Capacity-Achieving Codes That Mitigate Intercell Interference and Charge Leakage in Flash Memories

Yeow Meng Chee, *Senior Member, IEEE*, Johan Chrisnata, Han Mao Kiah, San Ling, Tuan Thanh Nguyen, and Van Khu Vu

***Abstract*— We investigate constant-composition constrained codes for the mitigation of intercell interference for multilevel cell flash memories with a dynamic threshold scheme. The first explicit formula for the maximum size of a q-ary $\mathcal{F}$-avoiding code with a given composition and certain families of substrings $\mathcal{F}$ is presented. In addition, we provide methods to determine the asymptotic rate for $\mathcal{F}$-avoiding codes with any composition ratio and to find the optimal composition ratio that maximizes the asymptotic rate. We also give the first efficient encoder/decoder for these q-ary constant-composition codes achieving the channel capacity, for all q values.**

***Index Terms*— Constrained codes, constant-composition codes, flash memories, intercell interference.**

I. Introduction

FLASH memories have become a popular nonvolatile storage of information owing to its advantage of high speed, low noise, low power consumption, compact form factor, and good physical reliability. The basic information storage element of a flash memory is called a *cell*, which consists of a floating-gate (FG) transistor. The amount of charge in an FG transistor is discretized into *charge levels* as a way to store information. The operation of injecting charge into an FG transistor to a desired level is called *programming*. In a single level cell (SLC) flash memory, each cell has two charge levels (corresponding to a charged or uncharged FG transistor), and hence can store one bit per cell. More recent multi-level cell (MLC[1]) flash memories have cells with $q > 2$ charge levels, with the ability to store $\log_2 q$ bits per cell. More specifically, we use qLC to refer to cells with q charge levels. The cells of a flash memory are further organized into *blocks*, each containing a constant number of cells. Hence, a block in a qLC flash memory stores a q-ary word (where symbol i is used to represent charge level i of a cell), and such a flash memory stores a collection of q-ary words.

MLC technology increases the storage density of flash memories. However, very precise programming is needed. There are two main challenges to reliable programming and storage:

(i) Intercell interference (ICI) caused by parasitic capacitance coupling between adjacent cells [3]. Such interference occurs when there are three adjacent cells c_1, c_2, c_3 and we want to increase the charge levels of the left-most and right-most cells, c_1 and c_3, while maintaining the charge level of the center cell c_2. Parasitic capacitance coupling can cause the charge level of the (victim) cell c_2 to increase when we increase the charge levels of its neighbouring cells c_1 and c_3. In the following example, the victim cell 0 increases unintentionally, therefore, the voltage misread the code as 01000111 while the original code was 01000101.

(ii) Charge leakage [4]. The charge in an FG transistor leaks away over time as a result of trap-assisted tunneling effect. This results in charge levels of cells drifting downwards over time, giving rise to asymmetric errors.

Different techniques have been explored to mitigate ICI. Physical methods, such as using low-κ dielectric material to reduce capacitative coupling [5], and programming methods such as proportional programming [6], have been investigated but one of the most effective approaches is the constrained coding method of Berman and Birk [7]–[9]. In their approach, certain words are forbidden to be stored, since the programming required to store such a word is highly unreliable, owing to ICI. For example, the quaternary word of length eight $(1, 2, 1, 3, 0, 3, 2, 0)$ should be avoided as the charge level of the fifth cell can be increased unintentionally during the programming of the fourth and sixth cells. More generally, Taranalli *et al.* [10] performed a comprehensive series of program/erase (P/E) cycling experiments to quantify ICI effects, and concluded that the words permitted for storage on a qLC flash memory should avoid containing any $(q-1, \sigma, q-1)$ as a substring, where $\sigma \in \{0, 1, \ldots, q-2\}$. Other studies of constrained codes mitigating ICI in flash memories include [11]–[13].

To mitigate the effect of charge leakage, a straightforward way is to adopt asymmetric error-correcting codes [14], [15]. Dynamic threshold techniques, introduced by Zhou *et al.* [16] for SLC and extended to MLC by Sala *et al.* [17], have been

Manuscript received March 14, 2018; revised July 25, 2018; accepted November 11, 2018. Date of publication November 3, 2018; date of current version May 20, 2019. Y. M. Chee, H. M. Kiah, and S. Ling were supported by the Singapore Ministry of Education under Grant MOE2015-T2-2-086. This paper was presented in part at the 2016 Proceedings of the IEEE International Symposium on Information Theory [1], [2].

The authors are with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore (e-mail: ymchee@ntu.edu.sg; johanchr001@e.ntu.edu.sg; hmkiah@ntu.edu.sg; lingsan@ntu.edu.sg; nguyentu001@ntu.edu.sg; vankhu001@ntu.edu.sg).

Communicated by A. Jiang, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2018.2884210

[1]MLC is commonly used to refer to the specific technology that allows four charge levels per cell. For lack of a better notion, we extend the use of "MLC" here to refer to technology allowing three or more charge levels per cell.

shown to be not only highly effective against asymmetric errors caused by charge leakage but also offer some protection against over-programming. In error-correcting schemes with dynamic threshold, the codes have constant composition. In particular, the case when the codes have both constant composition and balanced (where the number of times a symbol appears in a codeword is as close as possible) was studied in detail by Zhou *et al.* [16] and Sala *et al.* [17]

Recent approaches have combined constrained coding and dynamic threshold techniques [18], [19]. Before we give an account of these results, we introduce some necessary notations and terminologies.

A. Notations

Let $\Sigma \triangleq \{0, 1, \ldots, q-1\}$ be an alphabet of $q \geqslant 2$ symbols. A q-ary word of length n over Σ is an element $\mathsf{u} \in \Sigma^n$. The ith coordinate of u is denoted by u_i, so that $\mathsf{u} = (\mathsf{u}_1, \mathsf{u}_2, \ldots, \mathsf{u}_n)$. There is a natural correspondence between the data represented by the charge levels of a block of n cells in a qLC flash memory and a q-ary word $\mathsf{u} \in \Sigma^n$: u_i is the charge level of the ith cell in the block.

For a positive integer n, a *composition of n into q parts* is a q-tuple $\overline{w} = [w_0, w_1, \ldots, w_{q-1}]$ of nonnegative integers such that $\sum_{i=0}^{q-1} w_i = n$. A q-ary word $\mathsf{u} \in \Sigma^n$ is said to have *composition* $\overline{w}$ if the frequency of symbol i in u is w_i. The *weight* of a word $\mathsf{u} \in \Sigma^n$ with composition $\overline{w}$ is $w = \sum_{i=1}^{q-1} w_i$. A word $\mathsf{u} \in \Sigma^n$ is said to be *balanced* if it has composition $\overline{w}$ such that $w_i \in \{\lfloor n/q \rfloor, \lceil n/q \rceil\}$ for all $i \in \Sigma$.

A q-ary code of length n is a subset $\mathcal{C} \subseteq \Sigma^n$. Elements of $\mathcal{C}$ are called *codewords*. The size of $\mathcal{C}$ is the number of codewords in $\mathcal{C}$. A code $\mathcal{C}$ is said to have

(i) *constant weight* w, if each codeword in $\mathcal{C}$ has weight w;
(ii) *constant composition* $\overline{w}$, if each codeword in $\mathcal{C}$ has composition $\overline{w}$.

A code is *balanced* if each of its codewords is balanced.

A *substring* of a word u is a word $(\mathsf{u}_{i+1}, \mathsf{u}_{i+2}, \ldots, \mathsf{u}_{i+\ell}) \in \Sigma^\ell$, where $i \geqslant 0$ and $i + \ell \leqslant n$. Let $\mathcal{F}$ be a set of words over Σ. A word u is said to *avoid* $\mathcal{F}$ or *$\mathcal{F}$-avoiding* if no words in $\mathcal{F}$ is a substring of u. A code $\mathcal{C}$ is said to *avoid* $\mathcal{F}$ if every codeword in $\mathcal{C}$ avoids $\mathcal{F}$. For a fixed length n, we denote the set of all q-ary words that avoid $\mathcal{F}$ by $\mathcal{A}(n; \mathcal{F})$.

The *rate* of a code $\mathcal{C}$ is $R \triangleq \log_2 |\mathcal{C}|/n$. Intuitively, it measures the number of information bits stored in each multilevel cell. Henceforth, we adopt the notation log to mean logarithm base two.

Let $\mathcal{F}$ be a set of words over Σ. An *$\mathcal{F}$-avoiding channel* is a channel whose input codewords avoid $\mathcal{F}$. The *capacity* of an $\mathcal{F}$-avoiding channel or the *capacity of the $\mathcal{F}$-constraint* is given by the value

$$C(\mathcal{F}) \triangleq \limsup_{n\to\infty} \frac{\log |\mathcal{A}(n; \mathcal{F})|}{n}.$$

Recent approaches combine constrained coding and dynamic threshold techniques, leading to the consideration of codes that both avoid $\mathcal{F}$ and have constant composition. We denote an $\mathcal{F}$-avoiding code of length n with constant composition $\overline{w}$ by $\mathcal{C}(n; \overline{w}, \mathcal{F})$. The maximum size of a $\mathcal{C}(n; \overline{w}, \mathcal{F})$, that is, the size of the set of all $\mathcal{F}$-avoiding words of composition $\overline{w}$, is denoted by $A(n; \overline{w}, \mathcal{F})$ and the set is denoted by $\mathcal{A}(n; \overline{w}, \mathcal{F})$.

Let $\overline{\rho} = [\rho_0, \rho_1, \ldots, \rho_{q-1}]$ be a real-valued vector such that $\sum_{i=0}^{q-1} \rho_i = 1$. Let $(\overline{w}(n))_{n=1}^{\infty}$ be a sequence of compositions of n such that $w_i(n) = \lfloor \rho_i \cdot n \rfloor$ for all $i \in \Sigma \setminus \{0\}$ and $w_0(n) = n - \sum_{i=1}^{q-1} w_i(n)$. We define the *asymptotic information rate of* $(\overline{\rho}, \mathcal{F})$ to be

$$R(\overline{\rho}, \mathcal{F}) \triangleq \limsup_{n\to\infty} \frac{\log A(n; \overline{w}(n), \mathcal{F})}{n},$$

and refer to $\overline{\rho}$ as the *composition ratio*.

Notice for the family of balanced codes, the ratio $\overline{\rho} = [1/q, 1/q, \ldots, 1/q]$. In this case, we write $R([1/q, 1/q, \ldots, 1/q], \mathcal{F})$ simply as $R_{\mathrm{bal}}(\mathcal{F})$.

B. Previous Work

As mentioned earlier, a number of proposals for the avoidance set $\mathcal{F}$ have been put forth to mitigate the effects of ICI. In view of these proposals, we consider the following set of words over Σ. Fix $0 \leqslant a < b \leqslant q-1$ and let $\mathcal{I}(a, b) \triangleq \{(c_1, c_2, c_3) : 0 \leqslant c_2 \leqslant a \text{ and } b \leqslant c_1, c_3 \leqslant q-1\}$.

Taranalli *et al.* [10] proposed the avoidance set $\mathcal{I}_1(q) \triangleq \mathcal{I}(q-2, q-1)$, while Qin *et al.* [18] proposed the set $\mathcal{I}_2(q) \triangleq \mathcal{I}(0, q-1)$. We note that $\mathcal{I}_1(2) = \mathcal{I}_2(2)$. Therefore, in this work, we always consider $\mathcal{I}_1(q)$ for $q \geq 2$ and $\mathcal{I}_2(q)$ for $q \geq 3$.

