

Article

# Evaluating the Gilbert–Varshamov Bound for Constrained Systems <sup>†</sup>

Keshav Goyal  and Han Mao Kiah <sup>\*</sup> 

School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637121, Singapore; keshav002@ntu.edu.sg

<sup>\*</sup> Correspondence: hmkih@ntu.edu.sg

<sup>†</sup> The paper was presented in part at the 2022 IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, 26 Jun–1 July 2022.

**Abstract:** We revisit the well-known Gilbert–Varshamov (GV) bound for constrained systems. In 1991, Kolesnik and Krachkovsky showed that the GV bound can be determined via the solution of an optimization problem. Later, in 1992, Marcus and Roth modified the optimization problem and improved the GV bound in many instances. In this work, we provide explicit numerical procedures to solve these two optimization problems and, hence, compute the bounds. We then show that the procedures can be further simplified when we plot the respective curves. In the case where the graph presentation comprises a single state, we provide explicit formulas for both bounds.

**Keywords:** Gilbert–Varshamov bound; constrained codes; asymptotic rates; sliding window constrained codes

## 1. Introduction

From early applications in magnetic recording systems to recent applications in DNA-based data storage [1–4] and energy-harvesting [5–10], constrained codes have played a central role in enhancing reliability in many data storage and communications systems (see also [11] for an overview). Specifically, for most data storage systems, certain substrings are more prone to errors than others. Thus, by forbidding the appearance of such strings, that is, by imposing constraints on the codewords, the user is able to reduce the likelihood of error. We refer to the collection of words that satisfy the constraints as the *constrained space*  $\mathcal{S}$ .

To further reduce the error probability, one can impose certain distance constraints on the codebook. In this work, we focus on the *Hamming metric* and consider the maximum size of a codebook whose words belong to the constrained space  $\mathcal{S}$  and whose pairwise distance is at least of a certain value  $d$ . Specifically, we study one of the most well-known and fundamental lower bounds of this quantity—the *Gilbert–Varshamov (GV) bound*.

To determine the GV bound, one requires two quantities: the size of the constrained space,  $|\mathcal{S}|$ , and, also, the *ball volume*, that is, the number of words with a distance of at most  $d - 1$  from a “center” word. In the case where the space is unconstrained, i.e.,  $\mathcal{S} = \{0, 1\}^n$ , the ball volume does not depend on the center. Then, the GV bound is simply  $|\mathcal{S}|/V$ , where  $V$  is the ball volume of a center. However, for most constrained systems, the ball volume varies with the center. Nevertheless, Kolesnik and Krachkovsky showed that the GV lower bound can be generalized to  $|\mathcal{S}|/4\bar{V}$ , where  $\bar{V}$  is the *average ball volume* [12]. This was further improved by Gu and Fuja to  $|\mathcal{S}|/\bar{V}$  in [13] (see pp. 242–243 in [11] for additional details). In the same paper [12], they showed the asymptotic rate of average ball volume can be computed via an optimization problem. Later, Marcus and Roth modified the optimization problem by including an additional constraint and variable [14], and the resulting bound, which we refer to as *GV-MR bound*, improves the usual GV bound. Furthermore, in most cases, the improvement is strictly positive.



**Citation:** Goyal, K.; Kiah, H.M.

Evaluating the Gilbert–Varshamov Bound for Constrained Systems.

*Entropy* **2024**, *26*, 346. <https://doi.org/10.3390/e26040346>

Academic Editor: T. Aaron Gulliver

Received: 28 February 2024

Revised: 11 April 2024

Accepted: 17 April 2024

Published: 19 April 2024

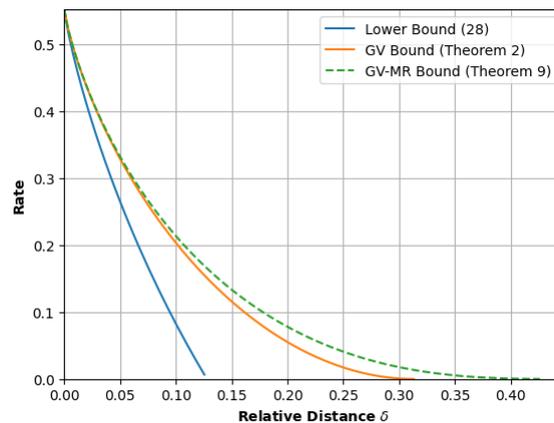


**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

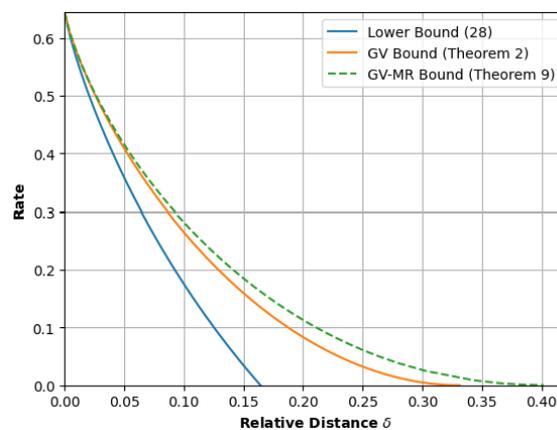
However, about three decades later, very few works have evaluated these bounds for specific constrained systems. To the best of our knowledge, in all works that numerically computed the GV bound and/or GV-MR bound, the constrained systems of interest have, at most, eight states [15]. In [15], the authors wrote that “evaluation of the bound required considerable computation”, referring to the GV-MR bound.

In this paper, we revisit the optimization problems defined by Kolesnik and Krachkovsky [12] and Marcus and Roth [14] and develop a suite of explicit numerical procedures that solve these problems. In particular, to demonstrate the feasibility of our methods, we evaluated and plotted the GV and GV-MR bounds for a constrained system involving 120 states in Figure 1b.

(a) Lower bounds for  $R(\delta; \mathcal{S})$  where  $\mathcal{S}$  is the class of (3,2)-SWCC



(b) Lower bounds for  $R(\delta; \mathcal{S})$  where  $\mathcal{S}$  is the class of (10,7)-SWCC



**Figure 1.** Lower bounds for optimal asymptotic code rates  $R(\delta; \mathcal{S})$  for the class of sliding-window constrained codes

We provide a high-level description of our approach. For both optimization problems, we first characterized the optimal solutions as roots of certain equations. Then, using the celebrated *Newton–Raphson* iterative procedure, we proceeded to find the roots of these equations. However, as the latter equations involved the largest eigenvalues of certain matrices, each *Newton–Raphson* iteration required the (partial) derivatives of these eigenvalues (in some variables). To resolve this, we made modifications to another celebrated iterative procedure—the *power iteration* method—and the resulting procedures computed the GV and GV-MR bounds efficiently for a specific relative distance  $\delta$ . Interestingly, if we plot the bounds for  $0 \leq \delta \leq 1$ , the numerical procedure can be further simplified. Specifically, by exploiting certain properties of the optimal solutions, we provided procedures that use less *Newton–Raphson* iterations.

Parts of this paper were presented in the IEEE International Symposium on Information Theory (ISIT 2022) [16]. In the next section, we provide the formal definitions and state the optimization problems that compute the GV bound.

### 2. Preliminaries

Let  $\Sigma = \{0, 1\}$  be the binary alphabet and let  $\Sigma^n$  denote the set of all words of length  $n$  over  $\Sigma$ . A labeled graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$  is a finite directed graph with states  $\mathcal{V}$ , edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ , and an edge labeling  $\mathcal{L} : \mathcal{E} \rightarrow \Sigma^s$  for some  $s \geq 1$ . Here, we use  $v_i \xrightarrow{\sigma} v_j$  to mean that there is an edge from  $v_i$  to  $v_j$  with label  $\sigma$ . The labeled graph  $\mathcal{G}$  is deterministic if, for each state, the outgoing edges have distinct labels.

A constrained system  $\mathcal{S}$  is, then, the set of all words obtained by reading the labels of paths in a labeled graph  $\mathcal{G}$ . We say that  $\mathcal{G}$  is a graph presentation of  $\mathcal{S}$ . We further denote the set of all  $n$ -length words  $\mathcal{S}$  by  $\mathcal{S}_n$ . Alternatively,  $\mathcal{S}_n$  is the set of all words obtained by reading the labels of  $(n/s)$ -length paths in  $\mathcal{G}$ . Then, the capacity of  $\mathcal{S}$ , denoted by  $\text{Cap}(\mathcal{S})$ , is given by  $\text{Cap}(\mathcal{S}) \triangleq \limsup_{n \rightarrow \infty} \log |\mathcal{S}_n| / n$ . It is well-known that  $\text{Cap}(\mathcal{S})$  corresponds to the largest eigenvalue of the adjacency matrix  $A_{\mathcal{G}}$  (see, for example, [11]). Here,  $A_{\mathcal{G}}$  is a  $(|\mathcal{V}| \times |\mathcal{V}|)$ -matrix whose rows and columns are indexed by  $\mathcal{V}$ . For each entry  $(u, v) \in \mathcal{V} \times \mathcal{V}$ , we set the corresponding entry to be one if  $(u, v)$  is an edge, and zero otherwise.

Every constrained system can be presented by a deterministic graph  $\mathcal{G}$ . Furthermore, any deterministic graph can be transformed into a primitive deterministic graph  $\mathcal{H}$  such that the capacity of  $\mathcal{G}$  is same as the capacity of the constrained system presented by some irreducible component (maximal irreducible subgraph) of  $\mathcal{H}$  (see, for example, Marcus et al. [11]). It should be noted that a graph  $\mathcal{G}$  is primitive if there exists a positive integer  $\ell$  such that  $(A_{\mathcal{G}})^\ell$  is strictly positive. Therefore, we henceforth assume that our graphs are deterministic and primitive. When  $|\mathcal{V}| = 1$ , we call this a single-state graph presentation and study these graphs in Section 5.

For  $x, y \in \mathcal{S}$ ,  $d_H(x, y)$  is the Hamming distance between  $x$  and  $y$ . We fix  $1 \leq d \leq n$ , and a fundamental problem in coding theory is finding the largest subset  $\mathcal{C}$  of  $\mathcal{S}_n$  such that  $d_H(x, y) \geq d$  for all distinct  $x, y \in \mathcal{C}$ . Let  $A(n, d; \mathcal{S})$  denote the size of largest subset  $\mathcal{C}$ .

In terms of asymptotic rates, we fix  $0 \leq \delta \leq 1$ , and our task is to find the highest attainable rate, denoted by  $R(\delta)$ , which is given by  $R(\delta; \mathcal{S}) \triangleq \limsup_{n \rightarrow \infty} \log A(n, \lfloor \delta n \rfloor; \mathcal{S}) / n$ .

#### 2.1. Review of Gilbert–Varshamov Bound

To define the GV bound, we need to determine the total ball size. Specifically, for  $x \in \mathcal{S}_n$  and  $0 \leq r \leq n$ , we define  $V(x, r; \mathcal{S}) \triangleq |\{y \in \mathcal{S}_n : d_H(x, y) \leq r\}|$ . We further define  $T(n, d; \mathcal{S}) = \sum_{x \in \mathcal{S}_n} V(x, d - 1; \mathcal{S})$ . Then, the GV bound, as given by Gu and Fuja [13,17], states that there exists an  $(n, d; \mathcal{S})$  code of size at least  $|\mathcal{S}_n|^2 / T(n, d; \mathcal{S})$ .

In terms of asymptotic rates, there exists a family of  $(n, \lfloor \delta n \rfloor; \mathcal{S})$  codes such that their rates approach

$$R_{\text{GV}}(\delta) = 2\text{Cap}(\mathcal{S}) - \tilde{T}(\delta), \tag{1}$$

where  $\tilde{T}(\delta) \triangleq \limsup_{n \rightarrow \infty} \log T(n, \lfloor \delta n \rfloor; \mathcal{S}) / n$ .

In this paper, our main task is to determine  $R_{\text{GV}}(\delta)$  efficiently. We observe that since  $\text{Cap}(\mathcal{S}) = \tilde{T}(0)$ , it suffices to find efficient ways of determining  $\tilde{T}(\delta)$ . It turns out that  $\tilde{T}(\delta)$  can be found via the solution of a convex optimization problem. Specifically, given a labeled graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ , we define its product graph  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}', \mathcal{L}')$  as follows:

- $\mathcal{V}' \triangleq \mathcal{V} \times \mathcal{V}$ .
- For  $(v_i, v_j), (v_k, v_\ell) \in \mathcal{V}'$ , and  $(\sigma_1, \sigma_2) \in \Sigma^s \times \Sigma^s$ , we draw an edge  $(v_i, v_j) \xrightarrow{(\sigma_1, \sigma_2)} (v_k, v_\ell)$  if and only if both  $v_i \xrightarrow{\sigma_1} v_k$  and  $v_j \xrightarrow{\sigma_2} v_\ell$  belong to  $\mathcal{E}$ .
- Then, we label the edges in  $\mathcal{E}'$  with the function  $\mathcal{L}' : \mathcal{E}' \rightarrow \mathbb{Z}_{\geq 0}$ , where  $\mathcal{L}'\left((v_i, v_j) \xrightarrow{(\sigma_1, \sigma_2)} (v_k, v_\ell)\right) = d_H(\sigma_1, \sigma_2) / s$ .

