The k-centre Problem for Classes of Cyclic Words

Duncan Adamson[1], Argyrios Deligkas[2], Vladimir V. Gusev[3,4], and Igor Potapov[4,5]*

[1] Department of Computer Science, Reykjavik University, Iceland duncana@ru.is
[2] Department of Computer Science, Royal Holloway, University of London
argyrios.deligkas@rhul.ac.uk
[3] Materials Innovation Factory, University of Liverpool, UK
Vladimir.Gusev@liverpool.ac.uk
[4] Department of Computer Science, University of Liverpool, UK
[5] potapov@liverpool.ac.uk

Abstract. The problem of finding k uniformly spaced points (centres) within a metric space is well known as the k-centre selection problem. In this paper, we introduce the challenge of k-centre selection on a class of objects of exponential size and study it for the class of combinatorial necklaces, known as cyclic words. The interest in words under translational symmetry is motivated by various applications in algebra, coding theory, crystal structures and other physical models with periodic boundary conditions. We provide solutions for the centre selection problem for both one-dimensional necklaces and largely unexplored objects in combinatorics on words - multidimensional combinatorial necklaces. The problem is highly non-trivial as even verifying a solution to the k-centre problem for necklaces can not be done in polynomial time relative to the length of the cyclic words and the alphabet size unless $P = NP$. Despite this challenge, we develop a technique of centre selection for a class of necklaces based on de-Bruijn Sequences and provide the first polynomial $O(k \cdot n)$ time approximation algorithm for selecting k centres in the set of 1D necklaces of length n over an alphabet of size q with an approximation factor of $O\left(1 + \frac{\log_q(k \cdot n)}{n - \log_q(k \cdot n)}\right)$. For the set of multidimensional necklaces of size $n_1 \times n_2 \times \ldots \times n_d$ we develop an $O(k \cdot N^2)$ time algorithm with an approximation factor of $O\left(1 + \frac{\log_q(k \cdot N)}{N - \log_q(k \cdot N)}\right)$ in $O(k \cdot N^2)$ time, where $N = n_1 \cdot n_2 \cdot \ldots \cdot n_d$ by approximating de Bruijn hypertori technique.

1 Introduction

The problem of finding k uniformly spaced points (centres) within a metric space is well known as the k-centre selection problem. So far, the problem has been intensely studied for finite, and explicitly given inputs like the k-centre problem for graphs, grids, or a set of strings, which are essential in the context of facility location and distribution [9,16,33].

* partially supported by ESPRC grant (EP/R018472/1)

The k-centre problem is also a tool in state space exploration, where cluster centres or equally spaced centres need to be selected to guarantee effective coverage of the configuration space. For algebraic and combinatorial structures with a state space of exponential size, sampling techniques have been used to generate equally probable objects [7]. However, while such sampling techniques can give uniform probability to the selection of any given object, there is a substantial gap in the problem of ensuring that k samples are representative. The k-centre problem is a natural means of modelling this objective, with the goal of ensuring that no object is significantly far from the set of samples under a distance based on some similarity metric. However, if the explicit representation of the whole class of objects is infeasible to store and process due to its exponential size, the k-centre selection problem requires new solutions and approaches.

In this work, we consider the class of combinatorial necklaces (also known as cyclic words). The study of 1D necklaces has been motivated by applications in the coding theory, free Lie algebras, and Hall sets [3,5,2,17,30,23,24]. Moreover, 2D necklaces have been recently studied for counting the number of toroidal codes in [6] and can be used in the construction of 2D Gray codes [8]. Algorithms for multidimensional combinatorial necklaces have remained a largely unexplored area in combinatorics on words [2,27,32]. A multidimensional necklace is an equivalence class of multidimensional words under *translational symmetry*, which is the natural generalisation of the shift operation in 1D, see Figure 1.

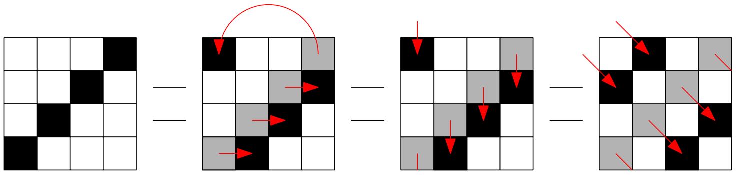

Fig. 1. An illustration of translational symmetry for a 4×4 word. Note that all 4 words presented here (out of a total of 16) correspond to the same necklace and can be reached from one another through some two-dimensional translation denoted (g_1, g_2). In red, the translation from the starting word to the new word is highlighted, with the original word overlaid in grey.

One natural use of multidimensional necklaces up to dimension three is the combinatorial representation of crystal structures. In computational chemistry, crystals are represented by periodic motives known as "unit cells". Informally, translational symmetry can be thought of as the equivalence of two crystals under translation in space. Intuitively this symmetry makes sense in the context of real structures, where two different "snapshots" of a unit cell both represent the same periodic and infinite global structure, see Figure 2.

Crystal Structure Prediction (CSP) is one of the most central and challenging problems in materials science and computational chemistry [4,1]. The objective is to find the "best" periodic three-dimensional structure of ions that yields the

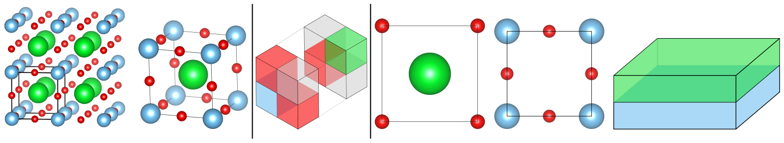

Fig. 2. The crystal of $SrTiO_3$ (left) and its 3D (middle) and 1D (right) necklace representations.

lowest interatomic potential energy. The aim of our k-centre selection algorithms for multidimensional necklaces is to replace the currently-used random generation approaches of unit cells [12] that often lead to identical crystal structures during the process of configuration-space exploration.