Example 1: $\mathcal{I}_1(2) = \mathcal{I}_2(2) = \{(1, 0, 1)\}$. $\mathcal{I}_1(4) = \{(3, 0, 3), (3, 1, 3), (3, 2, 3)\}$, while $\mathcal{I}_2(4) = \{(3, 0, 3)\}$.

In general, the capacity of the $\mathcal{F}$-constraint may be computed using the standard techniques detailed in [20]. For the purpose of mitigating ICI, the following results are known.

Proposition 1 [19], [21]:

(i) $C(\mathcal{I}_1(2)) = C(\mathcal{I}_2(2)) = \log_2 \lambda \approx 0.81137$*, where λ is the unique real root to the polynomial* $X^3 - 2X^2 + X - 1$.
(ii) $C(\mathcal{I}_1(4)) \approx 1.9374$.

For completeness, we state the following proposition without proof.

Selected capacities are computed and provided in Table I. In particular, the capacity values $C(\mathcal{I}_1(q))$ are presented in the fifth column of Table I for $2 \leq q \leq 8$ and the capacity values $C(\mathcal{I}_2(q))$ are presented in the ninth column of Table I for $3 \leq q \leq 8$.

Proposition 2: Fix q and $0 \leqslant a < b \leqslant q-1$. We have $C(\mathcal{I}(a, b)) = \log_2 \lambda_{a,b}$, where $\lambda_{a,b}$ is the maximum real root of the polynomial $X^3 - qX^2 + (q-b)(a+1)X - (q-b)(a+1)b$.

The asymptotic rate of balanced $\mathcal{I}_1(2)$-avoiding codes were investigated by Qin *et al.* and in the same paper, they documented the asymptotic rate of balanced $\mathcal{I}_2(3)$-avoiding codes.

Proposition 3 (Qin et al. *[18]):* $R_{\mathrm{bal}}(\mathcal{I}_1(2)) = (\log 3)/2 \approx 0.79428$ *and* $R_{\mathrm{bal}}(\mathcal{I}_2(3)) \approx 1.52576$.

Observe that the balanced $\mathcal{I}_1(2)$-avoiding codes have rates that fall short of over 2% of the capacity of the $\mathcal{I}_1(2)$-constraint. We state our question of interest: is there a ratio $\overline{\rho}$ where the asymptotic rate of $\mathcal{I}_1(2)$-avoiding codes with composition ratio $\overline{\rho}$ achieves capacity?

Next, we consider efficient encoding and decoding algorithms for constant-composition $\mathcal{F}$-avoiding codes.

Let $0 < p < 1$. Kayser and Siegel [19] constructed a family $\{\mathcal{C}_{n,m}\}_{n,m\geq 1}$ of constant-weight $\mathcal{I}_1(2)$-avoiding codes such that each $\mathcal{C}_{n.m}$ is an $\mathcal{I}_1(2)$-avoiding code of length $N = mn + o(mn)$ and constant composition $[(1-p)N, pN]$. Furthermore, Kayser and Siegel showed that there exists a p such that

$$\lim_{n\to\infty}\lim_{m\to\infty}\frac{\log_2|\mathcal{C}_{n,m}|}{mn+o(mn)} = C(\mathcal{I}_1(2)).$$

Unfortunately, for the encoder/decoder pair to work, an auxiliary codebook $\mathcal{C}_n$ of length n is required. Here, the size of the codebook $\mathcal{C}_n$ is exponential in n. In order to approach capacity, both m and n are required to be sufficiently large. Since the encoding and decoding complexity grows in terms of m and $|\mathcal{C}_n|$, we have that the encoding and decoding complexity is exponential in term of n (see [19, Remark 1] for more details).

For $q > 2$, no such concrete results are even known.

C. Our Contributions

Our first contribution is a closed formula for the number of $\mathcal{I}(a,b)$-avoiding words with composition $\overline{w}$.

Theorem 1: Fix q, n, $\mathcal{I}(a,b)$ with $a < b$ and $\overline{w}$. Then

$$\begin{aligned}A(n;\overline{w},\mathcal{I}(a,b)) &= \binom{s_1}{w_0,\cdots,w_a}\binom{s_2}{w_{a+1},\cdots,w_{b-1}}\binom{s_3}{w_b,\cdots,w_{q-1}}\\ &\quad\times\sum_{m=0}^{\min(s_2,s_3-1)}\binom{n-s_3-m}{s_1}B_n^{(m,s_3)},\end{aligned}$$

where $s_1 = \sum_{i=0}^{a} w_i$, $s_2 = \sum_{i=a+1}^{b-1} w_i$, $s_3 = \sum_{i=b}^{q-1} w_i$, and

$$B_n^{(m,s_3)} = \binom{s_3-1}{m}\sum_{i=0}^{s_3-m-1}\binom{s_3-m-1}{i}\binom{n-s_3-m-i+1}{n-s_3-m-2i}. \tag{1}$$

In the instance where $b = a+1$, we have $s_2 = 0$ and so we have only one summand in the outer summation. Therefore,

$$A(n;\overline{w},\mathcal{I}(a,b)) = \binom{s_1}{w_0,\cdots,w_a}\binom{s_3}{w_b,\cdots,w_{q-1}}B_n^{(0,s_3)}.$$

We defer the proof of Theorem 1 to Section II and explain the significance of the term $B_n^{(m,s_3)}$ therein.

While it is difficult to derive a closed expression for $R(\overline{p},\mathcal{I}(a,b))$ from Theorem 1 for general $\overline{p}$ and $\mathcal{I}(a,b)$, it is possible to compute *numerically* $R(\overline{p},\mathcal{I}(a,b))$ for specific values. From Theorem 1, we can compute $R(\overline{p},\mathcal{I}(a,b))$ for general $\overline{p}$ and $\mathcal{I}(a,b)$. Our next contributions are procedures that:

- determine the rates $R(\overline{p},\mathcal{I}(a,b))$ for any composition ratio $\overline{p}$;
- find optimal composition ratios $\overline{p}$ that maximize the rates $R(\overline{p},\mathcal{I}(a,b))$. Furthermore, we also show that these maximum asymptotic rates yield the channel capacity.

Section III provides a detailed description of the procedure and the numerical computations of certain rates.

Finally, we provide efficient encoding and decoding algorithms for binary constant-weight $\mathcal{I}_1(q)$-free codes and a special class of q-ary constant-composition $\mathcal{I}_1(q)$-free codes in Section IV. Therefore, our work gives the first efficient encoding and decoding of constant-composition codes that achieves channel capacity.

II. Proof of Theorem 1

We enumerate the set of all q-ary $\mathcal{I}(a,b)$-avoiding words of composition $\overline{w}$, and hence, prove Theorem 1. To do so, we first enumerate *binary* words that obey certain properties in Section II-A, and then provide a mapping from these binary words to q-ary $\mathcal{I}(a,b)$-avoiding words in Section II-B.

A. A Family of Binary Words

Let $0 \leqslant m \leqslant s_3$. Define $\mathcal{B}_n^{(m,s_3)}$ to be the set of words over the alphabet $\{\circ,\bullet\}$ of length n with the following properties:

(i) each word has exactly s_3 $\bullet$'s;
(ii) each word has exactly m substrings of the form $(\bullet,\circ,\bullet)$.

We demonstrate the following lemma.

Lemma 1: Let $0 \leqslant m \leqslant s_3 - 1$. Then

$$\sum_{n\geqslant 0}\frac{\left|\mathcal{B}_n^{(m,s_3)}\right|}{\binom{s_3-1}{m}}X^n = \frac{X^{s_3+m}(1-X+X^2)^{s_3-m-1}}{(1-X)^{s_3-m+1}}.$$

To prove this lemma, we map $\mathsf{u} \in \mathcal{B}_n^{(m,s_3)}$ to an integer-valued (s_3+1)-tuple $\boldsymbol{d}_{\mathsf{u}} = (d_1, d_2, \ldots, d_{s_3+1})$ such that $d_i \geq 0$ for all $1 \leq i \leq s_3+1$ and $\{t_j = \sum_{i=1}^{j} d_i : 1 \leqslant j \leqslant s_3\}$ is the set of coordinates where $\mathsf{u}_{t_j} = \bullet$, and $d_{s_3+1} = n - \sum_{i=1}^{s_3} d_i$.

Example 2: The word $\mathsf{u} = (\bullet,\circ,\bullet,\bullet,\circ,\bullet,\bullet,\circ)$ belongs to $\mathcal{B}_8^{(2,5)}$, where $m = 2$, $s_3 = 5$, $n = 8$. Hence, $\boldsymbol{d}_{\mathsf{u}} = (1,2,1,2,1,1)$ and $\{1,3,4,6,7\}$ is the set of coordinates where u has the symbol $\bullet$.

It is not difficult to see that $\boldsymbol{d}_{\mathsf{u}} = \boldsymbol{d}_{\mathsf{u}'}$ implies $\mathsf{u} = \mathsf{u}'$. We observe further that for $\mathsf{u} \in \mathcal{B}_n^{(m,s_3)}$, the (s_3+1)-tuple $\boldsymbol{d}_{\mathsf{u}}$ has the following properties:

(C1) the sum of entries in $\boldsymbol{d}_{\mathsf{u}}$ is n;
(C2) exactly m entries of $d_2, d_3, \ldots, d_{s_3}$ are two;
(C3) all entries except d_{s_3+1} of $\boldsymbol{d}_{\mathsf{u}}$ are positive, and d_{s_3+1} is nonnegative.

Conversely, for each (s_3+1)-tuple $\boldsymbol{c}$ that obeys the properties (C1), (C2) and (C3), there exists a $\mathsf{u} \in \mathcal{B}_n^{(m,s_3)}$ such that $\boldsymbol{d}_{\mathsf{u}} = \boldsymbol{c}$. Therefore, the cardinality of $\mathcal{B}_n^{(m,s_3)}$ is equal to the number of (s_3+1)-tuples satisfying these properties.

From (C1) and (C3), such (s_3+1)-tuples are compositions of n with s_3+1 parts and in general, the combinatorics of compositions have been well studied (see Heubach and Mansour [22] for a survey). If we impose restrictions for each part of the composition, we have what is known as *compositions with restricted parts* and the following theorem.

