Constructions and bounds for codes with restricted overlaps

Simon R. Blackburn, *Senior Member, IEEE,* Navid Nasr Esfahani, *Member, IEEE,* Donald L. Kreher, and Douglas R. Stinson

***Abstract*—Non-overlapping codes have been studied for almost 60 years. In such a code, no proper, non-empty prefix of any codeword is a suffix of any codeword. In this paper, we study codes in which overlaps of certain specified sizes are forbidden. We prove some general bounds and we give several constructions in the case of binary codes. Our techniques also allow us to provide an alternative, elementary proof of a lower bound on non-overlapping codes due to Levenshtein [9] in 1964.**

***Index Terms*—Non-overlapping codes, weakly mutually uncorrelated codes, cross-bifix-free codes.**

I. Introduction

Let u and v be (not necessarily distinct) words of length n over a specified alphabet. Let t be an integer such that $1 \leq t \leq n-1$. We say that u and v have a *t-overlap* if the prefix of u of length t is identical to the suffix of v of length t. A code C is *t-overlap-free* if no codewords u and v in C have a t-overlap. A code C is *non-overlapping* if it is t-overlap-free for all t such that $1 \leq t \leq n-1$.

S. R. Blackburn is with the Department of Mathematics, Royal Holloway, University of London, Egham, Surrey TW20 0EX,United Kingdom

N. N. Esfahani is with the Department of Computer Science, Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada

D. L. Kreher is with the Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931-1295, U.S.A.

D. R. Stinson is with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo ON, N2L 3G1, Canada

D. R. Stinson is also with the School of Mathematics and Statistics, Carleton University, Ottawa, Ontario, K1S 5B6, Canada

D. R. Stinson's research is supported by NSERC discovery grant RGPIN-03882.

Motivated by applications including frame synchronization, non-overlapping codes have been studied by numerous authors over the years, e.g., see [2], [3], [4], [5], [9], [12], [17].

Here we consider a less restrictive definition. Suppose that t_1 and t_2 are integers such that $1 \leq t_1 \leq t_2 \leq n-1$. We say that a code C is *(t_1, t_2)-overlap-free* if it is t-overlap-free for all t such that $t_1 \leq t \leq t_2$. Two special cases of interest are codes that are $(k, n-1)$-overlap-free (i.e., overlaps of size at least k are not allowed) and codes that are $(1, k)$-overlap-free (i.e., overlaps of size at most k are not allowed).

Motivated by applications in DNA-based storage systems and synchronization protocols, $(k, n-1)$-overlap-free codes were studied in [16] and termed k-weakly mutually uncorrelated codes. On the other hand, $(1, k)$-overlap-free codes could be useful in a setting where we have "approximate" synchronization, i.e., if we can assume that codewords will not "drift" too much. For example, suppose (see Figure 1) that we transmit blocks 1, 2, 3 and so on, each a codeword of the same length n. We consider channels where a received block might be corrupted, with bits changed and up to k bits inserted or deleted. We detect a loss of synchronization by checking if each block of n received bits is a codeword. If we use an $(n-k, n-1)$-overlap-free code, we are guaranteed to detect a loss of synchronization after $2n$ bits are received. If we use a $(1, k)$-overlap-free code, we are guaranteed to detect a loss of synchronization after $3n$ bits if there are inserted bits, but after only n bits are received if bits have been deleted. Thus, in channels where dele-

tions are more likely than insertions, $(1,k)$-overlap-free codes have an advantage over $(n-k,n-1)$-overlap-free codes.

We comment that codes for synchronization is a large and thriving area, which we cannot hope to cover comprehensively here. Notable related problems are codes designed to correct bursts of insertions or deletions [6], [10], [11], [15], and variants of the non-overlapping problem in two-dimensions [1].

In general, we wish to determine the maximum number of codewords in a (t_1,t_2)-overlap-free code. In Section II, we prove two upper bounds, on the size of $(k,n-1)$-overlap-free codes and $(1,k)$-overlap-free codes. Section III begins a study of constructions for $(1,k)$-overlap-free codes over a binary alphabet. Our first construction, the Doubling Construction, gives an inductive approach to the construction of these codes. Section IV introduces a graph-based interpretation of these codes. This approach is used to prove the optimality of our codes for $k \leq 6$. Section V presents an explicit construction that we term the m-minimum Construction, as well as the closely related Zero Block Construction. Both of these permit "good" codes to be constructed for specified values of k. The second of these two constructions can be analyzed by exploiting a connection with n-step Fibonacci numbers. We provide exact as well as asymptotic bounds; it is shown that the constructed codes are within a small constant factor of being optimal. Section VI revisits the classical problem of non-overlapping codes and discusses how our techniques apply to this problem. In particular, we provide an alternative, elementary proof of a lower bound on non-overlapping codes due to Levenshtein [9] in 1964. Finally, Section VII is a brief discussion and summary.

II. Two upper bounds

Chee *et al.* [5] proved that if C is a non-overlapping code over an alphabet of cardinality q, then $|C| \leq q^n/(2n-1)$. This bound can be proven using a simple combinatorial argument; see Blackburn [4]. Also, a stronger bound has been proven by Levenshtein [12] using analytic combinatorics.

For $(k,n-1)$-overlap-free codes, Yazdi *et al.* [16] proved that such a code C satisfies the inequality $|C| \leq q^n/(n-k+1)$. The following stronger bound can be proven using the argument from [4]. Note that the special case $k=1$ of Theorem II.1 is essentially the bound proven in [4].

Theorem II.1. *If C is a $(k,n-1)$-overlap-free code over an alphabet of cardinality q, then*

$$|C| \leq \frac{q^n}{2n-2k+1}.$$

Proof. Let C be a $(k,n-1)$-overlap-free code over an alphabet F of cardinality q. For $w \in F^{2n-2k+1}$ and $1 \leq i \leq 2n-2k+1$, define $w(i) = (w_i, w_{i+1}, \ldots, w_{i+n-1})$, where the subscripts are reduced modulo $2n-2k+1$. Thus, $w(i)$ is the cyclic subword of length n of w starting at w_i. Define

$$X = \{(w,i) : w \in F^{2n-2k+1}, \\ 1 \leq i \leq 2n-2k+1, w(i) \in C\}.$$

Suppose there exists $w \in F^{2n-2k+1}$ and i, i' such that $i \neq i'$ and $(w,i),(w,i') \in X$. We claim that (w,i) and (w,i') have an overlap of size at least k. This occurs because the overlap between (w,i) and (w,i') is at least

$$n + n - (2n-2k+1) = 2k-1,$$

and hence the overlap at one end is at least $\lceil (2k-1)/2 \rceil = k$. This violates the non-overlapping properties of C. Hence, for each $w \in F^{2n-2k+1}$ there is at most one i such that $(w,i) \in X$. Thus it follows that

$$|X| \leq q^{2n-2k+1}.$$

Also, $|X| = (2n-2k+1)|C|q^{n-2k+1}$, since there are $2n-2k+1$ choices for i, $|C|$ choices for $w(i)$, and q^{n-2k+1} choices for the remaining entries in w.

Hence,

$$(2n-2k+1)|C|q^{n-2k+1} \leq q^{2n-2k+1},$$

which immediately yields the stated upper bound on $|C|$. □

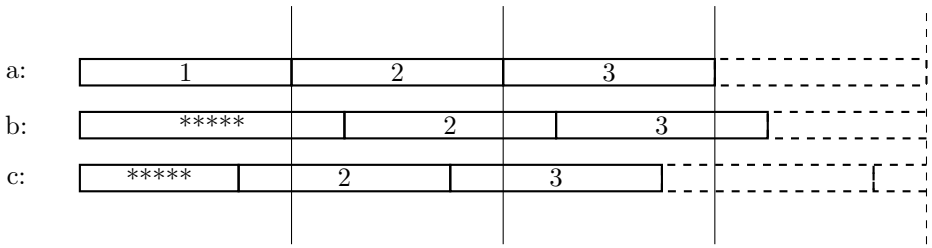

Fig. 1. Transmitting codewords when blocks are corrupted, and change length: (a) Transmitted data, (b) Inserted and corrupted bits, and (c) Deleted and corrupted bits.

It is natural to ask if there is a "related" upper bound for $(1,k)$-overlap-free codes.

Theorem II.2. *Let C be a $(1,k)$-overlap-free code, where $k \leq n/2$. Then*

$$|C| \leq \tfrac{1}{2k} q^n.$$

Proof. Let $k \leq n/2$ and let C be a $(1,k)$-overlap-free code. Let $\mathcal{X}$ be the set of all codewords with the middle $n-2k$ positions removed. Clearly, $|C| \leq q^{n-2k}|\mathcal{X}|$. The elements of $\mathcal{X}$ are q-ary words of length $2k$. We have $\mathcal{X} = \mathcal{Y} \cup \mathcal{Z}$, where the elements in $\mathcal{Y}$ have (cyclic) period strictly dividing $2k$, and the elements of $\mathcal{Z}$ have period exactly $2k$.

Suppose $y \in \mathcal{Y}$. If y has period p, where p strictly divides $2k$, then $p \leq k$. Then the first and last p elements of a corresponding codeword $w \in \mathcal{X}$ agree. This codeword has a p-overlap with itself, which contradicts the $(1,k)$-overlap-free property. We conclude that $\mathcal{Y} = \emptyset$.