Most of the existing methods for CSP require the exploration of different configurations of periodic structures that combine local exploration and selection of new random locations in unexplored configuration space. The number of unit cells of size $n_1 \times n_2 \times \ldots \times n_d$ in the d-dimensional integer lattice and considering them up to translational symmetry is exponential and it is larger than $\frac{q^N}{N}$, where q is the number of different ions and $N = \prod_{i=1}^{d} n_i$. The size of such an object makes it infeasible to represent this set explicitly in the form of a weighted graph. By extension, applying existing centre selection algorithms will lead to an EXPSPACE solution and therefore require new techniques for operating on implicitly represented combinatorial objects. The same problem exists when it may be required to construct k equally spaced code words from a set of multidimensional cyclic words.

Even the original k-centre problem on graphs is non-trivial. The k-centre problem is both NP-hard with respect to the size of a graph and is APX-hard [18], making a PTAS unlikely. Additionally, the k-centre problem is unlikely to be fixed-parameter tractable in the context of the most natural parameter k [13]. A different form of the k-centre problem appears in stringology with important applications in computational biology; for example, to find the approximate gene clusters for a set of words over the DNA alphabet [14,25,26].

This paper introduces the challenge of k-centre selection for implicitly represented sets. Notably, we aim for polynomial time algorithms in the length of the output rather than in the size of the graph. The length of the output corresponds to a logarithmic factor relative to the size of the graph, multiplied by some function on the number of centres. The k-centre problem for strings or words can be defined over various distance functions. In this paper we focus on the *overlap distance*, based on the *overlap coefficient* (well known in linguistic processing [11,29,28]). The overlap coefficient measures the similarity of two words relative to the number of common subwords. This measure can, in turn, be used as a proxy for the closeness of potential energy in crystals. However, it is not critical for our algorithmic results; all results could be reformulated using other distance functions at the cost of slightly different approximation bounds.

In particular we develop a technique of centre selection based on de-Bruijn Sequences and provide the first polynomial $O(k \cdot n)$ time approximation algorithm for selecting k centres in the set of 1D necklaces of length n over an alphabet of size q with an approximation factor of $O\left(1 + \frac{\log_q(k \cdot n)}{n - \log_q(k \cdot n)}\right)$. In the multidimensional case, the results on generating de Bruijn tori are highly limited, so we developed a technique to select centres by approximating de Bruijn hypertori. We present an algorithm that generates k centres for the set of multidimensional necklaces of size $n_1 \times n_2 \times \ldots \times n_d$ with an approximation of $O\left(1 + \frac{\log_q(k \cdot N)}{N - \log_q(k \cdot N)}\right)$ in $O(k \cdot N^2)$ time, where $N = n_1 \cdot n_2 \cdot \ldots \cdot n_d$. Moreover, we show that verifying a solution to the k-centre problem for necklaces can not be done in polynomial time relative to the length of the cyclic words and the alphabet size unless $P = NP$, indicating that the k-centre problem itself is likely to be at least NP-hard.

2 Preliminaries

Let Σ be a finite alphabet of size q. In this paper, we assume that Σ is linearly ordered. We denote by Σ^* the set of all words over Σ and by Σ^n the set of all words of length n. The length of a word $u \in \Sigma^*$ is denoted by $|u|$. We use u_i, for any $i \in \{1, \ldots, |u|\}$, to denote the i^{th} symbol of u.

Let $[n]$ return the ordered set of integers from 1 to n inclusive. Given 2 words $u, v \in \Sigma^*$ where $|u| = |v|$, $u = v$ if and only if $u_i = v_i$ for every $i \in [|u|]$. A word u is *lexicographically smaller* than v if there exists an $i \in [|u|]$ such that $u_1 u_2 \ldots u_{i-1} = v_1 v_2 \ldots v_{i-1}$ and $u_i < v_i$ or $|u| < |v|$ and $u_1 u_2 \ldots u_{|u|} = v_1 v_2 \ldots v_{|u|}$. For example, given the alphabet $\Sigma = \{a, b\}$ where $a < b$, the word $aaaba$ is smaller than $aabaa$ as the first 2 symbols are the same and a is smaller than b. For a given set of words $\mathbf{S}$, the rank of v with respect to $\mathbf{S}$ is the number of words in $\mathbf{S}$ that are smaller than v.

The *translation* (*cyclic shift*) of a word $w = w_1 w_2 \ldots w_n$ by $r \in [n]$ returns the word $w_{r+1} \ldots w_n w_1 \ldots w_r$, and is denoted by $\langle w \rangle_r$, i.e. $\langle w_1 w_2 \ldots w_n \rangle_r = w_{r+1} \ldots w_n w_1 \ldots w_r$. Under the translation operation, u is equivalent to v if $v = \langle w \rangle_r$ for some $r \in [n]$. The t^{th} power of a word w, denoted by w^t, is equal to w repeated t times. For example $(aab)^3 = aabaabaab$. A word w is *periodic* if there is some word u and integer $t \geq 2$ such that $u^t = w$. Equivalently, word w is *periodic* if there exists some translation $r \in [|w| - 1]$ where $w = \langle w \rangle_r$. A word is *aperiodic* if it is not periodic. The *period* of a word w is the length of the smallest word u for which there exists some value t for which $w = u^t$.

A *necklace* is an equivalence class of words under the translation operation. An aperiodic necklace is called a *Lyndon word*. For notation, a word w is written as $\tilde{\mathbf{w}}$ when treated as a necklace. Given a necklace $\tilde{\mathbf{w}}$, the *canonical form* of $\tilde{\mathbf{w}}$ is the lexicographically smallest element of the set of words in the equivalence class $\tilde{\mathbf{w}}$. The canonical form of $\tilde{\mathbf{w}}$ is denoted by $\langle \tilde{\mathbf{w}} \rangle$, and the r^{th} shift of the canonical form is denoted by $\langle \tilde{\mathbf{w}} \rangle_r$. Given a word w, $\langle w \rangle$ denotes the canonical form of the necklace containing w.

