DE BRUIJN SEQUENCES AND PERFECT FACTORS

CHRIS J. MITCHELL*

Abstract. In this paper we describe new constructions for de Bruijn sequences and Perfect Factors. These constructions are all based upon the idea of constructing one sequence (or set of sequences) from another. As a result of this fact, the sequences obtained from these construction methods possess simple decoding algorithms, based on decoding the sequences used to construct them. Such decoding algorithms are of importance in position location applications.

Key words. de Bruijn sequence, de Bruijn graph, window sequence, perfect factor

1. Introduction.

1.1. De Bruijn sequences, perfect factors and the decoding problem.

In this paper we address two main issues relating to the existence and decoding of Perfect Factors and de Bruijn sequences.

- Perfect Factors, i.e. sets of uniformly long cycles whose elements are drawn from an alphabet of size c and in which every possible v-tuple of elements occurs exactly once, are of significance for two main reasons.
  - They can be used to construct Perfect Maps (or two-dimensional de Bruijn arrays), see for example, [4, 9, 10], which are of practical importance in certain position-location applications.
  - They are special cases of Perfect Maps themselves, and hence their existence is of significance in deciding whether Perfect Maps exist for all parameter sets satisfying certain simple necessary conditions (it has recently been established that these necessary conditions are sufficient for prime power size alphabets, [12, 13]).

  They are also of combinatorial interest in their own right, [4].

  It has been conjectured, [6], that the simple necessary conditions for the existence of a Perfect Factor are sufficient for all finite alphabets and for all window sizes. This conjecture was established by Paterson for c a prime power, [11], and for $v < 5$ in [7]. In this paper we describe two new construction methods for Perfect Factors, yielding Perfect Factors with parameters not previously known to exist.
- The problem of *decoding* de Bruijn sequences and Perfect Maps, i.e. of finding the position within the sequence (or array) of any specified v-tuple (or subarray) is of fundamental importance in certain practical applications (see [2, 3, 14]). It has recently been shown that de Bruijn sequences can be constructed which have simple decoding methods, [8]; in this paper we present another construction method for de Bruijn sequences which also yields sequences with a simple decoding technique.

  In addition, it has been shown that Perfect Maps can be constructed using a combination of Perfect Factors and de Bruijn sequences, for which decoding the Perfect Map can be reduced to decoding its component sequences, [9]. The methods for constructing Perfect Factors presented here all allow simple decoding methods to be devised, and hence contribute to the simpler decoding of certain Perfect Maps.

*Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, England.

1.2. Notation. We first set up some notation which we will use throughout the paper.

We are concerned here with c-ary periodic sequences, where by the term c-ary we mean sequences whose elements are drawn from the set $\{0, 1, \ldots, c-1\}$. We refer throughout to c-ary cycles of period n, by which we mean periodic sequences $[s_0, s_1, \ldots, s_{n-1}]$ where $s_i \in \{0, 1, \ldots, c-1\}$ for every i, $(0 \leq i < n)$.

If $\boldsymbol{t} = (t_0, t_1, \ldots, t_{v-1})$ is a c-ary v-tuple (i.e. $t_i \in \{0, 1, \ldots, c-1\}$ for every i, $(0 \leq i < v)$), and $\boldsymbol{s} = [s_0, s_1, \ldots, s_{n-1}]$ is a c-ary cycle of period n $(n \geq v)$, then we say that $\boldsymbol{t}$ *occurs in* $\boldsymbol{s}$ *at position* j if and only if

$$t_i = s_{i+j}$$

for every i, $(0 \leq i < v)$, where $i + j$ is computed modulo n.

If $\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1}$ are t cycles of the same length, n say, and if

$$\boldsymbol{s}_i = [s_{i0}, s_{i1}, \ldots, s_{i(n-1)}] \quad (0 \leq i < t),$$

then $\mathcal{I}(\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1})$ denotes the t-fold interleaving of these cycles, i.e.

$$\mathcal{I}(\boldsymbol{s}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1}) = [s_{00}, s_{10}, \ldots, s_{(t-1)0}, s_{01}, s_{11}, \ldots, s_{(t-1)(n-1)}],$$

a cycle of length nt.

Given a cycle $\boldsymbol{s} = [s_i]$, $(0 \leq i < n)$, and any integer k, we define $\boldsymbol{T}_k(\boldsymbol{s})$ to be the *cyclic shift* of $\boldsymbol{s}$ by k places *to the right.* I.e. if we write $\boldsymbol{s}' = [s'_i] = \boldsymbol{T}_k(\boldsymbol{s})$ then

$$s'_{i+k} = s_i, \quad (0 \leq i < n)$$

where $i + k$ is calculated modulo n.

Suppose $\boldsymbol{s} = [s_0, s_1, \ldots, s_{n-1}]$ and $\boldsymbol{s}' = [s'_0, s'_1, \ldots, s'_{n'-1}]$ are c- ary cycles of periods n and n' respectively. Then define the *concatenation* of $\boldsymbol{s}$ and $\boldsymbol{s}'$, written

$$\boldsymbol{s}||\boldsymbol{s}'$$

to be the c-ary cycle of period $n + n'$

$$\boldsymbol{t} = [t_0, t_1, \ldots, t_{n+n'-1}] = \boldsymbol{s}||\boldsymbol{s}',$$

where

$$t_i = \begin{cases} s_i & \text{if } 0 \leq i < n \\ s'_{i-n} & \text{if } n \leq i < n + n' \end{cases}$$

In addition, if $\boldsymbol{s}$ is a cycle of length n, and $k > 0$, then $\boldsymbol{s}^k$ denotes the k-fold concatenation of $\boldsymbol{s}$ with itself, and hence $\boldsymbol{s}^k$ is a cycle of period nk.