Theorem 2 (Folklore, see [22, Ch. 3]): Let $\boldsymbol{P} = (P_1, P_2, \ldots, P_k)$ be an ordered collection of subsets of integers. Define $\mathrm{Comp}(n;\boldsymbol{P}) \triangleq \{\boldsymbol{c} = (c_1, c_2, \ldots, c_k) : \sum_{j=1}^{k} c_j = n$ *and $c_j \in P_j$ for $1 \leqslant j \leqslant k\}$. Then*

$$\sum_{n\geqslant 0}|\mathrm{Comp}(n;\boldsymbol{P})|X^n = \prod_{j=1}^{k}\sum_{i\in P_j}X^i.$$

For each (s_3+1)-tuple $\boldsymbol{c}$ satisfying properties (C1), (C2) and (C3), we have $\binom{s_3-1}{m}$ ways to choose exactly m entries of $c_2, c_3, \ldots, c_{s_3}$ to be two. Without loss of generality, we assume

$c_2 = c_3 = \cdots = c_{m+1} = 2$. Set $k = s_3 + 1$ and consider the ordered collection $\boldsymbol{P}$ be such that

$$P_j = \begin{cases} \mathbb{Z}_{\geqslant 1}, & \text{if } j = 1, \\ \{2\}, & \text{if } 2 \leqslant j \leqslant m+1, \\ \mathbb{Z}_{\geqslant 1} \setminus \{2\}, & \text{if } m+2 \leqslant j \leqslant s_3, \\ \mathbb{Z}_{\geqslant 0}, & j = s_3 + 1, \end{cases}$$

where $\mathbb{Z}_{\geqslant t}$ denote the set of integers at least t. Then, we have

$$\left|\mathcal{B}_n^{(m,s_3)}\right| = |\mathrm{Comp}(n;\, \boldsymbol{P})| \binom{s_3 - 1}{m}.$$

Since $\sum_{i \in \mathbb{Z}_{\geqslant t}} X^i = X^t/(1-X)$, we have

$$\begin{aligned} \sum_{n \geqslant 0} \frac{\left|\mathcal{B}_n^{(m,s_3)}\right|}{\binom{s_3-1}{m}} X^n &= \sum_{n \geqslant 0} |\mathrm{Comp}(n;\, \boldsymbol{P})|\, X^n \\ &= \left(\frac{X}{1-X}\right)\left(X^2\right)^m \left(X + \frac{X^3}{1-X}\right)^{s_3-m-1}\left(\frac{1}{1-X}\right) \\ &= \frac{X^{s_3+m}(1-X+X^2)^{s_3-m-1}}{(1-X)^{s_3-m+1}}. \end{aligned}$$

This completes the proof of Lemma 1. To compute $\left|\mathcal{B}_n^{(m,s_3)}\right|$, we extract the coefficient of X^n and multiply it by $\binom{s_3-1}{m}$. For convenience, we let $[X^j]\Big\{g(X)\Big\}$ denote the coefficient of X^j in $g(X)$. Hence,

$$\begin{aligned} &\left[X^n\right]\left\{X^{s_3+m}(1-X+X^2)^{s_3-m-1}(1-X)^{-s_3+m-1}\right\} \\ &= \left[X^{n-s_3-m}\right]\left\{(1-X+X^2)^{s_3-m-1}(1-X)^{-s_3+m-1}\right\} \\ &= \sum_{i=0}^{s_3-m-1}\binom{s_3-m-1}{i}\left[X^{n-s_3-m-2i}\right]\left\{(1-X)^{-2-i}\right\} \\ &= \sum_{i=0}^{s_3-m-1}\binom{s_3-m-1}{i}\binom{n-s_3-m-i+1}{n-s_3-m-2i}. \end{aligned}$$

Setting $B_n^{(m,s_3)} = \left|\mathcal{B}_n^{(m,s_3)}\right|$ yields (1).

B. Mapping to q-Ary Words

Finally, to complete the proof of Theorem 1, we take a word in $\mathcal{B}_n^{(m,s_3)}$ and replace the symbols in $\{\bullet, \circ\}$ with symbols in Σ. For convenience, we partition Σ into three parts:

$$\Sigma_1 = \{0, \ldots, a\}, \quad \Sigma_2 = \{a+1, \ldots, b-1\}, \\ \Sigma_3 = \{b, \ldots, q-1\}.$$

In addition, for $i = 1, 2, 3$, we consider $\mathcal{E}_i$ to be a set of words over Σ_i of length s_i such that $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ are the sets of all words with compositions $[w_0, \ldots, w_a]$, $[w_{a+1}, \ldots, w_{b-1}]$, and $[w_b, \ldots, w_{q-1}]$, respectively.

Example 3: Let $q = 5$, $a = 1$, $b = 4$. So, $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{2, 3\}$, and $\Sigma_3 = \{4\}$. Furthermore, let $n = 8$ with $\overline{w} = (1, 1, 1, 2, 3)$. Hence, $(s_1, s_2, s_3) = (2, 3, 3)$ and

$$\begin{aligned} \mathcal{E}_1 &= \{(0,1), (1,0)\}, \\ \mathcal{E}_2 &= \{(2,3,3), (3,2,3), (3,3,2)\}, \\ \mathcal{E}_3 &= \{(4,4,4)\}. \end{aligned}$$

For $\mathsf{u} \in \mathcal{B}_n^{(m,s_3)}$, we further define $T(\mathsf{u})$ to be the set of $n - s_3 - m$ coordinates such that $t \in T(\mathsf{u})$ implies that $\mathsf{u}_t = \circ$, but $(\mathsf{u}_{t-1}, \mathsf{u}_t, \mathsf{u}_{t+1}) \neq (\bullet, \circ, \bullet)$. In other words, $T(\mathsf{u})$ is the set of $n - s_3 - m$ $\circ$'s in u that do not belong to the substrings $(\bullet, \circ, \bullet)$. Let $\mathcal{D}(\mathsf{u})$ be the collection of all subsets of $T(\mathsf{u})$ of size s_1.

Example 4: Let $\mathsf{u} = (\bullet, \circ, \bullet, \circ, \bullet, \circ, \circ, \circ)$ with $n = 8$, $s_3 = 3$, $m = 2$. Then $T(\mathsf{u}) = \{6, 7, 8\}$ and for $s_1 = 2$, we have $\mathcal{D}(\mathsf{u}) = \{\{6,7\}, \{6,8\}, \{7,8\}\}$.

Next, we define the following collection of pairs:

$$\mathcal{D}_n^{(m,s_3)} \triangleq \Big\{(\mathsf{u}, D) : \mathsf{u} \in \mathcal{B}_n^{(m,s_3)}, D \in \mathcal{D}(\mathsf{u})\Big\}.$$

Observe that $\left|\mathcal{D}_n^{(m,s_3)}\right| = B_n^{(m,s_3)}\binom{n-s_3-m}{s_1}$ and consider the following map,

$$\Phi_1 : \mathcal{E}_1 \times \mathcal{E}_2 \times \mathcal{E}_3 \times \bigcup_{m=0}^{\min(s_2, s_3-1)} \mathcal{D}_n^{(m,s_3)} \to \mathcal{A}(n; \overline{w}, \mathcal{I}(a,b)), \\ (\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, \mathsf{u}, D) \mapsto \mathsf{v},$$

in which, $\mathsf{e}_i \in \mathcal{E}_i$ for $i = 1, 2, 3$, $\mathsf{u} \in \mathcal{B}_n^{(m,s_3)}$ and $D_1 \in \mathcal{D}(\mathsf{u})$ are given. Let D_2 be the set of coordinates of $\circ$ in u that do not belong to D_1 and $\Phi_1(\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, \mathsf{u}, D) = \mathsf{v} \in \mathcal{A}(n; \overline{w}, \mathcal{I}(a,b))$ is the q-ary word obtained by substituting

- the s_1 $\circ$'s of u at index set D_1 with e_1,
- the s_2 $\circ$'s of u at index set D_2 with e_2, and
- the s_3 $\bullet$'s of u with e_3.

Lemma 2: The map Φ_1 is a bijection.

Proof: We define the following map,

$$\Phi_2 : \mathcal{A}(n; \overline{w}, \mathcal{I}(a,b)) \to \mathcal{E}_1 \times \mathcal{E}_2 \times \mathcal{E}_3 \times \bigcup_{m=0}^{\min(s_2, s_3-1)} \mathcal{D}_n^{(m,s_3)} \\ \mathsf{v} \mapsto (\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, \mathsf{u}, D),$$

in which, $\mathsf{v} \in \mathcal{A}(n; \overline{w}, \mathcal{I}(a,b))$ is given and $\Phi_2(\mathsf{v}) = (\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, (\mathsf{u}, D))$ is determined as follows.

- e_i is the subsequence of v whose symbols belong to Σ_i for $i = 1, 2, 3$,
- u is the word obtained by substituting symbols in $\Sigma_1 \cup \Sigma_2$ with $\circ$ and symbols in Σ_3 with $\bullet$, and
- D is the set of indices with symbols in Σ_1.

We observe that $\Phi_1 \circ \Phi_2$ and $\Phi_2 \circ \Phi_1$ are identity maps on their respective domains.

Therefore, Φ_1 is a bijection. ■

Example 5: Let $q, a, b, n, \overline{w}$, and u be as defined in Examples 3 and 4. Consider $\mathsf{e}_1 = (0,1)$, $\mathsf{e}_2 = (3,2,3)$, $\mathsf{e}_3 = (4,4,4)$ and $D = \{6, 8\}$. Then $\Phi_1(\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, (\mathsf{u}, D)) = (4,3,4,2,4,0,3,1)$. Conversely, if we set $\mathsf{v} = (4,3,4,2,4,0,3,1)$, then $\Phi_2(\mathsf{v})$ recovers $\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3, \mathsf{u}$ and D.

Combining Lemmas 1 and 2 yields Theorem 1.

III. Rates of Constant-Composition $\mathcal{F}$-Avoiding Codes

In this section, we provide an efficient numerical procedure to determine the asymptotic information rates of $(\overline{\rho}, \mathcal{I}(a,b))$-pairs.

In what follows, we consider the entropy function $\mathsf{H}(p_1, p_2, \ldots, p_k) = -\sum_{i=1}^{k} p_i \log p_i$ and the binary entropy function $\mathsf{H}_2(p) = \mathsf{H}(1-p, p)$, where $\sum_{i=1}^{k} p_i = 1$ and p, p_i are nonnegative for all $1 \leqslant i \leqslant k$.

Theorem 3: Let $q \geqslant 3$ and $\overline{\rho} = (\rho_0, \rho_1, \ldots, \rho_{q-1})$. Let $\sum_{k=0}^{a} \rho_k = x_1$. Let $\sum_{k=a+1}^{b-1} \rho_k = x_2$. Let $\sum_{k=b}^{q-1} \rho_k = x_3$. For $0 \le i \le a$, let $p_i = \rho_i/x_1$. For $a+1 \le i \le b-1$, let $p_i = \rho_i/x_2$. For $b \le i \le q-1$, let $p_i = \rho_i/x_3$.

Define the function $F(\overline{\rho}, y, z)$ such that

$$\begin{aligned} F(\overline{\rho}, y, z) \triangleq\ & x_1\mathsf{H}(p_0, \ldots, p_a) + x_2\mathsf{H}(p_{a+1}, \ldots, p_{b-1}) \\ & + x_3\mathsf{H}(p_b, \ldots, p_{q-1}) \\ & + (1 - x_3 - x_3\ y)\mathsf{H}_2\left(\frac{x_1}{(1 - x_3 - x_3\ y)}\right) \\ & + x_3\mathsf{H}_2(y) + (x_3 - x_3 y)\mathsf{H}_2(z) \\ & + (1 - x_3 - x_3 y - z(x_3 - x_3 y)) \\ & \times \mathsf{H}_2\left(\frac{1 - x_3 - x_3 y - 2z(x_3 - x_3 y)}{1 - x_3 - x_3 y - z(x_3 - x_3 y)}\right). \end{aligned}$$

Then the asymptotic rate

$$R(\overline{\rho}, \mathcal{I}(a,b)) = \max_{\substack{0 \leqslant z \leqslant 1 \\ 0 \leqslant y \leqslant \min\{1, x_2/x_3\}}} F(\overline{\rho}, y, z).$$

Proof: For each n, let $\overline{w}(n) = (w_0, w_1, \ldots, w_{q-1})$ such that $w_i = \lfloor \rho_i \cdot n \rfloor$ for all $0 \leqslant i \leqslant q-2$ and $w_{q-1} = n - \sum_{i=0}^{q-2} w_i$. We verify that the sequence $\overline{w}(n)/n$ converges to $\overline{\rho}$ componentwise.