A stationary Markov chain  $P$  on a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$  is a probability distribution function  $P : \mathcal{E} \rightarrow [0, 1]$  such that  $\sum_{e \in \mathcal{E}} P(e) = 1$  and, for any state  $u \in \mathcal{G}$ , the sum of the probabilities of the outgoing edges equals the sum of the probabilities of the incoming edges. We denote by  $\mathcal{M}(\mathcal{G})$  the set of all stationary Markov chains on  $\mathcal{G}$ . For a state  $u \in \mathcal{V}$ , let  $\mathcal{E}_u$  denote the set of outgoing edges from  $u$  in  $\mathcal{G}$ . The state vector  $\pi^T = (\pi_u)_{u \in \mathcal{V}}$  of a stationary Markov chain  $P$  on  $\mathcal{G}$  is defined by  $\pi_u = \sum_{e \in \mathcal{E}_u} P(e)$ . The entropy rate of a stationary Markov chain is defined by

$$H(P) = - \sum_{u \in \mathcal{V}} \sum_{e \in \mathcal{E}_u} \pi_u P(e) \log(P(e))$$

Furthermore,  $\tilde{T}(\delta)$  can be obtained by solving the following optimization problem [12,14]:

$$\tilde{T}(\delta) = \sup \left\{ H(P) : P \in \mathcal{M}(\mathcal{G} \times \mathcal{G}), \sum_{e \in \mathcal{E}'} P(e) D(e) \leq \delta \right\}. \tag{2}$$

To this end, we consider the dual problem of (2). Specifically, we define a  $(|\mathcal{V}|^2 \times |\mathcal{V}|^2)$ -distance matrix  $T_{\mathcal{G} \times \mathcal{G}}(y)$  whose rows and columns are indexed by  $\mathcal{V}'$ . For each entry indexed by  $e \in \mathcal{V}' \times \mathcal{V}'$ , we set the entry to be zero if  $e \notin \mathcal{E}'$  and we set it to be  $y^{D(e)}$  if  $e \in \mathcal{E}'$ . Then, the dual problem can be stated in terms of the dominant eigenvalue of the matrix  $T_{\mathcal{G} \times \mathcal{G}}(y)$ .

By applying the reduction techniques from [14], we can reduce the problem size by a factor of two. Formally, in the case of  $s = 1$ , we define a  $\binom{|\mathcal{V}|+1}{2} \times \binom{|\mathcal{V}|+1}{2}$ -reduced distance matrix  $B_{\mathcal{G} \times \mathcal{G}}(y)$  whose rows and columns are indexed by  $\mathcal{V}^{(2)} \triangleq \{(v_i, v_j) : 1 \leq i \leq j \leq |\mathcal{V}|\}$  using the following procedure.

Two states  $s_1 = (v_i, v_j)$  and  $s_2 = (v_k, v_\ell)$  in  $\mathcal{G} \times \mathcal{G}$  are said to be *equivalent* if  $v_i = v_\ell$  and  $v_j = v_k$ . The matrix  $B_{\mathcal{G} \times \mathcal{G}}(y)$  is then obtained by merging all pairs of equivalent states  $s_1$  and  $s_2$ . That is, we add the column indexed by  $v_2$  to the column indexed by  $v_1$  and then remove the row and column which are indexed by  $v_2$ . It should be noted that it may be possible to reduce the size of this matrix  $B_{\mathcal{G} \times \mathcal{G}}(y)$  further. However, for the ease of exposition, we did not consider this case in this work.

Following this procedure, we observe that the entries in the matrix  $B_{\mathcal{G} \times \mathcal{G}}(y)$  can be described by the rules in Table 1. Moreover, the dominant eigenvalue of  $B_{\mathcal{G} \times \mathcal{G}}(y)$  is the same as that of  $T_{\mathcal{G} \times \mathcal{G}}(y)$ . Then, by strong duality, computing (2) is equivalent to solving the following dual problem [18,19] (see also, [20]):

$$\tilde{T}(\delta) = \inf \{ -\delta \log y + \log \Lambda(B_{\mathcal{G} \times \mathcal{G}}(y)) : 0 \leq y \leq 1 \}. \tag{3}$$

Here, we use  $\Lambda(M)$  to denote the dominant eigenvalue of matrix  $M$ . To simplify further, we write  $\Lambda(y; B) \triangleq \Lambda(B_{\mathcal{G} \times \mathcal{G}}(y))$ .

Since the objective function in (3) is convex, it follows from standard calculus that any local minimum solution  $y^*$  in the interval  $[0, 1]$  is also a global minimum solution. Furthermore,  $y^*$  is a zero of the first derivative of the objective function. If we consider the numerator of this derivative, then  $y^*$  is a root of the function

$$F(y) \triangleq y \Lambda'(y; B) - \delta \Lambda(y; B). \tag{4}$$

In Corollary 1, we showed that there is only one  $y^*$  such that  $F(y^*) = 0$  and  $F'(y)$  is strictly positive for all values of  $y$ . Therefore, to evaluate the GV bound for a fixed  $\delta$ , it suffices to determine  $y^*$ .

Later, Marcus and Roth [14] improved the GV bound (1) by considering certain subsets of the constrained space  $\mathcal{S}$ . This entails the inclusion of an additional constraint defined in the optimization problem (2), and, correspondingly, an additional variable in the dual problem (3). Specifically, they considered certain subsets  $\mathcal{S}(p) \subseteq \mathcal{S}$  where each symbol in the words of  $\mathcal{S}(p)$  appears with a certain frequency dependent on the parameter  $p$ . We describe this in more detail in Section 4.

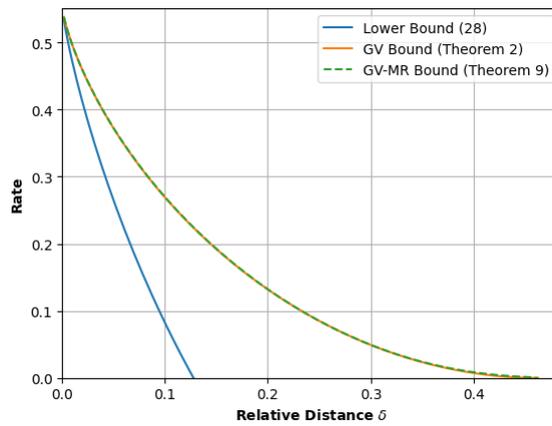
**Table 1.** We set the  $((v_i, v_j), (v_k, v_\ell))$  entry of the matrix  $B_{\mathcal{G} \times \mathcal{G}}(y)$  according to subgraph induced by the states  $v_i, v_j, v_k, v_\ell$  Gilbert–Varshamov  $v_\ell$ . Here,  $\bar{\sigma}$  denotes the complement of  $\sigma$ .

$B_{\mathcal{G} \times \mathcal{G}}(y)$ at Entry $((v_i, v_j), (v_k, v_\ell))$	Subgraph Induced by the States $\{v_i, v_j, v_k, v_\ell\}$					
0						
1						
y						
2y						

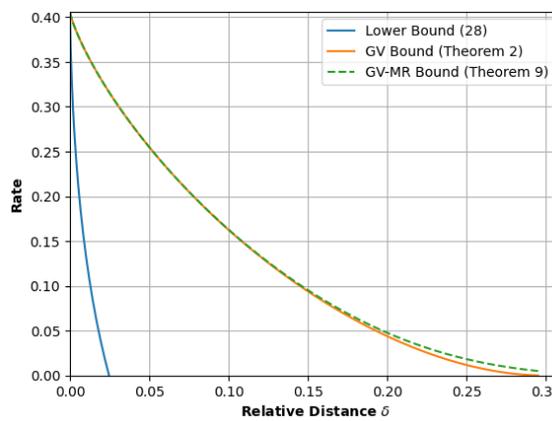
2.2. Our Contributions

- (A) In Section 3, we develop the numerical procedures to compute  $\tilde{T}(\delta)$  for a fixed  $\delta$  and, hence, determine the GV bound (1). Our procedure modifies the well-known *power iteration method* to compute the derivatives of  $\Lambda(y; \mathbf{B})$ . After that, using these derivatives, we apply the classical Newton–Raphson method to determine the root of (4). In the same section, we also study procedures to plot the GV curve, that is, the set  $\{(\delta, R_{GV}(\delta)) : 0 \leq \delta \leq 1\}$ . Here, we demonstrate that the GV curve can be plotted without any Newton–Raphson iterations.
- (B) In Section 4, we then develop similar power iteration methods and numerical procedures to compute the GV-MR bound. Similar to the GV curve, we also provide a plotting procedure that uses significantly less Newton–Raphson iterations.
- (C) In Section 5, we provide explicit formulas for the computation of the GV bound and GV-MR bound for graph presentations that have exactly one state but multiple parallel edges.
- (D) In Section 6, we validate our methods by computing the GV and the GV-MR bounds for some specific constrained systems. For comparison purposes, we also plot a simple lower bound that is obtained by using an upper estimate of the ball size. From the plots in Figures 1–3, it is also clear that the GV and GV-MR bounds are significantly better. We also observe that the GV bound and GV-MR bound for *subblock energy-constrained codes (SECCs)* obtained through our procedures improve the GV-type bound given by Tandon et al. (Proposition 12 in [21]).

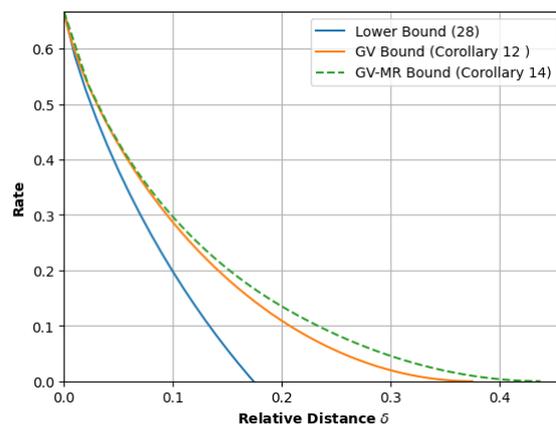
(a) Lower bounds for  $R(\delta; \mathcal{S})$  where  $\mathcal{S}$  is the class of (1,3)-RLL



(b) Lower bounds for  $R(\delta; \mathcal{S})$  where  $\mathcal{S}$  is the class of (3,7)-RLL



**Figure 2.** Lower bounds for optimal asymptotic code rates  $R(\delta; \mathcal{S})$  for the class of runlength limited codes.



**Figure 3.** Lower bounds for optimal asymptotic code rates  $R(\delta; \mathcal{S})$  where  $\mathcal{S}$  is the class of (3,2)-SECCs (subblock energy-constrained codes).

### 3. Evaluating the Gilbert–Varshamov Bound

In this section, we first describe a numerical procedure that solves (3) and, hence, determine  $R_{GV}(\delta)$  for fixed values of  $\delta$ . Then, we show that the procedure can be simplified when we compute the GV curve, that is, the set of points  $\{(\delta, R_{GV}(\delta)) : \delta \in [0, 1]\}$ . Here, we eschew notation and use  $[a, b]$  to denote the interval  $\{x : a \leq x \leq b\}$ , if  $a < b$ , and the interval  $\{x : b \leq x \leq a\}$  otherwise.

Below, we provide formal description of our procedure to obtain the GV bound for a fixed relative distance  $\delta$ .

**Procedure 1 (GV bound for fixed relative distance) .**

INPUT: Adjacency matrix  $A_{\mathcal{G}}$ , reduced distance matrix  $\mathbf{B}_{\mathcal{G} \times \mathcal{G}}(y)$ , and relative minimum distance  $\delta$

OUTPUT: GV bound, that is,  $R_{GV}(\delta)$  as defined in (1)

(1) Apply the Newton–Raphson method to obtain  $y^*$  such that  $F(y^*)$  is approximately zero.

- Fix the tolerance value  $\epsilon$ .
- Set  $t = 0$  and pick an initial guess  $0 \leq y_t \leq 1$ .
- While  $|y_t - y_{t-1}| > \epsilon$ ,
  - Compute the next guess  $y_{t+1}$  as follows:

$$y_{t+1} = y_t - \frac{F(y_t)}{F'(y_t)} = y_t - \frac{y_t \Lambda'(y_t; \mathbf{B}) - \delta \Lambda(y_t; \mathbf{B})}{(1 - \delta) \Lambda'(y_t; \mathbf{B}) + y_t \Lambda''(y_t; \mathbf{B})}.$$

- In this step, apply the power iteration method to compute  $\Lambda(y_t; \mathbf{B})$ ,  $\Lambda'(y_t; \mathbf{B})$ , and  $\Lambda''(y_t; \mathbf{B})$ .
- Increment  $t$  by one.

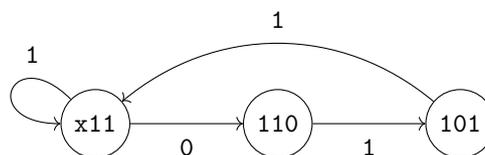
- Set  $y^* \leftarrow y_t$ .

(2) Determine  $R_{GV}(\delta)$  using  $y^*$ . Specifically, compute  $\tilde{T}(\delta) \triangleq -\delta \log y^* + \log \Lambda(y^*; \mathbf{B})$ ,  $\text{Cap}(\mathcal{S}) \triangleq \log \Lambda(A_{\mathcal{G}})$ , and  $R_{GV}(\delta) \triangleq 2\text{Cap}(\mathcal{S}) - \tilde{T}(\delta)$ .

Throughout Sections 3 and 4, we illustrate our numerical procedures via a running example using the class of *sliding window-constrained codes* (SWCCs). Formally, we fix a *window length*  $L$  and *window weight*  $w$ , and say that a binary word satisfies the  $(L, w)$ -*sliding window weight constraint* if the number of ones in every consecutive  $L$  bits is at least  $w$ . We refer to the collection of words that meet this constraint as an  $(L, w)$ -*SWCC constrained system*. The class of SWCCs was introduced by Tandon et al. for the application of simultaneous energy and information transfer [7,10]. Later, Immink and Cai [8,9] studied encoders for this constrained system and provided a simple graph presentation that uses only  $\binom{L}{w}$  states.

In the next example, we illustrate how the numerical procedure can be used to compute the GV bound for the value when  $\delta = 0.1$ .

