ON CODES WITH LOCAL JOINT CONSTRAINTS

Bruce Moision*

Jet Propulsion Laboratory, 4800 Oak Grove Drive, MS 238-420, Pasadena, CA 91109-8099

Alon Orlitsky

Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, California 92093-0407

Abstract

We study the largest number of sequences with the property that any two sequences do not contain specified pairs of patterns. We show that this number increases exponentially with the length of the sequences and that the exponent, or capacity, is the logarithm of the joint spectral radius of an appropriately defined set of matrices. We illustrate a new heuristic for computing the joint spectral radius and use it to compute the capacity for several simple collections. The problem of computing the achievable rate region of a collection of codes is introduced and it is shown that the region may be computed via a similar analysis.

Key words: joint spectral radius, generalized spectral radius, information capacity, forbidden words
PACS:

1 Introduction

In [1] collections of sequences with the property that the difference of any pair does not contain a pattern from a specified set were studied. Sequences with this property have been applied as the basis of codes in magnetic recording

* Corresponding Author.

Email addresses: bmoision@jpl.nasa.gov (Bruce Moision), alon@ece.ucsd.edu (Alon Orlitsky).
channels [2]. It was shown that the number of such sequences increases exponentially with their length and that the exponent, or \textit{capacity}, is the logarithm of the joint spectral radius of an appropriately defined set of matrices. In this paper we introduce two generalizations of this problem and provide a new heuristic for computing the joint spectral radius.

In the first generalization, we consider collections of sequences with the property that any pair does not contain a pair of patterns from a specified set. Extensions to disallow larger collections, e.g. triples, is straightforward. Sequences with this property would be better suited for channels with multiple-user or inter-track interference, e.g. [3,4], or channels whose performance is characterized by pairs or triples rather than differences, e.g. [5,6]. We show that the maximum growth rate of the number of such sequences is the joint spectral radius of a certain set of matrices, and determine descriptions of collections of sequences which achieve the maximum growth rate.

In the second generalization, we consider the achievable rates of a pair of codes such that the two codes do not jointly contain pairs of patterns from a specified set. We show that the upper bound on the sum of the rates is similarly given by the joint spectral radius of an appropriately defined set, and illustrate an algorithm for computing a tight lower bound on the rate region.

Underlying the solutions for these various problems is the computation of the joint spectral radius, which has been shown to be NP-hard [7]. We illustrate a new heuristic for computing the joint spectral radius, and use it to compute the capacity for several simple collections, giving new examples and extending prior results from [1].

The paper is organized as follows. In the next section we formally describe the first generalization. Section 3 states the connection to the joint spectral radius. In Section 4 we review known algorithms for determining the joint spectral radius and illustrate a new heuristic, computing the capacity for several collections of pairs. Finally, in Section 5 we introduce and discuss the rate pairs problem.

2 Notation and definitions

For simplicity, we assume sequences are binary. A \textit{pattern} is a finite string of bits. A \textit{joint pattern} is a set of two distinct equal-length patterns. Let \( \mathcal{J} \) denote a collection of joint patterns. An \( n \)-bit \textit{code} \( \mathcal{C} \) is a collection of \( n \)-bit sequences, or \textit{codewords}. \( \mathcal{C} \) avoids \( \mathcal{J} \) if for all \( u, v \in \mathcal{C} \) and all \( i \leq j \) in \([1, n]\),

\[
\{u[i,j], v[i,j]\} \notin \mathcal{J}
\]  \hspace{1cm} (1)
where, for all $i \leq j$, we use the notation

$$[i, j] \overset{\text{def}}{=} \{i, \ldots, j\}$$

and

$$u_{[i, j]} \overset{\text{def}}{=} u_i, \ldots, u_j.$$ 

We are interested in

$$\delta_n(\mathcal{J}) \overset{\text{def}}{=} \max\{|C| : C \text{ avoids } \mathcal{J}\},$$

the size of the largest $n$-bit code that avoids $\mathcal{J}$. It is easy to verify that $\delta_n(\mathcal{J})$ is *sub-multiplicative*, i.e.,

$$\delta_{n_1 + n_2}(\mathcal{J}) \leq \delta_{n_1}(\mathcal{J}) \cdot \delta_{n_2}(\mathcal{J})$$

for all $n_1, n_2 > 0$. Hence, by the Sub-Additivity Lemma, e.g., [8], we can define the *capacity* of $\mathcal{J}$ as the limit

$$\text{cap}(\mathcal{J}) \overset{\text{def}}{=} \log \left[ \lim_{n \to \infty} (\delta_n(\mathcal{J}))^{1/n} \right]. \quad (2)$$

We would like to determine the capacities of various sets $\mathcal{J}$ and find codes that achieve them.

We are primarily interested in finite collections. Without loss of generality we can assume from here on that all patterns in $\mathcal{J}$ have the same length $m$. Otherwise, let $m$ be the length of the longest pattern in $\mathcal{J}$ and replace every pair of length $m' < m$ by its $2^{m-m'}$ extensions of length $m$.

With this equal-length assumption, we restate constraint (1) and require that for all $u, v \in \mathcal{C}$ and all $i \in [1, n + m - 1]$,

$$\{u_{[i, i + m - 1]}, v_{[i, i + m - 1]}\} \notin \mathcal{J}$$

where, for $i$ and $n$ only, we let

$$i' \overset{\text{def}}{=} i + m - 1$$

and

$$n' \overset{\text{def}}{=} n - m + 1.$$ 

Note also that we use the term *pattern* to refer to strings of length $m$ and *sequence* for strings of length $n$.

