, i, j, \bar{\mathbf{f}})$ is computed as:

$$|\beta(\bar{w}, i, j, \bar{\mathbf{f}})| = \begin{cases} 0 & i = j, j > 0 \\ 1 & i = j = 0 \\ NS(\bar{w}, j, \bar{\mathbf{f}}) \cdot |\beta(\bar{w}, i-j-1, 0, \bar{\mathbf{f}})| + |\beta(\bar{w}, i, j+1, \bar{\mathbf{f}})| & \textit{Otherwise.} \end{cases}$$

This leaves the problem of computing $NS(\bar{w}, j, \bar{\mathbf{f}})$. This is done by considering two cases. First are the set of slices that belong to a necklace class greater than $\bar{w}_{j+1}$. The number of such necklaces is computed as $|\mathcal{N}_q^{(f_1,f_2,\ldots,f_{d-1})}| - RN(\bar{w}_j, f_1, f_2 \ldots, f_{d-1})$, i.e. the number of necklaces of dimensions $(f_1, f_2, \ldots, f_{d-1})$ minus the necklaces smaller than $\bar{w}_j$. To account for the number of possible translations of each necklace, it is easiest to use the sets of aperiodic words instead. These translations are counted by counting the number of atranslational words of dimensions $(f_1, f_2, \ldots, f_{i-1}, h_i, 1, \ldots, 1)$ for every $i \in [d]$ and factor h_i of f_i. This rank is then multiplied by the number of possible translations, given by $f_1 \cdot f_2 \cdot \ldots \cdot f_{i-1} \cdot h_i$, and $|\mathbf{G}(h, (f_1, f_2, \ldots, f_i))|$ to account for the number of necklaces corresponding to each word in $T(\bar{w}, f_1, f_2 \ldots, f_d)$. The second case to consider are translations of $\bar{w}_{j_1}$ greater than $TR(\bar{w}_{j+1})$. This is given by $TP(\bar{w}_{j+1}) - TR(\bar{w}_{j+1})$. This allows the number of necklaces greater than $\bar{w}_j$ along with the number of translations of these necklaces to be counted as:

$$NS(\bar{w}, j, \bar{\mathbf{f}}) = (TP(\bar{w}_{j+1}) - TR(\bar{w}_{j+1})) + \sum_{i \in [d-1]} \sum_{h_i | f_i} RA(\bar{w}_j, \overline{\mathbf{h}[\mathbf{i}]})) \cdot |\overline{\mathbf{h}[\mathbf{i}]}| \cdot |\mathbf{G}(h_i, (n_1, n_2, \ldots, n_i))|$$

Where $\overline{\mathbf{h}[\mathbf{i}]} = (f_1, \ldots, f_{i-1}, h_i, \ldots, 1)$ and $|\overline{\mathbf{h}[\mathbf{i}]}| = f_1 \cdot f_2 \cdot \ldots \cdot f_{i-1} \cdot h_i$.

Theorem 5. *The rank of a d-dimensional necklace with dimensions $\overline{\mathbf{n}}$ is computed in $O(N^5)$ time.*

Proof. Lemmas 13, 14, 15, 16, and 17 show that to rank $RN(\bar{w})$, the first step is to compute the size of $T(\bar{w}, f_1, f_2 \ldots, f_d)$. Following Lemma 14, to compute the size of $A(\bar{w}), f_1, f_2 \ldots, f_d$, the set $A(\bar{w}_{[1,l]}, f_1, f_2 \ldots, f_{d-1}, l)$ must be computed for every factor l of f_d, alongside the set $L(\bar{w}, f_1, f_2 \ldots, f_d)$ and $L(\bar{w}_1, f_1, f_2 \ldots, f_{d-1})$. Note that this requires at most $\log_2(n_d)$ sets to be computed. The size of the set $L(\bar{w}, f_1, f_2 \ldots, f_{d-1})$ is computed by computing the size of $T(V(\bar{w}, h_1, h_2, \ldots, h_d))$ where h_i is a factor of f_i. Therefore for $L(\bar{w}, f_1, f_2 \ldots, f_{d-1})$, the size of at most $\log_2(N)$ sets $T(\bar{u}, h_1, h_2 \ldots, h_d)$ must be computed.