Now we claim that no pair of distinct elements in $\mathcal{Z}$ are cyclic shifts of each other. For a contradiction, suppose that z_1, z_2 are a pair of distinct elements from $\mathcal{Z}$ that are cyclic shifts of each other. Let c_1, c_2 be the corresponding codewords in C. Write σ for the 'cyclic shift left by one position' operator, so

$$\sigma(a_1 a_2 \cdots a_{2k}) = a_2 a_3 \cdots a_{2k} a_1.$$

Then $z_1 = \sigma^j(z_2)$ for some $j \in \{1, \ldots, 2k-1\}$. Swapping z_1 and z_2 if needed, we may assume that j lies in the set $\{1, \ldots, k\}$ (as swapping replaces j by $2k-j$). But now the j-prefix of z_2 is equal to the j-suffix of z_1. So the j-prefix of c_2 is equal to the j-suffix of c_1. This contradicts our assumption that C is $(1,k)$-overlap-free, and so our claim follows.

We can partition the set of all q-ary sequences of length $2k$ and period exactly $2k$ into equivalence classes under cyclic shift. Each class contains $2k$ sequences, and so there are at most $q^{2k}/2k$ classes. The previous paragraph shows that no class contains two elements of $\mathcal{Z}$, and so $|\mathcal{Z}| \leq q^{2k}/2k$. Hence

$$\begin{aligned} |C| \leq q^{n-2k}|\mathcal{X}| &= q^{n-2k}(|\mathcal{Y}| + |\mathcal{Z}|) \\ &= q^{n-2k}|\mathcal{Z}| \\ &\leq \frac{q^n}{2k}. \end{aligned}$$

□

III. Constructions

In this section, and the next two sections, we investigate constructions and bounds for $(1,k)$-overlap-free codes. All of our constructions will be based on the following template.

Construction III.1. *Let F be an alphabet of size q and let n and t be positive integers such that $n \geq 2t$. Let $\mathcal{P}$ and $\mathcal{S}$ be two sets of t-tuples from F^t. Define*

$$C(\mathcal{P}, \mathcal{S}, n, t) = \{p \parallel x \parallel s : p \in \mathcal{P}, \\ s \in \mathcal{S}, x \in F^{n-2t}\}.$$

Thus a codeword $c \in C$ has a prefix chosen from $\mathcal{P}$, a suffix chosen from $\mathcal{S}$, and the remaining $n-2t$ elements are arbitrary symbols from F. We also observe that $|C(\mathcal{P}, \mathcal{S}, n, t)| = |\mathcal{P}| \times |\mathcal{S}| \times q^{n-2t}$.

The following Lemma is immediate.

Lemma III.2. *$C(\mathcal{P}, \mathcal{S}, n, t)$ is t-overlap-free if and only if $\mathcal{P} \cap \mathcal{S} = \emptyset$.*

Suppose $\mathcal{P}$ and $\mathcal{S}$ are two sets of k-tuples from F^k. Suppose t is a positive integer such that $t < k$. Define $\mathcal{P}|_t$ to be the set of all t-prefixes of tuples from $\mathcal{P}$ and $\mathcal{S}|_t$ to be the set of all t-suffixes of tuples from $\mathcal{S}$. So

$$\mathcal{P}|_t = \{(p_1, \ldots, p_t) : \text{ there exists } (p_1, \ldots, p_t, p_{t+1}, \ldots, p_k) \in \mathcal{P}\},$$

and

$$\mathcal{S}|_t = \{(s_{k-t+1}, \ldots, s_k) : \text{ there exists } (s_1, \ldots, s_{k-t}, s_{k-t+1}, \ldots, s_k) \in \mathcal{S}\}.$$

The following is a straightforward extension of Lemma III.2.

Theorem III.3. *$C(\mathcal{P}, \mathcal{S}, n, t)$ is a $(1, k)$-overlap-free code if and only if $\mathcal{P}|_t \cap \mathcal{S}|_t = \emptyset$ for $1 \leq t \leq k$.*

A. The Doubling Construction

Theorem III.3 suggests a way to build up $(1, k)$-overlap-free codes inductively. We will refer to this process as the Doubling Construction. For the rest of the paper, we consider the binary case, where $q = 2$.

We take $F = \{0, 1\}$. Suppose we begin with $k = 1$. Without loss of generality, we can define $\mathcal{P} = \{0\}$ and $\mathcal{S} = \{1\}$. So $C(\mathcal{P}, \mathcal{S}, n, 1)$ would consist of all 2^{n-2} binary n-tuples that begin with a 0 and end with a 1.

Next, we consider $k = 2$. We consider extensions of the solution for $k = 1$, where we append a symbol to a tuple in $\mathcal{P}$ and we prepend a symbol to a tuple in $\mathcal{S}$:

$\mathcal{P}$	$\mathcal{S}$
00	01
01	11

We cannot include 01 in both $\mathcal{P}$ and $\mathcal{S}$. Without loss of generality, we include 01 in $\mathcal{P}$ but not in $\mathcal{S}$. So we obtain the following solution for $k = 2$:

$$\mathcal{P} = \{00, 01\} \text{ and } \mathcal{S} = \{11\}.$$

Thus $C(\mathcal{P}, \mathcal{S}, n, 2)$ would consist of

$$2 \times 2^{n-4}$$

binary n-tuples.

We can use a similar process to proceed from $k = 2$ to $k = 3$. We append a symbol to each tuple in $\mathcal{P}$ and we prepend a symbol to each tuple in $\mathcal{S}$:

$\mathcal{P}$	$\mathcal{S}$
000	011
001	111
010	
011	

Now the 3-tuple 011 is duplicated. We retain it in $\mathcal{S}$ and delete it from $\mathcal{P}$ (this will lead to the largest code, since $3 \times 2 > 4 \times 1$. We obtain the following solution for $k = 3$: $\mathcal{P} = \{000, 001, 010\}$ and $\mathcal{S} = \{011, 111\}$. Thus $C(\mathcal{P}, \mathcal{S}, n, 3)$ consists of

$$3 \times 2 \times 2^{n-6} = 6 \times 2^{n-6}$$

binary n-tuples.

Now we proceed from $k = 3$ to $k = 4$. We get the following:

$\mathcal{P}$	$\mathcal{S}$
0000	0011
0001	1011
0010	0111
0011	1111
0100	
0101	

The 4-tuple 0011 is duplicated. Again, we retain it in $\mathcal{S}$ and delete it from $\mathcal{P}$. We obtain the following solution for $k = 4$:

$$\mathcal{P} = \{0000, 0001, 0010, 0100, 0101\}$$

and

$$\mathcal{S} = \{0011, 1011, 0111, 1111\}.$$

Thus $C(\mathcal{P}, \mathcal{S}, n, 4)$ consists of

$$5 \times 4 \times 2^{n-8} = 20 \times 2^{n-8}$$

binary n-tuples.

When we proceed from $k = 4$ to $k = 5$, we obtain the following:

$\mathcal{P}$	$\mathcal{S}$
00000	00011
00001	10011
00010	01011
00011	11011
00100	00111
00101	10111
01000	01111
01001	11111
01010	
01011	

Now there are two duplicated 5-tuples. We will retain both 5-tuples in $\mathcal{S}$ in order to balance the sizes of $\mathcal{P}$ and $\mathcal{S}$. So we obtain the following solution for $k = 5$:

$$\mathcal{P} = \begin{Bmatrix} 00000, 00001, 00010, 00100, \\ 00101, 01000, 01001, 01010 \end{Bmatrix}$$

and

$$\mathcal{S} = \begin{Bmatrix} 00011, 10011, 01011, 11011, \\ 00111, 10111, 01111, 11111 \end{Bmatrix}.$$

Thus $C(\mathcal{P}, \mathcal{S}, n, 5)$ consists of

$$8 \times 8 \times 2^{n-10} = 2^{n-4}$$

binary n-tuples.

We can make a few observations as to what happens when we increase k by one in the Doubling Construction.

1) First, we double the size of $\mathcal{P}$ and $\mathcal{S}$ by appending 0 and 1 to every tuple in $\mathcal{P}$ and prepending 0 and 1 to every tuple in $\mathcal{S}$.
2) Then we look for duplicates in $\mathcal{P}$ and $\mathcal{S}$. Note that a duplicate occurs in the new $\mathcal{P}$ and $\mathcal{S}$ whenever there was a k-tuple in the old $\mathcal{P}$ whose suffix of size $k-1$ is identical to a prefix of size $k-1$ of a k-tuple in the old $\mathcal{S}$. For example, when $k = 4$, we see that $0001 \in \mathcal{P}$ and $0011 \in \mathcal{S}$. The suffix of size 3 of 0001, namely 001, is the same as the prefix of size 3 of 0011. Thus, when we append 1 to 0001 and we prepend 0 to 0011, we obtain the duplicate string 00011.
3) Finally, we eliminate one copy of each duplicate so as to balance the resulting sizes of $\mathcal{P}$ and $\mathcal{S}$ as much as possible.