A generalization of the problem to allow $\mathcal{J}$ to be a collection of sets of arbitrary sizes is straightforward. In this paper, we address the case of sets of pairs to simplify the presentation.
3 From disallowed pairs to joint spectral radius

In this section we show that the capacity of a collection of joint patterns is the joint spectral radius of an appropriately defined set of matrices. The proof is presented via a sequence of lemmas in the following sections.

3.1 Disallowed joint patterns

Let $\mathcal{C}$ be an $n$-bit code. Let

$$P_i \eqdef \{ u_{i,i+m-1} : u \in \mathcal{C} \}$$

be the set of $m$-bit patterns present in columns $[i, i + m - 1]$, and let

$$M_i \eqdef \{0, 1 \}^m - P_i$$

be the set of $m$-bit patterns missing from those columns.

**Lemma 1** Let $\mathcal{C}$ be an $n$-bit code and let $M_1, \ldots, M_n$ be as defined in (3). Then $\mathcal{C}$ avoids $\mathcal{J}$ iff for all $i \in [1, n + m - 1]$ and all $J \in \mathcal{J}$,

$$J \cap M_i \neq \emptyset$$

(4)

**Proof:** If $\{p, p'\} \in \mathcal{J}$ is a disallowed joint pattern, then $p$ and $p'$ cannot both appear in any set $P_i$ of a code that avoids $\mathcal{J}$. Hence, $\{p, p'\} \cap M_i$ is not empty for all $i$.

Conversely, if (4) holds, then for every $u, v \in \mathcal{C}$ and all $i \in [1, n + m - 1]$, \{u_{i,i+m-1}, v_{i,i+m-1}\} $\notin \mathcal{J}$.

3.2 Representing sets

A set $M \subseteq \{0, 1 \}^m$ represents or is a representing set for $\mathcal{J}$ if it intersects every set in $\mathcal{J}$. It is minimal if, in addition, none of its strict subsets represents $\mathcal{J}$. Clearly, every representing set contains a minimal one. Let $\mathcal{M}(\mathcal{J})$ be the collection of all minimal representing sets for $\mathcal{J}$.

In general, finding the smallest size of a minimal representing set, and therefore finding all of them, is NP hard, e.g., [9, SP8]. However, in the cases we consider, $m$ is fixed and typically small, hence finding $\mathcal{M}(\mathcal{J})$ is usually not difficult.

Equation (4) implies the following lemma.
Lemma 2 If a code \( C \) avoids \( J \), then, for every \( i \in [1, n + m - 1] \), the set \( M_i \)
defined in (3) contains a set \( M'_i \in \mathcal{M}(J) \). □

We can think of the minimal representing sets as the smallest candidate sets of patterns which must be missing from columns \([i, i + m - 1]\) in a code that avoids \( J \).

3.3 Disallowed sets

Let \( M_1, \ldots, M_{n'} \subseteq \{0, 1\}^m \) be sets of \( m \)-bit patterns. An \( n \)-bit
sequence \( s_1, \ldots, s_n \) avoids the set sequence \( M_1, \ldots, M_{n'} \) if \( s_{[i,i+m-1]} \notin M_i \) for all \( i \in [1, n + m - 1] \). We think of these sets as missing from \( s_1, \ldots, s_n \). Let \( \mu(M_1, \ldots, M_{n'}) \)
be the number of \( n \)-bit sequences that avoid \( M_1, \ldots, M_{n'} \). Note that if \( M_i \supseteq M'_i \) for all \( i \in [1, n + m - 1] \), then

\[
\mu(M_1, \ldots, M_{n'}) \leq \mu(M'_1, \ldots, M'_{n'}).
\] (5)

If \( \mathcal{M} \) is a collection of sets in \( \{0, 1\}^m \), we let

\[
\mu_n(\mathcal{M}) \overset{\text{def}}{=} \max \{ \mu(M_1, \ldots, M_{n'}) : M_i \in \mathcal{M} \ \forall i \}
\]

be the largest number of \( n \)-bit sequences avoiding a sequence of sets in \( \mathcal{M} \).

Note that unlike disallowed pairs which constrain pairs of sequences, disallowed sets constrain individual sequences. We will show later that this type of constraint is easier to analyze, and we now prove that it leads to the same capacity.

Lemma 3 For every \( n \),

\[
\delta_n(J) = \mu_n(\mathcal{M}(J)).
\]

Proof: Consider a code \( C \) that avoids \( J \) and achieves \( \delta_n(J) \). For \( i \in [1, n + m - 1] \)
let \( M_i \) be the set defined in (3), and let \( M'_i \) be the sets in \( \mathcal{M}(J) \) indicated in
Lemma 2. Then, using (5),

\[
\delta_n(J) = |C| \leq \mu(M_1, \ldots, M_{n'}) \\
\leq \mu(M'_1, \ldots, M'_{n'}) \leq \mu_n(\mathcal{M}(J))
\]

where the first inequality follows as, by definition, each codeword in \( C \) avoids \( M_1, \ldots, M_{n'} \), the second because, as Lemma 2 showed, for each \( M_i \) there exists \( M'_i \) in \( \mathcal{M}(J) \) such that \( M'_i \supseteq M_i \) hence \( \mu(M_i, \ldots, M_{n'}) \leq \mu(M'_i, \ldots, M'_{n'}) \), and the third from the definition of \( \mu_n(\mathcal{M}(J)) \).
To establish the reverse inequality, note that if $M_1, \ldots, M_{n'} \in \mathcal{M}(\mathcal{J})$, then, by Lemma 1, the set of all $n$-bit sequences avoiding $M_1, \ldots, M_{n'}$ avoids $\mathcal{J}$. □

3.4 Bipartite and cascade graphs

In the previous subsection we reduced the constraint on pairs of sequences to a constraint on individual sequences. We now convert this problem to that of counting paths in graphs.