From Theorem 1, we know that

$$A(n; \overline{w}(n), \mathcal{I}(a,b)) = \sum_{m-0}^{\min\{s_2, s_3-1\}} \sum_{i=0}^{s_3-m-1} D_{i,m},$$

where

$$\begin{aligned} D_{i,m} = & \binom{s_1}{w_0, \cdots, w_a}\binom{s_2}{w_{a+1}, \cdots, w_{b-1}}\binom{s_3}{w_b, \cdots, w_{q-1}} \\ & \times \binom{n - s_3 - m}{s_1}\binom{s_3 - 1}{m}\binom{s_3 - m - 1}{i} \\ & \times \binom{n - s_3 - m - i + 1}{n - s_3 - m - 2i}. \end{aligned}$$

For $0 \leqslant m \leqslant \min\{s_2, s_3 - 1\}$ and $0 \leqslant i \leqslant s_3 - m - 1$, let $0 \leqslant y = m/s_3 \leqslant \min\{1, x_2/x_3\}$ and $0 \leqslant z = i/(s_3 - m) \leqslant 1$. Then by Stirling's approximation,

$$2^{nF(\overline{\rho}, y, z) - o(n)} \leqslant D_{i,m} \leqslant 2^{nF(\overline{\rho}, y, z) + o(n)} \quad \text{for all } i, m.$$

Hence, $A(n; \overline{w}(n), \mathcal{I}(a,b)) \geqslant D_{i,m} \geqslant 2^{nF(\overline{\rho}, y, z) - o(n)}$.

Taking logarithms, dividing by n and taking limits in n yields the inequality $R(\overline{\rho}, \mathcal{I}(a,b)) \geqslant F(\overline{\rho}, y, z)$ for all $0 \leqslant y \leqslant \min\{1, x_2/x_3\}$ and $0 \leqslant z \leqslant 1$. Let

$$F(\overline{\rho}, y^*, z^*) = \max_{\substack{0 \leqslant z \leqslant 1; \\ 0 \leqslant y \leqslant \min\{1, x_2/x_3\}}} F(\overline{\rho}, y, z).$$

Therefore, $R(\overline{\rho}, \mathcal{I}(a,b)) \geqslant F(\overline{\rho}, y^*, z^*)$.

On the other hand, we have

$$\begin{aligned} A(n; \overline{w}(n), \mathcal{I}(a,b)) & \leqslant \sum_{i,m} 2^{nF(\overline{\rho}, y, z) + o(n)} \\ & \leqslant n^2 \times 2^{nF(\overline{\rho}, y^*, z^*) + o(n)}. \end{aligned}$$

Again, taking logarithms, dividing by n and taking limits in n, we obtain $R(\overline{\rho}, \mathcal{I}(a,b)) \leqslant F(\overline{\rho}, y^*, z^*)$. Therefore, $R(\overline{\rho}, \mathcal{I}(a,b)) = F(\overline{\rho}, y^*, z^*)$, completing the proof. ■

So, for each composition ratio $\overline{\rho}$, we can find the asymptotic rate $R(\overline{\rho}, I(a,b))$ by maximizing the multivariate function $F(\overline{\rho}, y, z)$. The corollaries in the next subsection follow from direct application of Theorem 3.

A. Avoiding $\mathcal{I}_1(q)$

Corollary 1: Let $\overline{\rho} = (\rho_0, \rho_1, \ldots, \rho_{q-1})$. For $0 \le i \le q-2$, let $p_i = \rho_i / \sum_{k=0}^{q-2} \rho_k$ and $p_{q-1} = \rho_{q-1}$. Define the function F_1 so that

$$\begin{aligned} F_1(x) \triangleq\ & (1 - p_{q-1})\mathsf{H}(p_0, \ldots, p_{q-2}) + p_{q-1}\mathsf{H}_2(x) \\ & + (1 - p_{q-1} - p_{q-1}x)\mathsf{H}_2\left(\frac{1 - p_{q-1} - 2p_{q-1}x}{1 - p_{q-1} - p_{q-1}x}\right). \end{aligned}$$

Then the asymptotic rate $R(\overline{\rho}, \mathcal{I}_1(q))$ is given by $\max_{0 \leqslant x \leqslant 1} F_1(x)$.

Proof: Apply Theorem 3 with $a = q-2$ and $b = q-1$. Here, $y = 0$ since $x_2 = 0$. ■

Example 6: Let $q = 2$ and $\overline{\rho} = (1/2, 1/2)$. Then

$$F_1(x) = \frac{1}{2}\left(\mathsf{H}_2(x) + (1-x)\mathsf{H}_2\left(\frac{1-2x}{1-x}\right)\right).$$

Now, $F_1(x)$ is maximized when $x = 1/3$ and achieves the value $(\log 3)/2$. This yields $R_{\text{bal}}(\mathcal{I}_1(2))$ and recovers the result in Qin *et al.* [18]. Continuing this example, we compute the rates $R_{\text{bal}}(\mathcal{I}_1(q))$ for $2 \leqslant q \leqslant 8$ and tabulate these values in the second column of Table I.

B. Avoiding $\mathcal{I}_2(q)$

Corollary 2: Let $q \geqslant 3$ and $\overline{\rho} = (\rho_0, \rho_1, \ldots, \rho_{q-1})$. For $1 \le i \le q-2$, let $p_i = \rho_i / \sum_{k=1}^{q-2} \rho_k$. Define the function $F_2(y,z)$ such that

$$\begin{aligned} F_2(y,z) \triangleq\ & (1 - \rho_0 - \rho_{q-1})\mathsf{H}(p_1, \ldots, p_{q-2}) \\ & + (1 - \rho_{q-1} - \rho_{q-1}y)\mathsf{H}_2\left(\frac{\rho_0}{(1 - \rho_{q-1} - \rho_{q-1}y)}\right) \\ & + \rho_{q-1}\mathsf{H}_2(y) + (\rho_{q-1} - \rho_{q-1}y)\mathsf{H}_2(z) \\ & + (1 - \rho_{q-1} - \rho_{q-1}y - z(\rho_{q-1} - \rho_{q-1}y)) \\ & \times \mathsf{H}_2\left(\frac{1 - \rho_{q-1} - \rho_{q-1}y - 2z(\rho_{q-1} - \rho_{q-1}y)}{1 - \rho_{q-1} - \rho_{q-1}y - z(\rho_{q-1} - \rho_{q-1}y)}\right). \end{aligned}$$

Then the asymptotic rate

$$\begin{aligned} R(\overline{\rho}, \mathcal{I}_2(q)) = \max\{F_2(y,z) : 0 \leqslant z \\ \leqslant 1 \textit{ and } 0 \leqslant y \leqslant \min\{1, \frac{(1 - \rho_0 - \rho_{q-1})}{\rho_{q-1}}\}. \end{aligned}$$

Proof: Apply Theorem 3 with $a = 0$ and $b = q-1$. ■

As before, for $3 \leqslant q \leqslant 8$, we compute $R_{\text{bal}}(\mathcal{I}_2(q))$ and tabulate these results in the sixth column of Table I.

C. Capacity-Achieving Codes With Constant Composition

Consider the function F defined in Theorem 3. Since we are interested in constant-composition codes with high rates, we want to find the composition ratio $\overline{\rho}$ such that the rate $R(\overline{\rho}, \mathcal{I}(a,b))$ is maximal. A natural approach is to maximize $F(\overline{\rho}, y, z)$ in the variables $\overline{\rho}$, y and z, that have $q+2$ real components. However, the following theorem demonstrates that it suffices to maximize a function in four variables (independent of q). Furthermore, we also show that these codes achieve capacity when the rates are maximized.

Theorem 4: Given $q \geqslant 2$ and a, b such that $0 \leqslant a < b \leqslant q-1$. Define

$$\begin{aligned}F^*(x_1, x_3, y, z) \triangleq\ & x_1 \log(a+1) \\ &+ (1-x_1-x_3)\log(b-a-1) + x_3\mathsf{H}_2(y) \\ &+ (x_3 - x_3 y)\mathsf{H}_2(z) \\ &+ x_3 \log(q-b) + (1 - x_3 - x_3\ y) \\ &\times \mathsf{H}_2\left(\frac{x_1}{(1-x_3-x_3\ y)}\right) \\ &+ (1 - x_3 - x_3 y - z(x_3 - x_3 y)) \\ &\times \mathsf{H}_2\left(\frac{1 - x_3 - x_3 y - 2z(x_3 - x_3 y)}{1 - x_3 - x_3 y - z(x_3 - x_3 y)}\right).\end{aligned}$$

Then, $C(\mathcal{I}(a,b)) = \max\{F^*(x_1, x_3, y, z) : 0 \leqslant x_1, x_3, y, z \leqslant 1,\ y \leqslant (1-x_1-x_3)/x_3\}$.

Furthermore, if x_1^*, x_3^*, y^*, z^* satisfy $C(\mathcal{I}(a,b)) = F^*(x_1^*, x_3^*, y^*, z^*)$, then $R(\overline{\rho}, \mathcal{I}(a,b)) = C(\mathcal{I}(a,b))$ where $\rho_i = x_1^*/(a+1)$ for all $0 \leqslant i \leqslant a$, $\rho_i = (1-x_1^*-x_3^*)/(b-a-1)$ for all $a+1 \leqslant i \leqslant b-1$ and $\rho_i = x_3^*/(q-b)$ for all $b \leqslant i \leqslant q-1$.

Proof: Let $F^*(x_1^*, x_3^*, y^*, z^*) = \max\{F^*(x_1, x_3, y, z) : 0 \leqslant x_1, x_3, y, z \leqslant 1; y \leqslant (1-x_1-x_3)/x_3\}$.