A *subword* of the necklace w, denoted by $w_{[i,j]}$ is the word u of length $|w| + j - i - 1 \bmod |w|$ such that $u_a = w_{i-1+a \bmod |w|}$ for every $a \in |w|$. For notation, $u \sqsubseteq w$ denotes that u is a subword of w. Further, $u \sqsubseteq_i w$ denotes that u is a subword of w of length i. If a word u is a subword of w, then u is also a subword of the necklace $\langle w \rangle$. We denote that u is a subword of some necklace $\tilde{\mathbf{w}}$ by $u \sqsubseteq \tilde{\mathbf{w}}$, and that u is a subword of $\tilde{\mathbf{w}}$ of length i by $u \sqsubseteq_i \tilde{\mathbf{w}}$.

If $w = uv$, then u is a prefix and v is a suffix. For notation, the tuple $\mathbf{S}(v,\ell)$ is defined as the set of all subwords of v of length ℓ. Formally let $\mathbf{S}(v,\ell) = \{s \sqsubseteq v : |s| = \ell\}$. Further, $\mathbf{S}(v,\ell)$ is assumed to be in lexicographic order, i.e. $\mathbf{S}(v,\ell)_1 \geq \mathbf{S}(v,\ell)_2 \geq \ldots \mathbf{S}(v,\ell)_{|v|}$, where $\mathbf{S}(v,\ell)_i$ denotes the i^{th} entry of $\mathbf{S}(v,\ell)$. The set of necklaces of length n over an alphabet of size q is denoted by $\mathcal{N}_q^n$.

Multidimensional Words and Necklaces: In order to establish multidimensional necklaces, notation for *multidimensional words* must first be introduced. A *d-dimensional word* over Σ is an array of size $\vec{n} = (n_1, n_2, \ldots, n_d)$ of elements from Σ. In this work we tacitly assume that $n_1 \leq n_2 \leq \ldots \leq n_d$ unless otherwise stated. Let $|w|$ denote the vector of length d defining the size of the multidimensional word w. Given a size vector $\vec{n} = (n_1, n_2, \ldots, n_d)$, $\Sigma^{\vec{n}}$ is used to denote the set of all words of size $\vec{n}$ over Σ. Given a vector $\vec{n} = (n_1, n_2, \ldots, n_d)$ where every $n_i \geq 0$, $[\vec{n}]$ is used to denote the set $\{(x_1, x_2, \ldots, x_d) \in \mathbb{N}^d | \forall i \in [d], x_i \leq n_i\}$. Similarly $[\vec{m}, \vec{n}]$ is used to denote the set $\{(x_1, x_2, \ldots, x_d) \in \mathbb{N}^d \mid \forall i \in [d],\ m_i \leq x_i \leq n_i\}$.

For a d-dimensional word w, the notation $w_{(p_1,p_2,\ldots,p_d)}$ is used to refer to the symbol at position $(p_1,\ p_2,\ \ldots,\ p_d)$ in the array. Given two d-dimensional words w, u such that $|w| = (n_1, n_2, \ldots, n_{d-1}, a)$ and $|u| = (\ n_1,\ n_2,\ \ldots,\ n_{d-1},$ $b)$, the concatenation wu is performed along the last dimension, returning the word v of size $(n_1, n_2, \ldots, n_{d-1}, a + b)$ such that $v_{\vec{p}} = w_{\vec{p}}$ if $p_d \leq a$ and $v_{\vec{p}} = u_{(p_1,p_2,\ldots,p_{d-1},p_d-a)}$ if $p_d > a$.

A *multidimensional subword* of w of size $\vec{m}$ is denoted by $v \sqsubseteq_{\vec{m}} w$. As in the 1D case, a subword is defined by a start and an end position within the original word w. Let $w[\vec{i}, \vec{j}]$ for $\vec{i}, \vec{j} \in [\vec{n}]$ denote the subword u of size $(j_1 - i_1 + 1, j_2 - i_2 + 1, \ldots, j_d - i_d + 1)$. The symbol at position $\vec{p}$ of u equals the symbol at position $(i_1 + p_1, i_2 + p_2, \ldots, i_d + p_d)$ of w, i.e. $u_{\vec{p}} = w_{(i_1+p_1, i_2+p_2, \ldots, i_d+p_d)}$.

A *d-dimensional translation* r is defined by a d-tuple $r = (r_1, r_2, \ldots, r_d)$. The translation of the word $w \in \Sigma^{\vec{n}}$ by r, denoted by $\langle w \rangle_r$, returns the word $v \in \Sigma^{\vec{n}}$ such that $v_{\vec{p}} = w_{\vec{j}}$ for every position $\vec{p} \in [\vec{n}]$ where $\vec{j} = (p_1 + r_1 \bmod n_1, p_2 + r_2 \bmod n_2, \ldots, p_d + r_d \bmod n_d)$. It is assumed that $r_i \in [0, n_i - 1]$, so the set of translations $Z_{\vec{n}}$ is equivalent to the direct product of the cyclic groups, giving $Z_{\vec{n}} = Z_{n_1} \times Z_{n_2} \times \ldots \times Z_{n_d}$. Given two translations $r = (r_1, r_2, \ldots, r_d)$ and $t = (t_1, t_2, \ldots, t_d)$ in $Z_{\vec{n}}$, $t + r$ 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*** $\tilde{\mathbf{w}}$ *is an equivalence class of multidimensional words under the translation operation. The set of multidimensional necklaces over an alphabet of size q of size $n_1 \times n_2 \times \ldots \times n_d$ is denoted by $\mathcal{N}_q^{\vec{n}}$ where $\vec{n} = (n_1, n_2, \ldots, n_d)$.*

Proposition 1. *The number of multidimensional necklaces of size* $n_1 \times n_2 \times \ldots \times n_d$ *over an alphabet of size* q *is bounded by* $\frac{q^N}{N} \leq |\mathcal{N}_q^{\vec{n}}|$, *where* $N = \prod_{i=1}^{d} n_i$.

