On Bi-Modal Constrained Coding

Ron M. Roth, *Fellow, IEEE* Paul H. Siegel, *Life Fellow, IEEE*

***Abstract*—Bi-modal (respectively, multi-modal) constrained coding refers to an encoding model whereby a user input block can be mapped to two (respectively, multiple) codewords. In current storage applications, such as optical disks, multi-modal coding allows to achieve DC control, in addition to satisfying the runlength limited (RLL) constraint specified by the recording channel. In this work, a study is initiated on bi-modal fixed-length constrained encoders. Necessary and sufficient conditions are presented for the existence of such encoders for a given constraint. It is also shown that under somewhat stronger conditions, one can guarantee a bi-modal encoder with finite decoding delay.**

***Index Terms*—Approximate eigenvectors, Bi-modal encoders, Constrained codes, Multi-modal encoders, Parity-preserving encoders.**

I. Introduction

Runlength limited (RLL) coding is widely employed in magnetic and optical storage in order to mitigate the effects of inter-symbol interference and clock drifting [9]. The encoder typically takes the form of a finite-state machine, which maps a sequence of input p-bit blocks into a sequence of output q-bit codewords, so that the concatenation of the generated codewords satisfies the RLL constraint. In most applications, the coding scheme also provides DC control (or, more generally, suppression of the low frequencies). This is achieved by *bi-modal* (respectively, *multi-modal*) *encoding*, allowing some (or all) input p-blocks to be mapped by the encoder to two (respectively, multiple) codewords, and then, during the encoding process, selecting the codeword that yields the best DC suppression. In one implementation of this strategy, the input sequence is sub-divided into non-overlapping windows (each consisting of one or more p-blocks), and each window can be mapped by the encoder to two output sequences over $\{0,1\}$ that have different parities (i.e., modulo-2 sums). The generated output binary sequence is then transformed ("precoded") into sequences over the bipolar alphabet $\{+1,-1\}$, with the binary 1s corresponding to positions of the bipolar sign changes [9, p. 52]. DC control is achieved by selecting the output sequence that minimizes the DC contents [14, p. 29]. A *parity-preserving* encoder is one embodiment of this approach and is part of the Blu-ray standard. In such an encoder, the parity of the input sub-sequence within each window is preserved at the generated output sequence; one bit in each window is then reserved to set the parity of (the input window and) the output sequence, thereby controlling the DC contents [9, §11.4.3], [11], [15], [16], [17], [19]. Another implementation of DC control via multi-modal encoding maps the input sequence (without precoding) into bipolar sequences with different *disparities* (i.e., different signs of their sums of entries). Guided scrambling, which is a known paradigm for DC control enhancement (see, for example, [10]), can be combined with multi-modal encoding [9, Ch. 10].

Most constructions so far of multi-modal encoders and, in particular, of parity-preserving encoders, were obtained by ad-hoc methods. The purpose of this work is to initiate a study of bi-modal (and multi-modal) encoders, starting with the special case of fixed-length finite-state encoders. We provide a formal definition of our setting in Section I-B below, following a summary (in Section I-A) of some background and definitions which are taken from [14]. The main result of the paper, which we state in Section I-C, is a necessary and sufficient condition for the existence of a bi-modal encoder. We prove the necessity part in Section II, followed by a construction method of bi-modal encoders in Section III, thereby establishing sufficiency. The more general variable-length model is a subject of future work [18] and is briefly discussed in Section IV.

A. Background

A (finite labeled directed) graph is a graph $G=(V,E,L)$ with a nonempty finite state (vertex) set $V=V(G)$, finite edge set $E=E(G)$, and edge labeling $L:E\to\Sigma$. The constraint presented by G, denoted $S(G)$, is the set of words over Σ that are generated by finite paths in G. The set of words that are generated by finite paths that start at a given state $u\in V(G)$ is called the follower set of u and is denoted $\mathcal{F}_G(u)$.

A graph G is deterministic if all outgoing edges from a state are distinctly labeled. Every constraint has a deterministic presentation. A graph G is lossless if no two paths with the same initial state and the same terminal state generate the same word. The anticipation $\mathcal{A}(G)$ of G is the smallest nonnegative integer a (if any) such that all paths that generate any given word of length $a+1$ from any given state in G share the same first edge (thus, the anticipation of deterministic graphs is 0; generally, having finite anticipation is a stronger property than losslessness). A graph is (m,a)-definite if all paths that generate a given word of length $m+a+1$ share the same $(m+1)$st edge. A graph is said to have finite memory μ if μ is the smallest nonnegative integer (if any) such that all paths of length μ that generate the same word terminate in the same state.

This work was supported by Grants 2015816 and 2018048 from the United-States–Israel Binational Science Foundation (BSF), by NSF Grant CCF-BSF-1619053, and by Grant 1396/16 from the Israel Science Foundation. This work, under the title "On parity-preserving constrained coding," was presented in part at the IEEE Int'l Symposium on Information Theory, Vail, Colorado (June 2018).

Ron M. Roth is with the Computer Science Department, Technion, Haifa 320003, Israel. This work was done in part while R.M. Roth was visiting the Center for Memory and Recording Research (CMRR), UC San Diego. Email: ronny@cs.technion.ac.il

Paul H. Siegel is with the Department of Electrical and Computer Engineering and the Center for Memory and Recording Research (CMRR), University of California at San Diego, La Jolla, CA 92093 USA. Email: psiegel@ucsd.edu

A graph G is irreducible if it is strongly-connected. Every graph decomposes uniquely into irreducible components, and at least one such component must be an irreducible sink (i.e., with no outgoing edges to another component). The period of an irreducible graph (with at least one edge) is the greatest common divisor of the lengths of its cycles, and such a graph is primitive if its period is 1. A constraint S is irreducible if it can be presented by a deterministic irreducible graph (in which case the smallest such presentation is unique).

The power G^t of a graph G is the graph with the same set of states $V(G)$ and edges that are the paths of length t in G; the label of an edge in G^t is the length-t word generated by the path. For $S = S(G)$ the power S^t is defined as $S(G^t)$.

Given a constraint S over an alphabet Σ and a lossless presentation G of S, the capacity of S is defined by $\mathsf{cap}(S) = \lim_{\ell \to \infty} (1/\ell) \log_2 |S \cap \Sigma^\ell|$. The limit indeed exists, and it is known that $\mathsf{cap}(S) = \log_2 \lambda(A_G)$ where $\lambda(A_G)$ denotes the spectral radius (Perron eigenvalue) of the adjacency matrix A_G.

Given a constraint S and a nonnegative[1] integer n, an (S, n)-encoder is a lossless graph $\mathcal{E}$ such that $S(\mathcal{E}) \subseteq S$ and each state has out-degree n. An (S, n)-encoder exists if and only if $\log_2 n \le \mathsf{cap}(S)$. In a *tagged* (S, n)-encoder, each edge is assigned an input tag from a finite alphabet of size n, such that edges outgoing from the same state have distinct tags. The anticipation (if finite) of an encoder determines its decoding delay: given the current encoder state, the anticipation specifies how far one needs to look-ahead at a sequence generated from that state in order to recover the first edge (and, thus, the tag of that edge) in the encoder path that generated that sequence. A tagged encoder is (m, a)-sliding-block decodable if all paths that generate a given word of length $m+a+1$ share the same tag on their $(m+1)$st edges (thus, an (m, a)-definite encoder is (m, a)-sliding-block decodable for any tagging of its edges).

A (tagged) rate $p : q$ encoder for a constraint S is a (tagged) $(S^q, 2^p)$-encoder (the tags are then assumed to be from $\{0, 1\}^p$). A rate $p : q$ parity-preserving encoder for a constraint S over $\Sigma = \{0, 1\}$ is a tagged encoder for S in which the parity of the (length-q) label of each edge matches the parity of the (length-p) tag that is assigned to the edge (see also Section I-B below).

Given a square nonnegative integer matrix A and a positive integer n, an (A, n)-approximate eigenvector is a nonnegative nonzero integer vector $\boldsymbol{x}$ that satisfies the inequality $A\boldsymbol{x} \ge n\boldsymbol{x}$ componentwise. The set of all (A, n)-approximate eigenvectors will be denoted by $\mathcal{X}(A, n)$, and it is known that $\mathcal{X}(A, n) \ne \emptyset$ if and only if $n \le \lambda(A)$. Given a constraint S presented by a deterministic graph G and a positive integer n, the state-splitting algorithm provides a method for transforming G, through an (A_G, n)-approximate eigenvector, into an (S, n)-encoder with finite anticipation.

For a positive integer b, the set $\{0, 1, 2, \ldots, b-1\}$ will be denoted by $[b\rangle$.

[1]In all practical cases n needs to be strictly positive. Yet for the purpose of simplifying the wording of some results in the sequel, we find it convenient to allow n formally to be zero.

B. *Bi-modal encoders and parity-preserving encoders*

Let S be a constraint over an alphabet Σ, and fix a partition $\{\Sigma_0, \Sigma_1\}$ of Σ. The symbols in Σ_0 (respectively, Σ_1) will be referred to as the *even* (respectively, *odd*) symbols of Σ. The case where one of the partition elements is empty will turn out to be uninterestingly trivial, so we assume that Σ_0 and Σ_1 are both nonempty. The partition of Σ to two elements (only) and the choice of the terms even and odd follow from the above-mentioned parity-based DC control approach of constructing bi-modal encoders where an input p-bit block is mapped into two q-bit codewords with different parities. However, without much further effort, the definitions and the results can be extended to the multi-modal case, where Σ is partitioned into any number of partition elements.[2]

Given a graph H with labeling in Σ, for $\mathsf{b} \in [2\rangle$, we denote by H_{b} the subgraph of H containing only the edges with labels in Σ_{b}.

Let $S = S(G)$ be a constraint and n_0 and n_1 be nonnegative integers. An (S, n_0, n_1)-*encoder* $\mathcal{E}$ is an (S, n_0+n_1)-encoder such that for each $\mathsf{b} \in [2\rangle$, the subgraph $\mathcal{E}_{\mathsf{b}}$ is an (S, n_{b})-encoder. In other words, from each state in $\mathcal{E}$, there are n_0 outgoing edges with even labels and n_1 outgoing edges with odd labels. Thus, in the parity-based DC control scheme, a rate $p : q$ bi-modal encoder for a constraint S over the binary alphabet is an $(S^q, 2^{p-1}, 2^{p-1})$-encoder, where we partition the alphabet $S \cap \{0, 1\}^q$ of S^q into even and odd symbols according to their ordinary parity. In particular, a rate $p : q$ parity-preserving encoder for S is a (tagged) $(S^q, 2^{p-1}, 2^{p-1})$-encoder and, conversely, any $(S^q, 2^{p-1}, 2^{p-1})$-encoder can be tagged so that it is parity preserving. In the disparity-based DC control scheme, the constraint S is over the bipolar alphabet and we partition the set of length-q words of S according to the sign of their disparity (zero-disparity words can be treated arbitrarily, but see Footnote 2). In general, by a rate $p : q$ bi-modal encoder for a constraint S we hereafter mean an $(S^q, 2^{p-1}, 2^{p-1})$-encoder with respect to a prescribed partition of the set of length-q words of S.

Example 1. Let S be the constraint over $\Sigma = \{a, b, c, d\}$ which is presented by the graph $\mathcal{E}$ in Figure 1. The graph $\mathcal{E}$ is actually a deterministic $(S, 4)$-encoder. Moreover, it is an $(S, 2, 2)$ encoder if we assume the partition $\Sigma_0 = \{a, b\}$ and $\Sigma_1 = \{c, d\}$: the $(S, 2)$-encoders $\mathcal{E}_0$ and $\mathcal{E}_1$ are shown in Figures 2 and 3, respectively. $\square$

When studying $(S(G), n_0, n_1)$-encoders, there is no loss of generality in assuming that both G and the encoder are irreducible. Indeed, if $\mathcal{E}$ is an $(S(G), n_0, n_1)$-encoder, then an irreducible sink of $\mathcal{E}$ is an (S', n_0, n_1)-encoder, where S' is an irreducible constraint presented by some irreducible component of G (see [14, Lemma 2.9]). Note that G_0, G_1, $\mathcal{E}_0$, and $\mathcal{E}_1$ may still be reducible even when G and $\mathcal{E}$ are

[2]Moreover, we will also consider in Section III-D the extension where Σ_0 and Σ_1 are not necessarily disjoint (yet still cover all the elements of Σ), as this extension fits better the disparity-based DC control scheme: in this case, Σ_0 and Σ_1 represent sets of bipolar q-blocks of non-negative and non-positive disparities, respectively and, so, they may intersect on q-blocks with zero disparity.

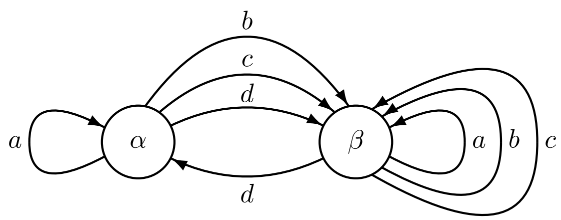

Fig. 1. Graph $\mathcal{E}$ for Example 1.