Throughout we will write $\boldsymbol{0}^i$ for the i-tuple of all zeros and $\boldsymbol{1}^i$ for the i-tuple of all ones.

Finally note that, throughout this paper, the notation (m, n) represents the *Greatest Common Divisor* of m and n (given that m, n are a pair of positive integers).

1.3. Fundamental definitions and results. We next define the objects of fundamental importance to this paper.

Definition 1.1. *If* $\boldsymbol{s} = (s_0, s_1, \ldots, s_{n-1})$ *is a c-ary cycle of period n, then we say that* $\boldsymbol{s}$ *is a* v-window sequence *if no c-ary v-tuple occurs in two distinct positions* *within a period of* $\mathbf{s}$*. Equivalently, it contains* n *distinct* v*-tuples in a period of the cycle.*

Using this definition we also have:

DEFINITION 1.2. *A* c*-ary* de Bruijn sequence of span v *is then simply a* v*-window sequence of period equal to* c^v*; equivalently every possible* c*-ary* v*-tuple occurs precisely once in a period of a de Bruijn sequence.*

A c*-ary* punctured de Bruijn sequence of span v *(sometimes called a* pseudorandom sequence*) is a* v*-window sequence in which every* c*-ary* v*-tuple except for* $\mathbf{0}^v$ *occurs, and so a punctured de Bruijn sequence has period* $c^v - 1$*. A span* v *de Bruijn sequence can be 'punctured' by deleting one of the zeros in* $\mathbf{0}^v$*, and a punctured de Bruijn sequence can be transformed into a de Bruijn sequence by adding a zero to any one of the* $c-1$ *occurrences of* $\mathbf{0}^{v-1}$*. sequence in which every* c*-ary* v*-tuple occurs except for* $\mathbf{0}^v$ *and* $\mathbf{1}^v$*, and hence a doubly punctured de Bruijn sequence has period* c^v-2*. A de Bruijn sequence can be 'doubly punctured' by first puncturing it and then deleting one of the ones in* $\mathbf{1}^v$*, and a doubly punctured de Bruijn sequence can be transformed into a de Bruijn sequence by adding a zero to any of the* $c-1$ *occurrences of* $\mathbf{0}^{v-1}$*, and adding a one to any of the* $c-1$ *occurrences of* $\mathbf{1}^{v-1}$*.*

We next have:

DEFINITION 1.3. *Suppose* n*,* c *and* v *are positive integers, where* $c \geq 2$*. An* (n,c,v)*–Perfect Factor, or simply a* (n,c,v)*–PF, is a collection of* c^v/n c*-ary cycles of period* n *with the property that every* c*-ary* v*-tuple occurs in one of these cycles.*

Note that, because we insist that a Perfect Factor contains exactly c^v/n cycles, and because there are clearly c^v different c-ary v-tuples, each v-tuple will actually occur exactly once somewhere in the collection of cycles (and hence all the cycles are distinct). Also observe that a (c^v,c,v)–PF is simply a c-ary span v de Bruijn sequence.

The following necessary conditions for the existence of a Perfect Factor are trivial to establish.

LEMMA 1.4. *Suppose* A *is a* (n,c,v)*–PF. Then*

1. $n|c^v$*, and*
2. $v < n \leq c^v$ *(or* $n = v = 1$*).*

It was conjectured in [6] that these necessary conditions are sufficient for the existence of a Perfect Factor. Paterson, [11] has shown that the conjecture holds if c is a prime power, and it has also been shown that the conjecture holds if $v < 5$, [7].

Finally we define a related set of combinatorial objects, first introduced in [6].

DEFINITION 1.5. *Suppose* n*,* k*,* c *and* v *are positive integers satisfying* $n|c^v$ *and* $c \geq 2$*. An* (n,k,c,v)*–Perfect Multi-factor, or simply a* (n,k,c,v)*–PMF, is a collection of* c^v/n c*-ary cycles of period* nk *with the property that for every* c*-ary* v*-tuple* $\mathbf{t}$*, and for every integer* j *in the range* $0 \leq j < k$*,* $\mathbf{t}$ *occurs at a position* $p \equiv j \pmod{k}$ *in one of these cycles.*

Note that, because a PMF contains exactly c^v/n cycles of length nk, and because there are c^v different c-ary v-tuples, each v-tuple will actually occur exactly k times in the collection of cycles, once in each of the possible position congruency classes (mod k). This also implies that all the cycles are distinct.

The following necessary conditions for the existence of a PMF are simple to establish.

LEMMA 1.6. *Suppose* A *is a* (n,k,c,v)*–PMF. Then*

1. $n|c^v$*, and*
2. $v < nk$ *(or* $v = nk$ *and* $n = 1$*).*

It has been shown, [6], that the above necessary conditions are sufficient if $k \geq v$.