TABLE I
RESULTS OBTAINED FROM THE DOUBLING CONSTRUCTION

k	$\lvert P_k\rvert$	$\lvert S_k\rvert$	$C(k,n) \geq$
2	2	1	$2\times 2^{n-4}$
3	3	2	$6\times 2^{n-6}$
4	5	4	$20\times 2^{n-8}$
5	8	8	$64\times 2^{n-10}$
6	15	14	$210\times 2^{n-12}$
7	26	27	$702\times 2^{n-14}$
8	50	50	$2500\times 2^{n-16}$
9	94	94	$8836\times 2^{n-18}$
10	180	179	$32220\times 2^{n-20}$
11	343	343	$117649\times 2^{n-22}$
12	659	659	$434281\times 2^{n-24}$
13	1267	1266	$1604022\times 2^{n-26}$
14	2444	2444	$5973136\times 2^{n-28}$
15	4726	4725	$22330350\times 2^{n-30}$
16	9157	9158	$83859806\times 2^{n-32}$
17	17779	17779	$316092841\times 2^{n-34}$
18	34575	34575	$1195430625\times 2^{n-36}$
19	67340	67339	$4534608260\times 2^{n-38}$
20	131323	131323	$17245730329\times 2^{n-40}$
21	256416	256416	$65749165056\times 2^{n-42}$
22	501208	501207	$251208958056\times 2^{n-44}$
23	980684	980684	$961741107856\times 2^{n-46}$

The results in Table I are obtained using the Doubling Construction. Note that here and elsewhere we denote the maximum size of a $(1, k)$-overlap-free code in $\{0, 1\}^n$ by $C(n, k)$.

IV. Optimal solutions—a graph-based approach

In this section, we discuss a graph-based approach that can (in principle) be used to prove that a solution is optimal. In practice, the method will only be feasible for small values of k. Again, we restrict our attention to the case $q = 2$ for convenience. Denote $F = \{0, 1\}$ and suppose k is a fixed positive integer.

We construct a bipartite graph G_k. The vertex set is $X \cup Y$, where $|X| = |Y| = 2^k$. We associate each vertex in X with a k-tuple from F^k, and similarly each vertex in Y corresponds to a k-tuple from F^k. The

vertices in X will be denoted by x_p, where $p \in F^k$, and the vertices in Y will be denoted by y_s, where $s \in F^k$. We will join vertices x_p and y_s by an edge if and only if a prefix of p is identical to a suffix of s. For example, the graph G_2 is depicted in Figure 2.

In general, the graph G_k records incompatible prefixes and suffixes. More precisely, if $x_p y_s$ is an edge of G_k, then there cannot exist two n-tuples in a $(1,k)$-overlap-free code where p is a k-prefix of an n-tuple and s is a k-suffix of a (not necessarily distinct) n-tuple.

The following lemma is immediate.

Lemma IV.1. *Suppose C is a $(1,k)$-overlap-free code. Let $\mathcal{P}$ denote all the k-prefixes of n-tuples in C and let $\mathcal{S}$ denote all the k-suffixes of n-tuples in C. Denote $X_C = \{x_p : p \in \mathcal{P}\}$ and $Y_C = \{y_s : s \in \mathcal{S}\}$. Then $X_C \cup Y_C$ is an independent set of vertices in G_k.*

Theorem IV.2. *Suppose $n \geq 2k$. Suppose that $X_C \cup Y_C$ is an independent set of vertices in G_k, where $X_C \subseteq X$ and $Y_C \subseteq Y$. Then there is a $(1,k)$-overlap-free code in F^n having size*

$$|X_C| \times |Y_C| \times 2^{n-2k}.$$

Proof. Suppose $X_C \cup Y_C$ is an independent set of vertices in G_k. Include all n-tuples of the form $p \parallel x \parallel s$ where $p \in \mathcal{P}$, $s \in \mathcal{S}$, and $x \in F^{n-2k}$. This is a $(1,k)$-overlap-free code having size $|X_C| \times |Y_C| \times 2^{n-2k}$. □

Theorem IV.3. *Suppose $n \geq 2k$. Suppose that $X_C \cup Y_C$ is an independent set of vertices in G_k, where $X_C \subseteq X$ and $Y_C \subseteq Y$, such that $|X_C| \times |Y_C|$ is maximized. Then the maximum size of any $(1,k)$-overlap-free code in F^n is exactly $|X_C| \times |Y_C| \times 2^{n-2k}$.*

Proof. Suppose C is a $(1,k)$-overlap-free code in F^n. Let $\mathcal{P}$ denote all the k-prefixes of n-tuples in C and let $\mathcal{S}$ denote all the k-suffixes of n-tuples in C. Lemma IV.1 asserts that $X_C \cup Y_C$ is an independent set of vertices in G_k. To maximize the size of C, we would include all n-tuples of the form $p \parallel x \parallel s$ where $p \in \mathcal{P}$, $s \in \mathcal{S}$, and $x \in F^{n-2k}$. From Theorem IV.2, this (optimal) code has size $|X_C| \times |Y_C| \times 2^{n-2k}$. □

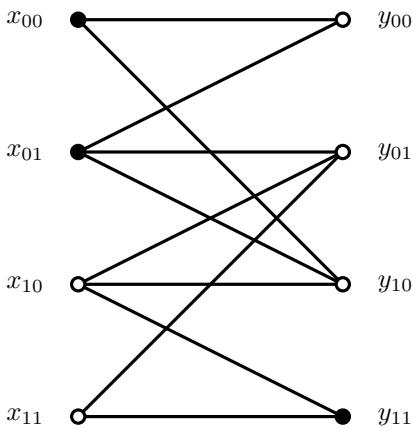

Fig. 2. The graph G_2, with nodes from an independent set highlighted

Example IV.1. Suppose $k = 2$. By examining the graph G_2 depicted in Figure 2, it is not hard to see that the only independent sets of size 4 are X and Y. Hence, the maximum value of $|X_C| \times |Y_C|$ is obtained when $|X_C| = 2$ and $|Y_C| = 1$ or when $|X_C| = 1$ and $|Y_C| = 2$. One optimal solution is $X_C = \{x_{00}, x_{01}\}$ and $Y_C = \{y_{11}\}$ (see the highlighted vertices in Figure 2). Therefore the maximum size of a $(1,2)$-overlap-free code in F^n is 2^{n-3}. In other words, the Doubling Construction is optimal for $k = 2$.

Remark IV.1. The proof of Theorem IV.3 uses the construction from Section III-A. In Section III-A, we inductively constructed independent sets $X_C \cup Y_C$ where we maximized $|X_C| \times |Y_C|$ at each step of the process. But it does not necessarily follow that the resulting values of $|X_C| \times |Y_C|$ are the maximum possible. In fact we will see situations where this is not the case.

The graph G_k has 2^{k+1} vertices. If we exhaustively search for an "optimal" independent set, this approach will quickly become infeasible as k increases. This can be done for a few small values of k, however. The approach we take is to identify some nice structure in optimal independent sets

for small k and then generalize the structure to larger values of k.

Suppose that $X_C \cup Y_C$ is an independent set of vertices in G_k, where $X_C \subseteq X$ and $Y_C \subseteq Y$. If $X_C \neq \emptyset$ and $Y_C \neq \emptyset$, then we say that $X_C \cup Y_C$ is a *non-trivial* independent set. Now we present an upper bound on the size of a non-trivial independent set in G_k.

Theorem IV.4. *A non-trivial independent set in G_k has size at most $2^{k-1}+1$.*

Proof. Define $X_i = \{x_p : p_1 = i\}$, for $i = 0, 1$. Also, define $Y_i = \{y_s : s_k = i\}$, for $i = 0, 1$. Thus X_i consists of all vertices in X corresponding to k-tuples beginning with i and Y_i consists of all vertices in Y corresponding to k-tuples ending with i. Suppose that $X_C \cup Y_C$ is a non-trivial independent set of vertices in G_k; hence $X_C \neq \emptyset$ and $Y_C \neq \emptyset$. Suppose without loss of generality that there is an $x_p \in X_0 \cap X_C$. Then $Y_C \cap Y_0 = \emptyset$ and hence $Y_C \subseteq Y_1$. Since $Y_C \neq \emptyset$, we have $X_C \cap X_1 = \emptyset$ and hence $X_C \subseteq X_0$.

Therefore, we can restrict our attention to the subgraph G' of G induced by the vertices in $X_0 \cup Y_1$. G' has 2^{k-1} vertices in each part of its partition. We show that G' contains a matching M of size $2^{k-1}-1$.

First, for the 2^{k-2} k-tuples p such that $p_1 = 0$ and $p_k = 1$, we match x_p with y_p. The remaining 2^{k-2} k-tuples p such that $x_p \in X_0$ have $p_1 = p_k = 0$ (call this set $\mathcal{P}'$), and the remaining 2^{k-2} k-tuples s such that $y_s \in Y_1$ have $s_1 = s_k = 1$ (call this set $\mathcal{S}'$). We ignore the all-0 k-tuple in $\mathcal{P}'$ and the all-1 k-tuple in $\mathcal{S}'$; there remain $2^{k-2}-1$ k-tuples in $\mathcal{P}'$ and $2^{k-2}-1$ k-tuples in $\mathcal{S}'$.