**Example 1.** Let  $L = 3$  and  $w = 2$ , and we consider a  $(3, 2)$ -SWCC constrained system. From [8], we have the following graph presentation with states  $x11$ ,  $101$ , and  $110$ :



Then, the corresponding adjacency and reduced distance matrices are as follows:

$$A_{\mathcal{G}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{B}_{\mathcal{G} \times \mathcal{G}}(y) = \begin{bmatrix} 1 & 2y & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & y & 0 \\ 1 & y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

To determine the GV bound at  $\delta = 0.1$ , we first approximate the optimal point  $y^*$  for which  $-\delta \log y + \log \Lambda(y; \mathbf{B})$  is minimized.

We apply the Newton–Raphson method to find a zero of the function  $F(y)$ . Now, with the initial guess  $y_0 = 0.3$ , we apply the power iteration method to determine

$$\Lambda(0.3; \mathbf{B}) = 1.659, \Lambda'(0.3; \mathbf{B}) = 0.694, \Lambda''(0.3; \mathbf{B}) = 0.183.$$

Then, we compute that  $y_1 \approx 0.238$ . Repeating the computations, we have that  $y_2 \approx 0.238$ . Since  $|y_2 - y_1|$  is less than the tolerance value  $10^{-5}$ , we set  $y^* = 0.238$ . Hence, we have that  $\tilde{T}(0.1) = 0.9$ . Applying the power iteration method to either  $\mathbf{A}_G$  or  $\mathbf{B}_{G \times G}(0)$ , we compute the capacity of the (3, 2)-SWCC constrained system to be  $\text{Cap}(\mathcal{S}) = 0.551$ . Then, the GV bound is given by  $R_{GV}(0.1) = 2(0.551) - 0.9 = 0.202$ .

We discuss the convergence issues arising from Procedure 1. We observe that there are two different iterative processes in Step 1, namely, (a) the power iteration method to compute the values  $\Lambda(y_t; \mathbf{B})$ ,  $\Lambda'(y_t; \mathbf{B})$ , and  $\Lambda''(y_t; \mathbf{B})$ , and (b) the Newton–Raphson method that determines the zero of  $F(y)$ .

- (a) We recall that  $\Lambda(y; \mathbf{B})$  is the largest eigenvalue of the reduced distance matrix  $\mathbf{B}_{G \times G}(y)$ . If we apply naive methods to compute this dominant eigenvalue, the computational complexity increases very rapidly with the matrix size. Specifically, if  $G$  has  $M$  states, then the reduced distance matrix has dimensions  $\Theta(M^2) \times \Theta(M^2)$  and finding its characteristic equation takes  $O(M^6)$  time. Even then, determining the exact roots of characteristic equations with at least five degrees is generally impossible. Therefore, we turn to the numerical procedures like the ubiquitous power iteration method [22]. However, the standard power iteration method is only able to compute the dominant eigenvalue  $\Lambda(y; \mathbf{B})$ . Nevertheless, we can modify the power iteration method to compute  $\Lambda(y; \mathbf{B})$  and its higher order derivatives. In Appendix A, we demonstrate that under certain mild assumptions, the modified power iteration method always converges. Moreover, using the sparsity of the reduced distance matrix, we have that each iteration can be completed in  $O(M^2)$  time.
- (b) Next, we discuss whether we can guarantee that  $y_t$  converges to  $y^*$  as  $t$  approaches infinity. Even though the Newton–Raphson method converges in all our numerical experiments, we are unable to demonstrate that it always converges for  $F(y)$ . Nevertheless, we can circumvent this issue if we are interested in plotting the GV curve. Specifically, if our objective is to determine the curve  $\{(\delta, R_{GV}(\delta)) : \delta \in [0, 1]\}$ , it turns out that we do not need to implement the Newton–Raphson iterations and we discuss this next.

We fix some constrained system  $\mathcal{S}$ . Let us define its corresponding GV curve to be the set of points  $\mathcal{GV}(\mathcal{S}) \triangleq \{(\delta, R_{GV}(\delta)) : \delta \in [0, 1]\}$ . Here, we demonstrate that the GV curve can be plotted without any Newton–Raphson iterations.

To this end, we observe that when  $F(y^*) = 0$ , we have that  $\delta = y^* \Lambda'(y^*; \mathbf{B}) / \Lambda(y^*; \mathbf{B})$ . Hence, we eschew notation and define the function

$$\delta(y) \triangleq y \Lambda'(y; \mathbf{B}) / \Lambda(y; \mathbf{B}). \tag{5}$$

We further define  $\delta_{\max} = \delta(1) = \Lambda'(1; \mathbf{B}) / \Lambda(1; \mathbf{B})$ . In this section, we prove the following theorem.

**Theorem 1.** *Let  $G$  be the graph presentation for the constrained system  $\mathcal{S}$ . If we define the function*

$$\rho_{GV}(y) \triangleq 2\text{Cap}(\mathcal{S}) + \delta(y) \log y - \log \Lambda(y; \mathbf{B}), \tag{6}$$

*then the corresponding GV curve is given by*

$$\mathcal{GV}(\mathcal{S}) = \{(\delta(y), \rho_{GV}(y)) : y \in [0, 1]\} \cup \{(\delta, 0) : \delta \geq \delta_{\max}\}. \tag{7}$$

Before we prove Theorem 1, we discuss its implications. It should be noted that to compute  $\delta(y)$  and  $\rho(y)$ , it suffices to determine  $\Lambda(y; \mathbf{B})$  and  $\Lambda'(y; \mathbf{B})$  using the modified power iteration methods described in Appendix A. In other words, no Newton–Raphson iterations are required. We also have additional computational savings, as we do not need to apply the power iteration method to compute the second derivative  $\Lambda''(y; \mathbf{B})$ .

**Example 2.** We continue our example and plot the GV curve for the (3, 2)-SWCC constrained system in Figure 1a. Before plotting, we observe that when  $y = 0$ , we have  $(\delta(0), \rho(0)) = (0, 0.551) = (0, \text{Cap}(\mathcal{S}))$ , as expected. When  $y = 1$ , we have  $\delta(1) = \delta_{\max} = 0.313$ . Indeed, both  $\rho(1)$  and  $R_{\text{GV}}(\delta_{\max})$  are equal to zero and we have that  $R_{\text{GV}}(\delta) = 0$  for  $\delta \geq \delta_{\max}$ .

Next, we compute a set of 100 points on the GV curve. If we apply Procedure 1 to compute  $R_{\text{GV}}(\delta)$  for 100 values of  $\delta$  in the interval  $[0, \delta_{\max}]$ , we require 275 Newton–Raphson iterations and 6900 power iterations to find these points. In contrast, applying Theorem 1, we compute  $(\delta(y), \rho(y))$  for 100 values of  $y$  in the interval  $[0, 1]$ . This does not require any Newton–Raphson iterations and involves only 2530 power iterations.

To prove Theorem 1, we demonstrate the following lemmas. Our first lemma is immediate from the definitions of  $R_{\text{GV}}$ ,  $\delta$ , and  $\rho$  in (1), (5), and (6), respectively.

**Lemma 1.**  $R_{\text{GV}}(\delta(y)) = \rho(y)$  for all  $y \in [0, 1]$ .

The next lemma studies the behaviour of both  $\delta$  and  $\rho$  as functions in  $y$ .

**Lemma 2.** In terms of  $y$ , the functions  $\delta(y)$  and  $\rho(y)$  are monotone increasing and decreasing, respectively. Furthermore, we have that  $(\delta(0), \rho(0)) = (0, \text{Cap}(\mathcal{S}))$ ,  $(\delta(1), \rho(1)) = (\delta_{\max}, 0)$  and  $R_{\text{GV}}(\delta) = 0$  for  $\delta \geq \delta_{\max}$ .

**Proof.** To simplify notation, we write  $\Lambda(y; \mathbf{B})$ ,  $\Lambda'(y; \mathbf{B})$ , and  $\Lambda''(y; \mathbf{B})$  as  $\Lambda$ ,  $\Lambda'$ , and  $\Lambda''$ , respectively.

First, we show that  $\delta'(y)$  is positive for  $0 \leq y < 1$ . Differentiating the expression in (5), we have that  $\delta'(y) > 0$  is equivalent to

$$\Lambda(\Lambda' + y\Lambda'') - y(\Lambda')^2 > 0. \tag{8}$$

We recall that (3) is a convex minimization problem. Hence, the second order derivative of the objective function is always positive. In other words,

$$\frac{\delta}{y^2} + \frac{\Lambda''\Lambda - (\Lambda')^2}{\Lambda^2} > 0.$$

Substituting  $\delta$  with  $y\Lambda'/\Lambda$  and multiplying by  $y\Lambda^2$ , we obtain (8), as desired.

Next, we show that  $\rho$  is monotone decreasing. We recall that  $\rho(y) = R_{\text{GV}}(\delta(y)) = \text{Cap}(\mathcal{S}) - \tilde{T}(\delta)$ . Since  $\tilde{T}(\delta)$  yields the asymptotic rate of the total ball size, we have that as  $y$  increases,  $\delta(y)$  increases and so,  $\tilde{T}(\delta)$  increases. Therefore,  $\rho(y)$  decreases, as desired.

Next, we show that  $\rho(1) = 0$ . When  $y = 1$ , we have from (6) that  $\rho(1) = 2\text{Cap}(\mathcal{S}) - \log \Lambda(1; \mathbf{B})$ . Now, we recall that  $B_{\mathcal{G} \times \mathcal{G}}(y)$  shares the same dominant eigenvalue as the matrix  $T_{\mathcal{G} \times \mathcal{G}}(y)$  [12]. Furthermore, it can be verified that when  $y = 1$ ,  $T_{\mathcal{G} \times \mathcal{G}}(1)$  is tensor product of  $A_{\mathcal{G}}$  and  $A_{\mathcal{G}}$ . That is,  $T_{\mathcal{G} \times \mathcal{G}}(1) = A_{\mathcal{G}} \otimes A_{\mathcal{G}}$ . It then follows from standard linear algebra that  $\Lambda(1; \mathbf{B}) = \Lambda(1; \mathbf{T}) = \Lambda(A_{\mathcal{G}})^2$ . Thus,  $\log \Lambda(1; \mathbf{B}) = 2\text{Cap}(\mathcal{S})$  and  $\rho(1) = 0$ . In this instance, we also have that  $\tilde{T}(\delta_{\max}) = 2\text{Cap}(\mathcal{S})$ .

Finally, for  $\delta \geq \delta_{\max}$ , we have that  $\tilde{T}(\delta_{\max}) = 2\text{Cap}(\mathcal{S})$  and thus,  $R_{\text{GV}}(\delta) = 0$ , as required.  $\square$

Theorem 1 is then immediate from Lemmas 1 and 2.

We have the following corollary that immediately follows from Lemma 2. This corollary then implies that  $y^*$  yields the global minimum for the optimization problem.

**Corollary 1.** *When  $0 \leq \delta \leq \delta_{max} = \frac{\Lambda'(1, \mathbf{B})}{\Lambda(1, \mathbf{B})}$ ,  $F(y) \triangleq y\Lambda'(y; \mathbf{B}) - \delta\Lambda(y; \mathbf{B})$  has a unique zero in  $[0, 1]$ . Furthermore,  $F'(y)$  is strictly positive for all  $y \in [0, 1]$ .*

#### 4. Evaluating Marcus and Roth’s Improvement of the Gilbert–Varshamov Bound

In [14], Marcus and Roth improved the GV lower bound for most constrained systems by considering subsets  $\mathcal{S}(p)$  of  $\mathcal{S}$  where  $p$  is some parameter. Here, we focus on the case  $s = 1$  and set  $p$  to be the normalized frequency of edges whose labels correspond to one. Specifically, we set  $\mathcal{S}(p) \triangleq \{x \in \mathcal{S} : \text{wt}(x) = \lfloor p|x| \rfloor\}$ .

Next, let  $\mathcal{S}_n(p)$  be the set of all words/paths of length  $n$  in  $\mathcal{S}(p)$  and we define  $S(p) \triangleq \limsup_{n \rightarrow \infty} \frac{1}{n} \log |\mathcal{S}_n(p)|$ .

Similar to before, we define  $\tilde{T}(p, \delta) = \limsup_{n \rightarrow \infty} \frac{1}{n} \log T(\lfloor \delta n \rfloor, n; \mathcal{S}_n(p))$ . Since  $\mathcal{S}_n(p)$  is a subset of  $\mathcal{S}_n$ , it follows from the usual GV argument that there exists a family of  $(n, \lfloor \delta n \rfloor; \mathcal{S})$  codes whose rates approach  $2S(p) - \tilde{T}(p, \delta)$  for all  $0 \leq p \leq 1$ . Therefore, we have the following lower bound on asymptotic achievable code rates:

$$R_{MR}(\delta) = \sup\{2S(p) - \tilde{T}(p, \delta) : 0 \leq p \leq 1\}. \tag{9}$$

Now, a key result from [14] is that both  $S(p)$  and  $\tilde{T}(p, \delta)$  can be obtained via two different convex optimization problems. For succinctness, we state the *dual* formulations of these optimization problems.

First,  $S(p)$  can be obtained from the following problem:

$$S(p) = \inf\{-p \log z + \log \Lambda(\mathbf{C}_g(z)) : z \geq 0\}. \tag{10}$$

Here,  $\mathbf{C}_g(z)$  is the following  $(|\mathcal{V}| \times |\mathcal{V}|)$  matrix  $\mathbf{C}_g(z)$  whose rows and columns are indexed by  $\mathcal{V}$ . For each entry indexed by  $e$ , we set  $(\mathbf{C}_g(z))_e$  to be zero if  $e \notin \mathcal{E}$ , and  $z^{\mathcal{L}(e)}$  if  $e \in \mathcal{E}$ .

