UNIVERSITY OF CALIFORNIA, SAN DIEGO

Coding and Capacities for Multidimensional Constraints

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Zsigmond Nagy

Committee in charge:

Professor Kenneth Zeger, Chair
Professor Edward Bender
Professor Adriano Garsia
Professor Alon Orlitsky
Professor Paul H. Siegel
Professor Jack K. Wolf

2002

The dissertation of Zsigmond Nagy is approved, and it is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2002

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would have not been able to complete this work without help. I would like to thank my advisor Professor Ken Zeger for his guidance and for the many helpful ideas. I want to thank Professors Ed Bender, Adriano Garsia, Paul Siegel, and Jack Wolf for serving on my committee. Special thanks to Professor Alon Orlitsky for being on my committee and for the fruitful conversations.

I would also like to thank my family, my parents, my sister, and Ági. Even from a distance, they were always there for me. I'm grateful for their support, love, and patience.

The text of Chapter 2, in full, is a reprint of the material as it appears in: H. Ito, A. Kato, Zs. Nagy, and K. Zeger, Zero Capacity Region of Multidimensional Run Length Constraints, *Electronic Journal of Combinatorics*, 6(1)(R33), 1999. The text of Chapter 3, in full, is a reprint of the material as it appears in: Zs. Nagy and K. Zeger, Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel, *IEEE Transactions on Information Theory*, 46(3):1030–1033, May 2000. The text of Chapter 4, in full, has been submitted for publication as: Zs. Nagy and K. Zeger, Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions, *IEEE Transactions on Information Theory*, November 2002. The text of Chapter 5, in full, has been submitted for publication as: Zs. Nagy and K. Zeger, Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints, *IEEE Transactions on Information Theory*, July 2002. The text of Chapter 6, in full, has been submitted for publication as: Zs. Nagy and K. Zeger, Capacity Bounds for

the Hard-Triangle Model, *IEEE Transactions on Information Theory*, November 2002. With the exception of the first publication I was the primary researcher and the co-author Kenneth Zeger listed in these publications directed and supervised the research which forms the basis for this dissertation.

VITA

1997	M.S., Budapest University of Technology and Economics
1997-2002	Research Assistant, University of California, San Diego
2002	Ph.D., University of California, San Diego

PUBLICATIONS

H. Ito, A. Kato, Zs. Nagy, and K. Zeger. Zero Capacity Region of Multidimensional Run Length Constraints. *Electronic Journal of Combinatorics*, 6(1)(R33), 1999.

Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional $(0,1)$ Run Length Limited Channel. *IEEE Transactions on Information Theory*, 46(3):1030-1033, May 2000.

Zs. Nagy and K. Zeger. Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints. *IEEE Transactions on Information Theory* (submitted July 2002).

Zs. Nagy and K. Zeger. Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions. *IEEE Transactions on Information Theory* (submitted November 2002).

Zs. Nagy and K. Zeger. Capacity Bounds for the Hard-Triangle Model. *IEEE Transactions on Information Theory* (submitted November 2002).

ABSTRACT OF THE DISSERTATION

Coding and Capacities for Multidimensional Constraints

by

Zsigmond Nagy

Doctor of Philosophy in Electrical Engineering

(Communication Theory and Systems)

University of California, San Diego, 2002

Professor Kenneth Zeger, Chair

A one-dimensional binary sequence satisfies the (d, k) run length constraint if the number of consecutive 0s is at most k, and between any two 1s in the sequence are at least d 0s. An n-dimensional binary array satisfies the (d, k) run length constraint if the one-dimensional (d, k) constraint is satisfied along every direction parallel to the coordinate axes. Other classes of constraints can also be defined. For example, a two-dimensional checkerboard constraint is a bounded measurable set $S \subset \mathbf{R}^2$ that contains the origin. A binary labeling of $\mathbf{Z}^2$ satisfies the checkerboard constraint S, if for every $t \in \mathbf{Z}^2$ that is labeled with 1, every other point of $\mathbf{Z}^2$ in $S + t$ is labeled with 0s. Constrained codes are used in digital recording applications such as magnetic and optical data storage systems, where the constraints model certain physical properties of the recording device.