Any k-tuple in $\mathcal{P}'$ can be written uniquely in the form $p = 0 \parallel \mathbf{a} \parallel 1 \parallel \mathbf{b} \parallel 0$, where $\mathbf{a}$ is a (possibly empty) sequence of 0's and $\mathbf{b}$ is an arbitrary binary sequence. For each such k-tuple, we observe that there is an edge in G' from x_p to y_s, where $s = 1 \parallel \mathbf{b} \parallel 0 \parallel \mathbf{a} \parallel 1$, because p begins with $0 \parallel \mathbf{a} \parallel 1$ and s ends with $0 \parallel \mathbf{a} \parallel 1$. This creates $2^{k-2}-1$ additional matching edges.

We have constructed a matching of size $2^{k-1}-1$. Since there are two unmatched vertices in G', this immediately implies that the maximum size of a non-trivial independent set in G' (and hence in G_k) is at most $2^{k-1}+1$. □

TABLE II
EXACT VALUES OF $I(k)$ AND $C(k,n)$ FOR $k \leq 6$

k	$I(k)$	$C(k,n)$
1	2	2^{n-2}
2	3	$2\times 2^{n-4}$
3	5	$6\times 2^{n-6}$
4	9	$20\times 2^{n-8}$
5	16	$64\times 2^{n-10}$
6	30	$216\times 2^{n-12}$

Remark IV.2. *The bound proven in Theorem IV.4 is tight. This can be seen by observing that $\{00\cdots 0\} \cup Y_1$ is an independent set of size $2^{k-1}+1$.*

Corollary IV.5. *For $k \geq 2$, it holds that*

$$C(k,n) \leq (2^{k-2}+1) \times 2^{k-2} \times 2^{n-2k} = 2^{n-4} + 2^{n-k-2}.$$

Proof. This is a straightforward application of Theorems IV.3 and IV.4. When $k \geq 2$, the value $2^{k-1}+1$ is odd. Therefore we maximize the product $|X_C| \times |Y_C|$ by taking

$$|X_C| = 2^{k-2}+1$$

and

$$|Y_C| = 2^{k-2}$$

(or vice versa). □

We note that the upper bound proven in Corollary IV.5 is weaker than the bound proven in Theorem II.2.

A. Results for small values of k

Let $I(k)$ denote the maximum size of a non-trivial independent set in G_k. Table II summarizes the exact values of $I(k)$ and $C(k,n)$ for $k \leq 6$.

It is clear that $I(1) = 2$ and the Doubling Construction is optimal for $k = 1$. Corollary IV.5 shows that the Doubling Construction is optimal for $2 \leq k \leq 4$, and it also yields the exact values of $I(k)$ for these k.

For $k = 5$, an exhaustive search shows that $I(5) = 16$. From this, it follows that

$C(5,n) \leq 8\times 8\times 2^{n-10} = 64\times 2^{n-10}$. On the other hand, from the Doubling Construction, $C(5,n) \geq 2^{n-4} = 64 \times 2^{n-10}$, and so the Doubling Construction is again optimal. For $k = 6$, Theorem IV.4 shows that $I(6) \leq 33$ and Corollary IV.5 states that

$$C(n,6) \leq 2^{n-4} + 2^{n-8} = 272 \times 2^{n-12}.$$

However, this is not a tight bound, as we discuss below. The Doubling Construction yields a non-trivial independent set of size 29 with 14 vertices in one part and 15 vertices in the other part. Hence,

$$C(6,n) \geq 15 \times 14 \times 2^{n-12} = 210 \times 2^{n-12}.$$

But it turns out that there is a non-trivial independent set of size 30 with 12 vertices in one part and 18 vertices in the other part. This leads to a larger $(1,6)$-overlap-free code because $18\times 12 > 15\times 14$. The resulting lower bound is

$$C(6,n) \geq 18 \times 12 \times 2^{n-12} = 216 \times 2^{n-12}.$$

This solution is in fact optimal, as was verified by an exhaustive search. Here are the 6-tuples in the sets $\mathcal{P}$ and $\mathcal{S}$:

$\mathcal{P}$
000000, 000001, 000010, 000011,
000100, 000101, 000110, 000111,
001000, 001001, 001010, 001011

$\mathcal{S}$
001101, 001111, 010011, 010101,
010111, 011011, 011101, 011111,
100111, 101011, 101101, 101111,
110011, 110101, 110111, 111011,
111101, 111111

V. The m-minimum Construction

For $k \geq 7$, exhaustive searches appear to be infeasible. So we have tried various techniques to find useful lower bounds. We first describe the m-minimum Construction, which has enabled us to find some good solutions.

Construction V.1 (m-minimum Construction)**.** *Suppose k is a given positive integer. For $m = 1, 2, \ldots, 2^{k-1}$, we construct a code D_m as follows:*

- *Let $\mathcal{P}$ consist of the first m non-negative integers, represented as binary k-tuples (padded on the left with 0's if necessary, i.e., in big-endian form). Define $X_C = \{x_p : p \in \mathcal{P}\}$.*
- *Let Y_C consist of all vertices in Y that are adjacent to no vertices in X_C. Define $\mathcal{S} = \{s : y_s \in Y_C\}$.*
- *Output the sets $\mathcal{P}$ and $\mathcal{S}$ for the code D_m that maximizes the value of $|\mathcal{P}| \times |\mathcal{S}|$. The resulting $(1,k)$-overlap-free code will have size $|\mathcal{P}| \times |\mathcal{S}| \times 2^{n-2k}$.*

TABLE III
Results obtained from the m-minimum Construction

k	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{S}\rvert$	$C(k,n) \geq$
2	1	2	$2\times 2^{n-4}$
3	2	3	$6\times 2^{n-6}$
4	4	5	$20\times 2^{n-8}$
5	8	8	$64\times 2^{n-10}$
6	12	18	$216\times 2^{n-12}$
7	24	31	$744\times 2^{n-14}$
8	44	60	$2640\times 2^{n-16}$
9	64	149	$9536\times 2^{n-18}$
10	128	274	$35072\times 2^{n-20}$
11	256	504	$129024\times 2^{n-22}$
12	512	927	$474624\times 2^{n-24}$
13	960	1823	$1750080\times 2^{n-26}$
14	1792	3644	$6530048\times 2^{n-28}$

Table III summarizes results obtained from the m-minimum Construction. For $k \geq 6$, these are all improvements over the Doubling Construction. The optimal solution for $k = 6$ that we presented in Section IV-A is precisely the code D_{12} obtained from the m-minimum Construction. For $k = 7$, D_{24} is the code found by the m-minimum Construction; it has $|\mathcal{P}| = 24$ and $|\mathcal{S}| = 31$:

$\mathcal{P}$
0000000, 0000001, 0000010, 0000011,
0000100, 0000101, 0000110, 0000111,
0001000, 0001001, 0001010, 0001011,
0001100, 0001101, 0001110, 0001111,
0010000, 0010001, 0010010, 0010011,
0010100, 0010101, 0010110, 0010111

and

$\mathcal{S}$
0011011, 0011101, 0011111, 0100111,
0101011, 0101101, 0101111, 0110011,
0110101, 0110111, 0111011, 0111101,
0111111, 1001101, 1001111, 1010011,
1010101, 1010111, 1011011, 1011101,
1011111, 1100111, 1101011, 1101101,
1101111, 1110011, 1110101, 1110111,
1111011, 1111101, 1111111

This yields the lower bound

$$C(7, n) \geq 744 \times 2^{n-14}.$$

A. The Zero Block Construction

We now present the Zero Block Construction, which is closely related to the m-minimum Construction, and is inspired by the classical construction of non-overlapping codes due to Gilbert and Levenshtein [8], [9], [12] which we discuss in Section VI.

Construction V.2 (Zero Block Construction)**.** *Suppose k is a given positive integer. For $z = 1, \ldots, k-1$, we construct a code C_z from a certain X_C and Y_C as follows:*

- *Let $\mathcal{P}$ consist of the first 2^{k-z} non-negative integers, represented as binary k-tuples. Note that every $p \in \mathcal{P}$ begins with a block of (at least) z consecutive 0's. Define $X_C = \{x_p : p \in \mathcal{P}\}$.*
- *Let $\mathcal{S}$ consist of all binary k-tuples s ending with a 1 that do not contain z consecutive 0's. Define $Y_C = \{y_s : s \in \mathcal{S}\}$.*
- *Output the sets $\mathcal{P}$ and $\mathcal{S}$ for the code C_z that maximizes the value of $|\mathcal{P}| \times |\mathcal{S}|$. The resulting $(1, k)$-overlap-free code will have size*

$$|\mathcal{P}| \times |\mathcal{S}| \times 2^{n-2k} = |\mathcal{S}| \times 2^{n-k-z}.$$

Lemma V.3. *For X_C and Y_C as defined in Construction V-A, no vertex in Y_C is adjacent to any vertex in X_C.*

Proof. Suppose $x_p \in X_C$ and $y_s \in Y_C$. We consider two cases. If $\ell \leq z$, then the ℓ-prefix of p consists of ℓ 0's. However, s ends in a 1, so the ℓ-suffix of s is not the same as the ℓ-prefix of p. The second case is when $\ell \geq z + 1$. Here an ℓ-prefix of p begins with z 0's. However, no ℓ-suffix of s contains z consecutive 0's, so the ℓ-suffix of s is not the same as the ℓ-prefix of p. □

Thus, for any fixed value of z, the set Y_C defined in Construction V-A is a subset of the set that would be chosen in Construction V.1 (the m-minimum Construction). So the Zero Block Construction cannot improve on the m-minimum Construction; however, it is an explicit construction and potentially easier to analyze. We will consider a general bound that can be proven, as well as numerical computations for various values of k.