Following the above observations, $T(\bar{w}, n_1, n_2 \ldots, n_d)$ is computed by determining the size of $B(\bar{w}, g, j, n_1, n_2 \ldots, n_{d-1})$ using n_d^2 combinations of j and g. For each pair j and g, the size of $\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_{d-1})$ must be computed for some value of i. This is done in a dynamic programming approach. Starting with $i = j$, the size of $|\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)|$ is computed using the previously computed values as a basis. As such, the size of $|\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)|$ for every pair i and j is computed in n_d^2 time multiplied by the complexity of computing $NS(\bar{w}, j, n_1, n_2 \ldots, n_d)$. To compute $NS(\bar{w}, j, n_1, n_2 \ldots, n_d)$, $d \cdot \frac{\log_2 N}{d} = \log_2 \frac{N}{n_d}$ words of dimensions $d-1$ must be ranked.

As there are n_d^2 values of $\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)$, and $\log_2(\frac{N}{n_d})$ words of dimensions $d-1$ must be ranked for each of the n_d^2 values of $\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)$, to precompute every value of $\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)$ $n_d^2 \cdot \log_2(\frac{N}{n_d})$ time is needed, multiplied by the cost of ranking a $d-1$ word. If $d = 2$, then the rank at this step is computed in $O(n_1^2)$ time using existing algorithms due to Sawada and Williams [33]. Hence the size of $\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)$ for every value of i and j is computed in the two dimensional case in $O(n_d \cdot N \cdot \log_2(\frac{N}{n_d}) \cdot n_1^2) = O(N^2 \cdot \log_2(\frac{N}{n_d}))$ time. To get the rank of a two dimensional word, a further n_2^2 time is needed to compute the size of $T(\bar{w}, n_1, n_2 \ldots, n_d)$, with $\log_2(N)$ sets of $T(\bar{w})$ to be computed. Therefore the rank of a two dimensional word is computed in $O(n_2^2 \cdot \log_2(N) N^2 \cdot \log_2(\frac{N}{n_d}))$.

Similarly in the three dimensional case, the set of all values of $\beta(\bar{w}, i, j, n_1, n_2 \ldots, n_d)$ is computed in $O(n_3^2 \cdot n_2^2 \cdot \log_2(N) \frac{N^2}{n_3^2} \cdot \log_2(n_1)) = O(N^2 \cdot n_2^2 \cdot \log_2(N) \cdot \log_2(n_1))$. Thus the complexity of ranking a three dimensional word is $O(n_3^2 \cdot \log_2(N) \cdot N^2 \cdot n_2^2 \cdot \log_2(N) \cdot \log_2(n_1))$ time. In the more general case, a total of $n_d^2 \cdot \log_2(N)$ words of dimension $d-1$ must be ranked. Using the two and three dimensional cases as a base, the total complexity of ranking a d dimensional word is $O(\left(\prod_{i=2}^{d} n_i^4 \cdot \log_2(n_i)\right) n_1^2) \leq O(N^5)$. □

5.3 Generating and Unranking Multidimensional Necklaces

In this section we provide efficient algorithms for two further fundamental operations for necklaces: *generation*, where the task is to generate all the necklaces of a given size over an alphabet Σ, and *unranking*, where the task is to find a necklace of a given rank.

The idea presented here is based on generation of lower dimensional necklaces, generalising the 1D techniques to the higher dimensional setting. For the 1D setting, there have been several approaches for the generation of necklaces in constant amortised time, notably those of Cattell et. al. [9] and of Fredricksen and Maiorana [15]. A tempting approach would be to make an alphabet of size equal to the number of necklaces with dimensions $(n_1, \ldots, n_{d-1})$ and to generate the 1D necklaces from that. While this approach would generate a set of necklaces, as each d-dimensional necklace is comprised of a set of $d-1$-dimensional necklaces, it would also miss any in which one or more slices are translated by any degree. Similarly, representing every slice under each translation would generate words that are not necklaces. Let us illustrate it for a set of necklaces over a binary alphabet with dimensions $(2, 2)$. The complete set of necklaces is given in Figure 8. Of particular interest is the necklace represented by $\begin{bmatrix} A & B \\ B & A \end{bmatrix}$. While the first row, AB, is the canonical form of a 1D necklace, BA is not as it is equal to AB after a cyclic shift. Despite AB occurring as the necklace representation multiple times prior to this, BA only occurs at this point. As such, the situations where some slice may or may not be translated need to be understood and taken into account in order to generate the set of necklaces.