2. A span-dividing construction for Perfect Factors.

In this section we describe a novel method for constructing a Perfect Factor from a Perfect Multi- factor. This method involves reducing the span and at the same time increasing the alphabet size. The method is of practical interest because a simple decoding algorithm for the Perfect Factor can be derived from a decoding algorithm for the Perfect Multi-factor used to construct it.

2.1. The construction method.

CONSTRUCTION 2.1. *Suppose c, k, n and v are positive integers where $c \geq 2$, $n|c^v$ and $k|v$, and let $A = \{\boldsymbol{a}_i : 0 \leq i < c^v/n\}$ be an (n, k, c, v)-PMF.*

Now define $D = \{\boldsymbol{d}_i : 0 \leq i < c^v/n\}$ to be the set of c^v/n c^k-ary cycles of period n defined so that $\boldsymbol{d}_i$ is obtained from $\boldsymbol{a}_i$ by dividing $\boldsymbol{a}_i$ into disjoint k-tuples and regarding each k-tuple as the c-ary representation of an element from an alphabet of size c^k.

THEOREM 2.2. *Suppose c, k, n, v and A satisfy the conditions of Construction 2.1. If D is constructed from A using Construction 2.1, then D is a $(n, c^k, v/k)$-PF.*

Proof. Let $u = v/k$ and suppose $\boldsymbol{e}$ and $\boldsymbol{e}'$ are u-tuples from D occurring at positions p and p' in cycles $\boldsymbol{d}_i$ and $\boldsymbol{d}_{i'}$ respectively ($0 \leq p, p' < n$ and $0 \leq i, i' < c^v/n$). We need to show that these tuples are distinct unless $p = p'$ and $i = i'$.

Now if $\boldsymbol{e} = \boldsymbol{e}'$ then $\boldsymbol{f} = \boldsymbol{f}'$, where $\boldsymbol{f}$ and $\boldsymbol{f}'$ are c-ary v-tuples derived respectively from $\boldsymbol{e}$ and $\boldsymbol{e}'$ by substituting every c^k-ary element with a c-ary k-tuple (inverting the procedure used to derive D in Construction 2.1). Now $\boldsymbol{f}$ and $\boldsymbol{f}'$ occur at positions kp and kp' in cycles $\boldsymbol{a}_i$ and $\boldsymbol{a}_{i'}$ respectively. Hence, since A is a Perfect Multi- factor and $kp \equiv kp' \pmod{k}$, we have

$$i = i' \text{ and } kp \equiv kp' \pmod{nk}$$

and the desired result follows. □

2.2. An example.

We now give a simple example.

EXAMPLE 2.3. *Let $n = v = 4$ and $c = k = 2$. Also let*

$$A = \left\{ \begin{array}{ll} \boldsymbol{a}_0 = \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right], & \boldsymbol{a}_1 = \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} \right], \\ \boldsymbol{a}_2 = \left[\begin{array}{cccccccc} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array} \right], & \boldsymbol{a}_3 = \left[\begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right] \end{array} \right\},$$

a (4,2,2,4)-PMF.

Then, using the above construction, we obtain

$$D = \{ \boldsymbol{d}_0 = [\,0\ 0\ 3\ 3\,],\ \boldsymbol{d}_1 = [\,2\ 0\ 1\ 3\,], \boldsymbol{d}_2 = [\,1\ 1\ 2\ 2\,],\ \boldsymbol{d}_3 = [\,0\ 2\ 3\ 1\,] \},$$

a (4,4,2)-PF.

2.3. A decoding algorithm.

We now present a simple algorithm for decoding cycles which have been obtained using Construction 2.1; the algorithm is based on the use of a partial decoder for the Perfect Multi- factor A.

THEOREM 2.4. *Suppose c, k, n, v and A satisfy the conditions of Construction 2.1, and D has been constructed from A using Construction 2.1. Suppose also that the pair of functions (E_1, E_2) acts as a partial decoder for A, i.e. if $\boldsymbol{x}$ is a c-ary v-tuple then $0 \leq E_1(\boldsymbol{x}) < c^v/n$, $0 \leq E_2(\boldsymbol{x}) < n$, and $\boldsymbol{x}$ occurs at position $kE_2(\boldsymbol{x})$ in cycle $\boldsymbol{a}_{E_1(\boldsymbol{x})}$ of A. I.e. the partial decoder will find the unique location of the specified tuple in a position congruent to 0 modulo k.*

Then the pair (E_1, E_2) *is a decoder for* D*, i.e. if* $\boldsymbol{y}$ *is a* c^k*-ary* u*-tuple, then* $\boldsymbol{y}$ *occurs at position* $E_2(\boldsymbol{y})$ *in cycle* $\boldsymbol{d}_{E_1(\boldsymbol{y})}$.

Proof. This result follows immediately from the way in which D is constructed. □

2.4. Constructing suitable PMFs. We now consider the problem of constructing PMFs with parameters suitable for use in Construction 2.1. We first observe that, using Constructions 6.1 and 6.4 of [6], we have:

THEOREM 2.5. *Suppose* c*,* m*,* n*,* s *and* v *are positive integers where* $c \geq 2$*,* $m|n$ *and* $(s, m) = 1$*, and suppose also that there exists an* (n, c, v)*–PF. Then an* $(m, ns/m, c, v)$*–PMF can be constructed.*

REMARK 2.6. *Examination of the construction methods in [6] reveals that a decoding algorithm for the PMF can very easily be derived from a decoding algorithm for the PF used to construct it.*

Note also that the (4,2,2,4)–PMF of Example 2.3 was obtained from an (8,2,4)–PF using exactly this method.