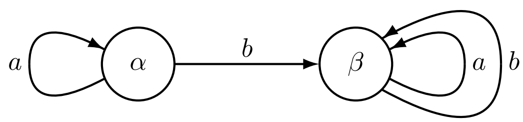

Fig. 2. Graph $\mathcal{E}_0$ for Example 1.

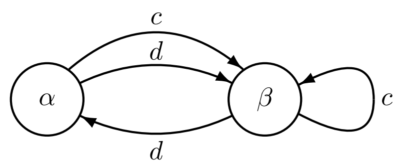

Fig. 3. Graph $\mathcal{E}_1$ for Example 1.

irreducible (e.g., in Example 1, the graph $\mathcal{E}$ is irreducible, while $\mathcal{E}_0$ is not).

C. Statement of main result

Next is a statement of our main result.

Theorem 1. *Let S be an irreducible constraint, presented by an irreducible deterministic graph G, and let n_0 and n_1 be positive integers. Then there exists an (S, n_0, n_1)-encoder if and only if $\mathcal{X}(A_{G_0}, n_0) \cap \mathcal{X}(A_{G_1}, n_1) \neq \emptyset$.*

We prove Theorem 1 in Sections II (necessity) and III (sufficiency). Hereafter, we use the notation $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)$ for the intersection $\mathcal{X}(A_{G_0}, n_0) \cap \mathcal{X}(A_{G_1}, n_1)$.

In view of Theorem 1, finding the possible pairs (n_0, n_1) for which an $(S(G), n_0, n_1)$-encoder exists for a given G and partition $\{\Sigma_0, \Sigma_1\}$ requires a method for deciding whether $\mathcal{X}(A_{G_0}, n_0)$ and $\mathcal{X}(A_{G_1}, n_1)$ share common vectors. This decision problem can be recast as a linear programming problem, namely, deciding whether there is a *real* vector $\boldsymbol{x}$ that satisfies the following constraints:

$$\begin{array}{rcl} (A_{G_0} - n_0 I)\, \boldsymbol{x} & \geq & \mathbf{0} \\ (A_{G_1} - n_1 I)\, \boldsymbol{x} & \geq & \mathbf{0} \\ \boldsymbol{x} & \geq & \mathbf{0} \\ \mathbf{1}^\top \cdot \boldsymbol{x} & = & 1 \,, \end{array} \tag{1}$$

where $\mathbf{0}$ and $\mathbf{1}$ stand for the all-zero and all-1 column vectors and $(\cdot)^\top$ denotes transposition. Since all the coefficients in (1) are integers, if there is a real feasible solution $\boldsymbol{x}$ then there is also a rational solution, and, therefore, there is a nonzero integer solution that satisfies the (first) three inequalities in (1). There are known polynomial-time algorithms for solving linear programming problems, such as Karmarkar's algorithm [12], but it would be interesting to find a more direct method, tailored specifically to the constraints (1), for determining whether $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)$ is nonempty (see also the modified Franaszek algorithm in Figure 6 in the sequel). In comparison, recall that in the context of ordinary (S, n)-encoders, the question of interest is whether $\mathcal{X}(A_G, n)$ is nonempty, which, in turn, is equivalent to asking whether $n \leq \lambda(A_G)$.

D. Going to powers of the constraint

Next, we discuss the effect of going to powers of a constraint, namely, attempting to construct (S^t, n_0, n_1)-encoders, for increasing values of t. To this end, we first need to define the even and odd symbols in Σ^t, which is the alphabet[3] of S^t, given a partition $\{\Sigma_0, \Sigma_1\}$ of Σ. We adopt a definition that fits the parity-based scheme for DC control: we say that $\boldsymbol{w} \in \Sigma^t$ is even (respectively, odd), if it contains an even (respectively, odd) number of symbols from Σ_1 (i.e., the parity of $\boldsymbol{w}$ is the modulo-2 sum of the parities of the symbols in $\boldsymbol{w}$). The set of even (respectively, odd) words in Σ^t will be denoted by $(\Sigma^t)_0$ (respectively, $(\Sigma^t)_1$).

It turns out that in most cases, we can approach the capacity of S with bi-modal encoders if we let t increase. Note, however, that such an increase may sometimes be *necessary*, even when the capacity of S is $\log_2(n_0{+}n_1)$ (see Example 2 below). This presents a distinction between bi-modal encoders and ordinary ones: when $\mathsf{cap}(S) = \log_2 n$, capacity is always attained by ordinary encoders already for $t = 1$.

Specifically, we have the following result.

Theorem 2. *Let G be a deterministic primitive graph, having at least one edge with an even label and one edge with an odd label. Then there exists an infinite sequence of nonnegative integers $n^{(1)}, n^{(2)}, \cdots$ such that $(S(G^t), n^{(t)}, n^{(t)})$-encoders exist and*

$$\lim_{t \to \infty} (\log_2 n^{(t)})/t = \mathsf{cap}(S(G)) \,.$$

Proof. By assumption, there are edges $e_0 = v_0 \xrightarrow{a} v_0'$ and $e_1 = v_1 \xrightarrow{b} v_1'$ in G such that a is even and b is odd (v_0 and v_1 may be the same state, and so may v_0' and v_1'). Also, since G is primitive, there exists a real $\kappa > 0$ such that $(A_G^\ell)_{u,v} > \kappa \cdot \lambda^\ell$ for any sufficiently large ℓ and for every $u, v \in V(G)$, where $\lambda = \lambda(A_G)$ (see [14, §3.3.4]).

Fix u^* to be some G-state. We show that there are two paths in G of the same length ℓ^* that start at u^* and have different parities. Let ℓ be sufficiently large such that $A_G^\ell > 0$ (componentwise), and consider all paths of length ℓ starting at u^*. If two of them have different parities, we are done, with ℓ^* taken as ℓ. Otherwise, for $\mathsf{b} \in [2\rangle$, let π_{b} be a path of length ℓ in G that starts at u^* and terminates in v_{b}. The paths $\pi_0 e_0$ and $\pi_1 e_1$, of length $\ell^* = \ell{+}1$, then have different parities.

Now, if $t{-}\ell^*$ is sufficiently large, there are at least $n^{(t)} = \lceil \kappa \cdot \lambda^{t-\ell^*} \rceil$ paths of length $t{-}\ell^*$ from any state $u \in V(G)$ to u^*. It follows that from any state $u \in V(G)$, there are at least $n^{(t)}$ paths of length t that generate even words,

[3]The *effective* alphabet of S^t is the subset $S \cap \Sigma^t$ of Σ^t.

and at least $n^{(t)}$ paths of that length that generate odd words. This means that G^t contains a subgraph which is an $(S(G^t), n^{(t)}, n^{(t)})$-encoder. The result follows by noticing that $\lim_{t\to\infty}(\log_2 n^{(t)})/t = \log_2 \lambda = \mathsf{cap}(S(G))$. $\square$

Remark 1. The requirement in Theorem 2 of having both even and odd labels in G is necessary. Indeed, if all the labels in G were even, then so would have to be all the labels in the encoder, implying that $n^{(t)} = 0$. Similarly, if all the labels in G were odd, then all the labels in the encoder would have to be even (respectively, odd) for even (respectively, odd) t, again implying that $n^{(t)} = 0$. $\square$

For a deterministic graph G and a positive integer t, we denote by $n_{\max}(G, t)$ the largest integer n for which there exist $(S(G^t), n, n)$-encoders, and define the largest possible coding ratio attainable by such encoders by

$$\rho(G,t) = \frac{\log_2 (2\, n_{\max}(G,t))}{t} \; . \tag{2}$$

Example 2. Let G be the graph in Figure 4, where

$$\Sigma_0 = \{a, b\} \quad \text{and} \quad \Sigma_1 = \{c, d\} \; . \tag{3}$$

We have $\lambda(A_G) = 2$, and the matrices A_{G_0} and A_{G_1} are

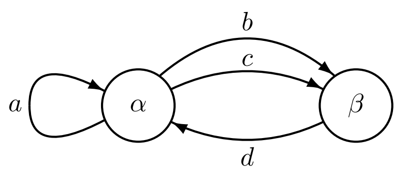

Fig. 4. Graph G for Example 2.

given by

$$A_{G_0} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad A_{G_1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \; .$$

Letting $S = S(G)$, we find conditions under which there exist (S^t, n, n)-encoders, for various values of t. Starting with $t = 1$, in this case, $\mathcal{X}(A_{G_0}, A_{G_1}, n, n) = \emptyset$ for every $n > 0$. Indeed, when $n > 0$, any $\boldsymbol{x} = (x_\alpha \; x_\beta)^\top \in \mathcal{X}(A_{G_0}, n)$ must have $x_\beta = 0$ (since β has no outgoing edges in G_0), but then $A_{G_1}\boldsymbol{x} \geq n\,\boldsymbol{x}$ implies that $x_\alpha = 0$. Hence, $n_{\max}(G, 1) = 0$ and $\rho(G, 1) = -\infty$.

Next, we turn to the second power of G. Here $S \cap (\Sigma^2)_0 = \{aa, ab, cd, dc\}$, $S \cap (\Sigma^2)_1 = \{ac, bd, da, db\}$, and, respectively,

$$A_{(G^2)_0} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_{(G^2)_1} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \; .$$

We must have $n \leq \lambda(A_{(G^2)_1}) = 2$. Checking first the case $n = 2$, any $\boldsymbol{x} \in \mathcal{X}(A_{(G^2)_1}, 2)$ must be a multiple of $(1\;1)^\top$, which is a (true) eigenvector of $A_{(G^2)_1}$ associated with the eigenvalue 2 [14, Theorem 5.4]. Yet $(1\;1)^\top \notin \mathcal{X}(A_{(G^2)_0}, 2)$, so we must have $n \leq 1$, and it is easily seen that $(1\;1)^\top \in \mathcal{X}(A_{(G^2)_0}, A_{(G^2)_1}, 1, 1) \neq \emptyset$. Therefore, by Theorem 1 there exists an $(S^2, 1, 1)$-encoder (i.e., a rate $1:2$ bi-modal encoder for S with respect to the partition $\{(\Sigma^2)_0, (\Sigma^2)_1\}$). In this case, $n_{\max}(G, 2) = 1$ and $\rho(G, 2) = 1/2$.

Turning to the third power of G, here

$$\begin{aligned} S \cap (\Sigma^3)_0 &= \{aab, cda, cdb, dcd, acd, bdc, dac, dbd\} \; , \\ S \cap (\Sigma^3)_1 &= \{aac, abd, cdc, dcd, bda, bdb, daa, dab\} \; , \end{aligned}$$

and, respectively,

$$A_{(G^3)_0} = \begin{pmatrix} 3 & 3 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad A_{(G^3)_1} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \; .$$

We must have $n \leq \lambda(A_{(G^3)_1}) = 4$, and we again rule out the case $n = 4$ by observing that $\mathcal{X}(A_{(G^3)_0}, 4)$ consists of multiples of $(3\;\;1)^\top$, yet this vector is not in $\mathcal{X}(A_{(G^3)_1}, 4)$. Hence, we must have $n \leq 3$, and since $(3\;\;1)^\top \in \mathcal{X}(A_{(G^3)_0}, A_{(G^3)_1}, 3, 3)$ we get by Theorem 1 that there exists an $(S^3, 3, 3)$-encoder. Thus, $n_{\max}(G, 3) = 3$ and $\rho(G, 3) = (\log_2 6)/3$.

In general, using the equalities

$$A_{(G^t)_0} = A_{G_0} A_{(G^{t-1})_0} + A_{G_1} A_{(G^{t-1})_1}$$

and

$$A_{(G^t)_1} = A_{G_0} A_{(G^{t-1})_1} + A_{G_1} A_{(G^{t-1})_0} \; ,$$

it can be shown by induction on t that

$$A_{(G^t)_0} = \frac{1}{6} \begin{pmatrix} 2^{t+1} + 3 + (-1)^t & 2^{t+1} - 2\,(-1)^t \\ 2^t - 3 - (-1)^t & 2^t + 2\,(-1)^t \end{pmatrix}$$

and

$$A_{(G^t)_1} = \frac{1}{6} \begin{pmatrix} 2^{t+1} - 3 + (-1)^t & 2^{t+1} - 2\,(-1)^t \\ 2^t + 3 - (-1)^t & 2^t + 2\,(-1)^t \end{pmatrix} \; ,$$

with $\lambda(A_{(G^t)_0}) = \lambda(A_{(G^t)_1}) = 2^{t-1}$ and respective eigenvectors

$$\boldsymbol{x}_0 = \begin{pmatrix} 2^{t+1} - 2\,(-1)^t \\ 2^t - 3 - (-1)^t \end{pmatrix} \quad \text{and} \quad \boldsymbol{x}_1 = \begin{pmatrix} 2^{t+1} - 2\,(-1)^t \\ 2^t + 3 - (-1)^t \end{pmatrix} \; .$$

Clearly, each of the sets $\mathcal{X}(A_{(G^t)_0}, n)$ and $\mathcal{X}(A_{(G^t)_1}, n)$ is empty if and only if $n > 2^{t-1}$. Their intersection, however, is empty also when $n = 2^{t-1}$: for $\mathsf{b} \in [2\rangle$, the elements of $\mathcal{X}(A_{(G^t)_\mathsf{b}}, 2^{t-1})$ are (true) eigenvectors of $A_{(G^t)_\mathsf{b}}$ associated with the Perron eigenvalue 2^{t-1}, namely, they are scalar multiples of $\boldsymbol{x}_\mathsf{b}$, yet $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are linearly independent. We therefore conclude that $\mathcal{X}(A_{(G^t)_0}, A_{(G^t)_1}, n_0, n_1) \neq \emptyset$ only when $\max\{n_0, n_1\} \leq 2^{t-1}$ and $n_0 + n_1 < 2^t$; in particular, there can be no (S^t, n_0, n_1)-encoder when $\log_2(n_0 + n_1) = t$ (and this holds also when $n_0 \neq n_1$). It follows that there can be no rate $t:t$ bi-modal encoder for S (with respect to the partition $\{(\Sigma^t)_0, (\Sigma^t)_1\}$), for any positive integer t.