Proof. Given any word $w \in \Sigma^{\vec{n}}$, there are at most $N-1$ words equivalent to w under the translation operation, giving $|\mathcal{N}_q^{\vec{n}}| \geq \frac{|\Sigma^{\vec{n}}|}{N} = \frac{q^N}{N}$. □

3 The *k*-centre problem for necklaces

In this section, we formally define the k-centre problem for a set of necklaces. The input to our problem is some positive integer k, an alphabet Σ, and positive integer length n. The goal is to choose a set $\mathbf{S}$ of k centres from the implicitly defined set of necklaces such that the maximum distance between any member of the input set and the set of centres $\mathbf{S}$ is minimised. For example, when the problem is defined over the set $\mathcal{N}_q^n$ of q-ary necklaces of length n, the problem is to select some subset $\mathbf{S} \subseteq \mathcal{N}_q^n$ such that $|\mathbf{S}| = k$ and the distance between each necklace $\tilde{\mathbf{w}} \in \mathcal{N}_q^n$ and the necklace $\tilde{\mathbf{u}} \in \mathbf{S}$ that is closest to $\tilde{\mathbf{w}}$ is minimised.

The remainder of this section formalises the k centre problem for necklaces. Section 3.1 defines the *overlap distance* between necklaces. At a high-level, the overlap distance between two necklaces is the inverse of the *overlap coefficient* between them, in this case, 1 minus the overlap coefficient. This distance can be seen as a natural distance based on "bag-of-words" techniques used in machine learning [15]. Section 3.2 uses the overlap distance to define the k-centre problem for classes of necklaces. Along with a problem definition, we provide preliminary results on the complexity of the k-centre problem for necklaces, as well as theoretical lower bounds on the optimal solution in the necklace setting.

3.1 The Overlap Distance and the k-centre Problem

Our definition of the overlap distance depends on the well-studied *overlap coefficient*, defined for a pair of sets A and B as $\mathfrak{C}(A, B) = \frac{|A \cap B|}{\min(|A|,|B|)}$. In the context of necklaces $\mathfrak{C}(\tilde{\mathbf{w}}, \tilde{\mathbf{v}})$ is defined as the overlap coefficient between the multisets of all subwords of $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{v}}$. For some necklace $\tilde{\mathbf{w}}$ of size $\vec{n}$, the multiset of subwords of size $\vec{\ell}$ contains all $u \sqsubseteq_{\vec{\ell}} w$. For each subword u appearing m times in $\tilde{\mathbf{w}}$, m copies of u are added to the multiset. This gives a total of N subwords of size $\vec{\ell}$ for any $\vec{\ell}$, 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 size vector, with a total of N^2 subwords; see Figure 3.

To use the overlap coefficient as a distance between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{v}}$, the overlap coefficient is inverted so that a value of 1 means $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{v}}$ share no common subwords while a value of 0 means $\tilde{\mathbf{w}} = \tilde{\mathbf{v}}$. The overlap distance (see example in Figure 3) between two necklaces $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{v}}$ is $\mathfrak{D}(\tilde{\mathbf{w}}, \tilde{\mathbf{v}}) = 1 - \mathfrak{C}(\tilde{\mathbf{w}}, \tilde{\mathbf{v}})$. Proposition 2 shows that this distance is a metric distance.

	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

Fig. 3. 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 $\mathfrak{O}(ababab, abbabb) = \frac{25}{36}$.

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

$\tilde{\mathbf{w}} \backslash \tilde{\mathbf{v}}$	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

Fig. 4. Example of the overlap distance $\mathfrak{O}(\langle\tilde{\mathbf{w}}\rangle, \langle\tilde{\mathbf{v}}\rangle)$ for all necklaces in $\mathcal{N}_2^4$.

Proposition 2. *The overlap distance for necklaces is a metric distance.*

Proof. Let $\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}} \in \mathcal{N}_q^{\vec{n}}$, for some arbitrary vector $\vec{n} \in \mathbb{N}^d$ and $q \in \mathbb{N}$. The overlap distance is metric if and only if $\mathfrak{O}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) \leq \mathfrak{O}(\tilde{\mathbf{a}}, \tilde{\mathbf{c}}) + \mathfrak{O}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}})$. Rewriting this gives $1 - \mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) \leq 2 - \mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) - \mathfrak{C}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}})$ which can be rewritten in turn as $\mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) + \mathfrak{C}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}}) \leq 1 + \mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}})$. Observe that if $\mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{c}}) + \mathfrak{C}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}}) > 1$ then $\frac{|\tilde{\mathbf{a}} \cap \tilde{\mathbf{c}}|}{N^2} + \frac{|\tilde{\mathbf{b}} \cap \tilde{\mathbf{c}}|}{N^2} > 1$, meaning that $|\tilde{\mathbf{a}} \cap \tilde{\mathbf{c}}| + |\tilde{\mathbf{b}} \cap \tilde{\mathbf{c}}| > N^2$. This implies that $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ share at least $|\tilde{\mathbf{a}} \cap \tilde{\mathbf{c}}| + |\tilde{\mathbf{b}} \cap \tilde{\mathbf{c}}| - N^2$ subwords. Therefore $\mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{n}})$ must be at least $\mathfrak{C}(\tilde{\mathbf{a}}, \tilde{\mathbf{n}}) + \mathfrak{C}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}}) - 1$. Hence $\mathfrak{O}(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) \leq \mathfrak{O}(\tilde{\mathbf{a}}, \tilde{\mathbf{c}}) + \mathfrak{O}(\tilde{\mathbf{b}}, \tilde{\mathbf{c}})$. □

3.2 The *k*-centre Problem.

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

Problem 1. k-centre problem for necklaces.