Every constraint reduces the average amount of information that can be stored

per unit area. The average number of information bits that can be stored per position in a constrained code is upper bounded by the capacity associated with the constraint, and the capacity can be achieved asymptotically as the number of information bits grows. In this dissertation we investigate some problems related to two and higher dimensional (d, k) run length constrained codes, two-dimensional checkerboard codes, and a constrained code defined on a two-dimensional triangular grid. In each problem we determine the value of or bounds on the capacity associated with the given constraint, and also efficient encoding algorithms.

Chapter 1

Introduction

1.1 Definitions

We consider communication systems in which bits are sent across a perfect channel (i.e. no errors) but for which only certain input patterns are acceptable. These channels are called *constrained channels*. Shannon in his 1948 paper [16] wrote (on page 9):

> "Definition: The capacity C of a discrete channel is given by
>
> $$C = \lim_{T\to\infty} \frac{\log N(T)}{T}$$
>
> where $N(T)$ is the number of allowed signals of duration T."

Then he proved the following important result (on page 27):

> "*Theorem 9*: Let a source have entropy H (bits per symbol) and a channel have a capacity C (bits per second). Then it is possible to encode the output of the source in such a way as to transmit at the average rate $\frac{C}{H} - \epsilon$ symbols per second over the channel where ϵ is arbitrarily small. It is not possible to transmit at an average rate greater than $\frac{C}{H}$."

This result has provided the basis for all noiseless constrained coding research in the last 50 years. In particular, it applies to the run length constraints studied in this dissertation. Although Shannon's Theorem 9 is stated for one-dimensional channels (where time T is the parameter), it can easily be extended to higher dimensional channels, such as those studied in this dissertation.

A one-dimensional *block code* of length m is an arbitrary subset $\mathcal{A}$ of $\{0,1\}^m$, and the elements of $\mathcal{A}$ are the *codewords*. Codes are used to store or to transmit information, and there are many aspects one has to consider in the design of a code. We will concentrate on codes with large numbers of codewords given particular constraints. A *constrained code* is a code that the constrained channel accepts. As an example, suppose a channel accepts all binary sequences of length m not containing the pattern 10. Then examples of constrained codes are $\mathcal{A}_1 = \{0,1\}$, $\mathcal{A}_2 = \{00,01,11\}$, $\mathcal{A}_3 = \{000,001,011,111\}$. The *rate* of a code $\mathcal{A}$ of length m is defined as

$$\rho = \frac{\log_2 |\mathcal{A}|}{m}$$

assuming the elements of $\mathcal{A}$ are equiprobable. The rate describes how many information bits are sent on average per transmitted bit in a block code. The *capacity* of a one-dimensional constrained channel is thus

$$C = \limsup_{m\to\infty} \max_{\mathcal{A}_m} \frac{\log_2 |\mathcal{A}_m|}{m}$$

where for each m, the maximization is over all codes $\mathcal{A}_m \subseteq \{0,1\}^m$ that the constrained channel accepts.[1] One class of codes we will consider in this dissertation is defined below.

[1] In practice, "limsup" can usually be replaced by "lim".

Definition 1.1. *A binary code $\mathcal{A}_m$ of length m is (d,k)-constrained (or "run length constrained") if for every codeword in $\mathcal{A}_m$ there are at least d consecutive zeros between every two ones, and the length of any run of zeros is at most k.*[2]

Let $N_m^{(d,k)}$ denote the number of (d,k)-constrained binary sequences of length m. If $d = 1$ and $k = \infty$ then any m-bit binary word that ends in a 0 is compatible with any preceding $m-1$ bits that satisfy the $(1,\infty)$ constraint, and any m-bit word ending in a 1 must actually end in a 01, and is therefore compatible with any preceding $m-2$ bits that satisfy the $(1,\infty)$ constraint. Thus we get the recursion