Let $D_{\max}(n) = \max\{A(n; \overline{w}, \mathcal{F}) : \sum w_i = n\}$ for all n. Since $A(n; \overline{w}(n), \mathcal{I}(a,b)) \leqslant n^2 \cdot 2^{n \max_{y,z} F(\overline{\rho}, y, z) + o(n)}$, we have that

$$D_{\max}(n) \leqslant n^2 \cdot 2^{n \max_{\overline{\rho}, y, z} F(\overline{\rho}, y, z) + o(n)}.$$

From the definition of capacity,

$$\begin{aligned}C(\mathcal{I}(a,b)) &= \limsup_{n\to\infty} \frac{\log |\mathcal{A}(n; \mathcal{I}(a,b))|}{n} \\ &= \limsup_{n\to\infty} \frac{\log \sum_{\sum w_i = n} A(n, \overline{w}(n), \mathcal{I}(a,b))}{n} \\ &\leqslant \limsup_{n\to\infty} \frac{\log n^q \max_{\sum w_i = n} A(n, \overline{w}, \mathcal{I}(a,b))}{n} \\ &\leqslant \limsup_{n\to\infty} \frac{\log D_{\max}(n)}{n} \leqslant \max_{\overline{\rho}, y, z} F(\overline{\rho}, y, z).\end{aligned}$$

Now, $\mathsf{H}(p_0, \ldots, p_a) \leqslant \log(a+1)$, $\mathsf{H}(p_{a+1}, \ldots, p_{b-1}) \leqslant \log(b-a-1)$, and $\mathsf{H}(p_b, \ldots, p_{q-1}) \leqslant \log(q-b)$. Therefore, $F(\overline{\rho}, y, z) \leqslant F^*(x_1, x_2, x_3, y, z)$, where $x_1 = \sum_{i=0}^{a} \rho_i$ and $x_3 = \sum_{i=b}^{q-1} \rho_i$. So,

$$C(\mathcal{I}(a,b)) \leqslant \max_{\overline{\rho}, y, z} F(\overline{\rho}, y, z) \leqslant F^*(x_1^*, x_3^*, y^*, z^*).$$

On the other hand, $C(\mathcal{I}(a,b)) \geq R(\overline{\rho}, \mathcal{I}(a,b))$ for all composition ratios $\overline{\rho}$.

Choose $\overline{\rho} = (\rho_0, \ldots, \rho_{q-1})$ such that $\rho_i = x_1^*/(a+1)$ for all $0 \leqslant i \leqslant a$, $\rho_i = (1 - x_1^* - x_3^*)/(b-a-1)$ for all $a+1 \leqslant i \leqslant b-1$, and $\rho_i = x_3^*/(q-b)$ for all $b \leqslant i \leqslant q-1$. Then $F(\overline{\rho}, y, z) = F^*(x_1^*, x_3^*, y, z)$ and applying Theorem 3, we have

$$\begin{aligned}R(\overline{\rho}, \mathcal{I}(a,b)) &= \max_{\substack{0\leqslant z\leqslant 1 \\ 0\leqslant y\leqslant \min\{1,(1-x_1-x_3)/x_3\}}} F(\overline{\rho}, y, z) \\ &= \max_{\substack{0\leqslant z\leqslant 1 \\ 0\leqslant y\leqslant \min\{1,(1-x_1-x_3)/x_3\}}} F(\overline{\rho}, y, z) \\ &= F^*(x_1^*, x_3^*, y^*, z^*).\end{aligned}$$

Therefore, $C(\mathcal{I}(a,b)) \geqslant F^*(x_1^*, x_3^*, y^*, z^*)$ and we conclude $C(\mathcal{I}(a,b)) = F^*(x_1^*, x_3^*, y^*, z^*)$. ■

Applying Theorem 4 for $\mathcal{I}_1(q)$-avoiding codes and $\mathcal{I}_2(q)$-avoiding codes, we obtain the following corollary.

Corollary 3: (i) Let

$$\begin{aligned}F_1^*(x, z) = (1-x)\log(q-1) + x\mathsf{H}_2(z) \\ + (1 - x - xz)\mathsf{H}_2\left(\frac{1-x-2xz}{1-x-xz}\right).\end{aligned}$$

Then $C(\mathcal{I}_1(q)) = \max\{F_1^*(x,z) : 0 \leqslant x, z \leqslant 1\}$ for $q \geqslant 2$. Moreover, for $\overline{\rho} = ((1-x^*)/(q-1), (1-x^*)/(q-1), \ldots, (1-x^*)/(q-1), x^*)$, where $(x^*, z^*) \in \arg\max\{F_1^*(x,z) : 0 \leqslant x, z \leqslant 1\}$, the asymptotic rate $R(\overline{\rho}, \mathcal{I}_1(q)) = C(\mathcal{I}_1(q))$.

(ii) Let

$$\begin{aligned}&F_2^*(x_0, x, y, z) \\ &\quad = (1 - x_0 - x)\log(q-2) \\ &\qquad + (1 - x - xy)\mathsf{H}_2\left(\frac{x_0}{(1-x-xy)}\right) \\ &\qquad + x\mathsf{H}_2(y) + (x - xy)\mathsf{H}_2(z) \\ &\qquad + (1 - x - xy - z(x - xy)) \\ &\qquad \times \mathsf{H}_2\left(\frac{1 - x - xy - 2z(x - xy)}{1 - x - xy - z(x - xy)}\right).\end{aligned}$$

Then $C(\mathcal{I}_2(q)) = \max\{F^*(x_0, x, y, z) : 0 \leqslant x_0, x, y, z \leqslant 1,\ y \leqslant (1 - x_0 - x)/x\}$ for $q \geq 3$. Moreover, for $\overline{\rho} = (x_0^*, (1 - x_0^* - x^*)/(q-2), \ldots, (1 - x_0^* - x^*)/(q-2), x^*)$, where $(x_0^*, x^*, y^*, z^*) \in \arg\max\{F_1^*(x_0, x, y, z) : 0 \leqslant x_0, x, y, z \leqslant 1; y \leqslant (1 - x_0 - x)/x\}$, the asymptotic rate $R(\overline{\rho}, \mathcal{I}_2(q)) = C(\mathcal{I}_2(q))$.

We apply Corollary 3 to determine composition ratios $\overline{\rho}$ that allow $\mathcal{I}_1(q)$-avoiding and $\mathcal{I}_2(q)$-avoiding codes to achieve capacity. In the case of $\mathcal{I}_1(q)$-avoiding codes, the values of asymptotic rates $R(\overline{\rho}, \mathcal{I}_1(q))$ for $2 \leq q \leq 8$ are listed in the fourth column of Table I. Here, the composition ratio is $\overline{\rho} = [\rho, \rho, \ldots, \rho, \rho_{q-1}(\mathcal{I}_1(q))]$, where $\rho = (1 - \rho_{q-1}(\mathcal{I}_1(q)))/(q-1)$ and the corresponding values of $\rho_{q-1}(\mathcal{I}_1(q))$ are listed in the third column of Table I. In the case of $\mathcal{I}_2(q)$-avoiding codes, we observe that the function $F_2^*(x_0, x, y, z)$ is maximized when $x_0^* = (1 - x_0^* - x^*)/(q-2) = (1 - x^*)/(q-1)$. For $3 \leq q \leq 8$, we computed $x^* = \rho_{q-1}(\mathcal{I}_2(q))$ and listed in the seventh column of Table I. The corresponding asymptotic rates $R(\overline{\rho}, \mathcal{I}_2(q))$ are listed in the eighth column of Table I.

TABLE I
RATES OF $\mathcal{I}_1(q)$ AND $\mathcal{I}_2(q)$-AVOIDING CODES WITH CONSTANT COMPOSITION

q	$R_{\mathrm{bal}}(\mathcal{I}_1(q))$	$\rho_{q-1}(\mathcal{I}_1(q))$	$R(\overline{\rho},\mathcal{I}_1(q))$	$C(\mathcal{I}_1(q))$	$R_{\mathrm{bal}}(\mathcal{I}_2(q))$	$\rho_{q-1}(\mathcal{I}_2(q))$	$R(\overline{\rho},\mathcal{I}_2(q))$	$C(\mathcal{I}_2(q))$
2	0.79248	0.41150	0.81137	0.81137				
3	1.46127	0.25653	1.48353	1.48353	1.52576	0.29308	1.53145	1.53145
4	1.92207	0.19425	1.93743	1.93743	1.97589	0.22989	1.97758	1.97758
5	2.26928	0.15865	2.27945	2.27945	2.30984	0.18867	2.31046	2.31046
6	2.54732	0.13496	2.55420	2.55420	2.57805	0.15967	2.57832	2.57832
7	2.77921	0.11782	2.78403	2.78403	2.80304	0.13827	2.80317	2.80317
8	2.97821	0.10475	2.98169	2.98169	2.99713	0.12181	2.99719	2.99719

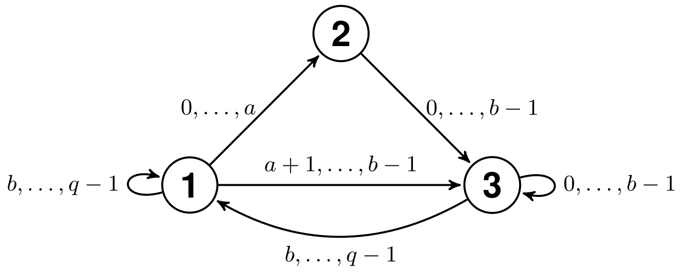

Fig. 1. Primitive graph G presenting the $\mathcal{I}(a,b)$-avoiding channel.

D. Markov Chain Approach

In this subsection, we present an alternative approach to compute the asymptotic information rate of q-ary $\mathcal{I}(a,b)$-avoiding constant-composition codes. The approach was communicated to us after the conference version [1] of the paper was presented. For completeness, we present this approach here and remark that the connection between the Markov chain approach and the approach using enumerative techniques remains unclear.

Recently, Roth and Siegel independently computed the asymptotic information rate of the ICI-free balanced codes using Markov chain approach [27]. The technique is presented in [25] and in [28, Ch. 9].

Here, we mimic the argument in [27] to compute the rate $R(\overline{\rho},\mathcal{I}(a,b))$. Figure 1 shows the minimum primitive graph G that presents the constraint of avoiding all patterns in the set $\mathcal{I}(a,b)$. Recall that $\overline{\rho}=(\rho_0,\rho_1,\dots,\rho_{q-1})$ is the composition ratio. Define the vector indicator function $\boldsymbol{I}_W : E_G \mapsto \mathbb{R}^{q-1}$ where $\boldsymbol{I}_W=(I_1,I_2,\dots,I_{q-1})$ with I_m denoting the indicator function for the symbol $m\in\{1,\dots,q-1\}$. We then consider stationary Markov chains such that $E(\boldsymbol{I}_W)=\boldsymbol{p}=(\rho_1,\dots,\rho_{q-1})$.

Theorem 5: Let $\boldsymbol{x}=(x_1,x_2,\dots,x_{q-1})$ be a vector of real numbers. Let $\boldsymbol{p}=(\rho_1,\dots,\rho_{q-1})$ and set the composition ratio to be $\overline{\rho}=(1-\sum_{i=1}^{q-1}\rho_i,\rho_1,\dots,\rho_{q-1})$. The maximum asymptotic rate of q-ary $\mathcal{I}(a,b)$-avoiding constant-composition constraint is given by

$$R(\overline{\rho},\mathcal{I}(a,b))=\inf_{\boldsymbol{x}\in\mathbb{R}^{q-1}}\{\boldsymbol{x}\cdot\boldsymbol{p}+\log\lambda(A_{G;I_W}(\boldsymbol{x}))\},$$

where

$$A_{G;I_W}(\boldsymbol{x})=\begin{bmatrix}\sum_{i=b}^{q-1}2^{-x_i} & 1+\sum_{i=1}^{a}2^{-x_i} & \sum_{i=a+1}^{b-1}2^{-x_i}\\ 0 & 0 & 1+\sum_{i=1}^{b-1}2^{-x_i}\\ \sum_{i=b}^{q-1}2^{-x_i} & 0 & 1+\sum_{i=1}^{b-1}2^{-x_i}\end{bmatrix}.$$

However, it is not trivial to solve the above optimization problem in Theorem 5 in general, especially when q is large. In [27], Roth and Siegel presented numerical result in the case $q=3$ and $a=0, b=q-1$ for balanced codes. However, using the approach from Theorem 3, we calculated the results $R_{bal}(\mathcal{I}_2(q))$ for all $3\le q\le 8$ in the sixth column in Table I. Furthermore, we also computed the optimal composition ratios $\overline{\rho}_0$ such that the rate $R(\overline{\rho}_0,\mathcal{I}(a,b))$ achieves the capacity of the $\mathcal{I}(a,b)$-avoiding channel $C(\mathcal{I}(a,b))$ for some specific cases in Table I.