There are two simple ways in which we can combine Theorem 2.5 with our new construction method.

- First suppose that $n = c^v$, $k|v$, and $(k, c^v) = 1$, and put $m = n$ and $s = k$ (and hence $(s, m) = 1$). Then, starting with a (c^v, c, v)–PF (a c-ary span v de Bruijn sequence), we can obtain a (c^v, k, c, v)–PMF. Now, since $k|v$, we can apply Construction 2.1 to obtain a $(c^v, c^k, v/k)$–PF, i.e. a c^k-ary span v/k de Bruijn sequence. Most significantly this new de Bruijn sequence can be trivially decoded using a decoder for the de Bruijn sequence used to construct it.
- Second suppose $k|v$ and $n|c^v$, and put $m = n/(k, n)$ and $s = k/(k, n)$ (and hence $(s, m) = 1$). Then, starting with a (n, c, v)–PF, we can obtain a $(n/(k, n), k, c, v)$–PMF. Now, since $k|v$, we can apply Construction 2.1 to obtain an $(n/(k, n), c^k, v/k)$–PF. Again, this new PF can be trivially decoded using a decoder for the PF used to construct it.

REMARK 2.7. *Note that, in the first case considered immediately above, we could replace the initial de Bruijn sequence with any* c*-ary* v*-window sequence of period* n*, as long as* $(n, k) = 1$ *and* $k|v$*. We would then obtain a* c^k*-ary* (v/k)*-window sequence, also of period* n*. Thus if* $(c^v - 1, k) = 1$ *then we could start with a punctured* c*-ary span* v *de Bruijn sequence, in which case the final sequence would also be a punctured de Bruijn sequence.*

2.5. Example. EXAMPLE 2.8. *Let* $v = 4$*,* $c = k = 2$ *and* $n = c^v - 1 = 15$ *(and hence* $(k, n) = (2, 15) = 1$*). Also let*

$$\boldsymbol{a}' = [\ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ ],$$

a 2-ary span 4 punctured de Bruijn sequence.

Then, using Constructions 6.1 and 6.4 of [6], we obtain

$$\boldsymbol{a} = [\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ ].$$

Using Construction 2.1 we obtain

$$\boldsymbol{d} = [\ 0\ \ 1\ \ 0\ \ 3\ \ 1\ \ 1\ \ 3\ \ 2\ \ 0\ \ 2\ \ 1\ \ 2\ \ 2\ \ 3\ \ 3\ ],$$

a 4-ary span 2 punctured de Bruijn sequence.

3. Constructing Perfect Factors by interleaving.

We now present another method for constructing Perfect Factors with a simple decoding algorithm. It also enables the construction of Perfect Factors for parameter sets for which the existence question was previously unanswered (examples of new parameter sets are given in Section 3.4 below).

3.1. The construction method.

We start by describing the method of construction.

Construction 3.1. *Suppose c, n, t, v are positive integers satisfying $c \geq 2$ and $t | n^{t-1}$. Moreover suppose that*

$$A = \{\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{c^v/n-1}\}$$

is an (n, c, v)-PF.

Consider the set S of all n-ary t-tuples $(x_0, x_1, \ldots, x_{t-1})$ with the property that $\sum_{i=0}^{t-1} x_i \equiv n-1 \pmod n$. If $\boldsymbol{x}, \boldsymbol{y} \in S$ then write $\boldsymbol{x} \sim \boldsymbol{y}$ if and only if $\boldsymbol{x}$ can be obtained from $\boldsymbol{y}$ by a cyclic shift operation. It is straightforward to verify that $\sim$ is an equivalence relation on S which partitions S into n^{t-1}/t classes each of size t. Now let

$$X = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n^{t-1}/t-1}\}$$

be a set of elements of S chosen so that X contains precisely one element of each equivalence class under $\sim$.

Next let

$$U = \{(\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}) \ : \ \boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}} \in A\}$$

be the set of all t-tuples of elements of A, and hence $|U| = c^{tv}/n^t$.

Finally let B the set of all interleaved cycles of the form

$$\mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{i_0}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{i_1}), \boldsymbol{T}_{x_0+x_1}(\boldsymbol{a}_{i_2}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{i_{t-1}})),$$

where $(x_0, x_1, \ldots, x_{t-1}) \in X$ and $(\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}) \in U$. Hence $|B| = |X|.|U| = (n^{t-1}/t)(c^{tv}/n^t) = c^{tv}/tn$.

We can now state and prove the following result.