$$N_m^{(1,\infty)} = N_{m-1}^{(1,\infty)} + N_{m-2}^{(1,\infty)}$$

which is a Fibonacci sequence with initial conditions $N_1^{(1,\infty)} = 2$ and $N_2^{(1,\infty)} = 3$. The capacity can thus be exactly computed as $C_{1,\infty} = \log_2 \frac{1+\sqrt{5}}{2}$. It can be shown more generally [18] that $N_m^{(d,k)}$ satisfies the recursion

$$N_m^{(d,k)} = N_{m-d-1}^{(d,k)} + N_{m-d-2}^{(d,k)} + \ldots + N_{m-k-1}^{(d,k)}$$

for $k < \infty$ and $m > d+k$, and

$$N_m^{(d,k)} = N_{m-1}^{(d,k)} + N_{m-d-1}^{(d,k)}$$

[2] Some slightly different definitions of a (d,k) constraint have appeared in the literature [4], [7], [10], [15], [19], [20], [21]. Such definitions include the following: (1) A binary sequence is (d,k)-constrained, if the number of zeros between consecutive ones is at least d and at most k; (2) A binary sequence satisfies a (d,k) run length constraint if every run of zeros has length at least d and at most k (if two ones are adjacent in the sequence we say that a run of zeros of length zero is between them); (3) Binary (d,k) sequences are described by two integers, d and k, $0 \le d < k$, such that there are at least d and at most k zeros before and after every one. However, these definitions lead to peculiarities such as the following. Definition (2) implies that every $m \times m$ two-dimensional array is not valid whenever $d \ge 2$ and $m > k$, and definitions (1) and (3) imply that every all-zeros string is valid.

for $k = \infty$ and $m > d + 1$. The corresponding capacity $C_{d,k}$ is given by

$$C_{d,k} = \log_2 \lambda$$

where λ is the largest real root of the polynomial

$$x^{k+1} - x^{k-d} - x^{k-d-1} - \ldots - x - 1$$

for $k < \infty$, and

$$x^{d+1} - x^{d} - 1$$

for $k = \infty$.

In many magnetic recording systems the information is written and read along tracks, and therefore the bits are viewed as a one-dimensional binary sequence. To increase the reliability of these data storage systems, a typical requirement is that the sequence of data bits satisfy certain constraints, such as a (d, k) run length limited constraint. Alternative storage media (e.g. holographic memory) motivates the study of two and higher dimensional constraints. A tutorial on some of these topics is given in [4]. The following definition extends the definition of (d, k)-constrained codes and capacities to higher dimensions.

Definition 1.2. *A binary n-dimensional $m_1 \times m_2 \times \cdots \times m_n$ array is (d, k)-constrained if it is (one-dimensional) (d, k)-constrained in each of the n coordinate axis directions.*

Figure 1.1 shows an example of a two-dimensional $(1, 3)$-constrained codeword. The *capacity* corresponding to the n-dimensional (d, k) constraint is defined as

$$C_{d,k}^{(n)} = \lim_{m_1, m_2, \ldots, m_n \to \infty} \frac{\log_2 N_{m_1, m_2, \ldots, m_n}^{(n;d,k)}}{m_1 m_2 \cdots m_n}$$

0	0	1	0
0	1	0	0
1	0	1	0
0	0	0	1

Figure 1.1: Example of $(1, 3)$-constrained codeword on a 4×4 array.

where $N^{(n;d,k)}_{m_1,m_2,\ldots,m_n}$ denotes the number of (d, k)-constrained $m_1 \times m_2 \times \cdots \times m_n$ arrays. The existence of $C^{(2)}_{d,k}$ has been shown in [6], and a similar proof can be used to show that $C^{(n)}_{d,k}$ exists. The capacity $C^{(n)}_{d,k}$ is an upper bound on the average number of information bits that can be stored per unit volume in the n-dimensional space without violating the (d, k) constraint.

For $n \geq 2$, the exact value of $C^{(n)}_{d,k}$ is only known in a few cases [9]. For example, $C^{(n)}_{d,k} = 0$ if $k = d$ (for every $n \geq 1$), and $C^{(n)}_{0,k} > 0$ if $k \geq 1$ (for every $n \geq 1$). In one dimension the capacity is positive whenever $k > d \geq 0$. The capacity is a monotonically nonincreasing function of n and d, and a monotonically nondecreasing function of k. The following result was shown in [6].