As before, we simplify notation by writing  $\Lambda(z; \mathbf{C}) \triangleq \Lambda(\mathbf{C}_g(z))$ . Again, following the convexity of (10), we are interested in finding the zero of the following function:

$$G_1(z) \triangleq z\Lambda'(z; \mathbf{C}) - p\Lambda(z; \mathbf{C}). \tag{11}$$

Next,  $\tilde{T}(p, \delta)$  can be obtained via the following optimization:

$$\tilde{T}(p, \delta) = \inf\{-2p \log x - \delta \log y + \log \Lambda(\mathbf{D}_{g \times g}(x, y)) : x \geq 0, 0 \leq y \leq 1\}. \tag{12}$$

Here,  $\mathbf{D}_{g \times g}(x, y)$  is a  $\binom{|\mathcal{V}|+1}{2} \times \binom{|\mathcal{V}|+1}{2}$ -reduced distance matrix indexed by  $\mathcal{V}^{(2)}$ . To define the entry of matrix  $\mathbf{D}_{g \times g}(x, y)$  indexed by  $((v_i, v_j), (v_k, v_\ell))$ , we look at the vertices  $v_i, v_j, v_k$ , and  $v_\ell$  and follow the rules given in Table 2.

Again, we write  $\Lambda(x, y; \mathbf{D}) \triangleq \Lambda(\mathbf{D}_{g \times g}(x, y))$ . Furthermore, following the convexity of (12), we have that if the optimal solution is obtained at  $x$  and  $y$ , then

$$G_2(x, y) \triangleq x\Lambda_x(x, y; \mathbf{D}) - 2p\Lambda(x, y; \mathbf{D}) = 0. \tag{13}$$

$$G_3(x, y) \triangleq y\Lambda_y(x, y; \mathbf{D}) - \delta\Lambda(x, y; \mathbf{D}) = 0. \tag{14}$$

To this end, we consider the function  $\Delta(x) = \Lambda_y(x, 1; \mathbf{D}) / \Lambda(x, 1; \mathbf{D})$  for  $x > 0$  and set  $\delta_{max} = \sup\{\Delta(x) : x > 0\}$ . As with the previous section, we develop a numerical procedure to solve the optimization problem (9). To this end, we have the following critical observation.

**Table 2.** We set the  $((v_i, v_j), (v_k, v_\ell))$  entry of the matrix  $D_{\mathcal{G} \times \mathcal{G}}(x, y)$  according to the subgraph induced by the states  $v_i, v_j, v_k$ , and  $v_\ell$ .

$D_{\mathcal{G} \times \mathcal{G}}(x, y)$ at Entry $((v_i, v_j), (v_k, v_\ell))$	Subgraph Induced by the States $\{v_i, v_j, v_k, v_\ell\}$		
0			
1			
$x^2$			
$xy$			
$2xy$			

**Theorem 2.** For a given  $\delta < \delta_{\max}$ , consider the optimization problem

$$\sup \left\{ -2p \log z + 2 \log \Lambda(z; \mathbf{C}) + 2p \log x + \delta \log y - \log \Lambda(x, y; \mathbf{D}) : G_1(z) = G_2(x, y) = G_3(x, y) = 0 \right\}.$$

If  $(p^*, x^*, y^*, z^*)$  is an optimal solution, then  $x^* = z^*$ . Furthermore, if  $0 \leq p^* \leq 1$ , then  $x^*, z^* \geq 0$  and  $0 \leq y^* \leq 1$ .

**Proof.** Let  $\lambda_1, \lambda_2$ , and  $\lambda_3$  be real-valued variables and we define  $L(p, x, y, z, \lambda_1, \lambda_2, \lambda_3) \triangleq G(p, x, y, z) + \lambda_1 G_1(z) + \lambda_2 G_2(x, y) + \lambda_3 G_3(x, y)$ . Using the Lagrangian multiplier theorem, we have that  $\partial L / \partial p = \partial L / \partial x = \partial L / \partial y = \partial L / \partial z = 0$  for any optimal solution. Solving these equations with the constraints  $G_1(z) = G_2(x, y) = G_3(x, y) = 0$ , we have that  $\lambda_1 = \lambda_2 = \lambda_3 = 0$  and  $x = z$  for any optimal solution.

Now, when  $p^* \in [0, 1]$ , using  $G_1(z) = 0$ , let us define  $z(p) \triangleq z \Lambda'(z; \mathbf{C}) / \Lambda(z; \mathbf{C})$ . Then, proceeding as with the proof of Lemma 2, we see that  $z(p)$  is monotone increasing with  $z(0) = 0$ . Therefore,  $z^* = z(p^*)$  is zero.

Similarly, given  $p^*$  and  $x^*$ , we use  $G_3(x^*, y) = 0$  to define  $\delta(y) = y \Lambda_y(x^*, y; \mathbf{D}) / \Lambda(x^*, y; \mathbf{D})$ . Again, we can proceed as with the proof of Lemma 2 to show that  $\delta(y)$  is monotone increasing. Furthermore, since  $\delta(y^*) < \delta_{\max} = \delta(1)$ , we have that  $y^* \in [0, 1]$ .  $\square$

Therefore, to determine  $R_{\text{MR}}(\delta)$  for any fixed  $\delta$ , it suffices to find  $x, y, z$ , and  $p$  such that  $G_1(z) = G_2(x, y) = G_3(x, y) = 0$  and  $x = z$ .

Now, the optimization in Theorem 2 does not constrain the values of  $p$ . Furthermore, for certain constrained systems, there are instances where  $p$  falls outside the interval  $[0, 1]$ . In this case, instead of solving the optimization problem (9), we set  $p$  to be either zero or one, and we solve the corresponding optimization problems (10) and (12). Specifically, if

we have  $p^* < 0$ , then we set  $p^* = 0$  and  $x^* = 0$ , or if  $p^* > 1$ , then we set  $p^* = 1$  and  $x^* = \infty$ . Hence, the resulting rates that we obtain are a *lower bound* for the GV-MR bound.

**Procedure 2** ( $R_{MR}(\delta)$  for fixed  $\delta \leq \delta_{max}$ ).

INPUT: Matrices  $C_G(x), D_G(x, y)$

OUTPUT:  $R_{MR}(\delta)$  or  $R_{LB}(\delta)$ , where  $R_{MR}(\delta) \geq R_{LB}(\delta)$ .

(1) Apply the Newton–Raphson method to obtain  $p^*, x^*$ , and  $y^*$  such that  $G_1(x^*), G_2(x^*, y^*)$ , and  $G_3(x^*, y^*)$  are approximately zero. Specifically, do the following:

- Fix a tolerance value  $\epsilon$
- Set  $t = 0$  and pick an initial guess  $p_t \geq 0, x_t \geq 0, 0 \leq y_t \leq 1$ .
- While  $|p_t - p_{t-1}| + |x_t - x_{t-1}| + |y_t - y_{t-1}| > \epsilon$ ,
  - Compute the next guess  $p_{t+1}, x_{t+1}, y_{t+1}$ :

$$\begin{bmatrix} p_{t+1} \\ x_{t+1} \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} p_t \\ x_t \\ y_t \end{bmatrix} - \begin{bmatrix} \frac{\partial G_1}{\partial p} & \frac{\partial G_1}{\partial x} & \frac{\partial G_1}{\partial y} \\ \frac{\partial G_2}{\partial p} & \frac{\partial G_2}{\partial x} & \frac{\partial G_2}{\partial y} \\ \frac{\partial G_3}{\partial p} & \frac{\partial G_3}{\partial x} & \frac{\partial G_3}{\partial y} \end{bmatrix}^{-1} \begin{bmatrix} G_1(x_t) \\ G_2(x_t, y_t) \\ G_3(x_t, y_t) \end{bmatrix}.$$

- Here, apply the power iteration method to compute  $\Lambda(x_t; C), \Lambda'(x_t; C), \Lambda''(x_t; C), \Lambda(x_t, y_t; D), \Lambda_x(x_t, y_t; D), \Lambda_y(x_t, y_t; D), \Lambda_{xx}(x_t, y_t; D), \Lambda_{yy}(x_t, y_t; D)$ , and  $\Lambda_{xy}(x_t, y_t; D)$ .
- Increment  $t$  by one.

- Set  $p^* \leftarrow p_t, x^* \leftarrow x_t, y^* \leftarrow y_t$ .

(2A) If  $0 \leq p^* \leq 1$ , set  $R_{MR}(\delta) \leftarrow 2 \log \Lambda(x^*; C) + \delta \log y^* - \log \Lambda(x^*, y^*; D)$ .

(2B) Otherwise,

- If  $p^* < 0$ , set  $p^* \leftarrow 0, x^* \leftarrow 0$ , and  $y^* \leftarrow$  solution of  $G_3(0, y) = 0$ .
- If  $p^* > 1$ , set  $p^* \leftarrow 1, x^* \leftarrow \infty$ , and  $y^* \leftarrow$  solution of  $G_3(\infty, y) = 0$ .

Finally, set  $R_{LB}(\delta) \leftarrow 2 \log \Lambda(x^*; C) + \delta \log y^* - \log \Lambda(x^*, y^*; D)$ .

**Remark 1.** Let  $p^*$  be the value computed at Step 1. When  $p^*$  falls outside the interval  $[0, 1]$ , we set  $p^* \in \{0, 1\}$ , and we argued earlier that the value returned  $R_{LB}(\delta)$  (at Step 2B) is, at most,  $R_{MR}(\delta)$ . Nevertheless, we conjecture that  $R_{LB}(\delta) = R_{MR}(\delta)$ .

As before, we develop a plotting procedure that minimizes the use of Newton–Raphson iterations.

We note that we have three scenarios for  $\Delta(x)$ . If  $\Delta(x)$  is monotone decreasing, then  $\delta_{max} = \lim_{x \rightarrow 0} \Delta(x)$  and we set  $x^\# = 0$ . If  $\Delta(x)$  is monotone increasing, then  $\delta_{max} = \lim_{x \rightarrow \infty} \Delta(x)$  and we set  $x^\# = \infty$ . Otherwise,  $\Delta(x)$  is maximized for some positive value and we set  $x^\#$  to be this value. Next, to obtain the GV-MR curve (see Remark 2); we iterate over  $x \in [1, x^\#]$ . It should be noted that if  $y(x^\#) < 1$  or, equivalently,  $\delta(x^\#) < \delta_{max}$ , we obtain a lower bound on the GV-MR curve by iterating over  $y \in [y(x^\#), 1]$ . Similar to Theorem 1, we define

$$\rho_{MR}(x) \triangleq 2 \log \Lambda(x; C) + \delta(x) \log y(x) - \log \Lambda(x, y(x); D), \tag{15}$$

and

$$\rho_{LB}(y) \triangleq 2 \log \Lambda(x^\#; C) + \delta(y) \log y - \log \Lambda(x^\#, y; D). \tag{16}$$

Finally, we state the following analogue of Theorem 1.

**Theorem 3.** We define  $\delta_{max}, x^\#$  as before. For  $x \in [1, x^\#]$ , we set

$$\begin{aligned}
 p(x) &\leftarrow x\Lambda'(x; \mathbf{C}) / \Lambda(x; \mathbf{C}), \\
 y(x) &\leftarrow \text{solution of } G_2(x, y) = 0, \\
 \delta(x) &\leftarrow y(x)\Lambda_y(x, y(x); \mathbf{D}) / \Lambda(x, y(x); \mathbf{D}),
 \end{aligned}$$

If  $y(x^\#) < 1$ , then for  $y \in [y(x^\#), 1]$ , we set

$$\delta(y) \leftarrow y\Lambda_y(x^\#, y; \mathbf{D}) / \Lambda(x^\#, y; \mathbf{D}),$$

then, the corresponding GV-MR curve is given by

$$\{(\delta(x), \rho_{\text{MR}}(x)) : x \in [1, x^\#]\} \cup \{(\delta(y), \rho_{\text{LB}}(y)) : y \in [y(x^\#), 1]\} \cup \{(\delta, 0) : \delta \geq \delta_{\text{max}}\}. \tag{17}$$

where  $\rho_{\text{MR}}$  and  $\rho_{\text{LB}}$  are defined in (15) and (16), respectively.

**Example 3.** We continue our example and evaluate the GV-MR bound for the (3, 2)-SWCC constrained system. In this case, the matrices of interest are

$$\mathbf{C}_{\mathcal{G}}(z) = \begin{bmatrix} z & 1 & 0 \\ 0 & 0 & z \\ z & 0 & 0 \end{bmatrix} \text{ and } \mathbf{D}_{\mathcal{G} \times \mathcal{G}}(x, y) = \begin{bmatrix} x^2 & 2xy & 0 & 1 & 0 & 0 \\ 0 & 0 & x^2 & 0 & xy & 0 \\ x^2 & xy & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x^2 \\ 0 & 0 & x^2 & 0 & 0 & 0 \\ x^2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Here, we observe that  $\Delta(x)$  is a monotone decreasing function and so, we set  $x^\# = 0.01$  and  $\delta_{\text{max}} = \lim_{x \rightarrow 0} \Delta(x) \approx 0.426$ . If we apply Procedure 2 to compute  $R_{\text{MR}}(\delta)$  for 100 points in  $[0, \delta_{\text{max}}]$ , we require 437 Newton–Raphson iterations and 85,500 power iterations. In contrast, we use Theorem 3 to compute  $(\delta(x), \rho_{\text{MR}}(x))$  for 100 values of  $x$  in the interval  $[1, x^\#]$ . This requires 323 Newton–Raphson iterations and involves 22,296 power iterations. The resulting GV-MR curve is given in Figure 1a.

**Remark 2.** Strictly speaking, the GV-MR curve described by (17) may not be equal to the curve defined by the optimization problem (15). Nevertheless, the curve provides a lower bound for the optimal asymptotic code rates and we conjecture that the GV-MR curve described by (17) is a lower bound for the curve defined by the optimization problem (15).