It remains to specify an appropriate value for z and to investigate the size of $\mathcal{S}$. It turns out that the number of binary ℓ-tuples s that do not contain n consecutive 0's is given by an *n-step Fibonacci number*. For a given value of $n \geq 2$, the *n-step Fibonacci sequence* is defined recursively as follows.

$$F_i^{(n)} = \begin{cases} 0 & \text{if } -n+2 \leq i \leq 0 \\ 1 & \text{if } i = 1 \\ \displaystyle\sum_{j=1}^{n} F_{i-j}^{(n)} & \text{if } i \geq 2. \end{cases} \tag{1}$$

That is, each term in this sequence is the sum of the n previous terms. It is easy to see that

$$F_i^{(n)} = 2^{i-2}$$

for $2 \leq i \leq n+1$. Also, it is easily verified that

$$F_{n+2}^{(n)} = 2^n - 1 \quad \text{and} \quad F_{n+3}^{(n)} = 2^{n+1} - 3.$$

For additional information about these sequences, see [7], [14].

The following result is well-known. We provide a proof for completeness.

Lemma V.4. *The number of binary ℓ-tuples that do not contain z consecutive 0's is $F_{\ell+2}^{(z)}$.*

Proof. Denote the number of binary ℓ-tuples that do not contain z consecutive 0's by $g(\ell, z)$. Then it is clear that $g(\ell, z) = 2^\ell$, if $1 \leq \ell < z$, and $g(z, z) = 2^z - 1$. Thus $g(\ell, z) = F_{\ell+2}^{(z)}$ if $1 \leq \ell \leq z$.

Next, consider $g(\ell, z)$ for some $\ell > z$. We partition the set of all binary ℓ-tuples that do

not contain z consecutive 0's into z disjoint subsets, denoted by $W_i, i = 1, \ldots, z$. For $1 \leq i \leq z$, the set W_i consists of all the ℓ-tuples that end with a 1 followed by $i-1$ 0's. It is clear that $|W_i| = g(\ell - i, k)$ for $1 \leq i \leq z$. Hence,

$$g(\ell, z) = \sum_{i=1}^{z} g(\ell - i, z)$$

whenever $\ell > z$. We can assume by induction that $g(\ell - i, z) = F^{(z)}_{\ell-i+2}$ for $1 \leq i \leq z$. So

$$g(\ell, z) = \sum_{i=1}^{z} F^{(z)}_{\ell-i+2} = F^{(z)}_{\ell+2},$$

from (1), as desired. □

The number of choices for $s \in \mathcal{S}$ is exactly $F^{(z)}_{k+1}$. Thus we have the following result.

Theorem V.5. *The size of the code obtained from the Zero Block Construction is*

$$\max\left\{F^{(z)}_{k+1} \times 2^{n-k-z} : 1 \leq z \leq k-1\right\}. \quad (2)$$

In order to obtain an explicit closed-form bound, it is probably more convenient to work with a simple lower bound on the values $F^{(z)}_{k+1}$.

Lemma V.6. *For $1 \leq z \leq k-1$, the following bound holds:*

$$F^{(z)}_{k+1} > (1 - k\,2^{-z})2^{k-1}.$$

Proof. Choose a binary word t of length $k-1$ randomly and uniformly and then append a 1. Let E_i be the 'bad' event that t contains 0^z, starting at position i. Note that t is of the desired form if and only if none of the events $E_1, E_2, \ldots, E_{k-1}$ occur. But the probability of E_i is at most 2^{-z} (indeed it is equal to this when $i \leq k - z$, and it is 0 otherwise). So the probability that one or more of the E_i's occurs is at most $(k-1)2^{-z}$. Hence the probability that none of the events $E_1, E_2, \ldots, E_{k-1}$ occur is at least $1 - (k-1)/2^z$. Since

$$1 - (k-1)2^{-z} > 1 - k\,2^{-z},$$

the stated bound follows. □

Now, using equation (2) from Theorem V.5, for a given value of z, we obtain a code of size at least

$$\begin{aligned}(1 - k\,2^{-z}) \times 2^{k-1} \times 2^{n-k-z} &= (1 - k\,2^{-z}) \times 2^{n-z-1} \\ &= (2^{-z}(1 - k\,2^{-z}))2^{n-1}.\end{aligned}$$

The function $f(z) = 2^{-z}(1 - k\,2^{-z})$ is maximized when $z = \log_2 2k$. Sadly, this is not always an integer. However, taking $z_0 = \lfloor \log_2 2k \rceil$ (i.e., rounding $\log_2 2k$ to the nearest integer), we have

$$\log_2 2k - 1/2 \leq z_0 \leq \log_2 2k + 1/2,$$

so $2^{z_0} \in [\sqrt{2}k, 2\sqrt{2}k]$. It then follows that

$$f(z_0) \geq \max\{f(\log_2 2k - \tfrac{1}{2}), f(\log_2 2k + \tfrac{1}{2})\}.$$

We have

$$f(\log_2 2k - \tfrac{1}{2}) = \frac{1}{\sqrt{2}k}\left(1 - \tfrac{1}{\sqrt{2}}\right) \approx \frac{1}{4.83k}$$

and

$$f(\log_2 2k + \tfrac{1}{2}) = \frac{1}{2\sqrt{2}k}\left(1 - \tfrac{1}{2\sqrt{2}}\right) \approx \frac{1}{4.38k}.$$

Hence, $f(z_0) \geq 1/(4.83k)$. Since the size of the resulting code is $f(z_0) \times 2^{n-1}$, we have the following theorem.

Theorem V.7. *There exists z such that*

$$|C_z| > (1/9.67k)2^n;$$

hence

$$C(k, n) \geq (1/9.67k)2^n.$$

We now incorporate two tweaks to improve Theorem V.7. The first is to define the events $E_1, E_2, \ldots$ used in the proof of Lemma V.6 a bit more carefully.

Lemma V.8. *For $1 \leq z \leq k-1$, the following bound holds:*

$$F^{(z)}_{k+1} \geq (1 - k\,2^{-z-1})2^{k-1}.$$

Proof. As before, choose a binary word t of length $k-1$ randomly and uniformly and then append a 1. We define E_1 as before. However, for $2 \leq i \leq k - z$, we now define E_i to be the event that there is a 1 in position $i-1$, followed by z 0's. It is not hard to see that if t contains z consecutive

zeroes, then one of the events $E_1, \ldots, E_{k-z}$ occurs. This is because the first occurrence of z consecutive 0's must immediately follow a 1, except when the first z positions are all 0's.

We have $\mathsf{Pr}[E_1] = 2^{-z}$ and $\mathsf{Pr}[E_i] = 2^{-z-1}$ for $2 \leq i \leq k-z$. Hence,

$$\begin{aligned}\mathsf{Pr}[E_1 \vee \cdots \vee E_{k-z}] &\leq 2^{-z} + (k-z-1)2^{-z-1} \\ &= 2^{-z-1}(2+k-z-1) \\ &\leq k\,2^{-z-1},\end{aligned}$$

since $z \geq 1$. Hence,

$$\mathsf{Pr}[\overline{E_1} \wedge \cdots \wedge \overline{E_{k-z}}] \geq 1 - k\,2^{-z-1}.$$

The stated bound follows. □

Using equation (2) from Theorem V.5, for a given value of z, we obtain a code of size at least

$$(2^{-z}(1 - k\,2^{-z-1}))2^{n-1}.$$

In order to maximize the size of the code, we choose z to maximize the function

$$g(z) = 2^{-z}(1 - k\,2^{-z-1}).$$

The maximum occurs when $z = \log_2 k$, which of course might not be an integer. We could consider an interval of length 1 whose centre is at $\log_2 k$ (similar to our argument above), but we can do slightly better by considering a different interval (this is our second tweak).

We choose z to be an integer in the interval $\left[\log_2 \frac{3k}{4}, \log_2 \frac{3k}{2}\right]$. Notice that this is again an interval of length 1. We obtain a slightly better bound because $g\left(\log_2 \frac{3k}{4}\right) = g\left(\log_2 \frac{3k}{2}\right)$. In fact,

$$g\left(\log_2 \frac{3k}{4}\right) = g\left(\log_2 \frac{3k}{2}\right) = \frac{4}{9k}.$$

We immediately obtain the following theorem, which improves Theorem V.7.

Theorem V.9. *There exists z such that*

$$|C_z| > (2/9k)2^n;$$

hence $C(k,n) \geq (2/9k)2^n$.

When k is a power of 2, the function $g(z)$ is maximized at the integral value $z = \log_2 k$. We obtain an improved result in this case.