IV. ENCODING AND DECODING ALGORITHMS

We consider efficient encoding and decoding algorithms for $\mathcal{I}_1(q)$-avoiding codes with certain constant compositions. In this section, we focus on the avoidance set $\mathcal{I}_1(q)$ as Taranalli *et al.* [10] have shown that avoiding $\mathcal{I}_2(q)$ is not enough to mitigate ICI effects. As such, we adopt the following notations in this section for convenience.

A q-ary code $\mathcal{C}\subseteq\Sigma^n$ is *ICI-free* if it avoids $\mathcal{I}_1(q)$. An *ICI channel* is a channel whose input codewords are ICI-free and the *capacity* of a q-ary ICI channel is $C_{\mathrm{ICI}}(q)=C(\mathcal{I}_1(q))$.

The set of all q-ary ICI-free words of length n with constant composition $\overline{w}$ is denoted $\mathcal{S}(n,\overline{w})$. Note that q, the size of the alphabet, is determined by the composition $\overline{w}$. In the case $q=2$, we further abbreviate $\mathcal{S}(n,[w_0,w_1])$ to $\mathcal{S}(n,w_1)$. The size of $\mathcal{S}(n,\overline{w})$ is denoted by $A_{\mathrm{ICI}}(n,\overline{w})$. Finally, we use $[\![n]\!]$ to denote the set of integers $\{1,2,\dots,n\}$.

A. A Recursive Construction for (Binary) $\mathcal{S}(n,w)$

Let $n\ge w\ge 2$ and define the map

$$\phi:\bigcup_{k\in[\![n-w+1]\!]\setminus\{2\}}\mathcal{S}(n-k,w-1)\to\mathcal{S}(n,w),$$

such that

$$\begin{aligned}\phi(\mathsf{u}_1,\mathsf{u}_2,\dots,\mathsf{u}_{n-k})&=(\mathsf{u}_1,\mathsf{u}_2,\dots,\mathsf{u}_{r(\mathsf{u})},\underbrace{0,0,\dots,0}_{k-1\ 0\text{'s}},1,\mathsf{u}_{r(\mathsf{u})+1},\\&\qquad\mathsf{u}_{r(\mathsf{u})+2},\dots,\mathsf{u}_{n-k}),\end{aligned}$$

where $r(\mathsf{u})$ is the position of the rightmost "1" in $\mathsf{u}=(\mathsf{u}_1,\mathsf{u}_2,\dots,\mathsf{u}_{n-k})$, that is,

$$r(\mathsf{u})=\max\{i\in[\![n-k]\!]\colon \mathsf{u}_i=1 \text{ and } \mathsf{u}_j=0 \text{ for all } j\ge i\}.$$

Theorem 6: The map ϕ is a bijection.

Proof: We first show injectivity of ϕ. If u and v are distinct elements of $\mathcal{S}(n-k, w-1)$ for some $k \in [\![n-w+1]\!] \setminus \{2\}$, then

- when $r(\mathsf{u}) = r(\mathsf{v})$, $\phi(\mathsf{u})$ and $\phi(\mathsf{v})$ must differ in some of their first $r(\mathsf{u})$ positions;
- when $r(\mathsf{u}) \neq r(\mathsf{v})$, $\phi(\mathsf{u})$ and $\phi(\mathsf{v})$ must differ in some of their last $n-k-r(\mathsf{u})$ positions.

If $\mathsf{u} \in \mathcal{S}(n-k_1, w-1)$ and $\mathsf{v} \in \mathcal{S}(n-k_2, w-1)$ for some $k_1, k_2 \in [\![n-w+1]\!] \setminus \{2\}$, where $k_1 \neq k_2$, then the rightmost occurrence of a substring of the form $(1, 0, 0, \ldots, 0, 1)$ in $\phi(\mathsf{u})$ and $\phi(\mathsf{v})$ has lengths $k_1 + 1$ and $k_2 + 2$, respectively.

To prove surjectivity of ϕ, consider $\mathsf{u} \in \mathcal{S}(n, w)$. Let $s(\mathsf{u})$ be the starting position of the rightmost substring in u of the form $(1, \underbrace{0, 0, \ldots, 0}_{k-1 \text{ 0's}}, 1)$, where $k \geq 1$. Deleting the substring $(\mathsf{u}_{s(\mathsf{u})+1}, \mathsf{u}_{s(\mathsf{u})+2}, \ldots, \mathsf{u}_{s(\mathsf{u})+k})$ from u gives an element $\mathsf{v} \in \mathcal{S}(n-k, w-1)$ such that $\phi(\mathsf{v}) = \mathsf{u}$. ■

Corollary 4:

$$A_{\text{ICI}}(n, w) = \sum_{k \in [\![n-w+1]\!] \setminus \{2\}} A_{\text{ICI}}(n-k, w-1). \tag{2}$$

Theorem 6 and Corollary 4, together with Proposition 4 below, give recurrence for the construction of $\mathcal{S}(n, w)$ and the determination of $A_{\text{ICI}}(n, w)$.

Proposition 4: For $n \geq 1$,

(i) $\mathcal{S}(n, 1)$ is the set of all words of weight one in Σ^n, and hence $A_{\text{ICI}}(n, 1) = n$;

(ii) $\mathcal{S}(n, n) = \{(1, 1, \ldots, 1)\}$, and hence $A_{\text{ICI}}(n, n) = 1$.

Example 7: We can construct $\mathcal{S}(5, 3)$ from $\mathcal{S}(4, 2)$ and $\mathcal{S}(2, 2)$. Moreover, $\mathcal{S}(4, 2)$ can in turn be constructed from $\mathcal{S}(3, 1)$ and $\mathcal{S}(1, 1)$. Hence, with the trivial codes $\mathcal{S}(3, 1) = \{100, 010, 001\}$, $\mathcal{S}(1, 1) = \{1\}$, and $\mathcal{S}(2, 2) = \{11\}$, we obtain

$$\begin{aligned} \mathcal{S}(4, 2) &= \phi(\mathcal{S}(3, 1)) \cup \phi(\mathcal{S}(1, 1)) \\ &= \{1100, 0110, 0011\} \cup \{1001\}, \end{aligned}$$

which in turn gives

$$\begin{aligned} \mathcal{S}(5, 3) &= \phi(\mathcal{S}(4, 2)) \cup \phi(\mathcal{S}(2, 2)) \\ &= \{11100, 01110, 00111, 10011\} \cup \{11001\}. \end{aligned}$$

Similarly,

$$\begin{aligned} A_{\text{ICI}}(5, 3) &= A_{\text{ICI}}(4, 2) + A_{\text{ICI}}(2, 2) \\ &= A_{\text{ICI}}(3, 1) + A_{\text{ICI}}(1, 1) + A_{\text{ICI}}(2, 2) \\ &= 5. \end{aligned}$$

In fact, Corollary 4 gives rise to a polynomial time algorithm, via dynamic programming, for computing the value of $A_{\text{ICI}}(n, w)$, for any given n and w. Let A be the $n \times w$ matrix whose (i, j)-th entry, $\mathsf{A}(i, j) = A_{\text{ICI}}(i, j)$. Prefill the first column so that $\mathsf{A}(i, 1) = i$ for all $i \in [\![n]\!]$, and the diagonal entries so that $\mathsf{A}(i, i) = 1$ for all $i \in [\![w]\!]$. Now fill the remaining entries $\mathsf{A}(i, j)$, where $i > j$, column wise from left to right (that is, by increasing value of j), and within each column j from top to bottom (that is, by increasing value of i), until we fill in the entry $\mathsf{A}(n, w)$, which gives the value of $A_{\text{ICI}}(n, w)$.

As the recurrence given by (2) involves $\Theta(n)$ terms and there are $\Theta(nw)$ entries in the matrix to fill, a naive implementation of the above algorithm requires $O(n^2 w)$ arithmetic operations. However, the number of arithmetic operations may be reduced to $O(nw)$ by considering an auxiliary $n \times w$ matrix.

Corollary 5: The set of values $\{A_{\text{ICI}}(i, j) : 1 \leqslant j \leqslant i \leqslant n\}$ can be computed with $O(nw)$ arithmetic operations, or running time $O(n^2 w)$, using $\Theta(nw)$ space.

Proof: Recall that A is the $n \times w$ matrix whose (i, j)-th entry, $\mathsf{A}(i, j) = A_{\text{ICI}}(i, j)$, and our objective is to compute $\mathsf{A}(i, j)$ for $1 \leqslant j \leqslant i \leqslant n$. To reduce the running time, we define an auxiliary $n \times w$ matrix B, whose (i, j)-th entry is $\mathsf{B}(i, j) = \mathsf{A}(i, j) + \mathsf{A}(i-2, j-1)$ for $2 \leqslant j < i \leqslant n$.

For $2 \leqslant j < i \leqslant n$, observe that $\mathsf{A}(i, j) + \mathsf{A}(i-2, j-1) = \sum_{k \in [\![i-j+1]\!]} \mathsf{A}(i-k, j-1) = \mathsf{B}(i-1, j) + \mathsf{A}(i-1, j-1)$. Therefore, we have the following recurrences,

$$\mathsf{B}(i, j) = \mathsf{B}(i-1, j) + \mathsf{A}(i-1, j-1), \tag{3}$$

$$\mathsf{A}(i, j) = \mathsf{B}(i, j) - \mathsf{A}(i-2, j-1). \tag{4}$$

Since the recurrences given by (3) and (4) involve $\Theta(1)$ terms and there are $\Theta(nw)$ entries in the matrices to fill, the required number of arithmetic operations to fill both A and B is $O(nw)$. Since $|\mathsf{A}(n, w)| = 2^{\Theta(n)}$, the running time is $O(n^2 w)$. ■

The building up of codewords in $\mathcal{S}(n, w)$ from shorter codewords in $\mathcal{S}(n-k, w-1)$ via ϕ leads also to an efficient ranking/unranking algorithm for codewords in $\mathcal{S}(n, w)$. We describe this next.

B. Ranking and Unranking $\mathcal{S}(n, w)$

A *ranking function* for a finite set S of cardinality N is a bijection

$$\text{rank}\colon\ S \to [\![N]\!].$$

There is a unique *unranking function* associated with the function rank:

$$\text{unrank}\colon\ [\![N]\!] \to S,$$

so that $\text{rank}(s) = i$ if and only if $\text{unrank}(i) = s$ for all $s \in S$ and $i \in [\![N]\!]$. In this section, we present an algorithm for ranking and unranking $\mathcal{S}(n, w)$.