### 5. Single-State Graph Presentation

In this section, we focus on graph presentations that have exactly one state. Here, we allow these single-state graph presentations to contain the parallel edges and their labels to be binary strings of length possibly greater than one. Now, for these constrained systems, the procedures to evaluate the GV bound and its MR improvements can be greatly simplified. This is because the matrices  $\mathbf{B}_{\mathcal{G} \times \mathcal{G}}(y)$ ,  $\mathbf{C}_{\mathcal{G}}(z)$ , and  $\mathbf{D}_{\mathcal{G} \times \mathcal{G}}(x, y)$  are all of dimensions one by one. Therefore, determining their respective dominant eigenvalues is straightforward and does not require the power iteration method. The results in this section follow directly from previous sections and our objective is to provide explicit formulas whenever possible.

Formally, let  $\mathcal{S}$  be the constrained system with graph presentation  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$  such that  $|\mathcal{V}| = 1$  and  $\mathcal{L} : \mathcal{E} \rightarrow \Sigma^s$  with  $s \geq 1$  (existing methods that determine the GV bound for constrained systems with  $|\mathcal{V}| \geq 1$  assume that the edge-labels have single letters, i.e.,  $s = 1$ . In other words, previous methods developed in [12,14] do not apply).

We further define  $\alpha_t \triangleq \#\{(x, y) \in \mathcal{L}(\mathcal{E})^2 : d_{\mathbf{H}}(x, y) = t\}$  for  $0 \leq t \leq s$ . Then, the corresponding adjacency and reduced distance matrices are as follows:

$$\mathbf{A}_{\mathcal{G}} = [|\mathcal{E}|] \text{ and } \mathbf{B}_{\mathcal{G} \times \mathcal{G}}(y) = [\sum_{t \geq 0} \alpha_t y^t].$$

Then, we compute the capacity using its definition as  $\text{Cap}(\mathcal{S}) = (\log |\mathcal{E}|) / s$ .

To compute  $\tilde{T}(\delta)$ , we consider the following extension of the optimization problem (3) for the case  $s \geq 1$ :

$$\begin{aligned} \tilde{T}(\delta) &= \frac{1}{s} \inf\{-\delta s \log y + \log \lambda(y; \mathbf{B}) : 0 \leq y \leq 1\} \\ &= \frac{1}{s} \inf\left\{-\delta s \log y + \log\left(\sum_{t \geq 0} \alpha_t y^t\right) : 0 \leq y \leq 1\right\}. \end{aligned} \tag{18}$$

As before, following the convexity of the objective function in (18), we have that the optimal  $y$  is the zero (in the interval  $[0, 1]$ ) of the function

$$F(y) \triangleq \sum_{t \geq 0} (t - \delta s) \alpha_t y^t. \tag{19}$$

So, for fixed values of  $\delta$ , we can use the Newton–Raphson procedure to compute the root  $y$  of (19), and, hence, evaluate  $R_{GV}(\delta)$ . It should be noted that the power iteration method is not required in this case.

On the other hand, to plot the GV curve, we have the following corollary of Theorem 1.

**Corollary 2.** *Let  $\mathcal{G}$  be the single-state graph presentation for a constrained system  $\mathcal{S}$ . Then, the corresponding GV curve is given by*

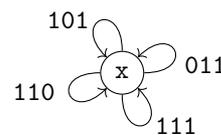
$$\mathcal{GV}(\mathcal{S}) \triangleq \{(\delta, R_{GV}(\delta)) : \delta \in [0, 1]\} = \{(\delta(y), \rho(y)) : y \in [0, 1]\} \cup \{(\delta, 0) : \delta \geq \delta_{\max}\}, \tag{20}$$

where

$$\begin{aligned} \delta_{\max} &= \frac{\sum_{t \geq 0} t \alpha_t}{s |\mathcal{E}|^2}, \\ \delta(y) &= \frac{\sum_{t \geq 0} t \alpha_t y^t}{s (\sum_{t \geq 0} \alpha_t y^t)}, \\ \rho(y) &= \frac{1}{s} \left( \log \frac{|\mathcal{E}|^2}{\sum_{t \geq 0} \alpha_t y^t} - \frac{\sum_{t \geq 0} t \alpha_t y^t}{\sum_{t \geq 0} \alpha_t y^t} \log y \right). \end{aligned}$$

We illustrate this evaluation procedure via an example of the class of *subblock energy-constrained codes* (SECCs). Formally, we fix a *subblock length*  $L$  and *energy constraint*  $w$ . A binary word  $x$  of length  $mL$  is said to satisfy the  $(L, w)$ -subblock energy constraint if we partition  $x$  into  $m$  subblocks of length  $L$ , then the number of ones in every subblock is at least  $w$ . We refer to the collection of words that meet this constraint as an  $(L, w)$ -SECC constrained system. The class of SECCs was introduced by Tandon et al. for the application of simultaneous energy and information transfer [7]. Later, in [21], a GV-type bound was introduced (see Proposition 12 in [21] and also, (28)) and we make comparisons with the GV bound (20) in the following example.

**Example 4.** *Let  $L = 3$  and  $w = 2$  and we consider a  $(3, 2)$ -SECC constrained system. It is straightforward to observe that the graph presentation is as follows with the single state  $x$ . Here,  $s = L = 3$ .*



Then, the corresponding adjacency and reduced distance matrices are as follows:

$$A_{\mathcal{G}} = [4], B_{\mathcal{G} \times \mathcal{G}}(y) = [4 + 6y + 6y^2].$$

First, we determine the GV bound at  $\delta = 1/3$ . We observe that  $F(y) = -4 + 6y^2$  and, so, the optimal point  $y$  for (18) is  $\sqrt{2/3}$  (the unique solution to  $F(y)$  in the interval  $[0, 1]$ ). Hence, we have that  $\tilde{T}(1/3) \approx 1.327$ . On the other hand, the capacity of a (3, 2)-SECC constrained system is  $\text{Cap}(\mathcal{S}) = 2/3$ . Therefore, the GV bound is given by  $R_{\text{GV}}(1/3) = 0.006$ .

In contrast, the GV-type lower bound given by Proposition 12 in [21] is zero for  $\delta > 0.174$ . Hence, the evaluation of the GV bound yields a significantly better lower bound. In fact, we can show that  $R_{\text{GV}}(\delta) > 0$  for all  $\delta \leq \delta_{\text{max}} = 3/8$ .

To plot the GV curve, using the fact that  $\delta_{\text{max}} = 3/8$ , we have that

$$\mathcal{GV}(\mathcal{S}) = \left\{ \left( \frac{y + 2y^2}{2 + 3y + 3y^2}, \frac{1}{3} \log \frac{8}{2 + 3y + 3y^2} + \frac{3y + 6y^2}{2 + 3y + 3y^2} \log y \right) : y \in [0, 1] \right\} \cup \left\{ (\delta, 0) : \delta \geq \frac{3}{8} \right\}.$$

We plot the curve in Section 6.

From this example, we see that our methods yield better lower bounds in terms of asymptotic coding rates for a specific pair of  $(L, w)$ . It is open to determine how much improvement can be achieved for general pairs of  $L$  and  $w$ .

Next, we evaluate the GV-MR bound. To this end, we consider some proper subset  $\mathcal{P} \subset \mathcal{E}$  and define

$$\begin{aligned} \alpha_t &\triangleq \#\{(\mathbf{x}, \mathbf{y}) \in \mathcal{L}(\mathcal{E})^2 : d_{\mathbf{H}}(\mathbf{x}, \mathbf{y}) = t, \mathbf{x}, \mathbf{y} \in \mathcal{P}\}, \\ \beta_t &\triangleq \#\{(\mathbf{x}, \mathbf{y}) \in \mathcal{L}(\mathcal{E}) : d_{\mathbf{H}}(\mathbf{x}, \mathbf{y}) = t, (\mathbf{x} \in \mathcal{P}, \mathbf{y} \notin \mathcal{P}) \text{ or } (\mathbf{x} \notin \mathcal{P}, \mathbf{y} \in \mathcal{P})\}, \\ \gamma_t &\triangleq \#\{(\mathbf{x}, \mathbf{y}) \in \mathcal{L}(\mathcal{E}) : d_{\mathbf{H}}(\mathbf{x}, \mathbf{y}) = t, \mathbf{x}, \mathbf{y} \notin \mathcal{P}\}. \end{aligned}$$

Then, we consider the following matrices:

$$\mathbf{C}_{\mathcal{G}}(z) = [|\mathcal{E}| - |\mathcal{P}| + |\mathcal{P}|z] \text{ and } \mathbf{D}_{\mathcal{G} \times \mathcal{G}}(x, y) = [\sum_{t \geq 0} (\alpha_t x^2 + \beta_t x + \gamma_t) y^t].$$

Setting  $p$  to be the normalized frequency of edges in  $\mathcal{P}$ , we obtain  $S(p)$  by solving the optimization problem (10).

Specifically, we have that

$$S(p) = \frac{1}{s} (H(p) + p + \log |\mathcal{P}| + (1 - p) \log (|\mathcal{E}| - |\mathcal{P}|)), \tag{21}$$

and this value is achieved when

$$z = \frac{p(|\mathcal{E}| - |\mathcal{P}|)}{(1 - p)|\mathcal{P}|}. \tag{22}$$

To compute  $\tilde{T}(p, \delta)$ , we consider the following extension of the optimization problem (12) for the case  $s \geq 1$ .

$$\begin{aligned} \tilde{T}(p, \delta) &= \frac{1}{s} \inf\{-2p \log x - \delta s \log y + \log \lambda(y; \mathbf{D}) : 0 \leq y \leq 1\} \\ &= \frac{1}{s} \inf\left\{-2p \log x - \delta s \log y + \log \left(\sum_{t \geq 0} (\alpha_t x^2 + \beta_t x + \gamma_t) y^t\right) : 0 \leq y \leq 1\right\}. \end{aligned} \tag{23}$$

As before, following the convexity of the objective function in (23), we have that the optimal  $x$  and  $y$  are the zeroes (in the interval  $[0, 1]$ ) of the functions

$$\begin{aligned} G_2(x, y) &\triangleq 2(1 - p) \left(\sum_{t \geq 0} \alpha_t y^t\right) x^2 + (1 - 2p) \left(\sum_{t \geq 0} \beta_t y^t\right) x - 2p \left(\sum_{t \geq 0} \gamma_t y^t\right) \\ G_3(x, y) &\triangleq \sum_{t \geq 0} (t - \delta s) (\alpha_t x^2 + \beta_t x + \gamma_t) y^t \end{aligned} \tag{24}$$

So, for fixed values of  $p$  and  $\delta$ , we can use the Newton–Raphson procedure to compute the roots  $x$  and  $y$  of (24), and, hence, evaluate  $R_{GV}(p, \delta)$ . It should be noted that the power iteration method is not required in this case. We find  $x^\#$  as defined in Section 4 and set

$$\rho_{MR}(x) \triangleq 2 \log(|\mathcal{E}| - |\mathcal{P}| + |\mathcal{P}|x) + \delta(x) \log y(x) - \log \sum_{t \geq 0} (\alpha_t x^2 + \beta_t x + \gamma_t) y(x)^t. \quad (25)$$

Furthermore, if  $y(x^\#) < 1$ , we set

$$\rho_{LB}(y) \triangleq 2 \log(|\mathcal{E}| - |\mathcal{P}| + |\mathcal{P}|x^\#) + \delta(y) \log y - \log \sum_{t \geq 0} (\alpha_t (x^\#)^2 + \beta_t x^\# + \gamma_t) y^t. \quad (26)$$

Next, to plot the GV-MR curve, we have the following corollary of Theorem 3.

**Corollary 3.** *Let  $\mathcal{G}$  be the single-state graph presentation for a constrained system  $\mathcal{S}$ . For  $x \in [1, x^\#]$ , we set*

$$p(x) = \frac{|\mathcal{P}|x}{(|\mathcal{E}| - |\mathcal{P}|) + |\mathcal{P}|x},$$

$$\delta(x) = \frac{\sum_{t \geq 1} t(\alpha_t x^2 + \beta_t x + \gamma_t) y(x)^t}{s \sum_{t \geq 0} (\alpha_t x^2 + \beta_t x + \gamma_t) y(x)^t},$$

where  $y(x)$  is the smallest root of the equation

$$2(|\mathcal{E}| - |\mathcal{P}|)(\sum_{t \geq 0} \alpha_t y^t)x + (|\mathcal{E}| - |\mathcal{P}| - |\mathcal{P}|x)(\sum_{t \geq 0} \beta_t y^t) - 2|\mathcal{P}|(\sum_{t \geq 0} \gamma_t y^t) = 0.$$

If  $y(x^\#) < 1$ , then for  $y \in [y(x^\#), 1]$ , we set

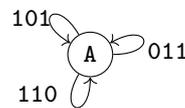
$$\delta(y) = \frac{\sum_{t \geq 1} t(\alpha_t (x^\#)^2 + \beta_t x^\# + \gamma_t) y^t}{s \sum_{t \geq 0} (\alpha_t (x^\#)^2 + \beta_t x^\# + \gamma_t) y^t}.$$

Then, the corresponding GV-MR curve is given by

$$\{(\delta(x), \rho_{MR}(x)) : x \in [1, x^\#]\} \cup \{(\delta(y), \rho_{LB}(y)) : y \in [y(x^\#), 1]\} \cup \{(\delta, 0) : \delta \geq \delta_{\max}\}. \quad (27)$$

where  $\rho_{MR}$  and  $\rho_{LB}$  are defined in (25) and (26), respectively.