Theorem V.10. *If $k = 2^i$ for a positive integer i, then*

$$|C_i| > (1/4k)2^n;$$

hence $C(k,n) \geq (1/4k)2^n$ for these values of k.

We note that the upper bound from Theorem II.2 is $C(k,n) \leq (1/2k)2^n$, which is roughly a factor of two greater than the lower bound from Theorem V.10 (when k is a power of two).

It is also possible to obtain asymptotic bounds which are stronger than the explicit general bounds discussed above. We pursue this now.

Let ℓ and z be integers, with $1 \leq z < \ell$. For an integer k with $0 \leq k < \ell$, define $\phi(\ell, k, z)$ to be the number of binary sequences of length ℓ and weight k such that any two cyclically consecutive ones are separated by at least z zeros. The following lemma gives bounds for $\phi(\ell, k, z)$ that are good when k and z are small compared to ℓ:

Lemma V.11. *Define ℓ, z, k and $\phi(\ell, k, z)$ as above. Then*

$$\binom{\ell}{k} - kz\binom{\ell}{k-1} \leq \phi(\ell, k, z) \leq \binom{\ell}{k}.$$

Proof. The lemma follows trivially in the case when $k \leq 1$, since $\phi(\ell, 0, z) = 1$ and $\phi(\ell, 1, z) = \ell$. So we may assume that $k \geq 2$.

The upper bound follows since $\binom{\ell}{k}$ is the number of weight k binary sequences of length ℓ. The lower bound follows if we can show that there are at most $kz\binom{\ell}{k-1}$ weight k binary sequences of length ℓ that have a zero run of length less than z. But all such sequences can be obtained (possibly more than once) in the following three-stage process. In Stage 1, choose a set of $k-1$ positions in the sequence to be equal to 1. In Stage 2, choose one of these $k-1$ positions, say position i. In Stage 3, choose a position $i + a \bmod \ell$ where $1 \leq a \leq z$ and set this position equal to 1; set the remaining positions to be zero. There are at most $\binom{\ell}{k-1}$ choices in the first stage,

there are $k-1$ choices in the second stage and at most z choices in the third stage. So

$$\begin{aligned}\phi(\ell,k,z) &\geq \tbinom{\ell}{k} - \tbinom{\ell}{k-1}(k-1)z \\ &\geq \tbinom{\ell}{k} - kz\tbinom{\ell}{k-1},\end{aligned}$$

as required. □

Corollary V.12. *Define ℓ, z, k and $\phi(\ell,k,z)$ as above. Then*

$$\frac{1}{k!} - \frac{2kz}{\ell} \leq \frac{\phi(\ell,k,z)}{\ell^k} \leq \frac{1}{k!}$$

Proof. The upper bound follows from the upper bound of Lemma V.11 and the inequality

$$\tbinom{\ell}{k} \leq \ell^k/k!.$$

For the lower bound, we use the lower bound of Lemma V.11 and the same bound on a binomial coefficient to see that

$$\begin{aligned}\frac{\phi(\ell,k,z)}{\ell^k} &\geq \frac{\binom{\ell}{k}}{\ell^k} - \frac{zk}{\ell} \times \frac{\binom{\ell}{k-1}}{\ell^{k-1}} \\ &\geq \frac{\binom{\ell}{k}}{\ell^k} - \frac{kz}{(k-1)!\ell} \\ &\geq \frac{\binom{\ell}{k}}{\ell^k} - \frac{kz}{\ell}.\end{aligned}$$

The lower bound now follows since

$$\begin{aligned}\tbinom{\ell}{k} &\geq \tfrac{(\ell-k)^k}{k!} \geq \tfrac{\ell^k - k^2\ell^{k-1}}{k!} \\ &\geq \tfrac{\ell^k}{k!} - k\ell^{k-1} \geq \tfrac{\ell^k}{k!} - kz\ell^{k-1}.\end{aligned}$$ □

Theorem V.13. *For a positive integer a, define $\ell = 2^a$ and $z = a-1$ (so $2^{z+1} = \ell$). Let ν_a be the number of binary sequences of length ℓ that do not contain any cyclic runs of z or more consecutive zeros. Then $\lim_{a\to\infty} \nu_a/2^\ell = 1/e$ (where e is the base of the natural logarithm).*

Proof. Let X_i be the set of sequences

$$s = (s_0, s_1, \ldots, s_\ell)$$

such that s_i=1 and s_{i+1}=s_{i+2}=$\cdots$=s_{i+z}=0. (Here we take subscripts modulo ℓ.)

Note that s has no cyclic runs of z or more zeros if and only if s is non-zero and $s \notin X_i$ for $i \in L = \{0,1,2,...,\ell-1\}$. Hence

$$\nu_a = \left|\overline{\bigcup_{i\in L} X_i}\right| - 1. \tag{3}$$

By the principle of inclusion-exclusion,

$$\left|\overline{\bigcup_{i\in L} X_i}\right| = \sum_{k=0}^{\ell-1}(-1)^k \sum_{\substack{I\subseteq L\\ |I|=k}} \left|\bigcap_{i\in I} X_i\right|, \tag{4}$$

where the partial sums involving k on the right hand side are successively upper and lower bounds for the left hand side (this follows from the Bonferroni inequalities).

For a subset $I \subseteq L$, let

$$t^I = (t_0^I, t_1^I, \ldots, t_{\ell-1}^I)$$

be the indicator binary sequence for I, so

$$t_i^I = \begin{cases} 1 & \text{if } i \in I, \\ 0 & \text{otherwise.}\end{cases}$$

When $k \geq 1$ we see that

$$\left|\bigcap_{i\in I} X_i\right| = 2^{\ell-(z+1)k},$$

when any two cyclically consecutive ones in t^I are separated by at least z zeros, and is 0 otherwise. So using the notation above Lemma V.11, we may simplify (4) as

$$\begin{aligned}\left|\overline{\bigcup_{i\in L} X_i}\right| &= \sum_{k=0}^{\ell-1}(-1)^k \phi(\ell,k,z) 2^{\ell-(z+1)k} \\ &= 2^\ell \left(\sum_{k=0}^{\ell-1}(-1)^k \tfrac{\phi(\ell,k,z)}{\ell^k}\right),\end{aligned} \tag{5}$$

since $2^{z+1} = \ell$ by our choice of ℓ and z.

Define $b = \lfloor \ell^{1/4} \rfloor$. We noted above that successive partial sums in the right hand side of (5) are upper and lower bounds for the left hand side, so truncating this sum after $b+1$ terms, we see that

$$\begin{aligned}&\left|\left|\overline{\bigcup_{i\in L} X_i}\right| - 2^\ell \left(\sum_{k=0}^{b}(-1)^k \frac{\phi(\ell,k,z)}{\ell^k}\right)\right| \\ &\qquad \leq 2^\ell \frac{\phi(\ell,b+1,z)}{\ell^{b+1}} \leq \frac{2^\ell}{(b+1)!},\end{aligned} \tag{6}$$

the final inequality following by the upper bound of Corollary V.12. Now, $z < b$ when a is sufficiently large, since $z = a-1$ and

$b \geq 2^{a/4} - 1$. So, using the bounds in Corollary V.12,

$$\begin{aligned}\left| 2^\ell \left(\sum_{k=0}^{b} (-1)^k \, \tfrac{\phi(\ell,k,z)}{\ell^k} \right) - 2^\ell \sum_{k=0}^{b} \tfrac{(-1)^k}{k!} \right| & \\ \leq 2^\ell \sum_{k=0}^{b} \tfrac{2kz}{\ell} \leq 2^\ell \sum_{k=1}^{b} \tfrac{2b^2}{\ell} & \\ = 2^\ell \tfrac{2b^3}{\ell} & \quad (7) \\ < 2^\ell \tfrac{2}{\ell^{1/4}} & \quad (8)\end{aligned}$$

whenever a is sufficiently large. But the usual power series expansion for $1/e$ shows that

$$2^\ell \left| \frac{1}{e} - \sum_{k=0}^{b} \frac{(-1)^k}{k!} \right| \leq \frac{2^\ell}{(b+1)!}. \quad (9)$$

Combining equations (3), (6), (7) and (9) we see that $\nu_a = 2^\ell(1/e + \epsilon)$, where

$$|\epsilon| \leq \frac{1}{2^\ell} + \frac{2}{(b+1)!} + \frac{2}{\ell^{1/4}}$$

whenever a is sufficiently large. In particular ϵ tends to zero as $a \to \infty$, and so the theorem follows. □

We remark that Schoeny *et al.* [15, Subsection V.B] prove a bound on the number of binary sequences with no (zero or one) runs of length $\log(2n)$, using a probabilistic construction. We wonder whether their bounds could be improved using techniques similar to those in the proof of Theorem V.13.

Corollary V.14. *For a positive integer a, define $\ell = 2^a$ and $z = a - 1$. Then*

$$\lim_{a\to\infty} \frac{F^{(z)}_{\ell+2}}{2^\ell} = \frac{1}{e}$$

(where e is the base of the natural logarithm).