On the other hand, for $n^{(t)} = 2^{t-1}-1$, we do have $(2\;1)^\top \in \mathcal{X}(A_{(G^t)_0}, A_{(G^t)_1}, n^{(t)}, n^{(t)})$. Therefore, $\rho(G, t) = (1/t)\log_2(2^t - 2) \;\; (< 1)$, and $\lim_{t\to\infty} \rho(G, t) = 1 = \log_2 \lambda(A_G)$, so the rate can *approach* the capacity value of 1 as t increases (yet can never attain it). $\square$

Example 3. Let G be as in Example 2, except that now

$$\Sigma_0 = \{a\} \quad \text{and} \quad \Sigma_1 = \{b, c, d\} \; . \tag{4}$$

It can be shown by induction on t that in this case,

$$A_{(G^t)_0} = \frac{1}{3} \begin{pmatrix} 2^{t+1} + (-1)^t & 0 \\ 0 & 2^t + 2\,(-1)^t \end{pmatrix}$$

and

$$A_{(G^t)_1} = \frac{1}{3} \begin{pmatrix} 0 & 2^{t+1} - 2(-1)^t \\ 2^t - (-1)^t & 0 \end{pmatrix} .$$

Letting

$$n^{(t)} = (A_{(G^t)_0})_{\beta,\beta} = (1/3)\left(2^t + 2(-1)^t\right)$$

and

$$\lambda^{(t)} = \lambda\left(A_{(G^t)_1}\right) = (\sqrt{2}/3)\left(2^t - (-1)^t\right) ,$$

it can be verified that $\mathcal{X}(A_{(G^t)_0}, A_{(G^t)_1}, n_0, n_1) \neq \emptyset$ only when $n_0 \leq n^{(t)}$ and $n_1 \leq \lambda^{(t)}$. In particular, since $n^{(t)} + \lambda^{(t)} < 2^t$, there can be no (S^t, n_0, n_1)-encoder when $\log_2(n_0 + n_1) = t$.

On the other hand, when $t > 2$, we do have $(3\ 2)^\top \in \mathcal{X}(A_{(G^t)_0}, A_{(G^t)_1}, n^{(t)}, n^{(t)})$. Hence, $n_{\max}(G, t) = n^{(t)}$ and, similarly to Example 2, $\rho(G, t) = (1/t)\log_2(2n^{(t)})$ (< 1): we can approach the capacity value (of 1), although we will never attain it.

Note that while the graphs G^t are primitive for all positive integers t, the graphs $(G^t)_0$ in this example are reducible and $(G^t)_1$ are irreducible with period 2. □

The result of Theorem 2 can be generalized to the non-primitive case as follows. Let G be an irreducible deterministic graph with period p. Then G^{p} decomposes into p primitive irreducible components $G_0, G_1, \ldots, G_{\mathsf{p}-1}$, with each G_i being the induced subgraph on a congruence class C_i in $V(G)$. Suppose that for some i, there are two paths of length p that start at states in C_i and generate words with different parities. Under these conditions, the subgraph G_i will satisfy the conditions of Theorem 2.

Theorem 2 (along with the previous examples) focused on (S, n_0, n_1)-encoders where $n_0 = n_1$. While this case suits the motivation of rate $p : q$ bi-modal encoders (where $n_0 = n_1 = 2^{p-1}$), there seems to be a merit in studying the more general case as well (see Example 5 in Section III-B).

Example 4. Let G be a deterministic graph with the adjacency matrix

$$A_G = \begin{pmatrix} 3 & 3 \\ 5 & 5 \end{pmatrix} ,$$

and let Σ_0 and Σ_1 be such that

$$A_{G_0} = \begin{pmatrix} 1 & 2 \\ 2 & 0 \end{pmatrix} \quad \text{and} \quad A_{G_1} = \begin{pmatrix} 2 & 1 \\ 3 & 5 \end{pmatrix} .$$

Then

$$A_{(G^2)_0} = \begin{pmatrix} 12 & 9 \\ 23 & 32 \end{pmatrix} \quad \text{and} \quad A_{(G^2)_1} = \begin{pmatrix} 12 & 15 \\ 17 & 8 \end{pmatrix} .$$

It can be verified that $\lambda(A_{G^2}) = 64$, $\lambda(A_{(G^2)_0}) \approx 39.5$, and $\lambda(A_{(G^2)_1}) \approx 26.1$. Figure 5 shows the boundary of the region of all pairs (n_0, n_1) for which $\mathcal{X}(A_{(G^2)_0}, A_{(G^2)_1}, n_0, n_1) \neq \emptyset$. □

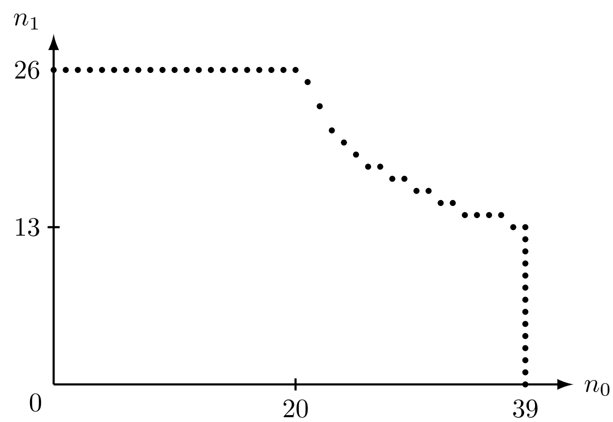

Fig. 5. Attainable pairs (n_0, n_1) for Example 4.

II. Necessary condition

In this section we prove the "only if" part in Theorem 1, recalled next.

Theorem 3. *Let S be an irreducible constraint, presented by an irreducible deterministic graph G, and let n_0 and n_1 be positive integers. Then, an (S, n_0, n_1)-encoder exists, only if $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1) \neq \emptyset$, namely only if there exists a nonnegative nonzero integer vector $\boldsymbol{x}$ such that*

$$A_{G_0}\boldsymbol{x} \geq n_0\boldsymbol{x} \quad \text{and} \quad A_{G_1}\boldsymbol{x} \geq n_1\boldsymbol{x} . \tag{5}$$

The proof of the theorem is a refinement of the proof of Theorem 3 in [13] (or Theorem 7.2 in [14]), where it was shown, *inter alia*, that the existence of an (S, n)-encoder implies the existence of an (A_G, n)-approximate eigenvector. Since the existence of an (S, n_0, n_1)-encoder implies the existence of $(S(G_{\mathsf{b}}), n_{\mathsf{b}})$-encoders for $\mathsf{b} \in [2\rangle$, it also implies that $\mathcal{X}(A_{G_0}, n_0)$ and $\mathcal{X}(A_{G_1}, n_1)$ are both nonempty sets (and, so, $n_0 \leq \lambda(A_{G_0})$ and $n_1 \leq \lambda(A_{G_1})$). Theorem 3 states that their *intersection* must be nonempty too.

Proof of Theorem 3. Suppose that an (S, n_0, n_1)-encoder exists, and fix $\mathcal{E}$ to be such an encoder. Next, we follow the three steps of the proof in [13]: step (a) is identical to the respective step in that proof, and so are the definitions of H', $\boldsymbol{c}$, and $\boldsymbol{x}$ in the remaining steps.

(a) Construct a deterministic graph H which presents $S(\mathcal{E})$. For any word $\boldsymbol{w}$ and state $v \in V(\mathcal{E})$, let $T_{\mathcal{E}}(\boldsymbol{w}, v)$ denote the subset of states in $\mathcal{E}$ which are accessible from v by paths in $\mathcal{E}$ which generate $\boldsymbol{w}$ (when $\boldsymbol{w}$ is the empty word define $T_{\mathcal{E}}(\boldsymbol{w}, v) = \{v\}$). The states of H are defined as the distinct nonempty subsets $\{T_{\mathcal{E}}(\boldsymbol{w}, v)\}_{\boldsymbol{w},v}$ of $V(\mathcal{E})$, and we endow H with an edge $Z \xrightarrow{a} Z'$, if and only if there exists a state $v \in V(\mathcal{E})$ and a word $\boldsymbol{w}$ such that $Z = T_{\mathcal{E}}(\boldsymbol{w}, v)$ and $Z' = T_{\mathcal{E}}(\boldsymbol{w}a, v)$. The graph H is deterministic by construction, and by [14, Lemma 2.1] we have $S(H) = S(\mathcal{E})$. For the rest of the proof, fix H' to be any irreducible sink of H.

(b) Define a positive integer vector $\boldsymbol{c}$ such that $A_{H'_0}\boldsymbol{c} = n_0\boldsymbol{c}$ and $A_{H'_1}\boldsymbol{c} = n_1\boldsymbol{c}$. Recalling that each state $Z \in V(H')$ is a subset of $V(\mathcal{E})$, let $c_Z = |Z|$ denote the number of $\mathcal{E}$-

states in Z and let $\boldsymbol{c}$ be the positive integer vector defined by $\boldsymbol{c} = (c_Z)_{Z \in V(H')}$. We claim that

$$A_{H'_0}\boldsymbol{c} = n_0\boldsymbol{c} \quad \text{and} \quad A_{H'_1}\boldsymbol{c} = n_1\boldsymbol{c} \ . \tag{6}$$

Fix a parity $\mathsf{b} \in [2\rangle$, and consider a state $Z \in V(H')$. Since $\mathcal{E}_\mathsf{b}$ has out-degree n_b, the number of edges in $\mathcal{E}_\mathsf{b}$ outgoing from the subset of states $Z \subseteq V(\mathcal{E})$ is $n_\mathsf{b}|Z|$ (this is also the number of outgoing edges from Z in $\mathcal{E}$ with labels from Σ_b). For $a \in \Sigma_\mathsf{b}$, let E_a denote the set of $\mathcal{E}_\mathsf{b}$-edges labeled a outgoing from the $\mathcal{E}$-states in Z, and let Z_a denote the set of terminal $\mathcal{E}$-states of these edges; note that $\{E_a\}_{a \in \Sigma_\mathsf{b}}$ forms a partition of the set of $\mathcal{E}_\mathsf{b}$-edges outgoing from Z. If $Z_a \neq \emptyset$, then clearly there is an edge $Z \xrightarrow{a} Z_a$ in H and, since H' is a sink, this edge is also contained in H', and, therefore, in H'_b. We now claim that any $\mathcal{E}$-state $u \in Z_a$ is accessible in $\mathcal{E}$ (and in $\mathcal{E}_\mathsf{b}$) by exactly one edge labeled a that starts at Z; otherwise if $Z = T_\mathcal{E}(\boldsymbol{w}, v)$, the word $\boldsymbol{w}a$ could be generated in $\mathcal{E}$ by two distinct paths which start at v and terminate in u, contradicting the losslessness of $\mathcal{E}$. Hence, $|E_a| = |Z_a|$ and, so, the entry of $A_{H'_\mathsf{b}}\boldsymbol{c}$ corresponding to the H'-state Z satisfies

$$\begin{aligned}(A_{H'_\mathsf{b}}\boldsymbol{c})_Z &= \sum_{Y \in V(H')} (A_{H'_\mathsf{b}})_{Z,Y} c_Y = \sum_{Y \in V(H')} (A_{H'_\mathsf{b}})_{Z,Y} |Y| \\ &= \sum_{a \in \Sigma_\mathsf{b}} |Z_a| = \sum_{a \in \Sigma_\mathsf{b}} |E_a| = n_\mathsf{b}|Z| = n_\mathsf{b} c_Z \ ,\end{aligned}$$

thus proving (6).