The basis of our ranking and unranking algorithms is the unfolding of the recurrence

$$\mathcal{S}(n, w) = \bigcup_{k \in [\![n-w+1]\!] \setminus \{2\}} \phi(\mathcal{S}(n-k, w-1)) \tag{5}$$

implied by Theorem 6, which yields a natural total ordering of codewords in $\mathcal{S}(n, w)$, given a total ordering of codewords in $\mathcal{S}(n, 1)$ and $\mathcal{S}(n, n)$. Throughout this paper, the reverse lexicographic order is used as a total ordering on $\mathcal{S}(n, w)$, so that the rank of $\mathsf{u} \in \mathcal{S}(n, 1)$ is the position of the symbol "1" in u. Note that the total ordering on $\mathcal{S}(n, n)$ is trivial since it contains only one element. Let us first illustrate the idea behind the unranking algorithm through an example.

Example 8: Consider $\mathcal{S}(7, 3)$. This code has size 18. Suppose we want to compute unrank(13). First, (5) gives

$$\mathcal{S}(7, 3) = \phi(\mathcal{S}(6, 2)) \cup \phi(\mathcal{S}(4, 2)) \cup \phi(\mathcal{S}(3, 2)) \cup \phi(\mathcal{S}(2, 2)),$$

where the codes in the union on the right hand side are ordered in decreasing length. Now, $|\mathcal{S}(6, 2)| = 11$, $|\mathcal{S}(4, 2)| = 4$, $|\mathcal{S}(3, 2)| = 2$, and $|\mathcal{S}(2, 2)| = 1$. We are interested in the 13th element of $\mathcal{S}(7,3)$. Since $|\mathcal{S}(6,2)| < 13 \le |\mathcal{S}(6,2)|+|\mathcal{S}(4,2)|$, the 13th element of $\mathcal{S}(7,3)$ is the $13-|\mathcal{S}(6,2)| = 2$-nd element of $\phi(\mathcal{S}(4,2))$, which can be obtained from the 2nd element of $\mathcal{S}(4,2)$. Recursing gives

$$\mathcal{S}(4,2) = \phi(\mathcal{S}(3,1)) \cup \phi(\mathcal{S}(1,1)),$$

where $|\mathcal{S}(3,1)| = 3$ and $|\mathcal{S}(1,1)| = 1$.

Hence the second element of $\mathcal{S}(4,2)$ is the second element of $\phi(\mathcal{S}(3,1))$, which can be obtained from the second element of $\mathcal{S}(3,1)$, namely 010. This gives

$$\begin{aligned}\text{unrank}(13) &= \phi^2(010)\\ &= \phi(0110)\\ &= 0110010.\end{aligned}$$

The formal unranking algorithm is described in Algorithm 1 below.

Algorithm 1 $\texttt{unrank}(n, w, m)$

```
Input: Integers $n \ge w \ge 1$ and $1 \le m \le A_{\rm ICI}(n,w)$.
Output: $(\mathsf{u}, r)$, where $\mathsf{u}$ is the codeword of rank $m$ in $\mathcal{S}(n,w)$, and $r$ is the position of the rightmost "1" in $\mathsf{u}$.
  if $w = n$ then
    return $(\mathsf{j}, n)$, where $\mathsf{j}$ is an all-ones vector of length $n$;
  if $w = 1$ then
    return $(\mathsf{e}_m, m)$, where $\mathsf{e}_m$ is a vector of length $n$ with "1" at its $m$th position and zero elsewhere;
  let $k \ge 1$ be such that
      $L = \sum_{i\in[\![k-1]\!]\setminus\{2\}} A_{\rm ICI}(n-i, w-1) < m \le \sum_{i\in[\![k]\!]\setminus\{2\}} A_{\rm ICI}(n-i, w-1);$   (6)
  $(\mathsf{u}, r) = \texttt{unrank}(n-k, w-1, m-L)$;
  return $((\mathsf{u}_1, \ldots, \mathsf{u}_r, \underbrace{0, 0, \ldots, 0}_{k-1\text{ 0's}}, 1, \mathsf{u}_{r+1}, \ldots, \mathsf{u}_{n-k}), r+k)$;
```

The values of $A_{\rm ICI}(n,w)$ required in Algorithm 1 can be precomputed using the dynamic programming method described at the end of the previous subsection.

The corresponding ranking algorithm for $\mathcal{S}(n,w)$ has a similar recursive structure and is described in Algorithm 2.

Example 9: Consider $\mathcal{S}(7,3)$ again. Suppose we want to compute rank$(7,3,0110010)$. First, we look for the rightmost "1" in 0110010 and set $k-1$ to be the number of zeroes preceding it. In other words, $k = 3$ and so,

$$\begin{aligned}\text{rank}(7,3,0110010) &= \text{rank}(4,2,0110) + A_{\rm ICI}(6,2)\\ &= \text{rank}(4,2,0110) + 11.\end{aligned}$$

To compute rank$(4,2,0110)$, we observe that $k = 1$ and we have

$$\text{rank}(4,2,0110) = \text{rank}(3,1,010).$$

Finally, since the weight of 010 is one, we have that rank$(3,1,010) = 2$. Therefore, rank$(7,3,0110010) = 2+11 = 13$, and we recover the rank of 0110010.

Algorithm 2 $\texttt{rank}(n, w, \mathsf{u})$

```
Input: Integers $n \ge w \ge 1$ and $\mathsf{u} \in \mathcal{S}(n,w)$.
Output: $m$, where $m = \text{rank}(\mathsf{u})$.
  if $w = n$ then
    return 1;
  if $w = 1$ then
    return $m$, where $m$ is the position of the rightmost "1" in $u$;
  let $r$ be the starting position of the rightmost substring in $\mathsf{u}$ of the form $(1, \underbrace{0, 0, \ldots, 0}_{k-1\text{ 0's}}, 1)$, where $k \ge 1$;
  $\mathsf{v} \leftarrow (\mathsf{u}_1, \mathsf{u}_2, \ldots, \mathsf{u}_r, \mathsf{u}_{r+k+1}, \mathsf{u}_{r+k+2}, \ldots, \mathsf{u}_n)$;
  return $\texttt{rank}(n-k, w-1, \mathsf{v}) + \sum_{i\in[\![k-1]\!]\setminus\{2\}} A_{\rm ICI}(n-i, w-1)$;
```

In summary, suppose that we are given a set of $A_{\rm ICI}(n,w)$ messages. To encode the mth message, we compute $\mathsf{u} = \text{unrank}(n,w,m)$ as described in Algorithm 1 and set u to be the codeword. On the other hand, to decode an ICI-free codeword u of length n and weight w, we compute $m = \text{rank}(n,w,\mathsf{u})$ and decode u to the mth message.

It remains to determine the running times of Algorithms 1 and 2. Here, we fix $0 < p < 1$ and set $w = \lfloor pn \rfloor$, and let $m \in [\![A_{\rm ICI}]\!]$. Since the routine rank recursively calls upon itself with a smaller weight, the routine rank(n,w,m) requires at most $w = \Theta(n)$ recursive calls. However, computing k in (6) requires $k = O(n)$ arithmetic operations and hence, this simple analysis implies that Algorithm 1 requires $O(n^2)$ arithmetic operations. Nevertheless, a refined analysis shows that Algorithm 1 requires $O(n)$ arithmetic operations. Similarly, we can show that Algorithm 2 can be computed with $O(n)$ arithmetic operations.

Theorem 7: Fix $0 < p < 1$ and let $w = \lfloor pn \rfloor$. Suppose that the set of values $\{A_{\rm ICI}(i,j) : 1 \leqslant j \leqslant i \leqslant n\}$ has been precomputed. Then Algorithms 1 and 2 can be computed with $O(n)$ arithmetic operations, or running time $O(n^2)$.

Proof: Suppose that unrank(n,w,m) makes J recursive calls. Specifically, for $j \in [\![J]\!]$, let k_j and L_j be the variables k and L in (6) that are computed by jth recursive call. In other words, for the Jth iteration, we make the recursive call to unrank $\left(n - \sum_{j\in[\![J]\!]} k_j, w-J, m - \sum_{j\in[\![J]\!]} L_j\right)$ and therefore, $\sum_{j\in[\![J]\!]} k_j \leqslant n$.

Now, in the jth recursive call, we require Ck_j arithmetic operations to compute k_j and L_j for some constant C. Hence, the total number of arithmetic operations required to compute unrank(n,w,m) is $\sum_{j\in[\![J]\!]} Ck_j \leqslant Cn$. In other words, Algorithm 1 requires $O(n)$ arithmetic operations. Since $|A_{\rm ICI}(n,w)| = 2^{\Theta(n)}$, the running time of Algorithm 1 is $O(n^2)$.

The running time analysis for Algorithm 2 is similar. Suppose that rank(n,w,u) makes J recursive calls. For $j \in [\![J]\!]$, let k_j be the variable k that is computed by jth recursive call, and we similarly verify that $\sum_{j\in[\![J]\!]} k_j \leqslant n$.

Since we require Dk_j arithmetic operations in in the jth recursive call for some constant D, the total number of arithmetic operations required to compute rank(n,w,m) is

$\sum_{j\in[\![J]\!]} Dk_j \leqslant Dn$. In other words, Algorithm 2 requires $O(n)$ arithmetic operations. Again, since $|A_{\rm ICI}(n,w)| = 2^{\Theta(n)}$, the running time of Algorithm 2 is $O(n^2)$. ■

C. Extension to $q > 2$

Let $\mathcal{C} \subseteq \Sigma^n$ and $\Omega \subseteq \Sigma$. Let $f : \Sigma \to \Omega$. By canonical extension, we have $f : \Sigma^n \to \Omega^n$, so that

$$\Sigma^n \ni (u_1, u_2, \ldots, u_n) \overset{f}{\mapsto} (f(u_1), f(u_2), \ldots, f(u_n)) \in \Omega^n.$$

The *restriction* of $\mathcal{C}$ by f is the code $f(\mathcal{C}) \subseteq \Omega^n$.

The idea behind the extension of our results in the previous section for binary codes to q-ary codes is based on the simple observation that if a q-ary code is ICI-free, then its restriction by $f : \Sigma \to \{0, 1\}$, where

$$f(\sigma) = \begin{cases} 1, & \text{if } \sigma = q-1, \\ 0, & \text{otherwise,} \end{cases}$$

is a binary ICI-free code. Hence, a binary ICI-free code $\mathcal{C} \subseteq \{0,1\}^n$ can be used as a template to construct a q-ary ICI-free code $\mathcal{C}' \subseteq \Sigma^n$: for each codeword $\mathsf{u} \in \mathcal{C}$, replace a coordinate with symbol "1" by $q-1$ and replace a coordinate with symbol "0" by all possible symbols from $\Sigma \setminus \{q-1\}$. Therefore, a binary codeword of weight w in $\mathcal{C}$ generates $(q-1)^{n-w}$ codewords in $\mathcal{C}'$.

We are concerned here with q-ary ICI-free codes of constant composition $[w_0, w_1, \ldots, w_{q-1}]$, where $w_0 = w_1 = \cdots = w_{q-2}$. We call codes of such composition *almost balanced*. The intuition behind this condition is that an ICI-free code avoids substrings of the form $q-1, \sigma, q-1$, for all $\sigma \in \Sigma \setminus \{q-1\}$, and so the symbol $q-1$ has a special status. Therefore, if we were to look for a constant-composition ICI-free code of maximum size, it would be a good strategy to look within almost balanced codes. Indeed, we have found ICI channel capacity-achieving ICI-free codes that are almost balanced (presented in Section III).