A bipartite graph $(L, R, E)$ consists of a set $L$ of left vertices, a set $R$ of right vertices, and a set $E$ of edges. Each edge $(l, r) \in E$ connects a left vertex $l \in L$ to a right vertex $r \in R$. Though we don’t draw their direction explicitly, we think of the edges as directed from left to right.

For $m \geq 2$, let $G_m$ be the bipartite graph where $L = R = \{0, 1\}^{m-1}$ and $(l_1, \ldots, l_{m-1}) \in L$ is connected to $(r_1, \ldots, r_{m-1}) \in R$ if $l_i = r_{i-1}$ for all $i = 2, \ldots, m-1$. We identify this edge with the $m$-bit sequence $l_1, l_2, \ldots, l_{m-1}, r_{m-1} = l_1, r_1, \ldots, r_{m-1}$. Fig. 1 illustrates $G_2$ and $G_3$.

For $M \subseteq \{0, 1\}^m$, define $G_M$ to be the bipartite graph obtained from $G_m$ by removing the edges corresponding to elements of $M$. Fig. 2 illustrates $G_{\{10\}}$ and $G_{\{101\}}$.

If $G_1, \ldots, G_{n'}$ are bipartite graphs with left vertex sets $L_1, \ldots, L_{n'}$ and right vertex sets $R_1, \ldots, R_{n'}$, respectively, such that $R_i = L_{i+1}$ for all $i \in [1, n'-1]$, we let

$$V_i \overset{\text{def}}{=} \begin{cases} \{1\} \times L_1 & \text{if } i = 1, \\ \{i\} \times R_{i-1} = \{i\} \times L_i & \text{if } 2 \leq i \leq n', \\ \{n'+1\} \times R_{n'} & \text{if } i = n'+1, \end{cases}$$

Fig. 1. $G_2$ and $G_3$
and define the cascade \([G_1, \ldots, G_{n'}]\) to be the graph whose vertex set is \(V_1 \cup \ldots \cup V_{n'+1}\) and where for \(i \in [1, n']\), the edges between \(V_i\) and \(V_{i+1}\) are the edges of \(G_i\), and there are no other edges. Drawing the vertices of each \(V_i\) vertically and to the left of the \(V_{i+1}\) vertices, we call the vertices \(V_1\) and \(V_{n'+1}\) leftmost and rightmost, respectively. Fig. 3 illustrates the cascade \([G_{\{10\}}, G_{\{01\}}, G_{\{10\}}]\).

A path in a cascade \([G_1, \ldots, G_{n'}]\) is a sequence \(v_1, \ldots, v_{n'+1}\) of vertices where each \(v_i \in V_i\), and \(v_i\) is connected to \(v_{i+1}\) for all \(i \in [1, n']\). Note that all paths connect a leftmost vertex to a rightmost vertex and proceed from left to right. We let \(\psi([G_1, \ldots, G_{n'}])\) be the total number of paths in the cascade. For example, in \([G_{\{10\}}, G_{\{01\}}, G_{\{10\}}]\) there are two paths from \((1, 0)\) to \((4, 0)\), three paths from \((1, 0)\) to \((4, 1)\), etc. Hence,

\[
\psi([G_{\{10\}}, G_{\{01\}}, G_{\{10\}}]) = 2 + 3 + 1 + 2 = 8.
\]

If \(M \subseteq \{0, 1\}^m\) and \(s \in \{0, 1\}^m\), then \(l = s_{[1, m-1]}\) and \(r = s_{[2, m]}\) are vertices of \(G_M\) with an edge from left node \(l\) to right node \(r\) if and only if \(s \notin M\), namely, \(s\) avoids \(M\). More generally, for \(n \geq m\), there is a bijection between \(n\)-bit sequences that avoid \(M_1, \ldots, M_{n'}\) and paths in the cascade \([G_{M_1}, \ldots, G_{M_{n'}}]\), hence

\[
\mu(M_1, \ldots, M_{n'}) = \psi([G_{M_1}, \ldots, G_{M_{n'}}]).
\]

Letting

\[
\psi_n(M) \overset{\text{def}}{=} \max\{\psi([G_{M_1}, \ldots, G_{M_{n'}}]) : M_i \in M \ \forall i\}
\]
we obtain

**Lemma 4**

\[ \mu_n(\mathcal{M}(\mathcal{J})) = \psi_n(\mathcal{M}(\mathcal{J})). \]

\[ \square \]

### 3.5 Adjacency matrices

Identifying the elements of \( L \) and \( R \) of a bipartite graph \( G = (L, R, E) \) with the intervals \([1, |L|]\) and \([1, |R|]\), respectively, we let the **adjacency matrix** \( A_G \) be the \(|L| \times |R|\) matrix whose \((l, r)\)th element is 1 if \((l, r) \in E\), and 0 otherwise.