Theorem 1.3. ***(Kato and Zeger, 1999)*** *For every $k > d \geq 1$,*

$$C^{(2)}_{d,k} = 0 \Leftrightarrow k = d + 1.$$

The special case when $d = 1$ and $k = \infty$ has been of particular interest. By exchanging the roles of 0 and 1 one can easily verify that $C^{(n)}_{0,1} = C^{(n)}_{1,\infty}$ for all $n \geq 1$. For $n = 1$, as noted earlier, $N^{(1;0,1)}_m$ is a Fibonacci sequence with initial conditions $N^{(1;0,1)}_1 = 2$ and $N^{(1;0,1)}_2 = 3$, and thus $C^{(1)}_{0,1} = \log_2 \frac{1+\sqrt{5}}{2} \approx 0.694242$. In two dimensions, Engel [3] and later Calkin and Wilf [2] gave close upper and lower bounds on the $(0, 1)$ capacity. The following upper and lower bounds are from [2].

Theorem 1.4. ***(Calkin and Wilf, 1998)*** *The capacity of the two-dimensional* $(0, 1)$ *constraint satisfies*

$$0.5879 \leq C_{0,1}^{(2)} \leq 0.5883.$$

Bounds on the capacity of the three-dimensional $(0, 1)$ constraint were presented in [10]. However, the proofs in [2] and [10] rely on matrix inequalities, and do not provide implementable encoding procedures. For a practical application, efficient encoding algorithms are needed that achieve a rate close to the value of the capacity. Such an algorithm for two-dimensional (d, ∞) constraints was proposed by Siegel and Wolf (1998), known as a "bit stuffing" algorithm [17]. The expected encoding rate of the algorithm was determined in [14].

Theorem 1.5. ***(Roth, Siegel, and Wolf, 1999)*** *The rate of the two-dimensional bit stuffing algorithm for the* $(1, \infty)$ *constraint is*

$$\rho = 0.583056.$$

The performance of the algorithm was further improved in a subsequent paper [15]. The authors mapped sequences of independent bits into new binary sequences with different probability distributions, using "distribution transformers", and achieved the following encoding rate [15].

Theorem 1.6. ***(Roth, Siegel, and Wolf, 2000)*** *The rate of the two-dimensional bit stuffing algorithm for the* $(1, \infty)$ *constraint with two distribution transformers is*

$$\rho = 0.587277.$$

The original bit stuffing algorithm can be generalized to three dimensions, and the corresponding rate was determined in [12].

In two dimensions, (d, k) constraints require that a binary labeling of the integer lattice have a specified minimum and maximum number of zeros between consecutive ones both "horizontally" and "vertically". Additional constraints, such as run length constraints along diagonals can also be imposed in order to more accurately model practical devices. The asymptotic behavior of constraints defined by open convex symmetric two-dimensional sets were examined in [11]. The following analogous result for the decay rate of the two-dimensional capacity $C^{(2)}_{d,\infty}$ (as d grows) was given in [6].

Theorem 1.7. ***(Kato and Zeger, 1999)***

$$\lim_{d\to\infty}\left(\frac{d}{\log_2 d}\right)\cdot C^{(2)}_{d,\infty} = 1.$$

Constrained codes can be defined not only on the two (or higher) dimensional integer lattice, but, more generally, on any point set. Different point sets with different mathematical properties (e.g. number of nearest neighbors, average number of points per unit area), can be used to model different practical applications. Binary constrained codes defined on the two-dimensional hexagonal lattice were studied in [1] and [8]. The definition of a hard-triangle constraint, and numerical bounds on the corresponding capacity can be found in [13].