(c) Construct from $\boldsymbol{c}$ a nonnegative nonzero integer vector $\boldsymbol{x}$ that satisfies (5). Recalling the definition and notation of a follower set from the beginning of Section I-A, let $\boldsymbol{x} = (x_u)_{u \in V(G)}$ be the nonnegative integer vector whose entries are defined for every $u \in V(G)$ by

$$x_u = \max\Big\{ c_Z \ : \ Z \in V(H') \text{ and } \mathcal{F}_{H'}(Z) \subseteq \mathcal{F}_G(u) \Big\} \ . \tag{7}$$

Also, denote by $Z(u)$ some particular H'-state Z for which the maximum is attained in (7); in case the maximum is over an empty set, define $x_u = 0$ and $Z(u) = \emptyset$. Since $S(H') \subseteq S(G)$, we get from [14, Lemma 2.13] that each follower set of an H'-state is contained in a follower set of some G-state; hence, $\boldsymbol{x}$ is necessarily nonzero.

Next, we show that $\boldsymbol{x}$ satisfies (5). Fix a parity $\mathsf{b} \in [2\rangle$, and let u be a G-state; if $x_u = 0$ then, trivially, $(A_{G_\mathsf{b}}\boldsymbol{x})_u \geq n_\mathsf{b} x_u$; so, we assume hereafter that $x_u \neq 0$. Let $\Sigma_\mathsf{b}(Z(u))$ denote the set of labels of edges in H'_b outgoing from $Z(u)$, and, for $a \in \Sigma_\mathsf{b}(Z(u))$, let $Z_a(u)$ be the terminal state in H' (and in H'_b) for an edge labeled a outgoing from the H'-state $Z(u)$. Since $\mathcal{F}_{H'}(Z(u)) \subseteq \mathcal{F}_G(u)$, there exists an edge labeled a in G from u which terminates in some G-state u_a. Now, since G and H' are both deterministic, we have $\mathcal{F}_{H'}(Z_a(u)) \subseteq \mathcal{F}_G(u_a)$ and, so, by (7) we get that $x_{u_a} \geq c_{Z_a(u)}$. Therefore,

$$\begin{aligned}(A_{G_\mathsf{b}}\boldsymbol{x})_u &\geq \sum_{a \in \Sigma_\mathsf{b}(Z(u))} x_{u_a} \geq \sum_{a \in \Sigma_\mathsf{b}(Z(u))} c_{Z_a(u)} \\ &= (A_{H'_\mathsf{b}}\boldsymbol{c})_{Z(u)} \stackrel{(6)}{=} n_\mathsf{b} c_{Z(u)} = n_\mathsf{b} x_u \ ,\end{aligned}$$

namely, $A_{G_\mathsf{b}}\boldsymbol{x} \geq n_\mathsf{b}\boldsymbol{x}$. $\square$

The following corollary parallels Corollary 1 in [13] (or Theorem 7.2 in [14]).

Corollary 4. *Let S be an irreducible constraint, presented by an irreducible deterministic graph G, and let n_0 and n_1 be positive integers. Then, for any (S, n_0, n_1)-encoder $\mathcal{E}$,*

$$|V(\mathcal{E})| \geq \min_{\boldsymbol{x} \in \mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)} \|\boldsymbol{x}\|_\infty \ ,$$

where $\|(x_u)_u\|_\infty = \max_u x_u$.

Proof. Given an (S, n_0, n_1)-encoder $\mathcal{E}$, construct the vector $\boldsymbol{x} \in \mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)$ as in the proof of Theorem 3. Then, by construction, each component of $\boldsymbol{x}$ is a size of a subset of $V(\mathcal{E})$ and, hence, bounds it from below. $\square$

The next corollary parallels Theorem 5 in [13] (or Theorem 7.15 in [14]) and is proved in the very same manner.

Corollary 5. *With S, G, n_0, n_1, and $\mathcal{E}$ as in Corollary 4,*

$$\mathcal{A}(\mathcal{E}) \geq \log_n \Big(\min_{\boldsymbol{x} \in \mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)} \|\boldsymbol{x}\|_\infty \Big) \ ,$$

where $n = \max\{n_0, n_1\}$.

The Franaszek algorithm is a known method for computing approximate eigenvectors [14, §5.2.2]. Figure 6 presents a modification of it for computing a vector in $\mathcal{X}(A_0, A_1, n_0, n_1)$, where A_0 and A_1 are nonnegative integer $k \times k$ matrices (a nonnegative integer k-vector $\boldsymbol{\xi}$ is provided as an additional parameter to the algorithm). The modified algorithm can be used to compute the lower bounds of Corollaries 4 and 5, and will turn out to be useful also when designing bi-modal encoders.

By slightly generalizing the proof of validity of the (ordinary) Franaszek algorithm (see [14, §5.2.2]) it follows that the algorithm in Figure 6 returns the largest (componentwise) vector $\boldsymbol{x} \in \mathcal{X}(A_0, A_1, n_0, n_1)$ that satisfies $\boldsymbol{x} \leq \boldsymbol{\xi}$; if no such vector exists, then the algorithm returns the all-zero vector.

$\boldsymbol{y} \leftarrow \boldsymbol{\xi}$;
$\boldsymbol{x} \leftarrow \boldsymbol{0}$;
while $(\boldsymbol{x} \neq \boldsymbol{y})$ {
  $\boldsymbol{x} \leftarrow \boldsymbol{y}$;
  $\boldsymbol{y} \leftarrow \min\Big\{ \Big\lfloor \frac{1}{n_0} A_0 \boldsymbol{x} \Big\rfloor, \Big\lfloor \frac{1}{n_1} A_1 \boldsymbol{x} \Big\rfloor, \boldsymbol{x} \Big\}$;
    /* apply $\lfloor \cdot \rfloor$ and $\min\{\cdot, \cdot\}$ componentwise */
}
return $\boldsymbol{x}$;

Fig. 6. Modified Franaszek algorithm.

By analyzing the complexity of Karmarkar's algorithm [12] (in terms of number of bit operations), one can conceptually infer an upper bound on the smallest possible largest entry of any vector in $\mathcal{X}(A_0, A_1, n_0, n_1)$, in terms of A_0, A_1, n_0, and n_1 (provided that we first use that algorithm to determine that $\mathcal{X}(A_0, A_1, n_0, n_1)$ is nonempty). Equivalently, such an analysis implies an upper bound on the smallest possible integer $\xi > 0$ such that running the algorithm in Figure 6 with $\boldsymbol{\xi} = \xi \cdot \boldsymbol{1}$ yields a nonzero output $\boldsymbol{x} \in \mathcal{X}(A_0, A_1, n_0, n_1)$. It is still open whether there is a more direct way for computing (efficiently) such an upper bound.

III. Sufficient condition

We start proving the "if" part in Theorem 1 by considering two special cases in Sections III-A and III-B. We then turn to the general case in Section III-C.

A. Deterministic encoders

If $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)$ contains a 0–1 vector, then a subgraph of G is an $(S(G), n_0, n_1)$-encoder with anticipation 0. By Corollary 5, the existence of such a vector is also a necessary condition for having a deterministic $(S(G), n_0, n_1)$-encoder. If, in addition, G has finite memory μ, then the resulting encoder is $(\mu, 0)$-definite and, therefore, $(\mu, 0)$-sliding-block decodable for any tagging.

B. Encoders with anticipation 1 obtained by state splitting

Suppose now that $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)$ does not contain a 0–1 vector, yet contains a vector $\boldsymbol{x} = (x_u)_u$ such that for each $\mathsf{b} \in [2\rangle$, an application of one $\boldsymbol{x}$-consistent state splitting round[4] to G_{b} (after deleting all states u of G with $x_u = 0$) results in an all-1 induced approximate eigenvector. For $\mathsf{b} \in [2\rangle$, let $\hat{\mathcal{E}}_{\mathsf{b}}$ be the resulting $(S(G_{\mathsf{b}}), n_0, n_1)$-encoder. Note that each state $u \in V(G)$ is transformed into x_u descendant states in each encoder $\hat{\mathcal{E}}_{\mathsf{b}}$; denote those states by $(u, i)_{\mathsf{b}}$, where $i \in [x_u\rangle$ (the order implied by the index i on the descendant states of a given u can be arbitrary).

Next, construct the following graph $\mathcal{E}$:

$$V(\mathcal{E}) = \Big\{ (u, i) \;:\; u \in V(G) \quad \text{and} \quad i \in [x_u\rangle \Big\} ,$$

and endow $\mathcal{E}$ with an edge $(u, i) \xrightarrow{a} (v, j)$ if and only if for some $\mathsf{b} \in [2\rangle$, the encoder $\hat{\mathcal{E}}_{\mathsf{b}}$ contains an edge $(u, i)_{\mathsf{b}} \xrightarrow{a} (v, j)_{\mathsf{b}}$. It follows from the construction that $\mathcal{E}_{\mathsf{b}} = \hat{\mathcal{E}}_{\mathsf{b}}$ for $\mathsf{b} \in [2\rangle$. In particular, from each $\mathcal{E}$-state there are n_{b} outgoing edges with labels from Σ_{b}, for each $\mathsf{b} \in [2\rangle$. Moreover, it can be readily seen that $\mathcal{F}_{\mathcal{E}}((u, i)) \subseteq \mathcal{F}_G(u)$ for every $u \in V(G)$ and $i \in [x_u\rangle$. Finally, $\mathcal{E}$ has anticipation 1: if a word $w_1 w_2$ is generated in $\mathcal{E}$ by a path π from state $(u, i) \in V(\mathcal{E})$, then the parent G-state u of (u, i) and the symbol w_1 uniquely identify the parent G-state, v, of the terminal state of the first edge in π, and the symbol w_2 then uniquely identifies the particular descendant state (v, j) of v in which that edge terminates.

In summary, $\mathcal{E}$ is an $(S(G), n_0, n_1)$-encoder with anticipation 1. Furthermore, if G has finite memory μ, then $\mathcal{E}$ is $(\mu, 1)$-definite and, therefore, $(\mu, 1)$-sliding-block decodable for any tagging.

Example 5. We consider the 16th power of the $(2, 10)$-RLL constraint, as found in the DVD standard [14, §1.7.3 and Example 5.7]. Let G be the graph presentation of that power, where the states are numbered from 0 to 10, and edges labeled with 16-bit words that end with a run of 0s of length $i \in [11\rangle$ terminate in state i. Also, let Σ_0 (respectively, Σ_1) be the set of all 16-bit words of even (respectively, odd) parity that satisfy the $(2, 10)$-RLL constraint. Then,

$$A_{G_0} = \begin{pmatrix} 42 & 28 & 19 & 12 & 8 & 6 & 5 & 4 & 3 & 2 & 1 \\ 62 & 42 & 28 & 19 & 12 & 8 & 6 & 5 & 4 & 3 & 2 \\ 90 & 62 & 42 & 28 & 19 & 12 & 8 & 6 & 5 & 4 & 3 \\ 89 & 61 & 41 & 27 & 18 & 12 & 8 & 6 & 5 & 4 & 3 \\ 88 & 60 & 40 & 26 & 17 & 11 & 8 & 6 & 5 & 4 & 3 \\ 86 & 59 & 39 & 25 & 16 & 10 & 7 & 6 & 5 & 4 & 3 \\ 82 & 57 & 38 & 24 & 15 & 9 & 6 & 5 & 5 & 4 & 3 \\ 75 & 53 & 36 & 23 & 14 & 8 & 5 & 4 & 4 & 4 & 3 \\ 65 & 46 & 32 & 21 & 13 & 7 & 4 & 3 & 3 & 3 & 3 \\ 50 & 36 & 25 & 17 & 11 & 6 & 3 & 2 & 2 & 2 & 2 \\ 29 & 21 & 15 & 10 & 7 & 4 & 2 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and

$$A_{G_1} = \begin{pmatrix} 41 & 29 & 21 & 15 & 10 & 7 & 4 & 2 & 1 & 1 & 1 \\ 60 & 41 & 29 & 21 & 15 & 10 & 7 & 4 & 2 & 1 & 1 \\ 87 & 60 & 41 & 29 & 21 & 15 & 10 & 7 & 4 & 2 & 1 \\ 85 & 59 & 41 & 29 & 21 & 15 & 10 & 6 & 4 & 2 & 1 \\ 82 & 57 & 40 & 29 & 21 & 15 & 10 & 6 & 3 & 2 & 1 \\ 78 & 54 & 38 & 28 & 21 & 15 & 10 & 6 & 3 & 1 & 1 \\ 73 & 50 & 35 & 26 & 20 & 15 & 10 & 6 & 3 & 1 & 0 \\ 67 & 45 & 31 & 23 & 18 & 14 & 10 & 6 & 3 & 1 & 0 \\ 59 & 39 & 26 & 19 & 15 & 12 & 9 & 6 & 3 & 1 & 0 \\ 47 & 31 & 20 & 14 & 11 & 9 & 7 & 5 & 3 & 1 & 0 \\ 28 & 19 & 12 & 8 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{pmatrix} .$$