Theorem 3.2. *Suppose c, n, t, v and A satisfy the conditions of Construction 3.1. If B is constructed from A using Construction 3.1 then B is a (tn, c, tv)-PF.*

Proof. Suppose $\boldsymbol{y}$ is any c-ary tv-tuple. We need to show that $\boldsymbol{y}$ occurs in one of the cycles of B. Suppose

$$\boldsymbol{y} = \mathcal{I}(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1})$$

where $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}$ are c-ary v-tuples. Now suppose that $\boldsymbol{z}_i$ occurs in cycle $\boldsymbol{a}_{\ell_i}$ at position k_i, for every i satisfying $0 \leq i < t$. In addition we define a further n-ary t-tuple $\boldsymbol{x} = (x_0, x_1, \ldots, x_{t-1})$ where $x_i \equiv k_i - k_{i+1} \pmod n$, for every i satisfying $0 \leq i < t-1$, and $x_{t-1} \equiv k_{t-1} - k_0 - 1 \pmod n$.

First observe that $\boldsymbol{x} \in S$, since

$$\sum_{i=0}^{t-1} x_i \equiv \sum_{i=0}^{t-2} (k_i - k_{i+1}) + (k_{t-1} - k_0 - 1) \equiv -1 \pmod n.$$

Hence there exists some cyclic shift of $\boldsymbol{x}$, say

$$\boldsymbol{T}_{t-u}(\boldsymbol{x}) = (x_u, x_{u+1}, \ldots, x_{t-1}, x_0, \ldots, x_{u-1}),$$

which is a member of X. Hence if we define the n-ary t-tuple $(v_0, v_1, \ldots, v_{t-1})$ by

$$v_i = \begin{cases} 0 & \text{if } i = 0 \\ \sum_{j=u}^{i+u-1} x_j \bmod n & \text{if } 0 < i \leq t-u \\ \sum_{j=u}^{t-1} x_j + \sum_{j=0}^{i+u-t-1} x_j \bmod n & \text{if } t-u < i \leq t-1 \end{cases}$$

then the following cycle is a member of B:

$$\boldsymbol{w} = \mathcal{I}(\boldsymbol{T}_{v_0}(\boldsymbol{a}_{\ell_u}), \boldsymbol{T}_{v_1}(\boldsymbol{a}_{\ell_{u+1}}), \ldots, \boldsymbol{T}_{v_{t-u-1}}(\boldsymbol{a}_{\ell_{t-1}}), \boldsymbol{T}_{v_{t-u}}(\boldsymbol{a}_{\ell_0}), \ldots, \boldsymbol{T}_{v_{t-1}}(\boldsymbol{a}_{\ell_{u-1}})).$$

Now z_{u+i} occurs in $\boldsymbol{T}_{v_i}(\boldsymbol{a}_{\ell_{u+i}})$ at position $k_{u+i} + v_i$, $(0 \leq i < t-u)$, and z_i occurs in $\boldsymbol{T}_{v_{i+t-u}}(\boldsymbol{a}_{\ell_i})$ at position $k_i + v_{t-u+i}$, $(0 \leq i \leq u-1)$. In addition, by definition of (x_i) we have

$$v_i = \begin{cases} 0 & \text{if } i = 0 \\ k_u - k_{u+i} \bmod n & \text{if } 0 < i < t-u \\ k_u - k_{u-t+i} - 1 \bmod n & \text{if } t-u \leq i \leq t-1 \end{cases}$$

Thus z_{u+i} occurs in $\boldsymbol{T}_{v_i}(\boldsymbol{a}_{\ell_{u+i}})$ at position k_u, $(0 \leq i < t-u)$, and z_i occurs in $\boldsymbol{T}_{v_{i+t-u}}(\boldsymbol{a}_{\ell_i})$ at position $k_u - 1$, $(0 \leq i \leq u-1)$. Hence $\boldsymbol{y}$ occurs in $\boldsymbol{w}$ at position $k_u t - u$ and the result follows. □

3.2. Examples. Before proceeding we give two simple examples of the construction method.

EXAMPLE 3.3. *Let $n = 4$ and $c = v = t = 2$. Then let A be the following $(4, 2, 2)$-PF (a de Bruijn sequence):*

$$\boldsymbol{a}_0 = [\ 0 \quad 0 \quad 1 \quad 1\ ]$$

Then

$$S = \{\ (\ 0 \quad 3\ ),\quad (\ 3 \quad 0\ ),\quad (\ 2 \quad 1\ ),\quad (\ 1 \quad 2\ )\ \}.$$

Then we can define

$$X = \{\ (\ 0 \quad 3\ ),\quad (\ 2 \quad 1\ )\ \}.$$

In addition

$$U = \{(\ \boldsymbol{a}_0, \quad \boldsymbol{a}_0\ )\}.$$

Hence

$$\begin{aligned} B &= \{\,\mathcal{I}(\ \boldsymbol{T}_0(\boldsymbol{a}_0),\ \boldsymbol{T}_0(\boldsymbol{a}_0)\ ),\ \mathcal{I}(\ \boldsymbol{T}_0(\boldsymbol{a}_0),\ \boldsymbol{T}_2(\boldsymbol{a}_0)\ )\,\} \\ &= \{\,\mathcal{I}(\ [\,0\ 0\ 1\ 1\,]\ ,\ [\,0\ 0\ 1\ 1\,]\ ),\ \mathcal{I}(\ [\,0\ 0\ 1\ 1\,]\ ,\ [\,1\ 1\ 0\ 0\,]\ )\,\} \\ &= \{\,[\,0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\,]\ ,\ [\,0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\,]\,\} \end{aligned}$$