1.2 Organization of the Dissertation

Each of the five Chapters 2-6 in this dissertation is a copy of a published or submitted co-authored journal paper. These are as follows:

Chapter 2	H. Ito, A. Kato, Zs. Nagy, and K. Zeger. Zero Capacity Region of Multidimensional Run Length Constraints. *Electronic Journal of Combinatorics*, 6(1)(R33), 1999.
Chapter 3	Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel. *IEEE Transactions on Information Theory*, 46(3):1030–1033, May 2000.
Chapter 4	Zs. Nagy and K. Zeger. Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions. *IEEE Transactions on Information Theory* (submitted November 2002).
Chapter 5	Zs. Nagy and K. Zeger. Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints. *IEEE Transactions on Information Theory* (submitted July 2002).
Chapter 6	Zs. Nagy and K. Zeger. Capacity Bounds for the Hard-Triangle Model. *IEEE Transactions on Information Theory* (submitted November 2002).

The characterization of the zero capacity region for three and higher dimensional run length constraints was given in [5]. Necessary and sufficient conditions for $C_{d,k}^{(n)}$ to converge to 0 as the dimension n goes to infinity were also established in [5]. These results are presented in Chapter 2. My contribution in [5] was the construction of n-dimensional $(d, d+2)$-constrained codes with positive coding rate, which are used in the proof of Theorem 2.1.

The capacity of the two-dimensional $(0,1)$ run length constraint played an important role in [6] for obtaining bounds on the capacities of other run length constraints. Bounds on the capacity of the three-dimensional $(0,1)$ constraint given in [10] can play a similar role for obtaining different three-dimensional bounds, and are also of theoretical interest. This result is given here in Chapter 3.

A generalization of the two-dimensional bit stuffing algorithm to three dimensions, and an analysis of the coding rate for both two and three dimensions have been presented in [12]. The algorithms and the analysis are presented in Chapter 4.

Another class of two-dimensional constraints, called checkerboard constraints, are defined by two-dimensional bounded measurable sets. The constraint specifies certain required 0's that must surround every 1 that appear in the binary labeling of the two-dimensional integer lattice. We examined the asymptotic behavior of open convex symmetric checkerboard constraints in [11], and determined the rate at which the capacity goes to zero as the area of the constraint grows. Chapter 5 presents these ideas.

The definition of the hard-triangle constraint, and upper and lower bounds on the capacity corresponding to this constraint can be found in [13]. The upper bound was calculated using transfer matrices, and the lower bound was obtained by generalizing the bit stuffing algorithm to the hard-triangle constraint. These results are presented here in Chapter 6. The last chapter of the dissertation suggests future problems in the field of constrained coding.

References

[1] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982.

[2] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[3] K. Engel. On the Fibonacci Number of an $M \times N$ Lattice. *Fibonacci Quarterly*, 28:72–78, 1990.

[4] K. A. S. Immink, P. H. Siegel, and J. K. Wolf. Codes for Digital Recorders. *IEEE Trans. Inform. Theory*, 44:2260–2299, October 1998.

[5] H. Ito, A. Kato, Zs. Nagy, and K. Zeger. Zero Capacity Region of Multidimensional Run Length Constraints. *Elec. J. Combinatorics*, 6(1)(R33), 1999.

[6] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[7] A. Kato and K. Zeger. Partial Characterization of the Positive Capacity Region of Two-Dimensional Asymmetric Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 46(7):2666–2670, November 2000.

[8] Z. Kukorelly and K. Zeger. The Capacity of some Hexagonal (d, k)-Constraints. *Proc. 2001 IEEE Intl. Symp. on Inform. Theory*, page 64, Washington, DC, June 2001.

[9] D. Lind and B. H. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, New York, 1995.

[10] Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel. *IEEE Trans. Inform. Theory*, 46(3):1030–1033, May 2000.

[11] Zs. Nagy and K. Zeger. Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints. *IEEE Trans. Inform. Theory* (submitted July 2002).

[12] Zs. Nagy and K. Zeger. Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions. *IEEE Trans. Inform. Theory* (submitted November 2002).

[13] Zs. Nagy and K. Zeger. Capacity Bounds for the Hard-Triangle Model. *IEEE Trans. Inform. Theory* (submitted November 2002).