**Example 5.** We continue our example and evaluate the GV-MR bound for the (3,2)-SECC constrained system. We have the following single-state graph presentation:



Then, the matrices of interest are:

$$C_{\mathcal{G}} = [1 + 3z], \quad D_{\mathcal{G} \times \mathcal{G}}(x, y) = [(3 + 6y^2)x^2 + 6xy + 1].$$

Since  $C_{\mathcal{G}}$  and  $D_{\mathcal{G} \times \mathcal{G}}(x, y)$  are both singleton matrices, we have  $\Lambda(z; C) = 1 + 3z$  and  $\Lambda(x, y; D) = (3 + 6y^2)x^2 + 6xy + 1$ . Then,  $G_1(z) = -p(1 + 3z) + 3z$ ,  $G_2(x, y) = 3(1 + 2y^2)x^2(1 - p) + 3xy(1 - 2p) - p$  and  $G_3(x, y) = 4x^2y^2 - 3\delta(1 + 2y^2)x^2 + 2xy(1 - 3\delta) - \delta$ . Now, we apply Theorem 2 and express  $p, y$ , and  $\delta$  in terms of  $x$  where  $x \in [1, x^\#]$  where  $x^\# \rightarrow \infty$ .

$$\begin{aligned}
 p &= \frac{3x}{1+3x} \\
 y &= \frac{x-1}{2x} \\
 \delta &= \frac{2x(x-1)}{9x^2-1}
 \end{aligned}$$

Now, we observe that we have  $y(x^\#) = 1/2$ . Since we can still increase  $y$  to 1, we apply the GV bound with  $p = 1$  and  $x = z = x^\#$  once we reach the boundary that is  $p = 1$ . Hence, at the boundary, we solve the following problem:

$$\begin{aligned}
 S(1) &= 2 \log 3 \\
 \tilde{T}(1, \delta) &= \inf \{ -2 \log x - 3\delta \log y + \log(3(1+2y^2)x^2 + 6xy + 1) : 1/2 \leq y \leq 1; x = x^\# \rightarrow \infty \} \\
 &= \inf \{ -3\delta \log y + \log 3 + \log(1+2y^2) : 1/2 \leq y \leq 1 \} \\
 R_{MR}(\delta) &= S(1) - \tilde{T}(1, \delta).
 \end{aligned}$$

By setting  $F(y) = -3\delta(1+2y^2) + 4y^2 = 0$ , we get  $\delta = 4y^2/3(1+2y^2)$  where  $y \in [1/2, 1]$  and we plot the respective curve.

### 6. Numerical Plots

In this section, we apply our numerical procedures to compute the GV and the GV-MR bounds for some specific constrained systems. In particular, we consider the  $(L, w)$ -SWCC constrained systems defined in Section 3, the ubiquitous  $(d, k)$ -runlength limited systems (see, for example, p. 3 in [11]) and the  $(L, w)$ -subblock energy constrained codes recently introduced in [7]. In addition to the GV and GV-MR curves, we also plot a simple lower bound. For each  $\delta \in [0, 1/2]$ , any ball size is at most  $2\mathbb{H}(\delta n)$ . So, for any constrained system  $\mathcal{S}$ , we have that  $\tilde{T}(\delta) \leq \text{Cap}(\mathcal{S}) + \mathbb{H}(\delta)$ . Therefore, we have that

$$R(\delta; \mathcal{S}) \leq \text{Cap}(\mathcal{S}) - \mathbb{H}(\delta). \tag{28}$$

From the plots in Figures 1–3, it is also clear that the computations of (7) and (17) yield a significantly better lower bound.

#### 6.1. $(L, w)$ -Sliding Window Constrained Codes

We fix  $L$  and  $w$ . We recall from Section 3 that a binary word satisfies the  $(L, w)$ -sliding window weight constraint if the number of ones in every consecutive  $L$  bits is at least  $w$  and the  $(L, w)$ -SWCC constrained system refers to the collection of words that meet this constraint. From [8,9], we have a simple graph presentation that uses only  $\binom{L}{w}$  states. To validate our methods, we choose  $(L, w) \in \{(3, 2), (10, 7)\}$  and the corresponding graph presentations have 3 and 120 states, respectively. Applying the plotting procedures described in Theorems 1 and 3, we obtain Figure 1.

#### 6.2. $(d, k)$ -Runlength Limited Codes

Next, we revisit the ubiquitous runlength constraint. We fix  $d$  and  $k$ . We say that a binary word satisfies the  $(d, k)$ -RLL constraint if each run of zeroes in the word has a length of at least  $d$  and at most  $k$ . Here, we allow the first and last runs of zeroes to have a length of less than  $d$ . We refer to the collection of words that meet this constraint as a  $(d, k)$ -RLL constrained system. It is well known that a  $(d, k)$ -RLL constrained system has the graph presentation with  $k + 1$  states (see, for example, [11]). Here, we choose  $(d, k) \in \{(1, 3), (3, 7)\}$  to validate our methods and apply Theorems 1 and 3 to obtain Figure 2. For  $(d, k) = (3, 7)$ , we corroborate our results with those derived in [15]. Specifically, Winick and Yang determined the GV bound (1) for the  $(3, 7)$ -RLL constraint and remarked that

the “evaluation of the (GV-MR) bound required considerable computation” for “a small improvement”. In Table 3, we verify this statement.

**Table 3.** Comparison of the GV-MR bound with lower bound [15] for (3,7)-RLL constrained systems.

$\delta$	GV-MR Bound (15)	GV Bound [15] (see Equation (1))
0	0.406	0.406
0.05	0.255	0.225
0.1	0.163	0.163
0.15	0.095	0.094
0.2	0.048	0.044
0.25	0.018	0.012

### 6.3. $(L, w)$ -Subblock Energy-Constrained Codes

We fix  $L$  and  $w$ . We recall from Section 5 that a binary word satisfies the  $(L, w)$ -subblock energy constraint if each subblock of length  $L$  has a weight of at least  $w$  and the  $(L, w)$ -SECC constrained system refers to the collection of words that meet this constraint. Then, the corresponding graph presentation has a single state  $x$  with  $\sum_{i=0}^w \binom{L}{i}$  edges, where each edge is labeled by a word of length  $L$  and weight at least  $w$ . We apply the methods in Section 5 to determine the GV and GV-MR bounds.

For the GV bound, we provide the explicit formula for  $\alpha_t$  and proceed as in Example 4.

$$\alpha_t = \binom{L}{t} (|\mathcal{E}| - \sum_{j=1}^t \sum_{k=0}^{\lceil \frac{j}{2} \rceil - 1} \binom{L-t}{w-j+k} \binom{t}{k}) \tag{29}$$

Similarly, for GV-MR bound, we provide the explicit formula for  $\alpha_t$ ,  $\beta_t$ , and  $\gamma_t$  and proceed as in Example 5.

$$\alpha_t = \binom{L}{w} \binom{L-w}{i/2} \binom{w}{i/2} \text{ if } t \text{ is even, otherwise, } \alpha_t = 0. \tag{30}$$

$$\beta_t = 2 \binom{L}{w} \sum_{j=1}^{\lfloor \frac{t}{2} \rfloor} \binom{L-w}{t-j} \binom{w}{j} - 2\alpha_t \tag{31}$$

$$\gamma_t = \binom{L}{t} (|\mathcal{E}| - \sum_{j=1}^t \sum_{k=0}^{\lceil \frac{j}{2} \rceil - 1} \binom{L-t}{w-j+k} \binom{t}{k}) - \alpha_t - \beta_t \tag{32}$$

In Figure 3, we plot the GV bound and GV-MR bounds. We remark that the simple lower bound (28) corresponds to Proposition 12 in [21].

**Author Contributions:** Conceptualization, K.G. and H.M.K.; software, K.G.; writing—original draft preparation, K.G.; writing—review and editing, K.G. and H.M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work of Han Mao Kiah was supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 Award under Grant MOE-T2EP20121-0007 and MOE AcRF Tier 1 Award under Grant RG19/23.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Acknowledgments:** The authors would also like to thank the assistant editor for her skillful handling and the anonymous reviewers for their valuable suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

### Appendix A. Power Iteration Method for Derivatives of Dominant Eigenvalues

Throughout this appendix, we assume that  $A$  is a diagonalizable matrix with dominant eigenvalue  $\lambda_1$  and whose corresponding eigenspace has dimension one. Let  $e_1$  be the unit eigenvector whose entries are positive in this space. Then, the power iteration method is a well-known numerical procedure that finds the dominant eigenvalue  $\lambda_1$  and the corresponding eigenvector  $e_1$  efficiently.

Now, in the preceding sections, the entries in the matrix  $A$  are given functions in either one or two variables and, thus, the dominant eigenvalue  $\lambda_1$  is a function in the same variables. Moreover, the numerical procedures in these sections require us to compute the higher order (partial) derivatives of this dominant eigenvalue function  $\lambda_1$ . To the best of our knowledge, we are unaware of any algorithms or numerical procedures that estimate the values of these derivatives. Hence, in this appendix, we modify the power iteration method to compute these estimates.

Formally, let  $A$  be an irreducible nonnegative diagonalizable square matrix with dominant eigenvalue  $\lambda_1$  and corresponding unit eigenvector  $e_1$ . Since  $A$  is diagonalizable,  $A$  has  $n$  eigenvectors  $e_1, e_2, \dots, e_n$  that form an orthonormal basis for  $\mathbb{R}^n$ . Let  $\lambda_1, \lambda_2, \dots, \lambda_n$  be the corresponding eigenvalues and, so, we have that

$$Ae_i = \lambda_i e_i \text{ for all } i = 1, 2, \dots, n. \quad (A1)$$

Since  $A$  is irreducible, the dominant eigenspace has dimension one and, also, the dominant eigenvalue is real and positive. Therefore, we can assume that  $\lambda_1 > |\lambda_2| \geq \dots \geq |\lambda_n|$ .

We first assume that the entries of  $A$  are functions in the variable  $z$ . Hence,  $\lambda_i$  and the entries of  $e_i$  are functions in  $z$  too. Then Power Iteration I then evaluates both  $\lambda_1$  and  $\lambda_1'$  for some fixed value of  $z$ , while Power Iteration II additionally evaluates the second order derivative  $\lambda_1''$ .

The case where the entries of  $A$  are functions in two variables  $x$  and  $y$  is discussed at the end of the appendix. Here, Power Iteration III evaluates higher order partial derivatives of  $\lambda_1$  for certain fixed values of  $x$  and  $y$ . For ease of exposition, we provide detailed proofs for the correctness of Power Iteration I and the proofs can be extended for Power Iteration II and Power Iteration III.

We continue our discussion where the entries of  $A$  are univariate functions in  $z$ . We differentiate each entry of  $A$  with respect to  $z$  to obtain the matrix  $A'$ . Furthermore, for all  $1 \leq i \leq n$ , we differentiate each entry of eigenvectors  $e_i$  and the eigenvalue  $\lambda_i$  to obtain  $e_i'$  and  $\lambda_i'$ , respectively. Specifically, it follows from (A1) that

$$A'e_i + Ae_i' = \lambda_i' e_i + \lambda_i e_i' \text{ for all } i = 1, 2, \dots, n. \quad (A2)$$

Then, the following procedure computes both  $\lambda_1$  and  $\lambda_1'$ .

#### Power Iteration I.

INPUT: Irreducible nonnegative diagonalizable matrix  $A$

OUTPUT: Estimates of  $\lambda_1$  and  $\lambda_1'$

(1) Initialize  $q^{(0)}$  such that all its entries are strictly positive.

- Fix a tolerance value  $\epsilon$ .
- While  $|q^{(k)} - q^{(k-1)}| > \epsilon$ ,
  - Set

$$\lambda^{(k)} = \|Aq^{(k-1)}\|,$$

$$q^{(k)} = \frac{Aq^{(k-1)}}{\lambda^{(k)}},$$

$$\begin{aligned} \mu^{(k)} &= \|A'q^{(k-1)} + Ar^{(k-1)} - \lambda^{(k)}r^{(k-1)}\|, \\ r^{(k)} &= \frac{Ar^{(k-1)} + A'q^{(k-1)} - \mu^{(k)}q^{(k-1)}}{\lambda^{(k)}}. \end{aligned}$$

– Increment  $k$  by one.

(2) Set  $\lambda_1 \leftarrow \lambda^{(k)}$  and  $\lambda'_1 \leftarrow \mu^{(k)}$ .

**Theorem A1.** *If  $A$  is an irreducible nonnegative diagonalizable matrix and  $q^{(0)}$  has positive components with unit norm, then, as  $k \rightarrow \infty$ , we have*

$$\lambda^{(k)} \rightarrow \lambda_1, q^{(k)} \rightarrow e_1, \mu^{(k)} \rightarrow \lambda'_1.$$

Here,  $q^{(k)} \rightarrow e_1$  means that  $\|q^{(k)} - e_1\| \rightarrow 0$  as  $k \rightarrow \infty$ .

Before we present the proof of Theorem A1, we remark that the usual power iteration method computes only  $\lambda^{(k)}$  and  $q^{(k)}$ . Then, it is well-known (see, for example, [22]) that  $\lambda^{(k)}$  and  $q^{(k)}$  tend to  $\lambda_1$  and  $e_1$ , respectively.

Now, since  $e_i$  spans  $\mathbb{R}^n$ , we can write  $q^{(0)} = \sum_{i=1}^n \alpha_i e_i$  for any initial vector  $q^{(0)}$ . The next technical lemma provides closed formulas for  $\lambda^{(k)}$ ,  $q^{(k)}$ ,  $\mu^{(k)}$ , and  $r^{(k)}$  in terms of  $\lambda_i$ ,  $e_i$  and  $\alpha_i$ .