Note that the \((l, r)\)th element of \( A_G \) is the number of edges from left node \( l \) to right node \( r \) in \( G \). Similarly, it can be shown that in the cascade \([G_1, \ldots, G_n]\), the number of left-to-right paths from leftmost vertex \( l \) to rightmost vertex \( r \) is the \((l, r)\)-th element of the product \( A_{G_1}A_{G_2} \cdots A_{G_n} \).

Letting

\[ \| A \|_1 = \sum_{l,r} |A_{l,r}| \]

(6)
de note the \( L_1 \) norm of the matrix \( A \), it follows that, for every \( M_1, \ldots, M_n \subseteq \{0, 1\}^m \),

\[ \psi([G_{M_1}, \ldots, G_{M_n}]) = \|A_{G_{M_1}} \cdots A_{G_{M_n}}\|_1. \]

Let

\[ \Sigma(\mathcal{J}) \overset{\text{def}}{=} \{ A_{G_{M}} : M \in \mathcal{M}(\mathcal{J}) \} \]

denote the set of adjacency matrices corresponding to the collection \( \mathcal{M}(\mathcal{J}) \) of minimal representing sets for the disallowed joint patterns \( \mathcal{J} \) (see sections 3.1 and 3.2 for definitions). Let

\[ \Sigma^n \overset{\text{def}}{=} \left\{ \prod_{i=1}^n A_i \in \Sigma \right\} \]

denote the set of products of \( n \) matrices in \( \Sigma \). Then, setting

\[ \hat{\rho}_n(\Sigma, \| \cdot \|_1) \overset{\text{def}}{=} \max \{ \| A \|_1 : A \in \Sigma^n \} \]

denote the maximum of the \( L_1 \) norm of \( n \) matrices in \( \Sigma \). For an arbitrary set \( \Sigma \subseteq \mathbb{C}^{n \times m} \), we get

**Lemma 5**

\[ \psi_n(\mathcal{M}(\mathcal{J})) = \hat{\rho}_n(\Sigma(\mathcal{J}), \| \cdot \|_1). \]

\[ \square \]

This suggests looking for algebraic methods to determine the capacity.
3.6 Matrix norms and spectral radius

A matrix norm for the set $\mathbb{C}^{n \times m}$ of complex square matrices is a mapping $\| \cdot \| : \mathbb{C}^{n \times m} \rightarrow [0, \infty)$ such that for all $A_1, A_2, A \in \mathbb{C}^{n \times m}$,

1. $\|A\| = 0$ iff $A = 0$,
2. $\|A_1 + A_2\| \leq \|A_1\| + \|A_2\|$, 
3. $\|A_1 \cdot A_2\| \leq \|A_1\| \cdot \|A_2\|$, 
4. $\|cA\| = |c| \cdot \|A\|$, $\forall c \in \mathbb{C}$.

**Example 1** Let $A \in \mathbb{C}^{n \times m}$. The $L_1$ norm, $\|A\|_1$, of $A$ was already defined in (6). The **maximum-column-sum norm** of $A$ is

$$\|A\|_\gamma \stackrel{\text{def}}{=} \max_{1 \leq j \leq m} \sum_{i=1}^{m} |A_{i,j}|,$$

and the **spectral norm** of $A \in \mathbb{C}^{n \times m}$ is

$$\|A\|_s \stackrel{\text{def}}{=} \max_{\|x\|_2=1} \|Ax\|_2,$$

where

$$\|x\|_2 \stackrel{\text{def}}{=} \left( \sum_{i=1}^{m} |x_i|^2 \right)^{1/2}$$

is the Euclidean norm of a vector $x \in \mathbb{C}^m$. It can be shown that $\| \cdot \|_1$, $\| \cdot \|_\gamma$, and $\| \cdot \|_s$ are all matrix norms.

One can show, e.g., [10, Theorem 5.4.4], that for any two matrix norms $\| \cdot \|_\alpha$, $\| \cdot \|_\beta$ there are constants $0 < c_1 < c_2$ such that for all $A \in \mathbb{C}^{n \times m}$,

$$c_1 \|A\|_\beta \leq \|A\|_\alpha \leq c_2 \|A\|_\beta.$$

(7)

By sub-multiplicativity of matrix norms, the limit

$$\hat{\rho}(A) \stackrel{\text{def}}{=} \lim_{n \to \infty} \|A^n\|^{1/n},$$

exists, and, by (7), is independent of the matrix norm $\| \cdot \|$. One can also define

$$\check{\rho}(A) \stackrel{\text{def}}{=} \max \{ |\lambda| : \lambda \text{ an eigenvalue of } A \}.$$ 

For any matrix norm and $A \in \mathbb{C}^{m \times m}$ we have, e.g., [10, Theorem 5.6.9],

$$\check{\rho}(A) \leq \|A\|.$$

(8)

It is also well known, e.g., [10, Corollary 5.6.14], that

$$\hat{\rho}(A) = \check{\rho}(A).$$
They are called the *spectral radius* of $A$ and denoted by $\rho(A)$.