Running the algorithm in Figure 6 with $A_0 = A_{G_0}$, $A_1 = A_{G_1}$, and $\boldsymbol{\xi} = 2 \cdot \mathbf{1}$ yields the result

$$\boldsymbol{x} = (1\ 1\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 0)^\top ,$$

for any $n_0 \le 173$ and $n_1 \le 178$ (running the algorithm with larger values of n_0 or n_1 yields the all-zero vector). This is also the vector obtained when running the (ordinary) Franaszek algorithm with $A_G = A_{G_0} + A_{G_1}$, $n = 351$, and $\boldsymbol{\xi} = 2 \cdot \mathbf{1}$. In both G_0 and G_1 we can merge states 2–5 into 5, states 6–9 into 9, and delete state 10 (see [14, §5.5.1]), resulting in graphs G'_0 and G'_1 with

$$A_{G'_0} = \begin{pmatrix} 42 & 28 & 45 & 14 \\ 62 & 42 & 67 & 18 \\ 86 & 59 & 90 & 22 \\ 50 & 36 & 59 & 9 \end{pmatrix}$$

and

$$A_{G'_1} = \begin{pmatrix} 41 & 29 & 53 & 8 \\ 60 & 41 & 75 & 14 \\ 78 & 54 & 102 & 20 \\ 47 & 31 & 54 & 16 \end{pmatrix} ,$$

and the respective $(A_{G'_0}, A_{G'_1}, n_0{=}173, n_1{=}178)$-approximate eigenvector is

$$\boldsymbol{x}' = (1\ 1\ 2\ 1)^\top .$$

Both G'_0 and G'_1 can be split in one round consistently with $\boldsymbol{x}'$, resulting in the all-1 induced approximate eigenvector and, therefore, in an $(S(G), 173, 178)$-encoder. The out-degree of the encoder that is actually used in the DVD is $2^p = 256$, where the set of input tags consists of all 8-bit tuples. Some of the input tags can be mapped to two possible 16-bit words with different parities[5], while the rest are mapped to unique 16-bit words. □

[4]Refer to [14, Ch. 5] for the description of the state-splitting algorithm and for the related terms used here.

[5]There can be at most $173 + 178 - 256 = 95$ input bytes of this type, but in practice their number is slightly smaller.

C. Construction using the stethering method

The technique used in Section III-B does not seem to generalize easily if the conditions therein—namely, being able to split G_0 and G_1 in one round and ending up with an all-1 induced approximate eigenvector—do not hold. In fact, due to the fact that the matrices G_0 and G_1 may be reducible, we may get stuck while attempting to split them.

Example 6. Let G be the graph with $V(G) = \{\alpha, \beta, \gamma\}$ whose even and odd subgraphs, G_0 and G_1, are shown in Figures 7 and 8 (note that G_1 is reducible). All the edges in G are assumed to have distinct labels. Assuming the ordering $\alpha < \beta < \gamma$ on the states, the adjacency matrices of the subgraphs are given by

$$A_{G_0} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{and} \quad A_{G_1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 2 \end{pmatrix} .$$

It is easy to see that $\boldsymbol{x} = (1\ 2\ 3)^\top$ is an eigenvector of both A_{G_0} and A_{G_1} associated with the Perron eigenvalue $n = 2$. Yet the subgraph G_1 cannot be split consistently with $\boldsymbol{x}$. $\square$

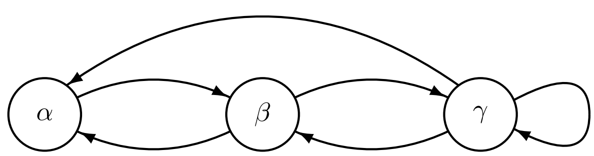

Fig. 7. Subgraph G_0 for Example 6.

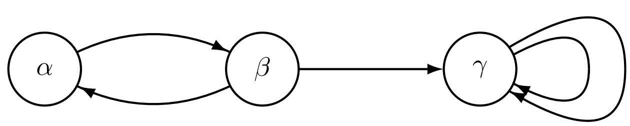

Fig. 8. Subgraph G_1 for Example 6.

Moreover, in Appendix A we present an example where multiple rounds of state splitting are required, which do end up with an all-1 approximate eigenvector, yet there is no way one can match the descendant states in $\mathcal{E}_0$ of a given G-state with the respective descendant states in $\mathcal{E}_1$ while maintaining finite anticipation.

Recognizing that the finite anticipation property is not guaranteed even when the state-splitting algorithm is used (at least in the manner we employed this algorithm in Section III-B), we resort to a more general framework of designing encoders, which includes the state-splitting algorithm and the stethering design method of [3] as special cases (see also [1] and [14, §6.2]). As we see, it will be rather easy to adapt the stethering method to design bi-modal encoders, even though finite anticipation can be guaranteed only under certain conditions.

Next we recall the stethering method, while tailoring it to our setting. Let G be a deterministic graph and $\{\Sigma_0, \Sigma_1\}$ be a partition of its label alphabet Σ, and let $\boldsymbol{x} = (x_u)_{u \in V(G)}$ be in $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)$. We assume that $\boldsymbol{x} > \boldsymbol{0}$, or else remove the zero-weight states from G (namely, the states u for which $x_u = 0$). For $u \in V(G)$, denote by $\Sigma_{\mathsf{b}}(u)$ the set of symbols from Σ_{b} that label edges outgoing from u. For $u \in V(G)$ and $a \in \Sigma_{\mathsf{b}}(u)$, denote by $\tau(u; a)$ the terminal G-state of the unique edge outgoing from u with label a.

For $\mathsf{b} \in [2\rangle$ and $u \in V(G)$, let

$$\Delta_{\mathsf{b}}(u) = \Big\{ (a, j) \ : \ a \in \Sigma_{\mathsf{b}}(u) \quad \text{and} \quad j \in \big[x_{\tau(u;a)} \big\rangle \Big\} .$$

Since $\boldsymbol{x} \in \mathcal{X}(A_{G_{\mathsf{b}}}, n_{\mathsf{b}})$ we have $|\Delta_{\mathsf{b}}(u)| = (A_{G_{\mathsf{b}}} \boldsymbol{x})_u \geq n_{\mathsf{b}} x_u$. Thus, we can partition (a subset of) $\Delta_{\mathsf{b}}(u)$ into x_u subsets

$$\Delta_{\mathsf{b}}^{(0)}(u),\ \Delta_{\mathsf{b}}^{(1)}(u),\ \ldots,\ \Delta_{\mathsf{b}}^{(x_u - 1)}(u) , \tag{8}$$

such that $|\Delta_{\mathsf{b}}^{(i)}(u)| = n_{\mathsf{b}}$ for each i. In what follows, we fix such a partition.

Next, construct the following graph $\mathcal{E}$:

$$V(\mathcal{E}) = \Big\{ (u, i) \ : \ u \in V(G) \quad \text{and} \quad i \in [x_u\rangle \Big\} ,$$

and for each $\mathsf{b} \in [2\rangle$, $u \in V(G)$, $i \in [x_u\rangle$, and $(a, j) \in \Delta_{\mathsf{b}}^{(i)}(u)$, we endow $\mathcal{E}$ with an edge $(u, i) \xrightarrow{a} (\tau(u; a), j)$.

Proposition 6. *The constructed graph $\mathcal{E}$ is an $(S(G), n_0, n_1)$-encoder.*

Proof. First, by construction, the number of outgoing edges from (u, i) with labels from Σ_{b} is $|\Delta_{\mathsf{b}}^{(i)}(u)| = n_{\mathsf{b}}$, for each $\mathsf{b} \in [2\rangle$.

Secondly, let

$$(u_0, i_0) \xrightarrow{w_1} (u_1, i_1) \xrightarrow{w_2} (u_2, i_2) \xrightarrow{w_3} \ldots \xrightarrow{w_\ell} (u_\ell, i_\ell) \tag{9}$$

be a path in $\mathcal{E}$. By construction, $u_{m+1} = \tau(u_m; w_{m+1})$ for every $m \in [\ell\rangle$. Hence, $S(\mathcal{E}) \subseteq S(G)$.

It remains to show that $\mathcal{E}$ is lossless. Consider the word $\boldsymbol{w} = w_1 w_2 \ldots w_\ell$ generated by the path (9). We show that the knowledge of $\boldsymbol{w}$, (u_0, i_0), and (u_ℓ, w_ℓ) uniquely determines the rest of the states along the path. Since G is deterministic, the component u_m of each state (u_m, i_m) along the path is uniquely determined. Suppose by induction that $(u_{m'}, i_{m'})$ has been uniquely determined for every $m' > m$, and let b_{m+1} be the parity of w_{m+1} (i.e., $w_{m+1} \in \Sigma_{\mathsf{b}_{m+1}}$). Since the subsets in (8) are disjoint for $(u, \mathsf{b}) = (u_m, \mathsf{b}_{m+1})$, there is a unique index $i \in [x_{u_m}\rangle$ for which $(w_{m+1}, i_{m+1}) \in \Delta_{\mathsf{b}_{m+1}}^{(i)}(u_m)$; that index must be $i = i_m$. $\square$

The number of states of the constructed encoder $\mathcal{E}$ (before any possible merging of states) equals the sum, $\|\boldsymbol{x}\|_1$, of the entries of $\boldsymbol{x}$. Thus, with this construction, we can obtain an encoder $\mathcal{E}$ such that

$$|V(\mathcal{E})| \leq \min_{\boldsymbol{x} \in \mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1)} \|\boldsymbol{x}\|_1$$

(compare with the lower bound of Corollary 4).

In stethering encoders, the subsets in (8) have a particular structure which we describe next. For $\mathsf{b} \in [2\rangle$ and $u \in V(G)$, assume some ordering on the elements of $\Sigma_{\mathsf{b}}(u)$. For $a \in \Sigma_{\mathsf{b}}(u)$, define $\phi_{\mathsf{b}}(u; a)$ by

$$\phi_{\mathsf{b}}(u; a) = \sum_{b \in \Sigma_{\mathsf{b}}(u) \,:\, b < a} x_{u_b} ,$$

where the sum is zero on an empty set. For $i \in [x_u\rangle$, let

$$\Delta_{\mathsf{b}}^{(i)}(u) = \Big\{ (a,j) \;:\; a \in \Sigma_{\mathsf{b}}(u) \;,\;\; j \in \big[x_{\tau(u;a)} \big\rangle \;, \\ \text{and } \; i\, n_{\mathsf{b}} \le \phi_{\mathsf{b}}(u;a) + j < (i{+}1) n_{\mathsf{b}} \Big\} \;. \quad (10)$$

In other words, for each $u, v \in V(G)$, $i \in [x_u\rangle$, $j \in [x_v\rangle$, and $a \in \Sigma_{\mathsf{b}}(u)$, we endow $\mathcal{E}$ with an edge $(u,i) \xrightarrow{a} (v,j)$, if and only if $v = \tau(u;a)$ and

$$i\, n_{\mathsf{b}} \le \phi_{\mathsf{b}}(u;a) + j < (i{+}1) n_{\mathsf{b}} \;.$$

This construction is illustrated in Figure 9, for a given G-state u and parity $\mathsf{b} \in [2\rangle$. The boxes in the top row in the figure represent the "descendants" of state u, namely, the states (u,i), for $i \in [x_u\rangle$, and the width of each box in the top row is one unit. The outgoing edges from each state (u,i) are shown as downward arrows, along with their labels, and an assignment of input tags, $(\mathsf{b},0), (\mathsf{b},1), \ldots, (\mathsf{b}, n_{\mathsf{b}}{-}1)$, is shown above the top row. The boxes at the bottom row are $1/n_{\mathsf{b}}$ units wide and represent the terminal states of the edges. The respective elements of $\Delta_{\mathsf{b}}(u)$ are written just below the bottom row, where we have also shown their grouping into the subsets (8) defined by (10). The double vertical lines group the edges according to their labels. So, for example, according to the figure, there is an outgoing edge labeled a' and tagged by $(\mathsf{b},0)$ from $(u, x_u{-}1)$ to $(v', x_{v'}{-}1)$, and that edge corresponds to the element $(a', x_{v'}{-}1) \in \Delta_{\mathsf{b}}(u)$.

Stethering encoders can have finite anticipation (and be sliding-block decodable if G has finite memory), provided that there is sufficient margin between the target encoder rate $p : q$ and the maximal coding ratio $\rho(G,q)$ (as defined in (2)). We demonstrate this next.

Suppose that $\boldsymbol{x} \in \mathcal{X}(A_{G_0}, A_{G_1}, n_0{+}1, n_1{+}1)$ (namely, we assume even and odd out-degrees larger by 1 than targeted). Using $\boldsymbol{x}$, we first construct a stethering $(S(G), n_0{+}1, n_1{+}1)$-encoder $\mathcal{E}^*$ and assign the input tags $(\mathsf{b},0), (\mathsf{b},1), \ldots, (\mathsf{b}, n_{\mathsf{b}})$ to the outgoing edges from each state, as in Figure 9. Then, from $\mathcal{E}^*$ we form a punctured $(S(G), n_0, n_1)$-encoder $\mathcal{E}$ by deleting all edges in $\mathcal{E}$ tagged by either $(0, n_0)$ or $(1, n_1)$.