Before generating the set of necklace, the idea of a *multidimensional prenecklace* must be established. A prenecklace is a word $\bar{w}$ of dimensions $(n_1, n_2, \ldots, n_d)$ such that there exists some necklace of dimensions $(n_1, n_2, \ldots, n_{d-1}, n_d + m)$ represented by a word $\bar{u}$ such that $\bar{u}_{[1,n_d]} = \bar{w}$. Note that every necklace is a prenecklace.

$$\begin{bmatrix} A & A \\ A & A \end{bmatrix} \rightarrow \begin{bmatrix} A & A \\ A & B \end{bmatrix} \rightarrow \begin{bmatrix} A & A \\ B & B \end{bmatrix} \rightarrow \begin{bmatrix} A & B \\ A & B \end{bmatrix} \rightarrow \begin{bmatrix} A & B \\ B & A \end{bmatrix} \rightarrow \begin{bmatrix} A & B \\ B & B \end{bmatrix} \rightarrow \begin{bmatrix} B & B \\ B & B \end{bmatrix}$$
$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ translated(2) \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Figure 8: An example of generation of $(2, 2)$ necklaces, over the alphabet (A, B). The following mapping from necklace to code has been used: $AA \rightarrow 1$, $AB \rightarrow 2$, $BB \rightarrow 3$.

Lemma 18. *Given $\widehat{w}, \widehat{u} \in |\mathcal{N}_q^{\overline{\mathbf{n}}}|$ such that $rank(\widehat{u}) = rank(\widehat{w}) + 1$, let $Pre(\bar{w}, \bar{u}) = \{\bar{v} \in \Sigma^{\overline{\mathbf{n}}} : \bar{u} > \bar{v} > \bar{w}, \bar{v}$ is a prenecklace$\}$. The size of $Pre(\bar{w}, \bar{u})$ is at most n_d.*

Proof. This is statement is proven constructively. Let $NextPrenecklace(\bar{u})$ return the smallest prenecklace greater than $\bar{u}$. Given some word $\bar{u}$, let p be the length of the longest prefix of $\bar{u}$ that is a necklace. If $p < n_d$, the word $\bar{u}'$ is defined $\bar{u}'_i = \bar{u}_{i \bmod p}$. If $\bar{u}' \neq \bar{u}$, then $\bar{u}'$ is the smallest prenecklace that is greater than $\bar{u}$. Otherwise, let i be the last slice of $\bar{u}$ such that $\bar{u}_i \neq \bar{Q}$. Note that $\bar{u}_{[1,i-1]}\bar{Q}^{n_d-i}$ is a necklace. The auxiliary function $NextSlice(\bar{v})$ is introduced as returning the subsequent word in the ordering defined in Section 2.

$$NextSlice(\bar{v}) = \begin{cases} translate(\bar{v}) & TR(\bar{v}) < TP(\bar{v}) \\ NextNecklace(\langle \bar{v} \rangle) & otherwise. \end{cases}$$

Here $NextNecklace$ is treated as a black box that returns the next necklace in the ordering. Note that $\bar{u}_{[1,i-1]} : NextSlice(\bar{u}_i))_{j \bmod i}$ must be a necklace as any suffix of $\bar{u}_{[1,i-1]} : NextSlice(\bar{u}_i))_{j \bmod i}$ must be greater than $\bar{u}_{[1,i-1]}$. The word $\bar{u}'$ is redefined as $\bar{u}'_j = (\bar{u}_{[1,i-1]} : NextSlice(\bar{u}_i))_{j \bmod i}$. As $\bar{u}_{[1,i-1]} : NextSlice(\bar{u}_i))_{j \bmod i}$ is a necklace, $\bar{u}'$ is a prenecklace. Therefore $\bar{u}'$ is returned. To determine the size of $Pre(\bar{w}, \bar{u})$, note that the slice at position $i + 1$ must be smaller than $\bar{Q}$, therefore by repeating this process at most n_d times, the necklace of rank $rank(\bar{w}) + 1$ is found, and hence the size of $Pre(\bar{w}, \bar{u})$ is at most n_d. □