3.7 Joint spectral radius

The quantities $\hat{\rho}$ and $\hat{\rho}$ can be generalized to sets of matrices. We begin with $\hat{\rho}$. Letting

$$\hat{\rho}_n(\Sigma, \| \cdot \|) \overset{\text{def}}{=} \sup \{ \| A \| : A \in \Sigma^n \}$$

for an arbitrary matrix norm $\| \cdot \|$ and set $\Sigma \subseteq \mathbb{C}^{m \times m}$, Rota and Strang [11] defined the *joint spectral radius* of $\Sigma$ to be

$$\hat{\rho}(\Sigma) \overset{\text{def}}{=} \lim_{n \to \infty} \hat{\rho}_n(\Sigma, \| \cdot \|)^{1/n},$$

where the limit exists by sub-multiplicativity, and (7) implies that it is independent of the norm $\| \cdot \|$.

Daubechies and Lagarias [12] defined the *generalized spectral radius* of $\Sigma$ to be

$$\check{\rho}(\Sigma) \overset{\text{def}}{=} \limsup_{n \to \infty} \hat{\rho}_n(\Sigma)^{1/n}$$

where

$$\hat{\rho}_n(\Sigma) \overset{\text{def}}{=} \sup \{ \hat{\rho}(A) : A \in \Sigma^n \}.$$

It follows from (8) that

$$\check{\rho}_n(\Sigma) \leq \hat{\rho}_n(\Sigma, \| \cdot \|)$$

for every $n$. Hence,

$$\check{\rho}(\Sigma) \leq \hat{\rho}(\Sigma),$$

and Daubechies and Lagarias conjectured that equality holds, namely

$$\check{\rho}(\Sigma) = \hat{\rho}(\Sigma),$$

as was proven by Berger and Wang [13] for all finite $\Sigma$. We denote this quantity by $\rho(\Sigma)$, and refer to it as the joint spectral radius.

Combining (2) and Lemmas 3 to 5, we obtain our main result:

**Theorem 1** For every finite $\mathcal{J}$,

$$\text{cap}(\mathcal{J}) = \log(\rho(\Sigma(\mathcal{J}))). \quad \Box$$

Namely, the capacity is the logarithm of the joint spectral radius of $\Sigma(\mathcal{J})$.

This equality generalizes known results on *constrained systems* where, instead of joint patterns, individual patterns are disallowed, and it is well known,
e.g., [14, Theorem 3.9], that the growth rate of the number of sequences, or 
Shannon capacity of the constraint, is \( \log(\hat{\rho}(\mathcal{A})) \), the logarithm of the spectral 
radius of a corresponding adjacency matrix \( A \).

The joint spectral radius measures the maximum growth rate of a product of 
matri ces drawn from the set \( \Sigma \). This concept appears in many applications. 
In addition to Rota and Strang’s original work in matrix theory [11], it has 
been used to study convergence of infinite products of matrices, e.g., [15], 
with applications to wavelets [12]. The concept is also related to the stability 
properties of discrete linear inclusions, e.g., [16,17], wherein the logarithm of 
the joint spectral radius is referred to as the Lyapunov indicator.

In the next section we describe several existing algorithms for computing the 
joint spectral radius and introduce a heuristic for determining the joint spec-
tral radius.

4 Computing the joint spectral radius

4.1 Computing the joint spectral radius is hard

Tsitsiklis and Blondel [7] have shown that approximating the joint spectral 
radius of a pair of matrices with \( \{0,1\} \) entries is NP-hard. In addition, they 
have shown [18] that determining whether \( \rho(\Sigma) \leq 1 \) when \( \Sigma \) is a set of nonneg-
avtive rational matrices is undecidable. Hence, the problem of determining the 
joint spectral radius of a set of nonnegative rational matrices is undecidable.

Note that \( \{\Sigma(\mathcal{J})\} \) is a subclass of the set of \( \{0,1\} \) matrices. It is currently 
unresolved whether or not determining the capacity of a collection of patterns 
is NP-hard. Nonetheless, Tsitsiklis and Blondel’s results point to the difficulty 
of finding efficient algorithms to determine the capacity of a given difference 
set.

4.2 Branch-and-bound algorithms

Because of the sub-multiplicativity of \( \hat{\rho}_n(\Sigma, \| \cdot \|) \),

\[
\rho(\Sigma) = \hat{\rho}(\Sigma) \leq \hat{\rho}_n(\Sigma, \| \cdot \|)^{1/n}
\]

for every \( n \). Furthermore as \( n \) increases, this upper bound generally better 
approximates the joint spectral radius in the sense that for every \( n \), there
exists an \( n' > n \) such that

\[
\hat{\rho}_n (\Sigma, \| \cdot \|)^{1/n'} \leq \hat{\rho}_n (\Sigma, \| \cdot \|)^{1/n}.
\]

Similarly, every \( \tilde{\rho}_n (\Sigma) \) lower bounds \( \rho(\Sigma) \), and as \( n \) increases, \( \tilde{\rho}_n (\Sigma) \) generally better approximates the joint spectral radius from below, in the sense that, for every \( n \),

\[
\tilde{\rho}_n (\Sigma)^{1/n} \leq \tilde{\rho}_{nk} (\Sigma)^{1/kn},
\]

for any \( k \geq 1 \).