We have the following result (compare the guaranteed upper bound on $\mathcal{A}(\mathcal{E})$ to the lower bound in Corollary 5).

Theorem 7. *Let G be a deterministic graph and let n_0 and n_1 be positive integers such that $\mathcal{X}(A_{G_0}, A_{G_1}, n_0{+}1, n_1{+}1) \ne \emptyset$. Then, there is an $(S(G), n_0, n_1)$-encoder $\mathcal{E}$, obtained by the (punctured) stethering method, such that $\mathcal{A}(\mathcal{E}) \le a$, where*

$$a = 1 + \min_{\boldsymbol{x} \in \mathcal{X}(A_{G_0}, A_{G_1}, n_0+1, n_1+1)} \Big\{ \big\lceil \log_{n+1} \|\boldsymbol{x}\|_\infty \big\rceil \Big\}$$

and $n = \min\{n_0, n_1\}$. Furthermore, if G has finite memory μ, then $\mathcal{E}$ is (μ, a)-definite, and hence any tagged $(S(G), n_0, n_1)$-encoder based on $\mathcal{E}$ is (μ, a)-sliding-block decodable.

Proof. The proof is essentially the same as that of Proposition 3 in [3], and we repeat it here (with the required modifications to handle the bi-modal setting) for completeness.

Let $\mathcal{C}_{\mathsf{b}}(u)$ denote Figure 9 drawn for a given $\mathsf{b} \in [2\rangle$ and $u \in V(G)$. Consider a path

$$\pi = (u_0, i_0) \xrightarrow{w_1} (u_1, i_1) \xrightarrow{w_2} (u_2, i_2) \xrightarrow{w_3} \ldots$$

in the encoder $\mathcal{E}^*$ (obtained prior to the puncturing), and let $(\mathsf{b}_1, s_1), (\mathsf{b}_2, s_2), (\mathsf{b}_3, s_3), \ldots$ be the respective sequence of input tags, where $w_m \in \Sigma_{\mathsf{b}_m}$. Envision an array $\mathcal{B}$ which is constructed as follows. Start with the figure $\mathcal{C}_{\mathsf{b}_1}(u_0)$; the edges labeled w_1 in the figure terminate in the descendant states of u_1, which appear as x_{u_1} boxes in the bottom row of $\mathcal{C}_{\mathsf{b}_1}(u_0)$. Up to down-scaling by a factor of n_{b_1}, these boxes are identical to the top row in $\mathcal{C}_{\mathsf{b}_2}(u_1)$. So, in $\mathcal{B}$, superimpose a down-scaled copy of $\mathcal{C}_{\mathsf{b}_2}(u_1)$ so that its top row coincides with the descendant states of u_1 in $\mathcal{C}_{\mathsf{b}_1}(u_0)$. Proceed in this manner by placing in $\mathcal{B}$ a copy of $\mathcal{C}_{\mathsf{b}_3}(u_2)$, down-scaled by a factor of $n_{\mathsf{b}_1} n_{\mathsf{b}_2}$, so that its top row coincides with the descendant states of u_2 in the bottom row of the (already inserted) down-scaled copy of $\mathcal{C}_{\mathsf{b}_2}(u_1)$. And so on.

The path π can be seen as a vertical line in $\mathcal{B}$ whose abscissa (i.e., the distance from the left margin of $\mathcal{B}$) has the mixed-base representation $i_0.s_{\mathsf{b}_1} s_{\mathsf{b}_2} s_{\mathsf{b}_3} \ldots$, where $i_0 \in [x_u\rangle$ and $s_{\mathsf{b}_m} \in [n_{\mathsf{b}_m}{+}1\rangle$. In other words, that abscissa equals

$$i_0 + \frac{s_{\mathsf{b}_1}}{n_{\mathsf{b}_1}{+}1} + \frac{s_{\mathsf{b}_2}}{(n_{\mathsf{b}_1}{+}1)(n_{\mathsf{b}_2}{+}1)} + \ldots$$

Due to the down-scaling process used to construct $\mathcal{B}$, a decoder can narrow down the uncertainty of that abscissa for each received symbol. Specifically, merely by the knowledge of u_0 that abscissa must be in the real interval $[0, x_{u_0})$ (which is the full width of $\mathcal{B}$). Upon receiving w_1, the length of that uncertainty (open) interval shrinks to $x_{u_1}/(n_{\mathsf{b}_1}{+}1)$; then w_2 reduces it to $x_{u_2}/((n_{\mathsf{b}_1}{+}1)(n_{\mathsf{b}_2}{+}1))$, and so forth. Hence, when the length ℓ of the path is such that

$$\frac{x_{u_\ell}}{\prod_{m=1}^{\ell} (n_{\mathsf{b}_m} + 1)} \le \frac{1}{(n_{\mathsf{b}_1}{+}1)(n_{\mathsf{b}_2}{+}1)} \;, \quad (11)$$

the length of the uncertainty interval reduces to at most the right-hand side of (11). At this point, the numerator in the expression

$$\frac{(n_{\mathsf{b}_2}{+}1) s_{\mathsf{b}_1} + s_{\mathsf{b}_2}}{(n_{\mathsf{b}_1}{+}1)(n_{\mathsf{b}_2}{+}1)}$$

(for the abscissa point $0.s_{\mathsf{b}_1} s_{\mathsf{b}_2}$) can be determined up to ± 1. Yet since the puncturing disallows s_{b_2} to take the value n_{b_2}, this means that s_{b_1} is uniquely determined. It is easy to see that (11) is satisfied for $\ell = a + 1 = 2 + \lceil \log_{n+1} \|\boldsymbol{x}\|_\infty \rceil$, where $n = \min\{n_0, n_1\}$, thereby proving the claimed upper bound on $\mathcal{A}(\mathcal{E})$. Moreover, if G has finite memory μ, then the decoder can recover u_0 by looking at a window of μ past symbols, i.e., $\mathcal{E}$ is (μ, a)-definite. $\square$

Recall that $n_{\max}(G,q)$ is the largest integer n for which $(S(G^q), n, n)$-encoders exist. If we use the punctured stethering method to construct rate $p : q$ bi-modal encoders, then we need to have $2^p + 1 \le n_{\max}(G,q)$. This inequality is satisfied whenever

$$\frac{p}{q} \le \frac{\log_2 n_{\max}(G,q)}{q} - \frac{\log_2(1 + 2^{-p})}{q} \;,$$

which, in turn, is satisfied whenever

$$\frac{p}{q} \le \rho(G,q) - \frac{\log_2 e}{2^p q}$$

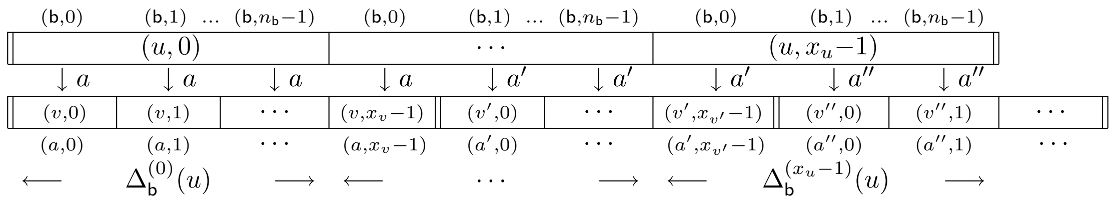

Fig. 9. Descendants of a G-state u in a subgraph $\mathcal{E}_b$ of a stethering encoder.

(see (2)). We conclude that finite anticipation (and sliding-block decodability when G has finite memory) can be guaranteed with a rate penalty of (no more than) $(\log_2 e)/(2^p q)$.

It is still an open problem whether finite anticipation can be guaranteed for any (n_0, n_1) for which $\mathcal{X}(A_{G_0}, A_{G_1}, n_0, n_1) \neq \emptyset$.

D. Extension to non-partition covers

We refer now to a more general setting where the even subset Σ_0 and the odd subset Σ_1 of the constraint alphabet Σ are not necessarily disjoint, but their union is still Σ (recall Footnote 2 for an application of this scenario). Thus, given a graph G, the subgraphs G_0 and G_1 may share edges. It turns out that most of our results hold also for this setting. In particular, the necessary condition stated in Theorem 3 holds as is, since the proof of the theorem does not assume that Σ_0 and Σ_1 are disjoint.

As for the sufficient condition shown in Section III-C, the proof of Proposition 6 holds as long as the partitions (8) satisfy the following condition for every $u \in V(G)$, $a \in \Sigma_0(u) \cap \Sigma_1(u)$, and $j \in [x_{\tau(u;a)}\rangle$:

$$(a, j) \in \Delta_0^{(i)}(u) \cap \Delta_1^{(i')}(u) \quad \Longrightarrow \quad i = i' \tag{12}$$

(this guarantees the uniqueness of the index $i = i_m$ in the last step of the proof of Proposition 6, even when the parity b_{m+1} is not uniquely determined by w_{m+1}). Assuming without loss of generality that $n_0 \leq n_1$, we can apply the following strategy to guarantee the condition (12). We first select an arbitrary partition (8) for $b = 0$ (where $|\Delta_0^{(i)}(u)| = n_0$ for each $i \in [x_u\rangle$). Then, for each $u \in V(G)$, $a \in \Sigma_0(u) \cap \Sigma_1(u)$, and $i \in [x_u\rangle$, we let all elements $(a, j) \in \Delta_0^{(i)}(u)$ be also elements of $\Delta_1^{(i)}(u)$. Finally, we fill in each subset $\Delta_1^{(i)}(u)$ from the remaining elements of $\Delta_1(u)$ to have $|\Delta_1^{(i)}(u)| = n_1$.

We note that, in general, the stethering partition (10) (which is used in the proof of Theorem 7) may be inconsistent with the condition (12). Still, we can always get consistency when $n_0 = n_1$, by selecting the ordering on $\Sigma_0(u)$ and $\Sigma_1(u)$ so that the elements in the intersection $\Sigma_0(u) \cap \Sigma_1(u)$ precede the rest.

IV. Variable-length encoders

So far in this work, we considered bi-modal *fixed-length* encoders at a fixed rate $p : q$, where all tags have the same length p, and all labels have the same length q (under the formulation of (S, n)-encoders, these lengths are 1). On the other hand, most ad-hoc constructions for parity-preserving encoders that were proposed have *variable length* (yet still with a *fixed coding ratio*). In this section, we show through examples that the flexibility of having variable-length (tags and) labels strictly increases the attainable range of the coding ratios of parity-preserving encoders, compared to the fixed-length case. A more thorough study of parity-preserving variable-length encoders is deferred to a subsequent work [18]. For more on variable-length encoders (in the non-parity-preserving setting), see [2], [4], [5], [6], [7], [8], [14, §6.4].

Example 7. Let S be the constraint over $\Sigma = \{a, b, c, d\}$ that is presented by the graph G in Figure 4. Consider the graph $\mathcal{E}$ in Figure 10, which has one state α and three edges: an edge labeled a and two edges of length 2, with labels bd and cd (namely, the length of an edge is the length of its label). For

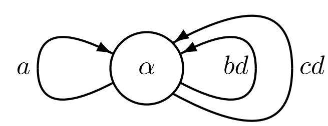

Fig. 10. Variable-length encoder $\mathcal{E}$ for the constraint presented by Figure 4.

the purpose of defining the words that can be generated by a variable-length graph such as $\mathcal{E}$, we view each length-ℓ edge as if it were a path of length ℓ (whose edges are connected by additional $\ell-1$ dummy states). Doing so, it is easy to see that every word that can be generated by $\mathcal{E}$ can also be generated from state α in the graph G of Figure 4. The graph $\mathcal{E}$ is *deterministic* in the sense that the set of labels is *prefix-free*: no label is a prefix of any other label. Hence, a word generated by $\mathcal{E}$ uniquely identifies the path that generates it.

We now assign tags over the (base tag) alphabet $\Upsilon = \{0, 1\}$ to the edges (labels) of $\mathcal{E}$, as shown in Table I. We get in this

TABLE I
Possible tag assignment for the encoder in Figure 10.

0	$\leftrightarrow$	a
10	$\leftrightarrow$	bd
11	$\leftrightarrow$	cd

manner an encoder that has a *coding ratio* of 1: the coding rate is $1 : 1$ when the input tag is 0, and $2 : 2$ when the input tag starts with a 1. Thus, this encoder is capacity-achieving. Moreover, this tag assignment is parity-preserving with respect to the partition $\{\Sigma_0, \Sigma_1\}$ defined in (3). In contrast, we showed in Example 2 that, for this partition, a coding ratio of 1 cannot be achieved by any bi-modal (and, *a fortiori*, parity-preserving) fixed-length encoder. □

Example 8. Considering the same constraint S as in the previous example, the graph $\mathcal{E}'$ in Figure 11 presents another (untagged) variable-length encoder. The coding rate at state α' is $3:3$, as it has eight outgoing edges with labels in Σ^3, and the coding rate at α'' and at β is $2:2$, as each state has four outgoing edges labeled from Σ^2; the coding ratio at each state is therefore 1, making $\mathcal{E}'$ capacity-achieving. However, $\mathcal{E}'$ is not deterministic (there are two edges labeled bda and two labeled cda outgoing from state α', two edges labeled aa outgoing from α'', and two labeled da from state β). Nevertheless, $\mathcal{E}'$ has finite anticipation and is therefore lossless: the first symbol of a label uniquely determines the length of the label as well as the initial state, and a label and the first symbol of the next label within a sequence uniquely determine the edge. One possible assignment of tags (over the alphabet $\Upsilon = \{0,1\}$) to the edges of $\mathcal{E}'$ is shown in Table II.