Input: A size vector of d-dimensions $\vec{n} \in \mathbb{N}^d$, an alphabet of size q, and an integer $k \in \mathbb{N}$.
Question: What is the set $\mathbf{S} \subseteq \mathcal{N}_q^{\vec{n}}$ of size k minimising $\mathfrak{O}(\mathbf{S}, \mathcal{N}_q^{\vec{n}})$?

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

Problem 2. The k-centre verification problem for necklaces.

Input: A d-dimensional size vector $\vec{n} \in \mathbb{N}$, an alphabet of size q, a rational distance $\ell \in \mathbb{Q}$, and a subset $\mathbf{S} \subseteq \mathcal{N}_q^{\vec{n}}$.
Question: Does there exist a necklace $\tilde{\mathbf{w}} \in \mathcal{N}_q^{\vec{n}}$ such that $\mathfrak{O}(\tilde{\mathbf{w}}, \mathbf{S}) \geq \ell$?

Theorem 1. *Given a set $\mathbf{S} \subseteq \mathcal{N}_q^{\vec{n}}$ and a distance ℓ, it is NP-hard to determine if there exists some necklace $\tilde{\mathbf{v}} \in \mathcal{N}_q^{\vec{n}}$ such that $\mathfrak{D}(\tilde{\mathbf{s}}, \tilde{\mathbf{v}}) > \ell$ for every $\tilde{\mathbf{s}} \in \mathbf{S}$.*

Proof. This claim is proven via a reduction from the Hamiltonian cycle problem on bipartite graphs to Problem 2 in 1D. 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 size $(n_1, 1, 1, \dots, 1)$. 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 $\tilde{\mathbf{w}} \in \mathcal{N}_q^{\vec{n}}$ such that the distance between $\tilde{\mathbf{w}}$ and every necklace $\tilde{\mathbf{v}} \in \mathbf{S}$ is greater than $1 - \frac{3}{n^2}$. If $\tilde{\mathbf{w}}$ shares a subword of length 2 with any necklace in $\mathbf{S}$ then $\tilde{\mathbf{w}}$ 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 $\tilde{\mathbf{w}}$ must correspond to a valid edge.

As $\tilde{\mathbf{w}}$ 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 $\tilde{\mathbf{w}}$. As G is bipartite, if any cycle represented by $\tilde{\mathbf{w}}$ 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 $1 - \frac{3}{n^2}$ from $\tilde{\mathbf{w}}$. Alternatively, if every cycle represented by $\tilde{\mathbf{w}}$ has length 2, there must be some vertex v that is represented at least 3 times in $\tilde{\mathbf{w}}$. Hence in this case $\tilde{\mathbf{w}}$ 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 $\mathcal{N}_q^{\vec{n}}$ relative to $\vec{n}$ and q makes finding an optimal solution for Problem 1 exceedingly unlikely.

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

Proof. Recall that the distance between the furthest necklace $\tilde{\mathbf{w}} \in \mathcal{N}_q^n$ and the optimal set $\mathbf{S}$ is bounded from bellow by determining an upper bound on the number of shared subwords between $\tilde{\mathbf{w}}$ and the words in $\mathbf{S}$. For the remainder of this proof let $\tilde{\mathbf{w}}$ 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 $\tilde{\mathbf{S}}$ of length $n \cdot k$. As the distance between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{S}}$ is no more than the distance between $\tilde{\mathbf{w}}$ and any $\tilde{\mathbf{v}} \in \mathbf{S}$, the distance between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{S}}$ provides a lower bound on the distance between $\tilde{\mathbf{w}}$ and $\mathbf{S}$.

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

For subwords of length 2, the problem becomes more complicated. 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 $\tilde{\mathbf{w}}$. 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 representation of necklaces is the number of necklaces of length 2, which in turn is bounded by $\frac{q^2}{2}$. Noting that the prefixes in $\tilde{\mathbf{S}}$ may overlap, to ensure that $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{w}}$ share at least one subword of length 2, the length of $\tilde{\mathbf{S}}$ must be at least $\frac{q^2}{2}$. Similarly, for $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{w}}$ to share i subwords of length 2, the length of $\tilde{\mathbf{S}}$ must be at least $\frac{i \cdot q^2}{2}$. Hence the maximum number of subwords of length 2 that $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{w}}$ 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. In order for $\tilde{\mathbf{S}}$ to share at least one subword of length j with $\tilde{\mathbf{w}}$, the length of $\tilde{\mathbf{S}}$ must be at least $\frac{q^j}{j}$. Further the maximum number of subwords of length j that $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{w}}$ 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.

The maximum length of a common subword that $\tilde{\mathbf{w}}$ can share with $\tilde{\mathbf{S}}$ 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 $\frac{q^l}{n} \leq n \cdot k$ as $l = 2 \log_q(n \cdot k)$. 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 $\tilde{\mathbf{S}}$ as $\tilde{\mathbf{S}}$ can not contain enough subwords to ensure that this is the case. This bound allows an upper bound number of shared subwords between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{S}}$ to be given by the summation

$$\sum_{i=1}^{2\log_q(n\cdot k)} \min(\lfloor \tfrac{i\cdot n\cdot k}{q^i} \rfloor, n) \leq n\cdot \log_q(n\cdot k) + \tfrac{\log_q(k\cdot n)}{q-1} \approx \tfrac{q\cdot n\log_q(k\cdot n)}{q-1} \approx n\log_q(k\cdot n).$$

Using this bound, the distance between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{S}}$ must be no less than $1 - \frac{\log_q(k\cdot n)}{n}$.