Lemma 19. *Let $\bar{w}$ be a word of dimensions $\overline{\mathbf{n}}$. $NextNecklace(\bar{w})$ returns the smallest word $\bar{u} > \bar{w}$ such that $\bar{u} = \langle \bar{u} \rangle$ in $O(N)$ time.*

Proof. Following Lemma 18, note that by applying the function $NextPrenecklace$ at most n_d times, the smallest necklace greater than $\bar{w}$ is determined. As each call to $NextPrenecklace$ requires $NextNecklace$ as a subroutine, to determine the next prenecklace of dimensions $d - 1$, n_{d-1} prenecklaces of dimensions $d - 2$ must be determined. Following this logic, to determine the next prenecklace of dimensions d at most $\frac{N}{n_d \cdot n_{d-1} \cdot \ldots \cdot n_{d-i+1}}$ prenecklaces of dimensions i must be considered. Therefore a total of $O(N)$ time is needed to compute all n_d prenecklaces. As it takes at most $O(N)$ time to determine if a word is a necklace, this process takes at most $O(N)$ time. □

Theorem 6. *Given an alphabet Σ of size q and set of dimensions $\overline{\mathbf{n}}$ there exists an algorithm to generate $\mathcal{N}_q^{\overline{\mathbf{n}}}$ in no more than $O(N)$ time per necklace.*

Proof. Let $\bar{a}$ be the smallest necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$. Following Lemma 19, it is possible to generate each necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ in at most $O(N)$ time per necklace. Hence it is possible to generate every necklace $\mathcal{N}_q^{\overline{\mathbf{n}}}$ in at most $O(|\mathcal{N}_q^{\overline{\mathbf{n}}}|N)$ time. □

Theorem 7. *The i^{th} necklace in $\mathcal{N}_q^{\overline{\mathbf{n}}}$ can be unranked in $O\left(N^{6(d+1)} \cdot \log^d(q)\right)$ time.*

Proof. The unranking procedure is done in a similar manner to the one dimensional case as presented by Sawada and Williams [33]. At a high level, the idea is to iteratively generate the necklace by generating prefixes of increasing length. Let $\bar{w}$ be the canonical representative of the i^{th} necklace. Further let $\bar{Q} = q^{(n_1, n_2, \ldots, n_{d-1})}$, the word of dimensions $(n_1, n_2, \ldots, n_{d-1})$ where every position is occupied by the symbol k. The first slice of $\bar{w}$ is determined through a binary search. Let $\bar{u}$ be the canonical representation of j^{th} necklace of dimensions $(n_1, n_2, \ldots, n_{d-1})$. Note that if $\bar{u}$ is the first slice of $\bar{w}$, then the rank of $\bar{w}$ must be between the rank of the smallest necklace starting with $\bar{u}$ and the greatest. These necklaces are determined using the same process as laid out in Lemma

4. Let $\bar{a}$ be the smallest such word and $\bar{b}$ the greatest. Therefore $\bar{u}$ is the first slice of $\bar{w}$ if and only if $RN(\bar{a}) \leq i \leq RN(\bar{b}$. Otherwise, depending on the value of i relative to $RN(\bar{a})$ and $RN(\bar{b})$ the next value of $\bar{u}$ is checked, with $\bar{u}$ determined by a binary search. Note that there are at most q^{N/n_d} necklaces of size $(n_1, n_2, \ldots, n_{d-1})$, the binary search requires at most $\log(q^{N/n_d}) = \frac{N}{n_d} \log k$ necklaces to be checked.

For the t^{th} slice, where $t \geq 2$, the process is slightly more complicated. As in the first case, to determine if the $\langle \bar{w}_t \rangle = \bar{u}$, the smallest and largest such words are determined and ranked. To that end, let $\bar{a}$ be the smallest possible word that is the canonical form of a necklace and has the prefix $\bar{w}_{[1,t-1]} : \langle \bar{u} \rangle_g$, and let $\bar{b}$ be the greatest. The value of $\bar{a}$ is computed in $O(N)$ time following the techniques outlined in Theorem 6. The word $\bar{b} = \bar{w}_{[1,t-1]} : \langle \bar{u} \rangle_g : \bar{Q}^{n_d - t}$ where g is the largest translation such that $\bar{u} \neq \langle \bar{u} \rangle_g$. Using these words, $\langle \bar{w}_t \rangle = \bar{u}$ if and only if $RN(\bar{a}) \leq i \leq RN(\bar{b})$.