This suggests approximating the joint spectral radius \( \rho(\Sigma) \) by computing the lower bounds \( \max_{1 \leq k \leq n} \tilde{\rho}_k (\Sigma)^{1/k} \) and upper bounds \( \min_{1 \leq k \leq n} \hat{\rho}_k (\Sigma, \| \cdot \|)^{1/k} \) for \( n = 1, 2, \ldots \). However, the number of matrix operations increases as \( |\mathcal{A}|^n \), such that determining \( \rho(\Sigma) \) with an arbitrary error may be computationally prohibitive.

Several steps have been taken to reduce the growth rate of the number of computations required to approximate \( \rho(\Sigma) \). Maesumi [19] has shown the number of matrix operations may be reduced from \( |\mathcal{A}|^n \) to \( |\mathcal{A}|^n / n \). Daubechies and Lagarias [12] proved the following result:

**Lemma 6** If \( \{ A_j \} \) is a set of building blocks for \( \Sigma \), i.e.,

a) each \( A_j \) is the product of \( n_j \) matrices drawn

from \( \Sigma \),

b) there exists some \( n_0 \geq 0 \) such that, if \( A \) is

a finite product of elements of \( \Sigma \), then

\[
A = A_{j_1} \cdots A_{j_k} Q,
\]

where \( Q \) is a product of

at most \( n_0 \) elements of \( \Sigma \),

then \( \rho(\Sigma) \leq \sup \| A_j \|^{1/n_j}. \)

Lemma 6 can be used to implement a recursive ‘branch-and-bound’ algorithm to upper bound \( \rho(\Sigma) \), e.g. [12,20,21]. Gripenberg [22] has provided an algorithm based on Lemma 6 that includes a sequence of lower bounds such that \( \rho(\Sigma) \) may be specified to lie within an arbitrarily small interval.

### 4.3 The pruning algorithm

In [1] a pruning algorithm was presented for bounding \( \rho(\Sigma) \) when all the matrices in \( \Sigma \) are non-negative. The method replaces the search for the largest norm among all (exponentially many) products of \( n \) matrices with a search
over a smaller set with the same largest norm. It can be applied to compute
\( \hat{\rho}_n(\Sigma) \) and \( \rho_n(\Sigma, \| \cdot \|) \) for several norms. We briefly describe the algorithm
here.

We write \( A \geq 0 \) if every element of \( A \) is nonnegative and \( A \geq B \) if every
element of \( A \) is at least as large as the corresponding element of \( B \). It can be
shown, e.g., [10, Theorem 8.1.18], that if \( A \geq B \geq 0 \) then
\[
\rho(A) \geq \rho(B). \quad (9)
\]

A matrix \( A \) dominates matrix \( B \) with respect to the norm \( \| \cdot \| \) if
\[
\| AM \| \geq \| BM \|
\]
for all \( M \geq 0 \). A subset \( S \) of \( \Sigma^n \) is dominating if every matrix in \( \Sigma^n \) is
dominated by some matrix in \( S \). Let \( \Psi_n \) be any dominating subset of \( \Sigma^n \). By
definition,
\[
\hat{\rho}_n(\Sigma, \| \cdot \|) = \max \{ \| A \| : A \in \Psi_n \}. \quad (10)
\]
Furthermore, it is easy to verify that if all matrices in \( \Sigma \) are non-negative then
\( \Psi_n \Sigma \) is a dominating subset of \( \Sigma^{n+1} \), namely,
\[
\Psi_{n+1} \subseteq \Psi_n \Sigma.
\]

Given a matrix norm one can therefore construct a recursive algorithm which
computes a dominating set \( \Psi_n \) from \( \Psi_{n-1} \) by considering all products in \( \Psi_{n-1} \Sigma \)
and ‘pruning’ those that are dominated by another product. The subsequent
growth rate of \( |\Psi_n| \) will depend on the condition for domination. Sufficient
conditions for domination for several norms are described in \([1]\).

4.4 A heuristic for computing the Joint Spectral Radius

In all cases we have observed, the lower bound doesn’t increase after a finite
depth. This suggests that the joint spectral radius is achieved for a finite
product. It has been conjectured that this is always the case \([23]\). However,
the upper bound may converge slowly. We propose the following heuristic for
increasing the convergence rate of the upper bound. It is a method of choosing
a good norm for an algorithm of the type in Section 4.2 when we suspect a
given matrix in \( \Sigma^n \) achieves \( \rho(\Sigma) \).

Suppose we observe that for some \( A \in \Sigma^n \), \( \hat{\rho}_k(\Sigma) = \hat{\rho}(A)^{1/n} \) for all computed
values of \( k \). We want to check whether \( A \in \Sigma^n \) achieves the joint spectral
radius, i.e., \( \rho(\Sigma) = \bar{\rho}(A)^{1/n} \). Let \( S \) be the nonsingular matrix such that \( S^{-1}AS \) is in Jordan form. We conjecture that \( \hat{\rho}_k(S^{-1}\Sigma S, \| \cdot \|) \) will converge more rapidly than \( \hat{\rho}_k(\Sigma, \| \cdot \|) \). The intuition behind the heuristic is that the growth rate of the norm of the product that achieves the joint spectral radius will be larger than the growth rates of any other product. If \( A \) is full rank, then \( S^{-1}AS \) will be diagonal and if \( S^{-1}AS \) achieves the upper bound \( \hat{\rho}_{kn}(S^{-1}\Sigma S, \| \cdot \|) \) with either the \( L_1 \) or spectral norms, then the algorithm will terminate. We note that since the pruning algorithm as proposed operates under the assumption that all matrices are non-negative, which may not be the case after the similarity transformation, we will use the heuristic with branch-and-bound algorithms such as that proposed in [22].