In the multidimensional case, let $\vec{m} = (\ m_1,\ m_2,\ \ldots,\ m_d)$ be a size vector of d-dimensions such that $M = m_1 \cdot m_2 \cdot \ldots \cdot m_d$. The largest value of M such that $\tilde{\mathbf{S}}$ can contain every subword with M positions is $2\log_q(n\cdot k)$. From Proposition 1, the lower bound on the number of necklaces of size $\vec{m}$ is $\frac{q^M}{M}$. The maximum number of shared subwords between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{S}}$ is $\sum_{i=1}^{M} i \cdot \frac{N\cdot k}{q^i} \leq \log_q(k\cdot N)$. Hence the distance between $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{s}}$ is at most $1 - \frac{\log_q(k\cdot N)}{N}$. □

The key idea behind our algorithms for approximating the k-centre problem on necklaces is to find the largest vector $\vec{\ell} = (l_1, l_2, \ldots, l_d)$ such that every word of size $\vec{\ell}$ appears as a subword within the set of centres. In this setting $\vec{m}$ is larger than $\vec{\ell}$ 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. *Given $\tilde{\mathbf{w}}, \tilde{\mathbf{v}} \in \mathcal{N}_q^{\vec{n}}$ sharing a common subword a of size $\vec{m}$, let $x_i = n_i \cdot m_i$ if $n_i = m_i$, and $x_i = \frac{m_i(m_i+1)}{2}$ otherwise. The distance between w and v is bounded by $\mathfrak{D}(w,v) \leq 1 - \frac{\prod_{i=1}^{d} x_i}{N^2} \leq 1 - \frac{M^2}{2N^2}$ where $N = \prod_{i\in[d]} n_i$ and $M = \prod_{i\in[d]} m_i$.*

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

4 Approximating the k-centre Problem for necklaces

In this section we provide our approximation algorithms. The main idea is to determine the longest 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.

Definition 2. *A* ***de Bruijn hypertorus*** *of order $\vec{n}$ is a cyclic d-dimensional word T containing as a subword every word in $\Sigma^{\vec{n}}$ exactly once.*

Lemma 3. *There exists an $O(n \cdot k)$ time algorithm for the k-centre problem on $\mathcal{N}_q^n$ returning a set of centres* $\mathbf{S}$ *such that* $\mathfrak{D}(\mathbf{S}, \mathcal{N}_q^n) \leq 1 - \frac{\log_q^2(k \cdot n)}{2n^2}$.

Proof. Our algorithm operates by partitioning a de Bruijn sequence S of order λ into a set of k 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 5.

Sequence:	0000001000011000101000111001001011001101001111010101110110111111
Centre	Word
1	000000100001100010100
2	101000111001001011001
3	110011010011110101011
4	000000 0101110110111111

Fig. 5. Example of how to split the de Bruijn sequence of order 6 between 4 centres. Highlighted parts are the shared subwords between two centres.

This leaves the problem of determining the largest value of λ such that $q^\lambda \leq k \cdot (n - \lambda + 1)$. Rearranging $q^\lambda \leq k \cdot (n - \lambda + 1)$ in terms of λ gives $\lambda \leq \log_q(k \cdot (n+1) - k \cdot \lambda)$. Noting that $\lambda \leq \log_q(k \cdot n)$, 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 each necklace in $\mathcal{N}_q^n$ and the set of centres 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 [31], the total complexity is at most $O(k \cdot n)$. □

Theorem 2. *The k-centre problem for $\mathcal{N}_q^n$ can be approximated in $O(n \cdot k)$ time with an approximation factor of* $1 + \frac{\log_q(k \cdot n)}{n - \log_q(k \cdot n)} - \frac{\log_q^2(k \cdot n)}{2n(n - \log_q(k \cdot n))}$.

Proof. Using the lower bound of $1 - \frac{\log_q^2(kn)}{2n^2}$ given by Lemma 3 gives $\frac{1 - \frac{\log_q^2(kn)}{2n^2}}{1 - \frac{\log_q(k \cdot n)}{n}}$ $= \frac{2n^2 - \log_q^2(kn)}{2n^2 - 2n\log_q(kn)} = 1 + \frac{2n\log_q(kn) - \log_q^2(kn)}{2n^2 - 2n\log_q(kn)} = 1 + \frac{\log_q(kn)}{n - \log_q(kn)} - \frac{\log_q^2(kn)}{2n(n - \log_q(kn))}$. □

Theorem 3. *Let T be a d-dimensional de Bruijn hypertorus of size $(x, x, \ldots, x)$. 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* $1 + \frac{\log_q(kN)}{N - \log_q(k \cdot N)} - \frac{\log_q^2(k \cdot N)}{2N(N - \log_q(k \cdot N))}$ *where $y^d = N$ and $x^d = log_N(y)$.*

Proof. Recall from Lemma 1 that the lower bound on the distance between the centre and every necklace in $\mathcal{N}_q^{\vec{n}}$ 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 size $m_1 = m_2 = \ldots = m_j = \sqrt[j]{N}$ for some $j \in [d]$, giving the total number of positions in the tori as M. Similarly, the centres is assumed to have size $n_1 = n_2 = \ldots = n_d = \sqrt[d]{N}$, giving N total positions.

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 $\mathcal{N}_q^{\vec{n}}$ is $1 - \frac{\log_q^2(kN)}{2N^2}$. Following the arguments from Theorem 2 gives a ratio of $1 + \frac{2 \cdot N \log_q(k \cdot N) - \log_q^2(k \cdot N)}{2 \cdot N^2 - 2 \cdot N \cdot \log_q(k \cdot N)} = 1 + \frac{\log_q(kN)}{N - \log_q(kN)} - \frac{\log_q^2(kN)}{2N(N - \log_q(kN))}$. □

While this provides a good starting point for solving the k-centre problem for $\mathcal{N}_q^{\vec{n}}$, this work is restricted by the limited results on generating de Bruijn hypertori, particularly in higher dimensions [10,19,20,21,22]. As such, we present an alternative approach below. The high-level idea is to reduce the problem from the multidimensional setting to the 1D problem, which we can approximate well using Theorem 2. Given a size vector $\vec{n}$, integer k and alphabet Σ our approach can be thought of as finding a set of $k \cdot n_1 \cdot \ldots \cdot n_{d-1}$ centres of length n_d over Σ, taking advantage of the added number of centres to increase the length of shared subwords.