The complexity of this process comes from the recursive nature of algorithm. In dimension d, n_d slices need to be computed, each requiring at most $\frac{N}{n_d} \cdot \log(q)$ necklaces to be ranked, the ranking having a complexity of N^5. Note that while determining the necklace that needs to be ranked has a complexity of N^2, this is not multiplicative with the complexity of ranking as each step is done independently. To determine each of these necklaces, a necklace of dimensions $(n_1, n_2, \ldots, n_{d-1})$ must be unranked, adding an additional complexity of $n_{d-1} \cdot \frac{N}{n_d \cdot n_{d-1}} \cdot \frac{N^5}{n_d^5} \cdot \log(q)$. As each dimension requires necklaces of the dimension one lower to be computed, the total complexity is $O\left(\prod_{i=0}^{d} \frac{N^6 \cdot \log(q)}{\prod_{j \in [1,i]} n_{d-j}^6}\right)$. In the worst case, where $n_1 = N$ and $n_i = 1$ for $i \in [2, d]$, this is simplified to $O\left(N^{6(d+1)} \cdot \log^d(q)\right)$. □

References

[1] Duncan Adamson, Vladimir V. Gusev, Igor Potapov, and Argyrios Deligkas. On the hardness of energy minimisation for crystal structure prediction. In *SOFSEM 2020*, volume 12011 of *Lecture Notes in Computer Science*, pages 587–596, 2020.

[2] Duncan Adamson, Vladimir V. Gusev, Igor Potapov, and Argyrios Deligkas. Ranking Bracelets in Polynomial Time. In Paweł Gawrychowski and Tatiana Starikovskaya, editors, *32nd Annual Symposium on Combinatorial Pattern Matching (CPM 2021)*, volume 191 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2021/13955`, `doi:10.4230/LIPIcs.CPM.2021.4`.

[3] F. S. Annexstein. Generating De Bruijn sequences: An efficient implementation. *IEEE Transactions on Computers*, 46(2):198–200, 1997.

[4] M. Anselmo, M. Madonia, and C. Selmi. Toroidal Codes and Conjugate Pictures. In *LATA 2019*, volume 11417 of *Lecture Notes in Computer Science*, pages 288–301, 2019.

[5] D. Antypov, A. Deligkas, V.V. Gusev, M. J. Rosseinsky, P. G. Spirakis, and M. Theofilatos. Crystal Structure Prediction via Oblivious Local Search. In *SEA 2020*, volume 160 of *LIPIcs*, pages 21:1–21:14, 2020.

[6] Jean Berstel, Dominique Perrin, Christophe Reutenauer, and Jean Berstel. *Codes and automata.* Encyclopedia of mathematics and its applications: volume 129. Cambridge University Press, 2009.

[7] Kellogg S. Booth. Lexicographically least circular substrings. *Information Processing Letters*, 10(4-5):240–242, jul 1980. `doi:10.1016/0020-0190(80)90149-0`.

[8] S. Bozapalidis and A. Grammatikopoulou. Picture codes. *RAIRO - Theoretical Informatics and Applications*, 40(4):537–550, 2006.

[9] K. Cattell, F. Ruskey, J. Sawada, M. Serra, and C.R. Miers. Fast Algorithms to Generate Necklaces, Unlabeled Necklaces, and Irreducible Polynomials over GF(2). *Journal of Algorithms*, 37(2):267–282, 2000.

[10] F. Chung, P. Diaconis, and R. Graham. Universal cycles for combinatorial structures. *Discrete Mathematics*, 110(1-3):43–59, 1992.

[11] C. Collins, G. R. Darling, and M.J. Rosseinsky. The Flexible Unit Structure Engine (FUSE) for probe structure-based composition prediction. *Faraday Discuss.*, 211:117–131, 2018.

[12] C. Collins, M. S. Dyer, M. J. Pitcher, G. F. S. Whitehead, M. Zanella, P. Mandal, J. B. Claridge, G. R. Darling, and M. J. Rosseinsky. Accelerated discovery of two crystal structure types in a complex inorganic phase field. *Nature*, 546(7657):280–284, 2017.