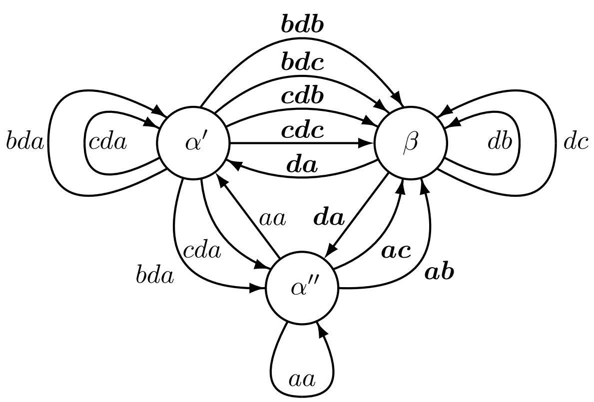

Fig. 11. Second variable-length encoder $\mathcal{E}'$ for the constraint presented by Figure 4.

TABLE II
POSSIBLE TAG ASSIGNMENT FOR THE ENCODER IN FIGURE 11.

State α'			State α''			State β		
$000, 011$	$\leftrightarrow$	bda	$00, 11$	$\leftrightarrow$	aa	$01, 10$	$\leftrightarrow$	$\boldsymbol{da}$
$101, 110$	$\leftrightarrow$	cda	01	$\leftrightarrow$	$\boldsymbol{ac}$	00	$\leftrightarrow$	db
001	$\leftrightarrow$	$\boldsymbol{bdb}$	10	$\leftrightarrow$	$\boldsymbol{ab}$	11	$\leftrightarrow$	dc
010	$\leftrightarrow$	$\boldsymbol{bdc}$						
100	$\leftrightarrow$	$\boldsymbol{cdb}$						
111	$\leftrightarrow$	$\boldsymbol{cdc}$						

Consider now the partition $\{\Sigma_0, \Sigma_1\}$ defined in (4). The labels in boldface in Figure 11 are the odd labels with respect to this partition. It can be readily verified that the tag assignment in Table II is parity-preserving. In contrast, recall that we have shown in Example 3 that there is no bi-modal fixed-length encoder at a coding ratio of 1 for this constraint and this partition.

The encoder in Figure 11 can be obtained by first splitting state α in Figure 4 into states α' and α'' which inherit, respectively, the outgoing edge sets $\{b,c\}$ and $\{a\}$. The resulting graph is an (ordinary) $(S,2)$-encoder, yet, for the above partition of Σ, the parities of the two outgoing edges from each state are the same. We then replace the outgoing edges from state α' with the eight paths of length 3 that start at that state; similarly, we replace the outgoing edges from each of the states α'' and β with the four paths of length 2 that start at the state. □

To summarize, for the two different partitions, (3) and (4), of the alphabet $\Sigma = \{a,b,c,d\}$, Examples 7 and 8 present respective (capacity-achieving) parity-preserving variable-length encoders with a coding ratio of 1: the first encoder is deterministic, while the other is not. In fact, we show in [18] that for the partition (4), one cannot achieve a coding ratio of 1 by any deterministic parity-preserving variable-length encoder (unless one uses a degenerate base tag alphabet containing only even symbols).

On the other hand, there exists such an encoder under some relaxation of the notion of fixed coding ratio, following the encoding model considered in [8]: the tagged encoder $\mathcal{E}^\circ$ in Figure 12 maintains a coding ratio of 1 *along each cycle*. It is easily seen that while at state α, each outgoing edge is uniquely determined by its first symbol, and while at state β, an outgoing edge is uniquely determined by its first two symbols.

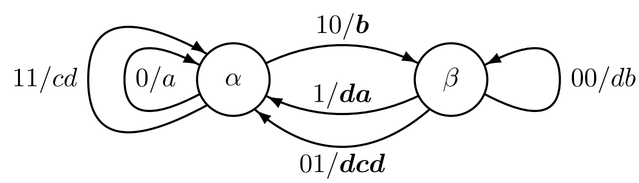

Fig. 12. Third variable-length encoder $\mathcal{E}^\circ$ for the constraint presented by Figure 4.

APPENDIX A
LIMITATIONS OF STATE SPLITTING

In contrast to what we have shown in Section III-B, we present here an example where the state-splitting algorithm yields encoders $\mathcal{E}_0$ and $\mathcal{E}_1$ with anticipation 3, yet any attempt to match between the descendant states in $\mathcal{E}_0$ of a given G-state and the respective descendant states in $\mathcal{E}_1$ results in an encoder that has no finite anticipation.[6]

Let G be the graph with $V(G) = \{\alpha,\beta,\gamma,\delta\}$ whose even and odd subgraphs, G_0 and G_1, are shown in Figures 13 and 14: the even-valued (respectively, odd-valued) hexadecimal digits form the set Σ_0 (respectively, Σ_1). Note that G_0 and G_1 are identical graphs except for the edge labeling and for a switch between states γ and δ.

[6]It is still open whether such an example exists where the anticipation of $\mathcal{E}_0$ and $\mathcal{E}_1$ is 2. It is not difficult to construct such an example where *some* matchings of the descendant states in $\mathcal{E}_0$ with those in $\mathcal{E}_1$ of the same G-state yield an encoder with infinite anticipation.

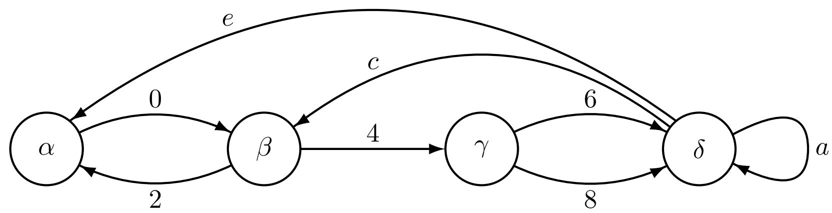

Fig. 13. Subgraph G_0.

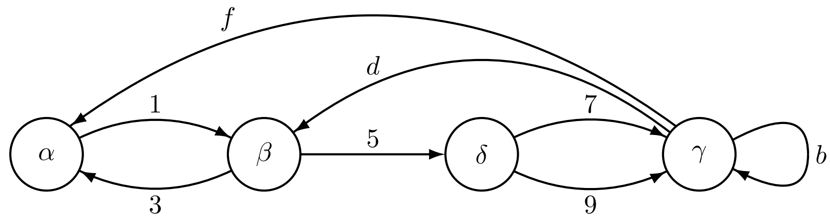

Fig. 14. Subgraph G_1.

Assuming the ordering $\alpha < \beta < \gamma < \delta$ on the states, the adjacency matrices of the subgraphs are given by

$$A_{G_0} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 1 & 1 & 0 & 1 \end{pmatrix} \quad \text{and} \quad A_{G_1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix} ,$$

and it is easily seen that $\boldsymbol{x} = (1\ 2\ 3\ 3)^\top$ is an eigenvector of both matrices associated with the Perron eigenvalue $n = 2$. We are interested in constructing an $(S(G), 2, 2)$-encoder.

Applying a first round of $\boldsymbol{x}$-consistent state splitting to G_0 allows splitting of state δ (and only that state in G_0) into two descendant states: $(\delta, 0)$, which has a weight (i.e., approximate eigenvector entry) of 1 and inherits the outgoing edge labeled c, and $(\delta, 1)$, of weight 2, which inherits the outgoing edges labeled a and e.

In the second round, state $(\delta, 1)$ can be split into $(\delta, 1, 0)$ and $(\delta, 1, 1)$ (each of weight 1), and state γ can be fully split into three descendant states, $(\gamma, \cdot, 0)$, $(\gamma, \cdot, 1)$, and $(\gamma, \cdot, 2)$, each of weight 1. The outgoing picture from the descendant states of γ is shown in Figure 15(a), where

$$(\gamma_{0;1}, \gamma_{0;2}, \gamma_{0;3}) = ((\gamma, \cdot, 0), (\gamma, \cdot, 1), (\gamma, \cdot, 2))$$

and

$$(\delta_{0;1}, \delta_{0;2}, \delta_{0;3}) = ((\delta, 0), (\delta, 1, 0), (\delta, 1, 1))$$

(the first subscript in $\gamma_{0;i}$ and $\delta_{0;j}$ indicates that we are splitting the graph G_0).

In the third round, state β is split into two descendant states, $(\beta, \cdot, \cdot, 0)$ and $(\beta, \cdot, \cdot, 1)$, each of weight 1. At this point, we get an $(S(G_0), 2)$-encoder $\mathcal{E}_0$.

Remark 2. State γ can alternatively be split only partially in the second round, yielding a descendant $(\gamma, \cdot, 0)$ of weight 1 and a descendant $(\gamma, \cdot, 1)$ of weight 2, deferring the splitting of $(\gamma, \cdot, 1)$ to the third round. In this case, the outgoing picture shown in Figure 15(b) is also possible, where now $(\gamma_{0;1}, \gamma_{0;2}, \gamma_{0;3}) = ((\gamma, \cdot, 0), (\gamma, \cdot, 1, 0), (\gamma, \cdot, 1, 1))$. □

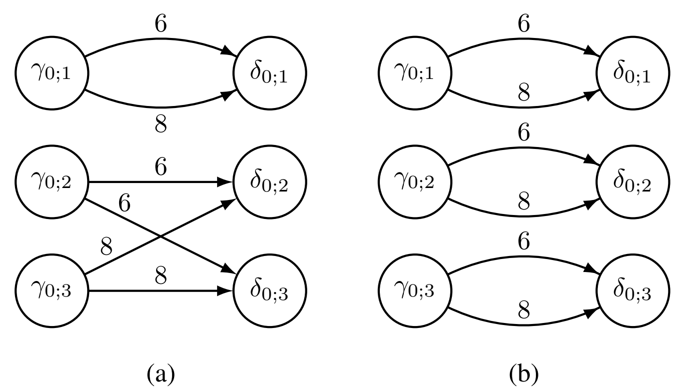

Fig. 15. Possible outgoing pictures from the descendants of state γ in $\mathcal{E}_0$.

Remark 3. In $\mathcal{E}_0$, there are three distinct paths labeled $0\,4$ from state α to the three descendant states ($\gamma_{0;1}$, $\gamma_{0;2}$, and $\gamma_{0;3}$) of γ. □

A respective splitting of G_1 yields an $(S(G_1), 2)$-encoder $\mathcal{E}_1$, in which the possible outgoing pictures from the descendant states of δ (namely, $\delta_{1;i}$) are shown in Figure 16.

Consider now the $(S(G), 2, 2)$-encoder $\mathcal{E}$ obtained by matching the descendant states in $\mathcal{E}_0$ of each given G-state with the respective descendant states in $\mathcal{E}_1$. In particular, we select some bijections

$$\varphi_0 : \{\gamma_{1;1}, \gamma_{1;2}, \gamma_{1;3}\} \to \{\gamma_{0;1}, \gamma_{0;2}, \gamma_{0;3}\} \tag{13}$$

and

$$\varphi_1 : \{\delta_{0;1}, \delta_{0;2}, \delta_{0;3}\} \to \{\delta_{1;1}, \delta_{1;2}, \delta_{1;3}\} \ . \tag{14}$$

Next, we show that for every such selection, there is an arbitrarily long word $\boldsymbol{w}$ that can be generated in $\mathcal{E}$ from two different descendant states of γ (in $\mathcal{E}_0$). And since both these

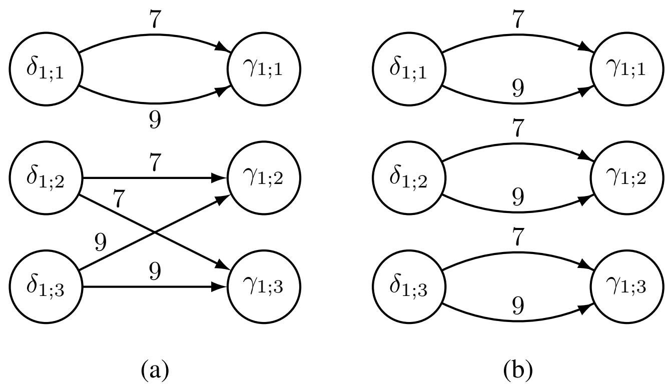

Fig. 16. Possible outgoing pictures from the descendants of state δ in $\mathcal{E}_1$.

states are reachable in $\mathcal{E}_0$ from state α by paths labeled $0\,4$, it will follow that $\mathcal{E}$ does not have finite anticipation.