Case 1, $q^{n_d} \geq k \cdot \frac{N}{n_d}$: In this case the set of centres is constructed by using $k' = \frac{k \cdot N}{n_d}$ centres of $\mathcal{N}_q^{n_d}$. The motivation behind this approach is to optimise the length of the 1D subwords that are shared by the centre and every other necklace in $\mathcal{N}_q^{\vec{n}}$. Let $\mathbf{S} \subseteq \mathcal{N}_q^{n_d}$ be a set of centres $k \cdot \frac{N}{n_d}$ from $\mathcal{N}_q^{n_d}$ constructed following the algorithm outlined in Lemma 3. Following the arguments from Lemma 3, every necklace in $\mathcal{N}_q^{n_d}$ must share a subword of length $\log_q(k \cdot N)$ with at least one centre in $\mathbf{S}$. As every subword of size $(1, 1, \ldots, 1, n_d)$ of any necklace in $\mathcal{N}_q^{\vec{q}}$ belongs to a necklace $\tilde{\mathbf{w}} \in \mathcal{N}_q^{n_d}$, by ensuring that every necklace in $\mathbf{S}$ appears as a subword in the centre $\mathbf{S}' \subseteq \mathcal{N}_q^{\vec{n}}$ it is ensured that $\tilde{\mathbf{w}}$ shares at least one subword of length $\log_q(k \cdot N)$ with some necklace in $\mathbf{S}'$. This can be done by simply splitting $\mathbf{S}$ into k sets of $\frac{N}{n_d}$ centres, each of which can be made into a word of size $\vec{n}$ through concatenation. From Lemma 2, the maximum distance between any necklace in $\mathbf{S}'$ and necklace in $\mathcal{N}_q^{\vec{n}}$ is $1 - \frac{\log_q^2(k \cdot N)}{2N^2}$. This equals the bound given by Lemma 3, giving the same approximation ratio.

Case 2, $q^{n_d} < k \cdot \frac{N}{n_d}$: Following the process outlined above, it is possible to represent every word of length n_d over Σ with some redundancy. In order to reduce the redundancy an alternative reduction from the 1D setting is constructed. The high-level idea is to construct a new alphabet such that each symbol corresponds to some word in $\Sigma^{\vec{m}}$ for some size vector $\vec{m}$.

The first problem is determining the size vector allowing for this reduction. Let $\Sigma(\vec{m})$ denote the alphabet of size $q^{m_1 \cdot m_2 \cdot \ldots \cdot m_d}$ such that each symbol in $\Sigma(\vec{m})$ corresponds to some word in $\Sigma^{\vec{m}}$. Given a word $w \in \Sigma(\vec{m})^{n_1/m_1, n_2/m_2, \ldots, n_d/m_d}$ a word $v \in \Sigma^{\vec{n}}$ can be constructed by replacing each symbol in w with the corresponding word in $\Sigma^{\vec{m}}$. Note that the largest value of $\vec{m}$ such that every symbol in $\Sigma(\vec{m})$ can be represented in k words from $\Sigma(\vec{m})^{n_1/m_1, n_2/m_2, \ldots, n_d/m_d}$ is bounded by the inequality $q^{m_1 \cdot m_2 \cdot \ldots \cdot m_d} \leq k \cdot \lfloor \frac{n_1}{m_1} \rfloor \cdot \lfloor \frac{n_2}{m_2} \rfloor \cdot \ldots \cdot \lfloor \frac{n_d}{m_d} \rfloor$. Letting $M = m_1 \cdot m_2 \cdot \ldots \cdot m_d$, this inequality can be rewritten as approximately $q^M \leq k \cdot \frac{N}{M}$. Treating M as being approximately N gives $M \leq \log_q(k)$.

Using this bound on M let $\vec{m}$ be some set of vectors such that $M = m_1 \cdot m_2 \cdot \ldots \cdot m_d$. We may assume without loss of generality that $m_d = 1$. The centres for $\mathcal{N}_q^{\vec{n}}$ are constructed by making a set $\mathbf{S}$ of $k \frac{N}{M \cdot n_d}$ centres for $\mathcal{N}_{q^M}^{n_d}$. Following the arguments from Lemma 3, every necklace in $\mathcal{N}_{q^M}^{n_d}$ must share a subword of length at least $\log_{q^M}(k \cdot \frac{N}{M}) = \frac{\log_q\left(k \cdot \frac{N}{M}\right)}{M} = \frac{\log_q\left(k \cdot \frac{N}{\log_q(k)}\right)}{\log_q(k)}$. Note further that, as each symbol in $\Sigma(\vec{m})$ corresponds to a word in $\Sigma^{\vec{m}}$, converting each word in $\mathbf{S}$ to a word of size $(m_1, m_2, \ldots, m_{d-1}, n_1)$ provides a set of centres such that every necklace in $\mathcal{N}_q^{(m_1, m_2, \ldots, m_{d-1}, n_1)}$ shares a subword of size $\left(m_1, m_2, \ldots, m_{d-1}, \frac{\log_q\left(k \cdot \frac{N}{\log_q(k)}\right)}{\log_q(k)}\right)$ with some centre. Converting this new set of centres into a set $\mathbf{S}' \subseteq \mathcal{N}_q^{\vec{n}}$ maintains the same size of shared subwords. From Lemma 2, the furthest distance between $\mathbf{S}'$ and any necklace in $\mathcal{N}_q^{\vec{n}}$ is bounded from above by $1 - \frac{\log_q^2(k) \cdot \frac{\log_q^2\left(k \cdot \frac{N}{\log_q(k)}\right)}{\log_q^2(k)}}{2N^2} = 1 - \frac{\log_q^2\left(k \cdot \frac{N}{\log_q(k)}\right)}{2N^2} \approx 1 - \frac{\log_q^2(k \cdot N)}{2N^2}$.