[13] M. S. Dyer, C. Collins, D. Hodgeman, P. A. Chater, A. Demont, S. Romani, R. Sayers, M. F. Thomas, J. B. Claridge, G. R. Darling, and M. J. Rosseinsky. Computationally assisted identification of functional inorganic materials. *Science*, 340(6134):847–852, 2013.

[14] A. E. Feldmann and D. Marx. The parameterized hardness of the k-center problem in transportation networks. *Algorithmica*, pages 1989–2005, 2020.

[15] H. Fredricksen and J. Maiorana. Necklaces of beads in k colors and k-ary de Bruijn sequences. *Discrete Mathematics*, 23(3):207–210, 1978.

[16] Guilhem Gamard, Gwenaël Richomme, Jeffrey Shallit, and Taylor J. Smith. Periodicity in rectangular arrays. *Information Processing Letters*, 118:58–63, 2017.

[17] Thomas Gärtner. A survey of kernels for structured data. *ACM SIGKDD explorations newsletter*, 5(1):49–58, 2003.

[18] D. Giammarresi, F. Venezia, and A. Restivo. Two-Dimensional Languages. *Handbook of Formal Languages, Vol. III*, pages 215–267, 1997.

[19] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete mathematics : a foundation for computer science*. Addison-Wesley, 1994.

[20] V. Horan and B. Stevens. Locating patterns in the de Bruijn torus. *Discrete Mathematics*, 339(4):1274–1282, 2016.

[21] G. Hurlbert and G. Isaak. On the de Bruijn Torus problem. *Journal of Combinatorial Theory, Series A*, 64(1):50–62, 1993.

[22] G. Hurlbert and G. Isaak. New constructions for De Bruijn tori. *Designs, Codes and Cryptography*, 6(1):47–56, 1995.

[23] G. H. Hurlbert, C. J. Mitchell, and K. G. Paterson. On the existence of de Bruijn Tori with two by two windows. *Journal of Combinatorial Theory. Series A*, 76(2):213–230, 1996.

[24] Yishan Jiao, Jingyi Xu, and Ming Li. On the k-closest substring and k-consensus pattern problems. In *Combinatorial Pattern Matching*, pages 130–144, 2004.

[25] T. Kociumaka, J. Radoszewski, and W. Rytter. Computing k-th Lyndon word and decoding lexicographically minimal de Bruijn sequence. In *Symposium on Combinatorial Pattern Matching*, pages 202–211. Springer International Publishing, 2014.

[26] S. Kopparty, M. Kumar, and M. Saks. Efficient indexing of necklaces and irreducible polynomials over finite fields. *Theory of Computing*, 12(1):1–27, 2016.

[27] M. Latteux and D. Simplot. Recognizable picture languages and domino tiling. *Theoretical Computer Science*, 178(1-2):275–283, 1997.

[28] Ming Li, Bin Ma, and Lusheng Wang. On the closest string and substring problems. *J. ACM*, 49(2):157–171, 2002.

[29] O. Matz. Regular expressions and context-free grammars for picture languages. In *Lecture Notes in Computer Science*, volume 1200, pages 283–294, 1997.

[30] D. Perrin. *Words*. Cambridge University Press, 2 edition, 1997.

[31] Chris J Pickard and R J Needs. Ab initiorandom structure searching. *Journal of Physics: Condensed Matter*, 23(5):053201, 2011.

[32] F. Ruskey and J. Sawada. Generating necklaces and strings with forbidden substrings. In *COCOON 2000*, volume 1858 of *Lecture Notes in Computer Science*, pages 330–339, 2000.

[33] J. Sawada and A. Williams. Practical algorithms to rank necklaces, Lyndon words, and de Bruijn sequences. *Journal of Discrete Algorithms*, 43:95–110, 2017.

[34] G. Stromoney, R. Siromoney, and K. Krithivasan. Abstract families of matrices and picture languages. *Computer Graphics and Image Processing*, 1(3):284–307, 1972.

[35] Y.Zhang, Z.Chang, F.Y.L.Chin, H.F.Ting, and Y.H.Tsin. Uniformly inserting points on square grid. *Inf. Process. Lett.*, 111:773–779, 2011.