is a ($8, 2, 4$*)-PF.*

EXAMPLE 3.4. *Let $n = c = t = 3$ and $v = 1$. Then let A be the following $(3,3,1)$-PF (a de Bruijn sequence):*

$$\boldsymbol{a}_0 = [\ 0 \quad 1 \quad 2\ ]$$

Then

$$\begin{aligned} S = \{ & (\ 0 \quad 0 \quad 2\ ),\ (\ 0 \quad 2 \quad 0\ ),\ (\ 0 \quad 1 \quad 1\ ), \\ & (\ 2 \quad 0 \quad 0\ ),\ (\ 2 \quad 2 \quad 1\ ),\ (\ 2 \quad 1 \quad 2\ ), \\ & (\ 1 \quad 0 \quad 1\ ),\ (\ 1 \quad 2 \quad 2\ ),\ (\ 1 \quad 1 \quad 0\ )\ \} \end{aligned}$$

Then we can define

$$X = \{\ (\ 0 \quad 0 \quad 2\ ),\ (\ 0 \quad 1 \quad 1\ ),\ (\ 2 \quad 2 \quad 1\ )\ \}.$$

In addition

$$U = \{(\ \boldsymbol{a}_0,\quad \boldsymbol{a}_0,\quad \boldsymbol{a}_0\ )\}.$$

Hence

$$\begin{aligned} B &= \{\mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_{0+0}(\boldsymbol{a}_0)), \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_{0+1}(\boldsymbol{a}_0)), \\ &\quad \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_0), \boldsymbol{T}_2(\boldsymbol{a}_0), \boldsymbol{T}_{2+2}(\boldsymbol{a}_0))\} \\ &= \{\mathcal{I}([0\,1\,2],[0\,1\,2],[0\,1\,2]), \mathcal{I}([0\,1\,2],[0\,1\,2],[2\,0\,1]), \mathcal{I}([0\,1\,2],[1\,2\,0],[2\,0\,1])\} \\ &= \{[0\,0\,0\,1\,1\,1\,2\,2\,2], [0\,0\,2\,1\,1\,0\,2\,2\,1], [0\,1\,2\,1\,2\,0\,2\,0\,1]\} \end{aligned}$$

is a $(9,3,3)$-PF.

3.3. A decoding algorithm. We next show how, given a Perfect Factor constructed using the above method, a simple decoding algorithm can be devised which reduces decoding the constructed Perfect Factors to decoding the Perfect Factor and the set of rotation vectors used as components in the construction.

ALGORITHM 3.5. *Suppose c, n, t, v and A satisfy the conditions of Construction 3.1, and B has been constructed from A using Construction 3.1. Suppose also that the pair of functions (E_1, E_2) acts as a decoder for A, i.e. if $\boldsymbol{z}$ is a c-ary v-tuple then $0 \leq E_1(\boldsymbol{z}) < c^v/n$ and $0 \leq E_2(\boldsymbol{z}) < n$ and $\boldsymbol{z}$ occurs at position $E_2(\boldsymbol{z})$ in cycle $\boldsymbol{a}_{E_1(\boldsymbol{z})}$ of A.*

We also need to define labellings for the sets U and B (defined in Construction 3.1). If $0 \leq i < c^{tv}/n^t$, then suppose $i_{t-1}i_{t-2}\ldots, i_1 i_0$ is the (c^v/n)-ary representation of i (with least significant digit i_0), i.e. $0 \leq i_j < c^v/n$ $(0 \leq j < t)$ and

$$i = \sum_{j=0}^{t-1} (c^v/n)^j i_j,$$

and let

$$\boldsymbol{u}_i = (\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}).$$

It should be clear that $U = \{\boldsymbol{u}_i\ :\ 0 \leq i < c^{tv}/n^t\}$.

Next, if $\boldsymbol{u}_i \in U$, say

$$\boldsymbol{u}_i = (\boldsymbol{a}_{i_0}, \boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{t-1}}),$$

and $\boldsymbol{x}_j \in X$*, say*

$$\boldsymbol{x}_j = (x_0, x_1, \ldots, x_{n^{t-1}/t-1}),$$

then put

$$\boldsymbol{b}_{ij} = \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{i_0}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{i_1}), \boldsymbol{T}_{x_0+x_1}(\boldsymbol{a}_{i_2}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{i_{t-1}})),$$

and hence $B = \{\boldsymbol{b}_{ij} \,:\, 0 \leq i < c^{tv}/n^t, 0 \leq j < n^{t-1}/t\}$.

Define the triple of functions

$$\begin{aligned} F_{11} :& T \rightarrow \{0, 1, \ldots, c^{tv}/n^t - 1\} \\ F_{12} :& T \rightarrow \{0, 1, \ldots, n^{t-1}/t - 1\} \\ F_2 :& \quad T \rightarrow \{0, 1, \ldots, nt - 1\} \end{aligned}$$

as follows, where T *is the set of all* c*-ary* tv*-tuples.*

First suppose $\boldsymbol{y} \in T$*, and suppose*

$$\boldsymbol{y} = \mathcal{I}(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1}).$$

Next put

$$\boldsymbol{w} = (w_0, w_1, \ldots, w_{t-1}) = (E_2(\boldsymbol{z}_0), E_2(\boldsymbol{z}_1), \ldots, E_2(\boldsymbol{z}_{t-1})),$$

and let

$$\boldsymbol{x}' = (x'_0, x'_1, \ldots, x'_{t-1}) = (w_0 - w_1, w_1 - w_2, \ldots, w_{t-2} - w_{t-1}, w_{t-1} - w_0 - 1).$$

Now $\boldsymbol{x}' \in S$ *(as defined in Construction 3.1) and hence suppose*

$$\boldsymbol{x}' = \boldsymbol{T}_r(\boldsymbol{x}_q),$$

for some $\boldsymbol{x}_q \in X$*, (where* $0 \leq r < t$*). We now put* $F_{12}(\boldsymbol{y}) = q$.