Theorem 4. *The k-centre problem for $\mathcal{N}_q^{\vec{n}}$ can be approximated in $O(N^2 k)$ time within a factor of $1 + \frac{\log_q(kN)}{N - \log_q(kN)} - \frac{\log_q^2(kN)}{2N(N - \log_q(kN))}$, where $N = \prod_{i=1}^{d} n_i$.*

Proof. Following the above construction, note that in both cases the distance between the set of centres $\mathbf{S}$ and the necklaces $\mathcal{N}_q^{\vec{n}}$ is bounded from above by $1 - \frac{\log_q^2(k \cdot N)}{2N^2}$. The approximation ratio of $1 + \frac{\log_q(kN)}{N - \log_q(kN)} - \frac{\log_q^2(kN)}{2N(N - \log_q(kN))}$ is derived using the same arguments as in Theorem 2. Regarding time complexity, in the first case the problem can be solved in $O(k \cdot N)$ time using Theorem 2. In the second case, a brute force approach to find to best value of $\vec{m}$ can be done in $O(N)$ additional time steps giving a total complexity of $O(k \cdot N^2)$. □

5 Acknowledgements

The authors thank the Leverhulme Trust via the Leverhulme Research Centre for Functional Materials Design at the University of Liverpool for their support.

References

1. D. Adamson, A. Deligkas, V. V. Gusev, and I. Potapov. 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. D. Adamson, A. Deligkas, V. V. Gusev, and I. Potapov. Combinatorial algorithms for multidimensional necklaces, 2021. URL: `https://arxiv.org/abs/2108.01990`, `doi:10.48550/ARXIV.2108.01990`.
3. Duncan Adamson. Ranking binary unlabelled necklaces in polynomial time. In Yo-Sub Han and György Vaszil, editors, *Descriptional Complexity of Formal Systems*, pages 15–29, Cham, 2022. Springer International Publishing.
4. Duncan Adamson, Argyrios Deligkas, Vladimir V. Gusev, and Igor Potapov. The Complexity of Periodic Energy Minimisation. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, volume 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2022/16806`, `doi:10.4230/LIPIcs.MFCS.2022.8`.
5. 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`.
6. 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.
7. László Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, page 164–174, New York, NY, USA, 1991. Association for Computing Machinery. `doi:10.1145/103418.103440`.
8. M.M. Bae and B. Bose. Gray codes for torus and edge disjoint hamiltonian cycles. In *Proceedings 14th International Parallel and Distributed Processing Symposium. IPDPS 2000*, pages 365–370, 2000. `doi:10.1109/IPDPS.2000.846007`.
9. D. Chakrabarty, P. Goyal, and R. Krishnaswamy. The non-uniform k-center problem. *ACM Trans. Algorithms*, 16(4), June 2020. `doi:10.1145/3392720`.
10. F. Chung, P. Diaconis, and R. Graham. Universal cycles for combinatorial structures. *Discrete Mathematics*, 110(1-3):43–59, 1992.
11. W. W. Cohen, P. Ravikumar, S. E. Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, volume 2003, pages 73–78, 2003.
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, 2017.
13. A. E. Feldmann and D. Marx. The parameterized hardness of the k-center problem in transportation networks. *Algorithmica*, pages 1989–2005, 2020.
14. M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30(2):113–119, 1997.

15. T. Gärtner. A survey of kernels for structured data. *ACM SIGKDD explorations newsletter*, 5(1):49–58, 2003.
16. L. Gasieniec, J. Jansson, and A. Lingas. Efficient approximation algorithms for the Hamming Center Problem. In *SODA 1999*, pages 905–906, 1999.
17. R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete mathematics : a foundation for computer science*. Addison-Wesley, 1994.
18. D. S. Hochbaum. Various notions of approximations: Good, better, best and more. *Approximation algorithms for NP-hard problems*, 1997.
19. V. Horan and B. Stevens. Locating patterns in the de Bruijn torus. *Discrete Mathematics*, 339(4):1274–1282, 2016.
20. G. Hurlbert and G. Isaak. On the de Bruijn Torus problem. *Journal of Combinatorial Theory, Series A*, 64(1):50–62, 1993.
21. G. Hurlbert and G. Isaak. New constructions for De Bruijn tori. *Designs, Codes and Cryptography*, 6(1):47–56, 1995.
22. 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.
23. 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.
24. 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.
25. J. K. Lanctot, M. Li, B. Ma, S. Wang, and L. Zhang. Distinguishing string selection problems. *Information and Computation*, 185(1):41–55, 2003.
26. M. Li, B. Ma, and L. Wang. On the closest string and substring problems. *J. ACM*, 49(2):157–171, 2002.
27. M. Lothaire. *Combinatorics on Words*. Cambridge Mathematical Library. Cambridge University Press, 2 edition, 1997. `doi:10.1017/CBO9780511566097`.
28. J. Piskorski, M. Sydow, and K. Wieloch. Comparison of string distance metrics for lemmatisation of named entities in polish. In *Language and Technology Conference*, pages 413–427, 2007.
29. G. Recchia and M. M. Louwerse. A comparison of string similarity measures for toponym matching. In *SIGSPATIAL 2013*, pages 54–61, 2013.
30. F. Ruskey, C. Savage, and T. Min Yih Wang. Generating necklaces. *Journal of Algorithms*, 13(3):414–430, 1992.
31. 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.
32. G. Siromoney, R. Siromoney, and T. Robinson. *KAHBI KOLAM AND CYCLE GRAMMARS*, pages 267–300. Springer-Verlag, 1987. URL: `https://www.worldscientific.com/doi/abs/10.1142/9789814368452_0017`, `arXiv:https://www.worldscientific.com/doi/pdf/10.1142/9789814368452_0017`, `doi:10.1142/9789814368452_0017`.
33. M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005. `arXiv:https://doi.org/10.1137/S0097539701388884`, `doi:10.1137/S0097539701388884`.