Proof. Recall that ν_a is the number of binary sequences of length ℓ that do not contain any cyclic runs of z or more consecutive zeros. Also, $F^{(z)}_{\ell+2}$ is the number of binary sequences of length ℓ that do not contain any runs of z or more consecutive zeros. Clearly

$$\nu_a \leq F^{(z)}_{\ell+2}.$$

A sequence with no (non-cyclic) runs of z or more consecutive zeros, but which contains a cyclic run of z or more zeros, must either start or end with at least $\lceil a/2 \rceil$ zeros. Hence $F^{(z)}_{\ell+2} - \nu_a \leq 2(2^{\ell - \lceil a/2 \rceil})$. Hence

$$2^{-\ell} \left| \nu_a - F^{(z)}_{\ell+2} \right| \leq 2^{1-\lceil a/2 \rceil}.$$

Since the right hand side of this inequality tends to 0 as $a \to \infty$, the corollary follows by Theorem V.13. □

We can now prove the following asymptotic lower bound on $C(k, n)$.

Theorem V.15.

$$\limsup_{k\to\infty} \frac{C(k,n)}{2^n} \geq \frac{1}{ek}.$$

Proof. Take $k = \ell + 1 = 2^a + 1$ in equation (2) from Theorem V.5. Since $z = a - 1$, we now have $2^{z+1} = k - 1 = \ell$. Then Theorem V.5 says that

$$\begin{aligned}\limsup_{k\to\infty} & \frac{C(k,n)}{2^n} \\ & \geq \lim_{k\to\infty} \frac{F^{(z)}_{k+1}}{2^{k+z}} \\ & = \lim_{\ell\to\infty} \frac{F^{(z)}_{\ell+2}}{2^{\ell+1+z}} \\ & = \frac{1}{e\,2^{z+1}} \quad \text{from Corollary V.14} \\ & = \frac{1}{e(k-1)} \\ & > \frac{1}{ek}.\end{aligned}$$

□

Finally, it is perhaps also of interest to compute the exact size of the codes obtained from the the Zero Block Construction for "small" values of k. We use the formula (2) from Theorem V.5. For a fixed "small" value of k, we choose z to maximize $F^{(z)}_{k+1} \times 2^{-z}$. This is easily done by iterating through the possible values of z to see which one gives the largest result. The exact values $F^{(z)}_{k+1}$ are computed very quickly from the recurrence relation (1).

We present some data in Table IV comparing the Zero Block Construction to the m-minimum Construction. For the Zero Block Construction, we also include the optimal value of z. Table V provides a summary of the constructions and bounds in this paper.

TABLE IV
COMPARISON OF LOWER BOUNDS OBTAINED FROM THE m-MINIMUM CONSTRUCTION AND THE ZERO BLOCK CONSTRUCTION

k	m-minimum Construction	Zero Block Construction	optimal value of z
2	$2\times2^{n-4}$	$2\times2^{n-4}$	1
3	$6\times2^{n-6}$	$6\times2^{n-6}$	2
4	$20\times2^{n-8}$	$20\times2^{n-8}$	2
5	$64\times2^{n-10}$	$64\times2^{n-10}$	2
6	$216\times2^{n-12}$	$208\times2^{n-12}$	2
7	$744\times2^{n-14}$	$704\times2^{n-14}$	3
8	$2640\times2^{n-16}$	$2592\times2^{n-16}$	3
9	$9536\times2^{n-18}$	$9536\times2^{n-18}$	3
10	$35072\times2^{n-20}$	$35072\times2^{n-20}$	3
11	$129024\times2^{n-22}$	$129024\times2^{n-22}$	3
12	$474624\times2^{n-24}$	$474624\times2^{n-24}$	3
13	$1750080\times2^{n-26}$	$1745920\times2^{n-26}$	3
14	$6530048\times2^{n-28}$	$6422528\times2^{n-28}$	3

k	Doubling	m-minimum	Zero block	Upper bound
2	$\mathbf{2\times2^{n-4}}$	$\mathbf{2\times2^{n-4}}$	$\mathbf{2\times2^{n-4}}$	$\mathbf{2\times2^{n-4}}$
3	$\mathbf{6\times2^{n-6}}$	$\mathbf{6\times2^{n-6}}$	$\mathbf{6\times2^{n-6}}$	$\mathbf{6\times2^{n-6}}$
4	$\mathbf{20\times2^{n-8}}$	$\mathbf{20\times2^{n-8}}$	$\mathbf{20\times2^{n-8}}$	$\mathbf{20\times2^{n-8}}$
5	$\mathbf{64\times2^{n-10}}$	$\mathbf{64\times2^{n-10}}$	$\mathbf{64\times2^{n-10}}$	$\mathbf{64\times2^{n-10}}$
6	$210\times2^{n-12}$	$\mathbf{216\times2^{n-12}}$	$208\times2^{n-12}$	$\mathbf{216\times2^{n-12}}$
7	$702\times2^{n-14}$	$\mathbf{744\times2^{n-14}}$	$704\times2^{n-14}$	$1170.3\times2^{n-14}$
8	$2500\times2^{n-16}$	$\mathbf{2640\times2^{n-16}}$	$2592\times2^{n-16}$	$4096\times2^{n-16}$
9	$8836\times2^{n-18}$	$\mathbf{9536\times2^{n-18}}$	$\mathbf{9536\times2^{n-18}}$	$14563.6\times2^{n-18}$
10	$32220\times2^{n-20}$	$\mathbf{35072\times2^{n-20}}$	$\mathbf{35072\times2^{n-20}}$	$52428.8\times2^{n-20}$
11	$117649\times2^{n-22}$	$\mathbf{129024\times2^{n-22}}$	$\mathbf{129024\times2^{n-22}}$	$190650.2\times2^{n-22}$
12	$434281\times2^{n-24}$	$\mathbf{474624\times2^{n-24}}$	$\mathbf{474624\times2^{n-24}}$	$699050.7\times2^{n-24}$
13	$1604022\times2^{n-26}$	$\mathbf{1750080\times2^{n-26}}$	$1745920\times2^{n-26}$	$2581110.2\times2^{n-26}$
14	$5973136\times2^{n-28}$	$\mathbf{6530048\times2^{n-28}}$	$6422528\times2^{n-28}$	$9586080.6\times2^{n-28}$

TABLE V
A SUMMARY OF THE LARGEST $(1,k)$-OVERLAP FREE CODES OBTAINED BY OUR THREE CONSTRUCTIONS. BOLD FONT INDICATES THE BEST OF THE THREE CONSTRUCTIONS, OR A TIGHT UPPER BOUND.

It is interesting to observe that the Zero Block Construction performs almost as well as the m-minimum Construction in all cases, and it gives the same result in many cases. However, the computations of the bounds for the Zero Block Construction are amazingly fast. For example, it is almost instantaneous to compute the lower bound

$$5745596237141382$$
$$78560878649953571642432 6$$
$$792561835200479232\times2^{n-200}$$

VI. NON-OVERLAPPING CODES

We can apply the techniques of Section V-A to the construction of "classic" non-overlapping codes. Again, we restrict our attention to the binary case. The following construction is due to Gilbert and Levenshtein; it has been re-discovered several times, and is used in many applications. See [5], [8], [9], [12], [13].

Construction VI.1 (Gilbert–Levenshtein Construction)**.** *Suppose n is a given positive integer. For $z=1,\ldots,k-1$, we construct a code L_z as follows:*

- *each codeword $c=(c_1,\ldots,c_n)\in L_z$ begins with a block of z consecutive 0's,*
- *$c_{z+1}=c_n=1$, and*
- *the sequence $(c_{z+1},\ldots,c_{n-1})$ does not contain z consecutive 0's.*

It is clear that $|L_z|$ equals the number of

binary sequences of length $n-z-2$ that do not contain z consecutive 0's. Hence, from Lemma V.4, we have the following.

Lemma VI.2. $|L_z| = F^{(z)}_{n-z}$.

Of course we would choose z to maximize $|L_z|$. Let $S(n)$ denote the size of the code obtained from the Gilbert–Levenshtein Construction. The following result is immediate.

Theorem VI.3.

$$S(n) = \max\left\{F^{(z)}_{n-z} : 1 \le z \le n-1\right\}. \quad (10)$$

We note that the connection between the Gilbert–Levenshtein Construction and the n-step Fibonacci numbers was pointed out by Chee *et al.* [5]. In fact, the entries in the third column of [5, Table 1] are computed using the formula (10).

We can use the techniques developed in Section V-A to give an explicit, non-asymptotic lower bound on $S(n)$.

Lemma VI.4.

$$F^{(z)}_{n-z} > (1 - n\,2^{-z-1})2^{n-z-2}.$$

Proof. If we take $k = n-z-1$ in Lemma V.8, we obtain

$$F^{(z)}_{n-z} \ge (1 - (n-z-1)\,2^{-z-1})2^{n-z-2}.$$

Clearly,

$$1 - (n-z-1)\,2^{-z-1} > 1 - n\,2^{-z-1},$$

so the stated bound follows. □

We now choose z to maximize the function $h(z) = (1 - n\,2^{-z-1})2^{-z-2}$. The maximum occurs when $z = \log_2 k$, which of course might not be an integer. Choose z to be an integer in the interval $\left[\log_2 \frac{3n}{4}, \log_2 \frac{3n}{2}\right]$. Then we have

$$h\left(\log_2 \frac{3n}{4}\right) = h\left(\log_2 \frac{3n}{2}\right) = \frac{1}{9n},$$

and we obtain the following theorem.