**Lemma A1.** *Let  $q^{(0)} = \sum_{i=1}^n \alpha_i e_i$ . Then,*

$$q^{(k)} = \frac{\sum_{i=1}^n \alpha_i \lambda_i^k e_i}{\|\sum_{i=1}^n \alpha_i \lambda_i^k e_i\|}, \tag{A3}$$

$$\lambda^{(k)} = \frac{\|\sum_{i=1}^n \alpha_i \lambda_i^k e_i\|}{\|\sum_{i=1}^n \alpha_i \lambda_i^{k-1} e_i\|}, \tag{A4}$$

$$r^{(k)} = \frac{\sum_{i=1}^n (\alpha_i e'_i + \alpha'_i e_i) \lambda_i^k + (k\lambda'_i - \sum_{j=1}^k \mu^{(j)}) \alpha_i \lambda_i^{k-1} e_i}{\|\sum_{i=1}^n \alpha_i \lambda_i^k e_i\|}, \tag{A5}$$

$$\mu^{(k)} = \frac{\left\| \sum_{i=1}^n (\alpha_i e'_i + \alpha'_i e_i) \lambda_i^{k-1} (\lambda_i - \lambda^{(k)}) + \alpha_i \lambda_i^{k-1} \lambda'_i e_i + ((k-1)\lambda'_i - \sum_{j=1}^{k-1} \mu^{(j)}) \alpha_i \lambda_i^{k-2} (\lambda_i - \lambda^{(k)}) e_i \right\|}{\|\sum_{i=1}^n \alpha_i \lambda_i^{k-1} e_i\|}. \tag{A6}$$

**Proof.** Since  $q^{(k)}$  is defined recursively as  $q^{(k)} = \frac{Aq^{(k-1)}}{\lambda^{(k)}} = \frac{Aq^{(k-1)}}{\|Aq^{(k-1)}\|}$ , we have that

$$q^{(k)} = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|}.$$

Then, it follows from Equation (A1) that

$$A^k q^{(0)} = A^k \sum_{i=1}^n \alpha_i e_i = \sum_{i=1}^n \alpha_i (A^k e_i) = \sum_{i=1}^n \alpha_i \lambda_i^k e_i, \tag{A7}$$

and, so, we obtain (A3). Similarly, from (A1), we have that

$$\lambda^{(k)} = \|Aq^{(k-1)}\| = \frac{\|A^k q^{(0)}\|}{\|A^{k-1} q^{(0)}\|} = \frac{\|\sum_{i=1}^n \alpha_i \lambda_i^k e_i\|}{\|\sum_{i=1}^n \alpha_i \lambda_i^{k-1} e_i\|},$$

as required for (A4).

Next, we note that  $\mathbf{r}^{(0)} = \sum_{i=1}^n \alpha_i \mathbf{e}'_i + \sum_{i=1}^n \alpha'_i \mathbf{e}_i$ . Then, using the recursive definition of  $\mathbf{r}^{(k)}$ , we have

$$\mathbf{r}^{(k)} = \frac{\mathbf{A}^k \mathbf{r}^{(0)} + \sum_{j=0}^{k-1} \mathbf{A}^j \mathbf{A}' \mathbf{A}^{k-j-1} \mathbf{q}^{(0)} - (\sum_{j=1}^k \mu^{(j)}) \mathbf{A}^{k-1} \mathbf{q}^{(0)}}{\|\mathbf{A}^k \mathbf{q}^{(0)}\|}. \tag{A8}$$

Then, from (A1), we have

$$\mathbf{A}^k \mathbf{r}^{(0)} = \mathbf{A}^k \left( \sum_{i=1}^n \alpha_i \mathbf{e}'_i + \sum_{i=1}^n \alpha'_i \mathbf{e}_i \right) = \sum_{i=1}^n \alpha_i (\mathbf{A}^k \mathbf{e}'_i) + \sum_{i=1}^n \alpha'_i \lambda_i^k \mathbf{e}_i. \tag{A9}$$

and, from (A2),

$$\mathbf{A}' \sum_{i=1}^n \alpha_i \lambda_i^{k-j-1} \mathbf{e}_i = \sum_{i=1}^n \alpha_i \lambda_i^{k-j-1} (\mathbf{A}' \mathbf{e}_i) = \sum_{i=1}^n \alpha_i \lambda_i^{k-j-1} (\lambda'_i \mathbf{e}_i + \lambda_i \mathbf{e}'_i - \mathbf{A} \mathbf{e}'_i).$$

Therefore, using (A1) again,

$$\begin{aligned} \sum_{j=0}^{k-1} \mathbf{A}^j \mathbf{A}' \sum_{i=1}^n \alpha_i \lambda_i^{k-j-1} \mathbf{e}_i &= \sum_{j=0}^{k-1} \mathbf{A}^j \sum_{i=1}^n \alpha_i \lambda_i^{k-j-1} (\lambda'_i \mathbf{e}_i + \lambda_i \mathbf{e}'_i - \mathbf{A} \mathbf{e}'_i) \\ &= k \sum_{i=1}^n \alpha_i \lambda_i^{k-1} \lambda'_i \mathbf{e}_i + \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{e}'_i - \sum_{i=1}^n \alpha_i (\mathbf{A}^k \mathbf{e}'_i). \end{aligned}$$

Therefore, we obtain (A5).

Finally, we recall that  $\mu^{(k)}$  is defined as

$$\mu^{(k)} = \|\mathbf{A}' \mathbf{q}^{(k-1)} + \mathbf{A} \mathbf{r}^{(k-1)} - \lambda^{(k)} \mathbf{r}^{(k-1)}\|.$$

Then, by replacing  $\mathbf{r}^{(k-1)}$  and  $\mathbf{q}^{(k-1)}$  from (A5) and (A3), respectively, and then using Equation (A2), we obtain (A6).  $\square$

Finally, we are ready to demonstrate the correctness of Power Iteration I.

**Proof of Theorem A1.** Since  $\mathbf{A}$  is an irreducible nonnegative diagonalizable matrix,  $\lambda_1$  is real positive and there exists  $0 < \epsilon < 1$  such that  $\frac{|\lambda_i|}{\lambda_1} < \epsilon$  for all  $i = 2, 3, \dots, n$  (see, for example, [11]). For purposes of brevity, we write

$$\Phi_k = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{e}_i \tag{A10}$$

and, so, we can rewrite (A3) as

$$\mathbf{q}^{(k)} = \frac{\Phi_k}{\|\Phi_k\|} = \frac{\lambda_1^k}{\|\Phi_k\|} \frac{\Phi_k}{\lambda_1^k} = \frac{\lambda_1^k}{\|\Phi_k\|} \left( \alpha_1 \mathbf{e}_1 + \sum_{i=2}^n \alpha_i \frac{\lambda_i^k}{\lambda_1^k} \mathbf{e}_i \right).$$

Now, since  $\lambda_i^k / \lambda_1^k \leq \epsilon^k$  for all  $i = 2, \dots, n$ , we have that

$$\left\| \frac{\Phi_k}{\lambda_1^k} - \alpha_1 \mathbf{e}_1 \right\| \leq C_1 \epsilon^k \text{ for some constant } C_1. \tag{A11}$$

Then, using the triangle inequality, we have that as  $k \rightarrow \infty$ ,  $\left| \frac{\|\Phi_k\|}{\lambda_1^k} - \alpha_1 \right| \rightarrow 0$  and, thus,  $\frac{\lambda_1^k}{\|\Phi_k\|} \rightarrow \frac{1}{\alpha_1}$ . Therefore,  $\|\mathbf{q}^{(k)} - \mathbf{e}_1\| \rightarrow 0$  as required.

It should be noted that since  $\frac{\lambda_1^k}{\|\Phi_k\|}$  tends to a finite limit, we have that  $\frac{\lambda_1^k}{\|\Phi_k\|}$  is bounded above by some constant. In other words, we have that

$$\frac{\lambda_1^k}{\|\Phi_k\|} \leq C_2 \text{ for some constant } C_2. \tag{A12}$$

Next, we show the following inequality:

$$|\lambda^{(k)} - \lambda_1| \leq C_3 \epsilon^{k-1} \text{ for some constant } C_3. \tag{A13}$$

Using (A4), we have that

$$\frac{\|\Phi_k - \lambda_1 \Phi_{k-1}\|}{\|\Phi_{k-1}\|} = \frac{\lambda_1^{k-1}}{\|\Phi_{k-1}\|} \frac{\sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{e}_i - \alpha_i \lambda_1 \lambda_i^{k-1} \mathbf{e}_i}{\lambda_1^{k-1}} = \left( \frac{\lambda_1^{k-1}}{\|\Phi_{k-1}\|} \right) \cdot \lambda_1 \cdot \sum_{i=2}^n \alpha_i \left( \frac{\lambda_i^k}{\lambda_1^k} - \frac{\lambda_i^{k-1}}{\lambda_1^{k-1}} \right) \mathbf{e}_i.$$

Now, observe that  $\left( \frac{\lambda_i^k}{\lambda_1^k} - \frac{\lambda_i^{k-1}}{\lambda_1^{k-1}} \right) \leq 2\epsilon^{k-1}$  for  $i = 2, \dots, n$ . Since  $\frac{\lambda_1^{k-1}}{\|\Phi_{k-1}\|} \leq C_2$ , we have (A13) after applying the triangle inequality.

Again, to reduce clutter, we introduce the following abbreviations:

$$\begin{aligned} D_k &= \sum_{i=1}^n (\alpha_i \mathbf{e}'_i + \alpha'_i \mathbf{e}_i) \lambda_i^{k-1} (\lambda_i - \lambda^{(k)}), \\ E_k &= \sum_{i=1}^n \alpha_i \lambda_i^{k-1} \lambda'_i \mathbf{e}_i, \\ F_k &= \sum_{i=1}^n \left( (k-1) \lambda'_i - \sum_{j=1}^{k-1} \mu^{(j)} \right) \alpha_i \lambda_i^{k-2} (\lambda_i - \lambda^{(k)}) \mathbf{e}_i. \end{aligned}$$

Thus, we can rewrite (A6) as

$$\mu^{(k)} = \frac{\|D_k + E_k + F_k\|}{\|\Phi_{k-1}\|} \leq \lambda'_1 + \frac{\|D_k\|}{\|\Phi_{k-1}\|} + \frac{\|E_k - \lambda'_1 \Phi_{k-1}\|}{\|\Phi_{k-1}\|} + \frac{\|F_k\|}{\|\Phi_{k-1}\|}.$$

Next, we bound each of the summands on the right-hand side. Specifically, we show the following inequalities:

$$\frac{\|D_k\|}{\|\Phi_{k-1}\|} + \frac{\|E_k - \lambda'_1 \Phi_{k-1}\|}{\|\Phi_{k-1}\|} \leq C_4 \epsilon^{k-1} \text{ for some constant } C_4, \tag{A14}$$

$$\frac{\|F_k\|}{\|\Phi_{k-1}\|} \leq C_5 (k-1) \epsilon^{k-1} + C_5 \left( \sum_{j=1}^{k-1} \mu^{(j)} \right) \epsilon^{k-1} \text{ for some constant } C_5. \tag{A15}$$

To demonstrate (A14), we consider

$$\frac{\|D_k\|}{\lambda_1^{k-1}} = \left\| \sum_{i=1}^n (\alpha_i \mathbf{e}'_i + \alpha'_i \mathbf{e}_i) \frac{\lambda_i^{k-1}}{\lambda_1^{k-1}} (\lambda_i - \lambda^{(k)}) \right\| \leq \|\alpha_1 \mathbf{e}'_1 + \alpha'_1 \mathbf{e}_1\| |\lambda_1 - \lambda^{(k)}| + \epsilon^{k-1} \sum_{i=2}^n \|\alpha_i \mathbf{e}'_i + \alpha'_i \mathbf{e}_i\| |\lambda_i - \lambda^{(k)}|.$$

We use (A13) to bound the first summand by some constant multiple of  $\epsilon^{k-1}$ . On the other hand, we have  $|\lambda_i - \lambda^{(k)}| \leq |\lambda_i - \lambda_1| + |\lambda_1 - \lambda^{(k)}| \leq \max\{|\lambda_i - \lambda_1| : 2 \leq i \leq n\} + C_3 \epsilon^{k-1}$  for  $2 \leq i \leq n$ . In other words, the second summand is also bounded by some constant multiple of  $\epsilon^{k-1}$ . Next, we consider

$$\frac{\|E_k - \lambda'_1 \Phi_{k-1}\|}{\lambda_1^{k-1}} = \left\| \sum_{i=1}^n \alpha_i \frac{\lambda_i^{k-1}}{\lambda_1^{k-1}} (\lambda'_i - \lambda'_1) \mathbf{e}_i \right\| \leq \epsilon^{k-1} \sum_{i=2}^n |\alpha_i (\lambda'_i - \lambda'_1)|.$$

and, so,  $\frac{\|E_k - \lambda'_1 \Phi_{k-1}\|}{\lambda_1^{k-1}}$  is also bounded by a multiple of  $\epsilon^{k-1}$ . Therefore, since  $\frac{\lambda_1^{k-1}}{\|\Phi_{k-1}\|} \leq C_2$ , we have (A14). Using similar methods, we can establish (A15).