**Example 2** As an example of applying the heuristic for computing the joint spectral radius of an arbitrary set, let

\[
\Sigma = \left\{ \begin{bmatrix} 3/5 & 0 \\ 1/5 & 3/5 \end{bmatrix}, \begin{bmatrix} 3/5 & -3/5 \\ 0 & -1/5 \end{bmatrix} \right\},
\]

which was considered in [21,22]. At a search depth of 243, taking into account the finite precision of computations, Gripenberg [22] could show that

\[ .6596789 < \rho(\sigma) < .6596924, \]

but the algorithm did not seem to give any smaller interval. However, we observe that

\[
\begin{bmatrix} 3/5 & -3/5 \\ 0 & -1/5 \end{bmatrix}^{12}
\]

achieves the lower bound. Using the heuristic and exact arithmetic, we find \( S \) which diagonalizes \( A \) and compute \( \rho(S^{-1}\Sigma S) \) using the spectral norm, yielding

\[
\rho(\Sigma) = \frac{3^{12/13}(5 + 2\sqrt{7})^{1/13}}{5} = .6596789 \ldots
\]

\[ \square \]

### 4.5 Examples

In this section we illustrate computations of the capacity for several collections of pairs. When all matrices in \( \Sigma \) are Hermitian, it follows, e.g. [10, 5.6.6], that

\[
\hat{\rho}_1(\Sigma) = \hat{\rho}_1(\Sigma, \| \cdot \|),
\]

hence,

\[
\rho(\Sigma) = \hat{\rho}_1(\Sigma, \| \cdot \|).
\]
For example, this can be used to provide a simple proof of $\text{cap}({\{00,11\}})$.  

**Example 3** For $\mathcal{J} = \{00,11\}$ we have  

$$\Sigma(\mathcal{J}) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},$$  

hence $\text{cap}(\mathcal{J}) = \log((1 + \sqrt{5})/2)$.

**Example 4** For $\mathcal{J} = \{010,001\}, \{110,101\}$, computation of $\Sigma(\mathcal{J})$ using the spectral norm and either a branch-and-bound [22] or pruning algorithm yields  

$$\text{cap}(\mathcal{J}) \in [.6942, .6948),$$  

and further computations are hindered by the growth rate of $|\Sigma^n(\mathcal{J})|$.  

However, we observe that  

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \in \Sigma(\mathcal{J}),$$  

which is full-rank, achieves the lower bound. Using the heuristic, we find $S$ which diagonalizes $A$, and compute $\rho(S^{-1}\Sigma(\mathcal{J})S)$ using a branch-and-bound algorithm [22] and the spectral norm, yielding  

$$\text{cap}(\mathcal{J}) = \log_2 \left( \frac{1 + \sqrt{5}}{2} \right) = .6942\ldots$$

**Example 5** For $\mathcal{J} = \{0101,1010\}$, computation of $\Sigma(\mathcal{J})$ using the spectral norm and either a branch-and-bound [22] or pruning algorithm yields  

$$\text{cap}(\mathcal{J}) \in [.9467, .9468),$$  

and again further computations are hindered by the growth rate of $|\Sigma^n(\mathcal{J})|$.  

15
However, we observe that

\[
A = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix} \in \Sigma_2(\mathcal{J})
\]

achieves that lower bound. Using the heuristic, we find \( S \) such that \( S^{-1}AS \) is in Jordan form, and compute \( \rho(S^{-1}\Sigma(\mathcal{J})S) \) using a branch-and-bound algorithm [22] and the spectral norm, yielding

\[
cap(\mathcal{J}) = \log_2 \left( 3 + \sqrt{3\zeta} + \sqrt{99 - 3\zeta + 234\sqrt{3}/\zeta} / 12 \right)
\]

\[
= .9467 \ldots,
\]

where \( \zeta = 11 - 56\beta + 4/\beta \), and \( \beta = (2/(-65 + 3\sqrt{1689}))^{1/3} \).

The following examples consider some classes of pairs \( \mathcal{J} \) such that all matrices in \( \Sigma(\mathcal{J}) \) are full-rank. The initial motivation for investigating these classes was an intuition that the heuristic described in the prior section would perform well if the product that achieves the lower bound is full-rank. However, the capacities follow from more straightforward inductive arguments.

**Example 6** Let \( \mathcal{J} \) be the collection of all \( m \)-bit pairs with difference \( 0^{(m-2)}11 \), \( m \geq 2 \). From Example 3, the case \( m = 2 \), we know \( \text{cap}(\mathcal{J}) \geq \log_2((1 + \sqrt{5})/2) \). By inspection of the bipartite graphs \( G_M, M \in \mathcal{M}(\mathcal{J}) \), one can show via an inductive argument that \( \delta_n(\mathcal{J}) \leq \delta_{n-1}(\mathcal{J}) + \delta_{n-2}(\mathcal{J}) \), which implies \( \text{cap}(\mathcal{J}) \leq \log_2((1 + \sqrt{5})/2) \). Hence \( \text{cap}(\mathcal{J}) = \log_2((1 + \sqrt{5})/2) \).