[14] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding for a Two-Dimensional Runlength-Limited Constraint. *Proc. of the SPIE - The Intl. Soc. for Optical Engineering*, 3802:8–17, Denver, CO, July 1999.

[15] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding Schemes for the Hard-Square Model. *IEEE Trans. Inform. Theory*, 47:1166–1176, March 2001.

[16] C. E. Shannon. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27(3 and 4):379–423 and 623–656, July and October 1948.

[17] P. H. Siegel and J. K. Wolf. Bit Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays. *Proc. 1998 IEEE Intl. Symp. on Inform. Theory*, page 323, MIT, Cambridge, MA, August 1998.

[18] D. T. Tang and L. R. Bahl. Block Codes for a Class of Constrained Noiseless Channels. *Information and Control*, 17:436–461, 1970.

[19] J. H. Weber and K. A. S. Abdel-Ghaffar. Cascading Runlength-Limited Sequences. *IEEE Trans. Inform. Theory*, 39(6):1976–1984, November 1993.

[20] J. K. Wolf. Constrained Sequences, Crossword Puzzles and Shannon. *IEEE Inform. Theory Soc. Newsletter*, 51(3):1–7, September 2001.

[21] Z. Ye and Z. Zhang. On the Capacity Rates of Two-Dimensional Runlength Limited Codes. *Proc. 1995 IEEE Intl. Symp. on Inform. Theory*, page 247, Whistler, BC, Canada, September 1995.

Chapter 2

Zero Capacity Region of Multidimensional Run Length Constraints

Abstract

For integers d and k satisfying $0 \leq d \leq k$, a binary sequence is said to satisfy a one-dimensional (d, k) run length constraint if there are never more than k zeros in a row, and if between any two ones there are at least d zeros. For $n \geq 1$, the n-dimensional (d, k)-constrained capacity is defined as

$$C_{d,k}^{(n)} = \lim_{m_1, m_2, \ldots, m_n \to \infty} \frac{\log_2 N_{m_1, m_2, \ldots, m_n}^{(n;d,k)}}{m_1 m_2 \cdots m_n}$$

where $N_{m_1, m_2, \ldots, m_n}^{(n;d,k)}$ denotes the number of $m_1 \times m_2 \times \cdots \times m_n$ n-dimensional binary rectangular patterns that satisfy the one-dimensional (d, k) run length constraint in the direction of every coordinate axis. It is proven for all $n \geq 2$, $d \geq 1$, and $k > d$ that $C_{d,k}^{(n)} = 0$ if and only if $k = d + 1$. Also, it is proven for every $d \geq 0$ and $k \geq d$ that $\lim_{n \to \infty} C_{d,k}^{(n)} = 0$ if and only if $k \leq 2d$.

2.1 Introduction

A binary sequence is (d,k)-*constrained* (or "runlength constrained") if there are at most k consecutive zeros and between every two ones there are at least d consecutive zeros. An n-dimensional pattern of zeros and ones arranged in an $m_1 \times m_2 \times \cdots \times m_n$ hyper-rectangle is (d,k)-*constrained* if it is (1-dimensional) (d,k)-constrained in each of the n coordinate axis directions. The n-dimensional (d,k)-*capacity* is defined as

$$C_{d,k}^{(n)} = \lim_{m_1,m_2,\dots,m_n \to \infty} \frac{\log_2 N_{m_1,m_2,\dots,m_n}^{(n;d,k)}}{m_1 m_2 \cdots m_n},$$

where $N_{m_1,m_2,\dots,m_n}^{(n;d,k)}$ denotes the number of (d,k)-constrained patterns on an $m_1 \times m_2 \times \cdots \times m_n$ hyper-rectangle. A simple proof was given in [5] that shows the existence of two-dimensional (d,k)-capacities, and a slight modification of the proof can show that the n-dimensional (d,k)-capacities exist. The capacity $C_{d,k}^{(n)}$ represents the maximum number of bits of information that can be stored asymptotically per unit volume in n-dimensional space without violating the (d,k) constraint.