Such a word $\boldsymbol{w}$ will be generated by paths that toggle between $\mathcal{E}_0$ and $\mathcal{E}_1$ after each symbol. For example, suppose that $\varphi_0(\gamma_{1;i}) = \gamma_{0;i}$ and $\varphi_1(\delta_{0;i}) = \delta_{1;i}$, for $i = 1, 2, 3$. Then the word $6\,7\,6\,7\,\ldots$ can be generated by the following two paths:

$$\gamma_{0;1} \xrightarrow{6} \begin{matrix}\delta_{0;1}\\ \delta_{1;1}\end{matrix} \xrightarrow{7} \begin{matrix}\gamma_{0;1}\\ \gamma_{1;1}\end{matrix} \xrightarrow{6} \begin{matrix}\delta_{0;1}\\ \delta_{1;1}\end{matrix} \xrightarrow{7} \cdots$$

and

$$\gamma_{0;2} \xrightarrow{6} \begin{matrix}\delta_{0;2}\\ \delta_{1;2}\end{matrix} \xrightarrow{7} \begin{matrix}\gamma_{0;2}\\ \gamma_{1;2}\end{matrix} \xrightarrow{6} \begin{matrix}\delta_{0;2}\\ \delta_{1;2}\end{matrix} \xrightarrow{7} \cdots$$

Lemma 8. *For any two bijections φ_0 and φ_1 as in (13)–(14) and for any positive integer ℓ, there are at least two paths of length ℓ in $\mathcal{E}$ that satisfy the following properties.*

(i) *The paths generate the same word.*
(ii) *The paths start at distinct descendants of γ in $\mathcal{E}_0$.*
(iii) *The states along each path alternate between descendants of γ in $\mathcal{E}_0$ and descendants of δ in $\mathcal{E}_1$.*

Proof. When both subgraphs G_0 and G_1 are split according to part (b) in Figures 15 and 16, then the word $6\,7\,6\,7\,\ldots$ can be generated in $\mathcal{E}$ from $\gamma_{0;1}$, $\gamma_{0;2}$, and $\gamma_{0;3}$. Hence, we assume from now on in the proof that at most one of the subgraphs is split according to part (b).

Our proof is by induction on ℓ. The case $\ell = 1$ is obvious, yet when G_0 is split according to Figure 15(b), we will also need to establish the case $\ell = 2$. Let i, j be distinct in $\{1, 2, 3\}$ such that

$$\{\varphi_1(\delta_{0;i}), \varphi_1(\delta_{0;j})\} \neq \{\delta_{1;2}, \delta_{1;3}\} \ .$$

By Figure 16, this selection guarantees that $\varphi_1(\delta_{0;i})$ and $\varphi_1(\delta_{0;j})$ share an outgoing label $w' \in \{7, 9\}$. Hence, if G_0 is split according to part (b), then the word $6\,w'$ (as well as the word $8\,w'$) can be generated in $\mathcal{E}$ both from $\gamma_{0;i}$ and from $\gamma_{0;j}$.

Turning to the induction step, assume that for some odd positive ℓ there exist paths π_1 and π_2 that satisfy properties (i)–(iii), and let $\boldsymbol{w}$ be the word generated by both paths; due to the symmetry between Figures 15 and 16, the proof is similar for even ℓ. Let $\gamma_{0;i}$ (respectively, $\gamma_{0;j}$) be the *penultimate* state visited along the path π_1 (respectively, π_2); note that $i \neq j$, or else $\mathcal{E}$ would not be lossless (by Remark 2). We now distinguish between two cases.

Case 1: G_0 is split according to Figure 15(a). In that figure $\gamma_{0;2}$ and $\gamma_{0;3}$ do not share any outgoing labels and, therefore, $\{i, j\} \neq \{2, 3\}$. Thus, i (say) is 1—and therefore π_1 terminates in $\delta_{0;1}$—and $j \in \{2, 3\}$, and, without loss of generality, π_2 terminates in $\delta_{0;2}$; moreover, there exists a path π_2' that differs from π_2 only in that it terminates in $\delta_{0;3}$ instead (and π_2' still generates the same word $\boldsymbol{w}$). Now, any descendant state of δ in $\mathcal{E}_1$ shares at least one outgoing label with at least one other descendant state of δ in $\mathcal{E}_1$; hence, $\varphi_1(\delta_{0;1})$ must have a common outgoing label $w' \in \{7, 9\}$ with either $\varphi_1(\delta_{0;2})$ or $\varphi_1(\delta_{0;3})$. Thus, π_1, as well as either π_2 or π_2', can be extended by an edge (of $\mathcal{E}_1$) labeled w', to produce two paths of length $\ell{+}1$ that satisfy properties (i)–(iii) (both paths generating the word $\boldsymbol{w}w'$).

Case 2: G_0 is split according to Figure 15(b). By our assumption this implies that G_1 is split according to Figure 16(a), so we can apply the analysis of Case 1 to length $\ell{-}1$ (with G_0 and G_1 switching roles). In particular, there exist paths π_1, π_2, and π_2' of length $\ell{-}1$, all generating the same word $\boldsymbol{w}$, such that π_1 and π_2 start at distinct descendants of γ (in $\mathcal{E}_0$), π_1 terminates in $\gamma_{1;1}$, and π_2 and π_2' differ only in their last edge: π_2 terminates in $\gamma_{1;2}$ while π_2' terminates in $\gamma_{1;3}$. Write $\gamma_{0;r} = \varphi_0(\gamma_{1;1})$, $\gamma_{0;s} = \varphi_0(\gamma_{1;2})$, and $\gamma_{0;t} = \varphi_0(\gamma_{1;3})$. Similarly to Case 1, $\varphi_1(\delta_{0;r})$ must have a common outgoing label $w' \in \{7, 9\}$ with either $\varphi_1(\delta_{0;s})$ or $\varphi_1(\delta_{0;t})$. Thus, π_1, as well as either π_2 or π_2', can be extended by two edges—the first of $\mathcal{E}_0$ labeled 6 (or 8) and the second of $\mathcal{E}_1$ labeled w'—to produce two paths of length $\ell{+}1$ that satisfy properties (i)–(iii) (both paths generating either the word $\boldsymbol{w}\,6\,w'$ or the word $\boldsymbol{w}\,8\,w'$). $\square$

References

[1] R.L. Adler, L.W. Goodwyn, B. Weiss, "Equivalence of topological Markov shifts," *Israel J. Math.,* 27 (1977), 49–63.
[2] R.L. Adler, J. Friedman, B. Kitchens, B.H. Marcus, "State splitting for variable-length graphs," *IEEE Trans. Inf. Theory,* 32 (1986), 108–113.
[3] J.J. Ashley, B.H. Marcus, R.M. Roth, "Construction of encoders with small decoding look-ahead for input-constrained channels," *IEEE Trans. Inf. Theory,* 41 (1995), 55–76.
[4] M.-P. Béal, "The method of poles: A coding method for constrained channels," *IEEE Trans. Inf. Theory,* 36 (1990), 763–772.
[5] M.-P. Béal, "Extensions of the method of poles for code construction," *IEEE Trans. Inf. Theory,* 49 (2003), 1516–1523.
[6] P.A. Franaszek, "Sequence-state coding for digital transmission," *Bell Sys. Tech. J.,* 47 (1968), 143–155.
[7] P.A. Franaszek, "On synchronous variable length coding for discrete noiseless channels," *Inform. Control,* 15 (1969), 155–164.
[8] C.D. Heegard, B.H. Marcus, P.H. Siegel, "Variable-length state splitting with applications to average runlength-constrained (ARC) codes," *IEEE Trans. Inf. Theory,* 37 (1991), 759–777.
[9] K.A.S. Immink, *Codes for Mass Data Storage Systems,* Second Edition, Shannon Foundation Publishers, Eindhoven, The Netherlands, 2004.
[10] K.A.S. Immink, J.-Y. Kim, S.-W. Suh, S.K. Ahn, "Efficient dc-free RLL codes for oprical recording," *IEEE Trans. Commun.,* 51 (2003), 326–331.
[11] J.A.H.M. Kahlman, K.A.S. Immink, "Device for encoding/decoding N-bit source words into corresponding M-bit channel words, and vice versa," US Patent 5,477,222, 1995.
[12] D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming,* Springer, New York, 2008.
[13] B.H. Marcus, R.M. Roth, "Bounds on the number of states in encoder graphs for input-constrained channels," *IEEE Trans. Inf. Theory,* 37 (1991), 742–758.

[14] B.H. Marcus, R.M. Roth, P.H. Siegel, *An Introduction to Coding for Constrained Systems,* Lecture Notes, 2001, available online at: ronny.cswp.cs.technion.ac.il/wp-content/uploads/sites/54/2016/05/chapters1-9.pdf

[15] T. Miyauchi, Y. Shinohara, Y. Iida, T. Watanabe, Y. Urakawa, H. Yamagishi, M. Noda, "Application of turbo codes to high-density optical disc storage using 17PP Code," *Jpn. J. Appl. Phys.,* 44 No. 5B (2005), 3471–3473.

[16] T. Narahara, S. Kobayashi, M. Hattori, Y. Shimpuku, G.J. van den Enden, J.A.H.M. Kahlman, M. van Dijk, R. van Woudenberg, "Optical disc system for digital video recording," *Jpn. J. Appl. Phys.,* 39 No. 2B (2000), 912–919.

[17] M. Noda, H. Yamagishi, "An 8-state DC-controllable run-length-limited code for the optical-storage channel," *Jpn. J. Appl. Phys.,* 44 No. 5B (2005), 3462–3466.

[18] R.M. Roth, P.H. Siegel, "On parity-preserving variable-length constrained coding," available online at https://arxiv.org/abs/2005.05455. See also *Proc. IEEE Int'l Symp. Inf. Theory* (2020), 682–687.

[19] W.Y.H. Wilson, K.A.S. Immink, X.B. Xi, C.T. Chong, "A Comparison of two coding schemes for generating DC-free runlength-limited sequences," *Jpn. J. Appl. Phys.,* 39 No. 2B (2000), 815–818.

Ron M. Roth (Fellow, IEEE) received the B.Sc. degree in computer engineering, the M.Sc. in electrical engineering, and the D.Sc. in computer science from Technion—Israel Institute of Technology, Haifa, Israel, in 1980, 1984, and 1988, respectively. Since 1988 he has been with the Computer Science Department at Technion, where he now holds the General Yaakov Dori Chair in Engineering. During the academic years 1989–91 he was a Visiting Scientist at IBM Research Division, Almaden Research Center, San Jose, California, and during 1996–97, 2004–05, and 2011–2012 he was on sabbatical leave at Hewlett–Packard Laboratories, Palo Alto, California. He is the author of the book *Introduction to Coding Theory*, published by Cambridge University Press in 2006. Dr. Roth was an associate editor for coding theory in IEEE TRANSACTIONS ON INFORMATION THEORY from 1998 till 2001, and he is now serving as an associate editor in *SIAM Journal on Discrete Mathematics*. His research interests include coding theory, information theory, and their application to the theory of complexity.

Paul H. Siegel (Life Fellow, IEEE) received the S.B. and Ph.D. degrees in mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1975 and 1979, respectively. He held a Chaim Weizmann Postdoctoral Fellowship with the Courant Institute, New York University, New York, NY, USA. He was with the IBM Research Division, San Jose, CA, USA, from 1980 to 1995. He joined the University of California at San Diego, CA, USA, in July 1995, as a Faculty Member, where he is currently a Distinguished Professor of electrical and computer engineering with the Jacobs School of Engineering. He is also affiliated with the Center for Memory and Recording Research, where he holds the Endowed Chair and served as the Director from 2000 to 2011. His research interests include information theory and communications, particularly coding and modulation techniques, with applications to digital data storage and transmission. He was a Member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996 and 2009 to 2014. He is a member of the National Academy of Engineering. He was the 2015 Padovani Lecturer of the IEEE Information Theory Society. He was a co-recipient of the 2007 Best Paper Award in signal processing and coding for data storage from the Data Storage Technical Committee of the IEEE Communications Society. He was a co-recipient of the 1992 IEEE Information Theory Society Paper Award and the 1993 IEEE Communications Society Leonard G. Abraham Prize Paper Award. He served as a Co-Guest Editor of the May 1991 Special Issue on Coding for Storage Devices of the IEEE TRANSACTIONS ON INFORMATION THEORY. He has served as an Associate Editor for Coding Techniques for the IEEE TRANSACTIONS ON INFORMATION THEORY from 1992 to 1995, and the Editor-in-Chief from July 2001 to July 2004. He was also a Co-Guest Editor of the May/September 2001 two-part issue on The Turbo Principle: From Theory to Practice and the February 2016 issue on Recent Advances in Capacity Approaching Codes of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.