Next put

$$\boldsymbol{g}' = (g'_0, g'_1, \ldots, g'_{t-1}) = (E_1(\boldsymbol{z}_0), E_1(\boldsymbol{z}_1), \ldots, E_1(\boldsymbol{z}_{t-1}))$$

and let

$$\boldsymbol{g} = (g_0, g_1, \ldots, g_{t-1}) = \boldsymbol{T}_r(\boldsymbol{g}').$$

Finally put

$$F_{11}(\boldsymbol{y}) = \sum_{i=0}^{t-1} g_i (c^v/n)^i,$$

and

$$F_2(\boldsymbol{y}) = tE_2(\boldsymbol{z}_r) - r.$$

THEOREM 3.6. *If* B *and* (F_{11}, F_{12}, F_2) *are defined as in Algorithm 3.1, then the pair* $((F_{11}, F_{12}), F_2)$ *is a decoder for* B*, i.e. if* $\boldsymbol{y}$ *is a* c*-ary* tv*-tuple, then* $\boldsymbol{y}$ *occurs at position* $F_2(\boldsymbol{y})$ *in cycle* $\boldsymbol{b}_{F_{11}(\boldsymbol{y}), F_{12}(\boldsymbol{y})}$.

Proof. Suppose $\boldsymbol{y}$, $(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{t-1})$, F_{11}, F_{12} and F_2 are as in the Algorithm. We need to show that $\boldsymbol{y}$ occurs at position $F_2(\boldsymbol{y})$ in cycle $\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})}$.

First observe that

$$F_{11}(\boldsymbol{y}) = \sum_{i=0}^{t-1} g_i(c^v/n)^i$$

and

$$\boldsymbol{x}_{F_{12}(\boldsymbol{y})} = (x_0, x_1, \ldots, x_{t-1}) \in X.$$

Now, by definition:

$$\begin{aligned}
\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})} &= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{g_0}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{g_1}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{g_{t-1}})) \\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{g'_r}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{g'_{r+1}}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{g'_{r-1}})) \\
&\qquad \text{(since } \boldsymbol{g} = \boldsymbol{T}_r(\boldsymbol{g}')\text{)} \\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{x_0}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \ldots, \boldsymbol{T}_{x_0+x_1+\cdots+x_{t-2}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})})) \\
&\qquad \text{(by definition of } \boldsymbol{g}'\text{)} \\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{x'_r}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \boldsymbol{T}_{x'_r+x'_{r+1}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})}), \ldots, \\
&\quad \boldsymbol{T}_{x'_r+x'_{r+1}+\cdots+x'_{r-2}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})})) \\
&\qquad \text{(since } \boldsymbol{x}' = \boldsymbol{T}_r(\boldsymbol{x}_q)\text{)} \\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{w_r-w_{r+1}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \boldsymbol{T}_{w_r-w_{r+2}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})}), \ldots, \\
&\quad \boldsymbol{T}_{w_r-w_{t-1}}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{t-1})}), \boldsymbol{T}_{w_r-w_0-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_0)}), \ldots, \boldsymbol{T}_{w_r-w_{r-1}-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})})) \\
&\qquad \text{(by definition of } \boldsymbol{x}'\text{)} \\
&= \mathcal{I}(\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)}), \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})}), \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+2})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})}), \\
&\quad \ldots, \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{t-1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{t-1})}), \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_0)-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_0)}), \ldots, \\
&\quad \boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r-1})-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})})) \\
&\qquad \text{(by definition of } \boldsymbol{w}\text{)}
\end{aligned}$$

Now, since $\boldsymbol{z}_i$ occurs at position $E_2(\boldsymbol{z}_i)$ in $\boldsymbol{a}_{E_1(\boldsymbol{x}_i)}$, $(0 \leq i < t)$, we have

- $\boldsymbol{z}_r$ occurs in $\boldsymbol{T}_0(\boldsymbol{a}_{E_1(\boldsymbol{z}_r)})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_{r+1}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+1})})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_{r+2}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r+2})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r+2})})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_{t-1}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{t-1})}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{t-1})})$ at position $E_2(\boldsymbol{z}_r)$,
- $\boldsymbol{z}_0$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_0)-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_0)})$ at position $E_2(\boldsymbol{z}_r) - 1$, and
- $\boldsymbol{z}_{r-1}$ occurs in $\boldsymbol{T}_{E_2(\boldsymbol{z}_r)-E_2(\boldsymbol{z}_{r-1})-1}(\boldsymbol{a}_{E_1(\boldsymbol{z}_{r-1})})$ at position $E_2(\boldsymbol{z}_r) - 1$.