**Example 7** Let \( \mathcal{J} \) be the collection of all \( m \)-bit pairs with difference \( 10^{(m-2)}1 \), \( m \geq 2 \). By inspection of the bipartite graphs \( G_M, M \in \mathcal{M}(\mathcal{J}) \), one can show via an inductive argument that \( \delta_n(\mathcal{J}) = \delta_{n-1}(\mathcal{J}) + \delta_{n-2}(\mathcal{J}) \), hence \( \text{cap}(\mathcal{J}) = \log_2((1 + \sqrt{5})/2) \).
5 Rate Pairs

Consider the following scenario. We have two sources operating independently and transmitting over a channel wherein the two sources interfere with one another, e.g., inter-track interference in a magnetic recording channel or multi-user interference in a wireless channel. The performance of our system is enhanced if we can guarantee that the two users don’t transmit a certain pair of patterns simultaneously. We would like to determine the achievable rate pairs for such a scheme.

This leads to the following modification of the problem. Let $\mathcal{J}$ be a collection of ordered pairs of possibly identical patterns. The $n$-bit codes $\mathcal{C}_1$ and $\mathcal{C}_2$ avoid $\mathcal{J}$ if, for all $u \in \mathcal{C}_1, v \in \mathcal{C}_2$ and all $i \leq j$ in $[1, n]$,

$$(u_{[i,j]}, v_{[i,j]}) \notin \mathcal{J}.$$  

A rate pair $(R_1, R_2)$ is achievable if there exist codes $\mathcal{C}_1$ and $\mathcal{C}_2$ which avoid $\mathcal{J}$ and have rates greater than or equal to $R_1, R_2$ respectively. The achievable rate region is the set of all achievable rate pairs. Of particular interest is

$$\delta_n(\mathcal{J}) \overset{\text{def}}{=} \max \{|\mathcal{C}_1| + |\mathcal{C}_2| : \mathcal{C}_1, \mathcal{C}_2 \text{ avoid } \mathcal{J}\}$$

the largest sum of two $n$-bit codebooks. We similarly define the capacity of $\mathcal{J}$ as the limit

$$\text{cap}(\mathcal{J}) \overset{\text{def}}{=} \log \left[ \lim_{n \to \infty} (\delta_n(\mathcal{J}))^{1/n} \right].$$  \hspace{1cm} (11)$$

The capacity is an upper bound on the sum of the rates of the two codes. By translating $\mathcal{J}$ into a set of product trellises’ reflecting the pairs of paths simultaneously allowed in the two codes, one can show that $\text{cap}(\mathcal{J})$ defined by (11) is the joint spectral radius of the corresponding set of adjacency matrices.

Example 8 For $\mathcal{J} = \{(11, 00)\}$, we have

$$\Sigma(\{(11, 00)\}) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

and $\text{cap}(\mathcal{J}) = 1 + \log_2 \left( \frac{1 + \sqrt{5}}{2} \right) \approx 1.6942$. The capacity may be achieved by leaving one source unconstrained and disallowing 11 in the second source. The rate region, illustrated in Figure 4, is achieved by time sharing. \hfill \Box
We note that if the codes were allowed to cooperate, the problem reduces to the that of a constrained code where, for example, one can show the capacity in Example 8 is $\log_2(3 + \sqrt{21}/2) \approx 1.923$.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{rate_region.png}
\caption{Achievable rate region for $\mathcal{J} = \{(11,00)\}$.}
\end{figure}

In general, the rate region is difficult to compute. We can, however, compute a tight lower bound by computing the rate pairs of all pairs of codes which avoid $\mathcal{J}$ and take the convex hull of the resulting region, connecting outlying points by time-sharing.

The computation is simplified by applying a pruning similar to that described in Section 4.3 to the tree search. Here, the leaves on the tree are pairs of products corresponding to the pair of codes. We say a pair $(A_1, A_2)$ dominates $(B_1, B_2)$ if $A_1 \geq B_1$ and $A_2 \geq B_2$. If $(A_1, A_2)$ dominates $(B_1, B_2)$, then the children of $(B_1, B_2)$ will fall within the rate region defined by the children of $(A_1, A_2)$.

**Example 9** For $\mathcal{J} = \{(11,10)\}$, we have $\text{cap}(\mathcal{J}) = 1 + 1 + \log_2 \left( \frac{1 + \sqrt{5}}{2} \right)$. A simple upper bound defined by $R_1 \leq 1, R_2 \leq 1, R_1 + R_2 \leq \text{cap}(\mathcal{J})$, and the lower bound obtained by the pruning described on the achievable rate region are illustrated in Fig. 5.

**Example 10** For $\mathcal{J} = \{(01,10)\}$, using the heuristic we can show $\text{cap}(\mathcal{J}) = \log_2(3) = 1.58496 \ldots$. We note that we were unable to compute the capacity exactly via straightforward computations using the branch-and-bound or pruning algorithms. A simple upper bound defined by $R_1 \leq 1, R_2 \leq 1, R_1 + R_2 \leq \text{cap}(\mathcal{J})$, and the lower bound obtained by the pruning described on the achievable rate region are illustrated in Fig. 6.
Fig. 5. Achievable rate region for $\mathcal{J} = \{(11,10)\}$.

Fig. 6. Achievable rate region for $\mathcal{J} = \{(01,10)\}$.

References