Theorem VI.5. $S(n) > (1/9n)2^n$.

When n is a power of 2, the maximum value of $h(z)$ occurs when $z = \log_2 n$, and so we do slightly better:

Theorem VI.6. *If n is a power of two, then $S(n) > (1/8n)2^n$.*

These bounds improve previous explicit bounds. In Bilotta, Pergola and Pinzani [3], an explicit construction based on Dyck paths was given. However, it was observed by Chee *et al.* [5] that this construction does not yield a lower bound of the form $S(n) > (c/n)2^n$ for any constant $c > 0$. Also, Blackburn [4] proved that $S(n) > (3/64n)2^n$; our lower bound from Theorem VI.5 is stronger.

As far as asymptotic bounds are concerned, Levenshtein [9] proved that

$$\limsup_{n\to\infty} S(n) \ge (1/2en)2^n \approx (1/5.436n)2^n.$$

Levenshtein's asymptotic bound also follows easily from Corollary V.14 and Theorem VI.3, as we now demonstrate.

Theorem VI.7. $\limsup_{n\to\infty} S(n) \ge (1/2en)2^n$.

Proof. We prove that

$$\limsup_{n\to\infty} \frac{2nS(n)}{2^n} \ge \frac{1}{e}.$$

From Theorem VI.3, we have

$$\frac{2nS(n)}{2^n} \ge \frac{2nF^{(z)}_{n-z}}{2^n}$$

for any z such that $1 \le z \le n-1$. Let $\ell = 2^a$, $z = a-1$ and $n = \ell + a + 1$ for a positive integer a. Then $n - z = \ell + 2$. For these values of n and z, we compute

$$\begin{aligned}\frac{2nF^{(z)}_{n-z}}{2^n} &= \frac{2(\ell+a+1)}{2^{a+1}} \times \frac{F^{(a-1)}_{\ell+2}}{2^\ell} \\ &= \frac{\ell+a+1}{\ell} \times \frac{F^{(a-1)}_{\ell+2}}{2^\ell}.\end{aligned}$$

It is clear that $(\ell+a+1)/\ell$ approaches 1 as $a \to \infty$, because $\ell = 2^a$. Also, from Corollary V.14, $F^{(a-1)}_{\ell+2}/2^\ell$ approaches $1/e$ as $a \to \infty$. The desired result follows. □

VII. Discussion and Summary

In this paper, we have mainly concentrated on $(1,k)$-overlap-free codes over a binary alphabet. Our constructions and bounds are actually quite close. There are many possible avenues for future research,

including studying variable-length analogs, studying codes over non-binary alphabets, or investigating codes with other forbidden overlaps. One direction that might be fruitful for applications is the investigation of codes which are simultaneously $(1, k)$-overlap-free and $(n-k, n-1)$-overlap-free, where $k < \frac{n}{2}$.

The Zero Block Construction is inspired by a classical construction of non-overlapping codes due to Gilbert and Levenshtein. It is surprising to us that the m-minimum Construction can sometimes yield better codes. Here is one specific question relating to these two constructions from Section V: Do the m-minimum Construction and Zero Block Construction give the same bound for infinitely many values of n?

Finally, we note that the constructions in Section VI are most effective when n is close to a power of two. We ask if there are constructions that are asymptotically better when n is not of this form, for example when $n = \lfloor 2^{(a+1)/2} \rfloor$ as $a \to \infty$?

References

[1] E. Barcucci, A. Bernini, S. Bilotta, and R. Pinzani, "A 2D non-overlapping code over a q-ary alphabet," *Cryptogr. Commun.*, vol. 10, no. 4, pp. 667–683, 2018.

[2] S. Bilotta, "Variable-length non-overlapping codes," *IEEE Trans. Inform. Theory*, vol. 63, no. 10, pp. 6530–6537, 2017.

[3] S. Bilotta, E. Pergola, and R. Pinzani, "A new approach to cross-bifix-free sets," *IEEE Trans. Inform. Theory*, vol. 58, no. 6, pp. 4058–4063, 2012.

[4] S. R. Blackburn, "Non-overlapping codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 9, pp. 4890–4894, 2015.

[5] Y. M. Chee, H. M. Kiah, P. Purkayastha, and C. Wang, "Cross-bifix-free codes within a constant factor of optimality," *IEEE Trans. Inform. Theory*, vol. 59, no. 7, pp. 4668–4674, 2013.

[6] L. Cheng, T. G. Swart, H. C. Ferreira, and K. A. S. Abdel-Ghaffar, "Codes for correcting three or more adjacent deletions or insertions," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, June 2014, pp. 1246–1250.

[7] G. P. B. Dresden and Z. Du, "A simplified Binet formula for k-generalized Fibonacci numbers," *J. Integer Seq.*, vol. 17, no. 4, Article 14.4.7, 2014.

[8] E. N. Gilbert, "Synchronization of binary messages," *IRE Trans. Inform. Theory*, pp. 470–477, 1960.

[9] V. I. Levenšteĭn, "Decoding automata which are invariant with respect to their initial state," *Probl. Cybern.*, vol. 12, pp. 125–136, 1964.

[10] ——, "Binary codes capable of correcting deletions, insertions and reversals," *Dokl. Akad. Nauk Tadzhik. SSR*, vol. 163, pp. 845–848, 1965, (in Russian).

[11] ——, "Asymptotically optimum binary codes with correction for losses of one or two adjacent bits," *Syst. Theory Res.*, vol. 19, pp. 298–304, 1970.

[12] ——, "Maximal number of words in codes without overlap," *Probl. Inf. Transm.*, vol. 6, pp. 355–357, 1970.

[13] M. Levy and E. Yaakobi, "Mutually uncorrelated codes for DNA storage," *IEEE Trans. Inform. Theory*, vol. 65, no. 6, pp. 3671–3691, 2019.

[14] T. Noe, T. Piezas III, and E. Weisstein, "Fibonacci n-step number," from *MathWorld*–A Wolfram Web Resource. [Online]. Available: https://mathworld.wolfram.com/Fibonaccin-StepNumber.html

[15] C. Schoeny, A. Wachter-Zeh, R. Gabrys, and E. Yaakobi, "Codes correcting a burst of deletions or insertions," *IEEE Trans. Inform. Theory*, vol. 63, no. 4, pp. 1971–1985, 2017.

[16] S. M. H. Tabatabaei Yazdi, H. M. Kiah, Gabrys, Ryan, and O. Milenkovic, "Mutually uncorrelated primers for DNA-based data storage," *IEEE Trans. Inform. Theory*, vol. 64, no. 9, pp. 6283–6296, 2018.

[17] G. Wang and Q. Wang, "q-ary non-overlapping codes: a generating function approach," *IEEE Trans. Inform. Theory*, vol. 68, no. 8, pp. 5154–5164, 2022.

Simon R. Blackburn (M'12, SM'19) was born in Beverley, Yorkshire, England in 1968. He received a BSc in Mathematics from Bristol in 1989, and a DPhil in Mathematics from Oxford in 1992.

He has worked in the Mathematics Department at Royal Holloway University of London since 1992, and is currently a Professor of Pure Mathematics. His research interests include algebra, combinatorics and associated applications in cryptography and communication theory.

Navid Nasr Esfahani (M'18) received the B.Sc. degree from the Isfahan University of Technology, Isfahan, Iran, in 2011, the M.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 2014, and the Ph.D. degree from the Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada in 2021. He then continued his research as a Post-Doctoral Fellow at the University of Waterloo. In 2023, he joined the Department of Computer Science at the Memorial University of Newfoundland, Canada, as an Assistant Professor.

His research interests include cryptography, information theory, information theoretic security, privacy, and combinatorics.

Donald L. Kreher (born in Albany, New York, U.S.A. in 1955) obtained a joint computer science and mathematics Ph.D. from the University of Nebraska in 1984 and held academic positions at Rochester Institute of Technology from 1984 to 1989, the University of Wyoming from 1989 to 1991, and Michigan Technological University from 1991 to 2020 when he retired as an emeritus professor.

In 1995, Professor Kreher was awarded the Marshall Hall Medal, awarded by the Institute of Combinatorics and its Applications.

His research interests include computational and algebraic methods for determining the structure and existence of combinatorial configurations, such as designs, graphs, error-correcting codes, cryptographic systems and extremal set systems.

Douglas R. Stinson (born in 1956 in Guelph, Ontario) is a Canadian mathematician and cryptographer, currently Professor Emeritus at the University of Waterloo. Stinson received his B.Math from the University of Waterloo in 1978, his M.Sc. from Ohio State University in 1980, and his Ph.D. from the University of Waterloo in 1981. He was at the University of Manitoba from 1981 to 1989 and the University of Nebraska-Lincoln from 1990 to 1998. Since 1998 he has been at the University of Waterloo, retiring in 2019. Professor Stinson was awarded the 1994 Hall Medal and the 2022 Stanton Medal by the Institute of Combinatorics and its Applications. In 2011, he was named as a Fellow of the Royal Society of Canada. His research interests include combinatorics, cryptography, algorithms and information security.