Hence $\boldsymbol{y}$ occurs in $\boldsymbol{b}_{F_{11}(\boldsymbol{y}),F_{12}(\boldsymbol{y})}$ at position $tE_2(\boldsymbol{z}_r) - r = F_2(\boldsymbol{y})$, and the result follows.
□

3.4. New parameter sets. We conclude our discussion of this method for constructing Perfect Factors by showing how it can be used to construct Perfect Factors with parameters for which the existence question was previously unresolved.

As has already been mentioned, in [7] the necessary conditions of Lemma 1.4 have been shown to be sufficient for the existence of a Perfect Factor when $v < 5$. Construction 3.1 does not help with any of the unresolved parameter sets for $v = 5$, and so we examine the case $v = 6$.

Now, by Theorem 7.1 of [6], Perfect Factors exist for all triples $(n, c, 6)$ satisfying the conditions of Lemma 1.4 with the possible exceptions of:

- $n = 10$, $c = 10d$ $(d \geq 1)$,
- $n = 12$, $c = 6d$ $(d \geq 1)$,
- $n = 15$, $c = 15d$ $(d \geq 1)$,
- $n = 20$, $c = 10d$ $(d \geq 1)$,
- $n = 30$, $c = 30d$ $(d \geq 1)$, and
- $n = 60$, $c = 30d$ $(d \geq 1)$.

Next observe that, by Theorem 26 of [7], the following Perfect Factors exist:

- $(6, 6d, 3)$–PFs, $d \geq 1$,
- $(10, 10d, 3)$–PFs, $d \geq 1$, and
- $(30, 30d, 3)$–PFs, $d \geq 1$.

Applying Construction 3.1 to all of these Perfect Factors (in each case with $t = 2$) we obtain Perfect Factors for precisely the parameter sets in the second, fourth and sixth of the cases listed above.

This means that the only unresolved cases for $v = 6$ are

- $n = 10$, $c = 10d$ $(d \geq 1)$,
- $n = 15$, $c = 15d$ $(d \geq 1)$, and
- $n = 30$, $c = 30d$ $(d \geq 1)$.

4. Summary and conclusions. Using recursive methods of construction we have made further progress towards proving the conjecture of [6], namely that Perfect Factors exist for all parameter sets satisfying the necessary conditions of Lemma 1.4. All the construction methods in this paper, both for de Bruijn sequences and for Perfect Factors, admit simple methods of decoding, making their use in practical applications advantageous.

Finally, it is interesting to observe that, when put together with the de Bruijn sequence construction methods in [8] and [11] (special case of Lemma 5.1), there exists a series of construction methods for building one de Bruijn sequence out of another. If it turns out that some or all of these construction methods have ‘complexity preserving properties’ (c.f. the Lempel construction, [5]), then there may exist the means to make further progress with the long- standing problem of discovering for which linear complexities there exist de Bruijn sequences (see, for example, [1]).

Acknowledgements. The author would like to thank an anonymous referee for invaluable comments regarding Construction 2.1, which have both improved and shortened the paper.

REFERENCES

[1] S. Blackburn, T. Etzion, and K. Paterson, *Permutation polynomials, de Bruijn sequences and linear complexity*, Journal of Combinatorial Theory (Series A), (to appear).

[2] J. Bondy and U. Murty, *Graph theory with applications*, Elsevier, 1976.

[3] J. Burns and C. Mitchell, *Coding schemes for two-dimensional position sensing*, in Cryptography and Coding III, M. Ganley, ed., Oxford University Press, 1993, pp. 31–66.

[4] T. Etzion, *Constructions for perfect maps and pseudo-random arrays*, IEEE Transactions on Information Theory, **34** (1988), pp. 1308–1316.

[5] A. Lempel, *On a homomorphism of the de Bruijn graph and its application to the design of feedback shift registers*, IEEE Transactions on Computers, **C-19** (1970), pp. 1204–1209.

[6] C. Mitchell, *Constructing c-ary perfect factors*, Designs, Codes and Cryptography, **4** (1994), pp. 341–368.

[7] ———, *New c-ary perfect factors in the de Bruijn graph*, in Codes and Cyphers, P. Farrell, ed., Formara Ltd., Southend, 1995, pp. 299–313. Proceedings of the fourth IMA Conference on Cryptography and Coding, Cirencester, December 1993.

[8] C. Mitchell, T. Etzion, and K. Paterson, *A method for constructing decodable de Bruijn sequences*, IEEE Transactions on Information Theory, (to appear).

[9] C. Mitchell and K. Paterson, *Decoding perfect maps*, Designs, Codes and Cryptography, **4** (1994), pp. 11–30.

[10] K. Paterson, *Perfect maps*, IEEE Transactions on Information Theory, **40** (1994), pp. 743–753.

[11] ———, *Perfect factors in the de Bruijn graph*, Designs, Codes and Cryptography, **5** (1995), pp. 115–138.

[12] ———, *New classes of perfect maps I*, Journal of Combinatorial Theory (Series A), 73 (1996), pp. 302–334.

[13] ———, *New classes of perfect maps II*, Journal of Combinatorial Theory (Series A), 73 (1996), pp. 335–345.

[14] E. Petriu, *New pseudorandom/natural code conversion method*, ElectronicsLetters, **24** (1988), pp. 1358–1359.