To construct an almost balanced ICI-free code $\mathcal{C} \subseteq \Sigma^n$ of constant composition $[w_0, w_1, \ldots, w_{q-1}]$, we can start with $\mathcal{S}(n, w_{q-1})$ as a template and replace every occurrence of symbol "1" in each codeword $u \in \mathcal{S}(n, w_{q-1})$ by $q-1$. However, instead of replacing the remaining $n - w_{q-1}$ "0"s in u with all possible words in $(\Sigma \setminus \{q-1\})^{n-w_{q-1}}$, we replace them with codewords from a balanced $(q-1)$-ary code of length $n - w_{q-1}$ over $\Sigma \setminus \{q-1\}$.

Efficient encoder/decoder pairs for capacity-achieving balanced q-ary codes are known [23], [24]. We can combine the encoder/decoder for $\mathcal{S}(n, w)$ and that for a capacity-achieving balanced $(q-1)$-ary code $\mathcal{B}$ of length $n-w$ to give an efficient encoder/decoder for an almost balanced q-ary ICI-free code $\mathcal{C}$. The encoding algorithm is described in Algorithm 3.

The corresponding decoding algorithm is given in Algorithm 4.

From Theorem 7, it is not hard to see that the running time for both algorithms is $O(n^2)$, assuming the values of $A_{\rm ICI}$ have been precomputed.

Algorithm 3 `encode`(m)

```
Input: $0 \le m < |\mathcal{S}(n,w)| \cdot |\mathcal{B}|$.
Output: u, where u is an encoding of $m$ as a codeword in $\mathcal{C}$.
  let $m = s \cdot |\mathcal{B}| + t$, where $0 \le t < |\mathcal{B}|$;
  u ← encoding of $s$ as a codeword in $\mathcal{S}(n,w)$;
  v ← encoding of $t$ as a codeword in $\mathcal{B}$;
  w ← word obtained by replacing each occurrence of symbol "1" in $u$ by $q-1$ and all the other $n-w$ "0"s in u by the word v;
  return w;
```

Algorithm 4 `decode`(u)

```
Input: u ∈ $\mathcal{C}$.
Output: $m$, where u = encode($m$).
  v ← word obtained from u by deleting occurrences of symbol $q-1$;
  $t$ ← decoding of v ∈ $\mathcal{B}$;
  w ← word obtained from u by replacing each occurrence of symbol $q-1$ in u by "1" and all the other symbols by "0";
  $s$ ← decoding of w ∈ $\mathcal{S}(n,w)$;
  return $s \cdot |\mathcal{B}| + t$;
```

D. Application to the Case $q = 4$

Using Perron-Frobenius theory, the capacity of q-ary ICI channels can be determined to be $\log_2 \lambda$, where λ is the largest root of $x^3 - qx^2 + (q-1)x - (q-1)^2$ (see [20], [28]). This gives $C_{\rm ICI}(4) \approx 1.9374$. Taranalli *et al.* [10] gave an encoding/decoding algorithm for quaternary ICI-free codes that has rate 1.6942.

We have constructed almost balanced quaternary ICI-free codes of composition $[\alpha n, \alpha n, \alpha n, \beta n]$, where $\alpha \approx 0.268582$ and $\beta \approx 0.194254$, and showed them to be capacity-achieving (having rate 1.9374) in Section IV. These codes can be encoded and decoded with the algorithms described earlier in this section. Hence, we now have efficient encoding/decoding algorithms for quaternary constant-composition ICI-free codes that are capacity-achieving.

V. Conclusion

We enumerated the set of all $\mathcal{F}$-avoiding words with a fixed composition for certain avoidance sets $\mathcal{F} = \mathcal{I}(a,b)$. Using this formula, we presented procedures to determine the rates of $\mathcal{I}(a,b)$-avoiding codes with fixed composition ratios. We also determined the optimal composition ratios that maximize the rates of $\mathcal{I}(a,b)$-avoiding constant-composition codes and showed that these codes achieve the capacity of the $\mathcal{I}(a,b)$-avoiding channel. Efficient encoding and decoding algorithms for certain special classes of constant-composition $\mathcal{I}_1(q)$-avoiding codes are presented.

Acknowledgement

The authors are also grateful to Ron Roth and Paul H. Siegel for the helpful discussions and sharing the preprint [27]. They would like to thank Bei Xiaohui for his insightful comments in running time analysis. The authors thank the editor and the reviewers for their constructive feedback and helpful suggestions.

References

[1] Y. M. Chee, J. Chrisnata, H. M. Kiah, S. Ling, T. T. Nguyen, and V. K. Vu, "Rates of constant-composition codes that mitigate intercell interference," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 200–204.

[2] Y. M. Chee, C. Johan, H. M. Kiah, S. Ling, T. T. Nguyen, and V. K. Vu, "Efficient encoding/decoding of capacity-achieving constant-composition ICI-free codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 205–209.

[3] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on NAND flash memory cell operation," *IEEE Electron Device Lett.*, vol. 23, no. 5, pp. 264–266, May 2002.

[4] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, May 2012, pp. 521–526.

[5] D. Kang *et al.*, "The air spacer technology for improving the cell distribution in 1 giga bit NAND flash memory," in *Proc. 21st IEEE Non-Volatile Semiconductor Memory Workshop*, Feb. 2006, pp. 36–37.

[6] R. Fastow and S. Park, "Minimization of FG-FG coupling in flash memory," U.S. Patent 6 996 004 B1, Feb. 7, 2006.

[7] A. Berman and Y. Birk, "Mitigating inter-cell coupling effects in MLC NAND flash via constrained coding," in *Proc. Flash Memory Summit*, 2010.

[8] A. Berman and Y. Birk, "Error correction scheme for constrained inter-cell interference in flash memory," in *Proc. 2nd Annu. Non-Volatile Memories Workshop (NVMW)*, 2011, pp. 1–20.

[9] A. Berman and Y. Birk, "Constrained flash memory programming," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 2128–2132.

[10] V. Taranalli, H. Uchikawa, and P. H. Siegel, "Error analysis and inter-cell interference mitigation in multi-level cell flash memories," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 271–276.

[11] S. Buzaglo, P. H. Siegel, and E. Yaakobi, "Coding schemes for inter-cell interference in flash memory," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1736–1740.

[12] S. Buzaglo and P. H. Siegel, "Row-by-row coding schemes for inter-cell interference in flash memory," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4101–4113, Oct. 2017.

[13] O. Torii and P. H. Siegel, "Novel ICI-mitigation Codes for MLC NAND Flash Memory," *Presented in 8th Annu. Non-Volatile Memories Workshop (NVMW)*, Mar. 2017.

[14] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for asymmetric limited-magnitude errors with application to multilevel flash memories," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1582–1596, Apr. 2010.

[15] E. Yaakobi, P. H. Siegel, A. Vardy, and J. K. Wolf, "On codes that correct asymmetric errors with graded magnitude distribution," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 1056–1060.

[16] H. Zhou, A. Jiang, and J. Bruck, "Error-correcting schemes with dynamic thresholds in nonvolatile memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 2143–2147.

[17] F. Sala, R. Gabrys, and L. Dolecek, "Dynamic threshold schemes for multi-level non-volatile memories," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2624–2634, Jul. 2013.

[18] M. Qin, E. Yaakobi, and P. H. Siegel, "Constrained codes that mitigate inter-cell interference in read/write cycles for flash memories," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 836–846, May 2014.

[19] S. Kayser and P. H. Siegel, "Constructions for constant-weight ICI-free codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun./Jul. 2014, pp. 1431–1435.

[20] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Denver, CO, USA: Shannon Foundation, 2004.

[21] P. H. Siegel, "Constrained codes for multilevel flash memory," North Amer. School Inf. Theory, Tech. Rep., Aug. 2015.

[22] S. Heubach and T. Mansour, *Combinatorics of Compositions and Words*. Boca Raton, FL, USA: CRC Press, 2009.

[23] T. G. Swart and J. H. Weber, "Efficient balancing of q-ary sequences with parallel decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2009, pp. 1564–1568.

[24] T. G. Swart and K. A. S. Immink, "Prefixless q-ary balanced codes with ECC," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2013, pp. 1–5.

[25] B. H. Marcus and R. M. Roth, "Improved Gilbert-Varshamov bound for constrained systems," *IEEE Trans. Inf. Theory*, vol. 38, no. 4, pp. 1213–1221, Jul. 1992.

[26] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. Springer, 2010.

[27] R. Roth and P. H. Siegel, "A Markov chain approach to computing the capacity of ICI-free balanced codes," Tech. Rep.

[28] B. H. Marcus, R. M. Roth, and P. H. Siegel, *An Introduction to Coding for Constrained Systems*, 5th ed., 2001.

Yeow Meng Chee (SM'08) received the B.Math. degree in computer science and combinatorics and optimization and the M.Math. and Ph.D. degrees in computer science from the University of Waterloo, Waterloo, ON, Canada, in 1988, 1989, and 1996, respectively.

Currently, he is a Professor at the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Prior to this, he was Program Director of Interactive Digital Media R&D in the Media Development Authority of Singapore, Postdoctoral Fellow at the University of Waterloo and IBMs Zürich Research Laboratory, General Manager of the Singapore Computer Emergency Response Team, and Deputy Director of Strategic Programs at the Infocomm Development Authority, Singapore.

His research interest lies in the interplay between combinatorics and computer science/engineering, particularly combinatorial design theory, coding theory, extremal set systems, and electronic design automation.

Johan Chrisnata received his Bachelor degree in mathematics from Nanyang Technological University (NTU), Singapore in 2015. From August 2015 until August 2018, he was a research officer in NTU. Currently he is pursuing a Ph.D. degree in mathematics from School of Physical and Mathematical Sciences at Nanyang Technological University, Singapore. His research interest includes enumerative combinatorics and coding theory.

Han Mao Kiah received his Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2014. From 2014 to 2015, he was a Postdoctoral Research Associate at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 2015 to 2018, he was a Lecturer at the School of Physical and Mathematical Sciences (SPMS), NTU, Singapore. Currently he is an Assistant Professor at SPMS, NTU, Singapore. His research interests include combinatorial design theory, coding theory, and enumerative combinatorics.

San Ling received the B.A. degree in mathematics from the University of Cambridge and the Ph.D. degree in mathematics from the University of California, Berkeley.

Since April 2005, he has been a Professor with the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, in the Nanyang Technological University, Singapore. Prior to that, he was with the Department of Mathematics, National University of Singapore. His research fields include: arithmetic of modular curves and application of number theory to combinatorial designs, coding theory, cryptography and sequences.

Tuan Thanh Nguyen received his B.Sc. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2014. Currently he is pursuing the Ph.D. degree in mathematics from School of Physical and Mathematical Sciences at Nanyang Technological University, Singapore. His research interests include combinatorics, coding theory, and codes for DNA-based data storage.

Van Khu Vu received his B.Sc. degree in mathematics from Vietnam National University (VNU), Hanoi, in 2010 and the Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore in 2018. From 2010 to 2012, he was a lecturer at VNU College of Sciences, Hanoi. Currently, he is a Research Fellow at School of Physical and Mathematical Sciences, NTU, Singapore. His primary research interests lies in the areas of algorithms, combinatorics and coding theory.