Next, we apply (A14) and then recursively apply (A15) until the right-hand side is free of  $\mu^{(i)}$ s. Then, it follows that

$$\mu^{(k)} \leq \lambda'_1 + C_4 \epsilon^{k-1} + C_5(k-1)\epsilon^{k-1} + \prod_{j=2}^{k-1} (1 + C_5 \epsilon^{k-j}) + C_5 \epsilon^{k-1} \sum_{i=1}^{k-1} (\lambda'_1 + C_4 \epsilon^{k-i-1} C_5(k-i-1)\epsilon^{k-i-1}) \prod_{j=2}^i (1 + C_5 \epsilon^{k-j}). \tag{A16}$$

Furthermore, since  $i \leq k-1$ ,  $\prod_{j=2}^i (1 + C_5 \epsilon^{k-j}) \leq \prod_{j=2}^{k-1} (1 + C_5 \epsilon^{k-j})$ , we can rewrite (A16) as

$$\mu^{(k)} \leq \lambda'_1 + C_4 \epsilon^{k-1} + C_5(k-1)\epsilon^{k-1} + \prod_{j=2}^{k-1} (1 + C_5 \epsilon^{k-j}) \left( 1 + C_5 \epsilon^{k-1} \sum_{i=1}^{k-1} (\lambda'_1 + C_4 \epsilon^{k-i-1} C_5(k-i-1)\epsilon^{k-i-1}) \right). \tag{A17}$$

Next, it follows from standard calculus that  $\prod_{j=2}^{k-1} (1 + C_5 \epsilon^{k-j}) < e^{\frac{C_5}{1-\epsilon}}$ . Furthermore, since  $\epsilon < 1$ , we have  $\sum_{i=0}^{k-2} \epsilon^j < \frac{1}{1-\epsilon}$  and  $\sum_{i=0}^{k-2} j\epsilon^j < \frac{1}{(1-\epsilon)^2}$ . Putting everything together, we have

$$\mu^{(k)} \leq \lambda'_1 + C_4 \epsilon^{k-1} + C_5(k-1)\epsilon^{k-1} + C_5 \epsilon^{k-1} e^{\frac{C_5}{1-\epsilon}} \left( 1 + (k-1)\lambda'_1 + \frac{C_4}{1-\epsilon} + \frac{C_5}{(1-\epsilon)^2} \right). \tag{A18}$$

As  $k \rightarrow \infty$ , since  $\epsilon < 1$ , we have  $\epsilon^k \rightarrow 0$  and  $k\epsilon^k \rightarrow 0$ . Therefore,  $\lim_{k \rightarrow \infty} \mu^{(k)} \leq \lambda'_1$ . Using similar methods, we have that  $\lim_{k \rightarrow \infty} \mu^{(k)} \geq \lambda'_1$  and, so,  $\lim_{k \rightarrow \infty} \mu^{(k)} = \lambda'_1$ , as required.  $\square$

Next, we modify Power Iteration I so as to compute the higher order derivatives. We omit a detailed proof as it is similar to the proof of Theorem A1.

**Power Iteration II.**

INPUT: Irreducible nonnegative diagonalizable matrix  $A$

OUTPUT: Estimates of  $\lambda_1$ ,  $\lambda'_1$ , and  $\lambda''_1$

(1) Initialize  $\mathbf{q}^{(0)}$  such that all its entries are strictly positive.

- Fix a tolerance value  $\epsilon$ .
- While  $|\mathbf{q}^{(k)} - \mathbf{q}^{(k-1)}| > \epsilon$ ,

– Set

$$\lambda^{(k)} = \|\mathbf{A}\mathbf{q}^{(k-1)}\|,$$

$$\mathbf{q}^{(k)} = \frac{\mathbf{A}\mathbf{q}^{(k-1)}}{\lambda^{(k)}},$$

$$\mu^{(k)} = \|\mathbf{A}'\mathbf{q}^{(k-1)} + \mathbf{A}\mathbf{r}^{(k-1)} - \lambda^{(k)}\mathbf{r}^{(k-1)}\|,$$

$$\mathbf{r}^{(k)} = \frac{\mathbf{A}\mathbf{r}^{(k-1)} + \mathbf{A}'\mathbf{q}^{(k-1)} - \mu^{(k)}\mathbf{q}^{(k-1)}}{\lambda^{(k)}},$$

$$\nu^{(k)} = \|\mathbf{A}''\mathbf{q}^{(k-1)} + 2\mathbf{A}'\mathbf{r}^{(k-1)} + \mathbf{A}\mathbf{s}^{(k-1)} - \lambda^{(k)}\mathbf{s}^{(k-1)} - 2\mu^{(k)}\mathbf{r}^{(k-1)}\|,$$

$$\mathbf{s}^{(k)} = \frac{\mathbf{A}''\mathbf{q}^{(k-1)} + 2\mathbf{A}'\mathbf{r}^{(k-1)} + \mathbf{A}\mathbf{s}^{(k-1)} - 2\mu^{(k)}\mathbf{r}^{(k-1)} - \nu^{(k)}\mathbf{q}^{(k-1)}}{\lambda^{(k)}}.$$

– Increment  $k$  by one.

(2) Set  $\lambda_1 \leftarrow \lambda^{(k)}$ ,  $\lambda'_1 \leftarrow \mu^{(k)}$  and  $\lambda''_1 \leftarrow \nu^{(k)}$ .

**Theorem A2.** If  $A$  is an irreducible nonnegative diagonalizable matrix and  $\mathbf{q}^{(0)}$  has positive components with unit norm, then, as  $k \rightarrow \infty$ , we have

$$\lambda^{(k)} \rightarrow \lambda_1, \mathbf{q}^{(k)} \rightarrow \mathbf{e}_1, \mu^{(k)} \rightarrow \lambda'_1, \nu^{(k)} \rightarrow \lambda''_1.$$

Finally, we end this appendix with a power iteration method that computes the partial derivatives when the elements of the given matrix are bivariate functions.

**Power Iteration III.**

INPUT: Irreducible nonnegative diagonalizable matrix  $A$

OUTPUT: Estimates of  $\lambda_1, (\lambda_1)_x, (\lambda_1)_y, (\lambda_1)_{xx}, (\lambda_1)_{yy},$  and  $(\lambda_1)_{xy}$

(1) Initialize  $q^{(0)}$  such that all its entries are strictly positive.

- Fix a tolerance value  $\epsilon$ .
- While  $|q^{(k)} - q^{(k-1)}| > \epsilon,$ 
  - Set

$$\lambda^{(k)} = \|Aq^{(k-1)}\|,$$

$$q^{(k)} = \frac{Aq^{(k-1)}}{\lambda^{(k)}},$$

$$\lambda_x^{(k)} = \|A_x q^{(k-1)} + A q_x^{(k-1)} - \lambda q_x^{(k-1)}\|,$$

$$q_x^{(k)} = \frac{A_x q^{(k-1)} + A q_x^{(k-1)} - \lambda_x^{(k-1)} q^{(k-1)}}{\lambda^{(k)}},$$

$$\lambda_y^{(k)} = \|A_y q^{(k-1)} + A q_y^{(k-1)} - \lambda q_y^{(k-1)}\|,$$

$$q_y^{(k)} = \frac{A_y q^{(k-1)} + A q_y^{(k-1)} - \lambda_y^{(k-1)} q^{(k-1)}}{\lambda^{(k)}},$$

$$\lambda_{xx}^{(k)} = \|A_{xx} q^{(k-1)} + 2A_x q_x^{(k-1)} + A q_{xx}^{(k-1)} - \lambda^{(k-1)} q_{xx}^{(k-1)} - 2\lambda_x^{(k-1)} q_x^{(k-1)}\|,$$

$$q_{xx}^{(k)} = \frac{A_{xx} q^{(k-1)} + 2A_x q_x^{(k-1)} + A q_{xx}^{(k-1)} - 2\lambda_x^{(k-1)} q_x^{(k-1)} - \lambda_{xx}^{(k-1)} q^{(k-1)}}{\lambda^{(k)}}$$

$$\lambda_{yy}^{(k)} = \|A_{yy} q^{(k-1)} + 2A_y q_y^{(k-1)} + A q_{yy}^{(k-1)} - \lambda^{(k-1)} q_{yy}^{(k-1)} - 2\lambda_y^{(k-1)} q_y^{(k-1)}\|,$$

$$q_{yy}^{(k)} = \frac{A_{yy} q^{(k-1)} + 2A_y q_y^{(k-1)} + A q_{yy}^{(k-1)} - 2\lambda_y^{(k-1)} q_y^{(k-1)} - \lambda_{yy}^{(k-1)} q^{(k-1)}}{\lambda^{(k)}}$$

$$\lambda_{xy}^{(k)} = \|A_{xy} q^{(k-1)} + A_x q_y^{(k-1)} + A_y q_x^{(k-1)} + A q_{xy}^{(k-1)} - \lambda^{(k-1)} q_{xy}^{(k-1)} - \lambda_x^{(k-1)} q_y^{(k-1)} - \lambda_y^{(k-1)} q_x^{(k-1)}\|,$$

$$q_{xy}^{(k)} = \frac{A_{xy} q^{(k-1)} + A_x q_y^{(k-1)} + A_y q_x^{(k-1)} + A q_{xy}^{(k-1)} - \lambda_{xy}^{(k-1)} q^{(k-1)} - \lambda_x^{(k-1)} q_y^{(k-1)} - \lambda_y^{(k-1)} q_x^{(k-1)}}{\lambda^{(k)}}.$$

- Increment  $k$  by one.

- Set  $\lambda^{(k)} \leftarrow \lambda_1, \lambda_x^{(k)} \leftarrow (\lambda_1)_x, \lambda_y^{(k)} \leftarrow (\lambda_1)_y, \lambda_{xx}^{(k)} \leftarrow (\lambda_1)_{xx}, \lambda_{yy}^{(k)} \leftarrow (\lambda_1)_{yy}, \lambda_{xy}^{(k)} \leftarrow (\lambda_1)_{xy}.$

**Theorem A3.** If  $A$  is an irreducible nonnegative diagonalizable matrix and  $q^{(0)}$  has positive components with unit norm, then, as  $k \rightarrow \infty,$  we have  $\lambda_{xx}^{(k)} \rightarrow (\lambda_1)_{xx}, \lambda_{yy}^{(k)} \rightarrow (\lambda_1)_{yy}, \lambda_{xy}^{(k)} \rightarrow (\lambda_1)_{xy}.$

**References**

1. Yazdi, S.M.H.T.; Kiah, H.M.; Garcia-Ruiz, E.; Ma, J.; Zhao, H.; Milenkovic, O. DNA-Based Storage: Trends and Methods. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2015**, *1*, 230–248. [CrossRef]
2. Immink, K.A.S.; Cai, K. Efficient balanced and maximum homopolymer-run restricted block codes for DNA-based data storage. *IEEE Commun. Lett.* **2019**, *23*, 1676–1679. [CrossRef]
3. Nguyen, T.T.; Cai, K.; Immink, K.A.S.; Kiah, H.M. Capacity-Approaching Constrained Codes with Error Correction for DNA-Based Data Storage. *IEEE Trans. Inf. Theory* **2021**, *67*, 5602–5613. [CrossRef]
4. Kovačević, M.; Vukobratović, D. Asymptotic Behavior and Typicality Properties of Runlength-Limited Sequences. *IEEE Trans. Inf. Theory* **2022**, *68*, 1638–1650. [CrossRef]
5. Popovski, P.; Fouladgar, A.M.; Simeone, O. Interactive joint transfer of energy and information. *IEEE Trans. Commun.* **2013**, *61*, 2086–2097. [CrossRef]
6. Fouladgar, A.M.; Simeone, O.; Erkip, E. Constrained codes for joint energy and information transfer. *IEEE Trans. Commun.* **2014**, *62*, 2121–2131. [CrossRef]

7. Tandon, A.; Motani, M.; Varshney, L.R. Subblock-constrained codes for real-time simultaneously energy and information transfer. *IEEE Trans. Inf. Theory* **2016**, *62*, 4212–4227. [[CrossRef](#)]
8. Immink, K.A.S.; Cai, K. Block Codes for Energy-Harvesting Sliding- Window Constrained Channels. *IEEE Commun. Lett.* **2020**, *24*, 2383–2386. [[CrossRef](#)]
9. Immink, K.A.S.; Cai, K. Properties and Constructions of Energy-Harvesting Sliding-Window Constrained Codes. *IEEE Commun. Lett.* **2020**, *24*, 1890–1893. [[CrossRef](#)]
10. Wu, T.Y.; Tandon, A.; Varshney, L.R.; Motani, M. Skip-sliding window codes. *IEEE Trans. Commun.* **2021**, *69*, 2824–2836. [[CrossRef](#)]
11. Marcus, B.H.; Roth, R.M.; Siegel, P.H. *An Introduction to Coding for Constrained Systems*; Lecture Notes; 2001. Available online: <https://ronny.cswp.cs.technion.ac.il/wp-content/uploads/sites/54/2016/05/chapters1-9.pdf> (accessed on 1 October 2020).
12. Kolesnik, V.D.; Krachkovsky, V.Y. Generating functions and lower bounds on rates for limiting error-correcting codes. *IEEE Trans. Inf. Theory* **1991**, *37*, 778–788. [[CrossRef](#)]
13. Gu, J.; Fuja, T. A generalized Gilbert-Varshamov bound derived via analysis of a code-search algorithm. *IEEE Trans. Inf. Theory* **1993**, *39*, 1089–1093. [[CrossRef](#)]
14. Marcus, B.H.; Roth, R.M. Improved Gilbert-Varshamov bound for constrained systems. *IEEE Trans. Inf. Theory* **1992**, *38*, 1213–1221. [[CrossRef](#)]
15. Winick, K.A.; Yang, S.H. Upper bounds on the size of error-correcting runlength-limited codes. *Eur. Trans. Telecommun.* **1996**, *37*, 273–283. [[CrossRef](#)]
16. Goyal, K.; Kiah, H.M. Evaluating the Gilbert-Varshamov Bound for Constrained Systems. In Proceedings of the 2022 IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, 26 Jun–1 July 2022; pp. 1348–1353.
17. Tolhuizen, L.M.G.M. The generalized Gilbert-Varshamov bound is implied by Turan’s theorem. *IEEE Trans. Inf. Theory* **1997**, *43*, 1605–1606. [[CrossRef](#)]
18. Luenberger, D.G. *Introduction to Linear and Nonlinear Programming*; Addison-Wesley: Reading, MA, USA, 1973.
19. Rockafellar, T. *Convex Analysis*; Princeton University: Pressrinceton, NJ, USA, 1970.
20. Kashyap, N.; Roth, R.M.; Siegel, P.H. The Capacity of Count-Constrained ICI-Free Systems. In Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 1592–1596.
21. Tandon, A.; Kiah, H.M.; Motani, M. Bounds on the size and asymptotic rate of subblock-constrained codes. *IEEE Trans. Inf. Theory* **2018**, *64*, 6604–6619. [[CrossRef](#)]
22. Stewart, G.W. *Introduction to Matrix Computations*; Computer Science and Applied Mathematics; Academic Press: New York, NY, USA, 1973.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.