The study of 1-dimensional (d,k)-capacities was originally motivated by applications in magnetic storage. Interest in 2-dimensional (d,k)-capacities has recently increased due to emerging 2-dimensional optical recording devices, and the 3-dimensional (d,k)-capacities may play a role in future applications as well. A tutorial on these topics is given in [4]. Capacities in four and higher dimensions yield natural generalizations of interesting mathematical questions in lower dimensions.

In general, the exact values of the various n-dimensional (d,k)-capacities are not known except in a few cases [6]. For example, in all dimensions, if $k = d$ the capacity is zero, and if $d = 0$ the capacity is positive for all $k \geq 1$. In one dimension the capacity is positive whenever $k > d \geq 0$. The capacity is known to be a monotonically nonincreasing function of n and d and a monotonically nondecreasing function of k. It was recently shown [5] that whenever $k > d \geq 1$, the 2-dimensional capacity is zero if and only if $k = d + 1$. These facts are summarized in our Lemma 2.3.

Some interesting facts are known about the capacities for $d = 0$ and $k = 1$ in three and lower dimensions. In one dimension, $N_m^{(1;0,1)}$ is known [6] to be a Fibonacci sequence with initial conditions $N_1^{(1;0,1)} = 2$ and $N_2^{(1;0,1)} = 3$, and thus the 1-dimensional $(0,1)$-capacity is the logarithm of the golden mean, namely $C_{0,1}^{(1)} = \log_2 \frac{1+\sqrt{5}}{2} \approx 0.694$. Very tight upper and lower bounds on the $(0,1)$-capacity were given for two dimensions in [2] and for three dimensions in [7]. These two and three dimensional $(0,1)$-capacities are $C_{0,1}^{(2)} \approx 0.58789116$ and $C_{0,1}^{(3)} \approx 0.52$, given here to their known accuracies.

In this paper we present two main results that characterize the zero capacity region for finite dimensions and in the limit of large dimensions. The first result generalizes the zero capacity characterization in [5] to all dimensions greater than one. Namely it gives a necessary and sufficient condition on d and k for the capacity to equal zero. This condition turns out to be exactly the same as in dimension 2. The second result gives a necessary and sufficient condition on d and k, such that the capacity approaches zero in the limit as the dimension n grows to infinity. These results are summarized in the following two theorems.

Theorem 2.1. *For every $n \geq 2$, $d \geq 1$, and $k > d$,*

$$C_{d,k}^{(n)} = 0 \Leftrightarrow k = d + 1.$$

Theorem 2.2. *For every $d \geq 0$ and $k \geq d$,*

$$\lim_{n \to \infty} C_{d,k}^{(n)} = 0 \Leftrightarrow k \leq 2d.$$

The following lemma contains useful facts about capacities for various constraints and is used to establish Theorems 2.1 and 2.2.

Lemma 2.3.

(a) $C^{(n)}_{d,k+1} \geq C^{(n)}_{d,k}$; whenever $n \geq 1,\ 0 \leq d \leq k$

(b) $C^{(n)}_{d,k} \geq C^{(n)}_{d+1,k}$; whenever $n \geq 1,\ 0 \leq d < k$

(c) $C^{(n+1)}_{d,k} \leq C^{(n)}_{d,k}$; whenever $n \geq 1,\ 0 \leq d < k$

(d) $C^{(n)}_{d,d} = 0$; whenever $n \geq 1,\ d \geq 0$

(e) $C^{(n)}_{d,2d+1} \geq \frac{1}{2(d+1)}$; whenever $n \geq 1,\ d \geq 0$

(f) $C^{(n)}_{0,k} > 0$; whenever $n \geq 1,\ k \geq 1$

(g) $C^{(1)}_{d,k} > 0$; whenever $0 \leq d < k$

(h) $C^{(2)}_{d,k} = 0$ if and only if $k = d+1$; whenever $1 \leq d < k$.

Proof. (a) Follows from the fact that $N^{(n;d,k+1)}_{m_1,m_2,\dots,m_n} \geq N^{(n;d,k)}_{m_1,m_2,\dots,m_n}$ since any pattern that satisfies the (d,k) constraint also satisfies the $(d,k+1)$ constraint.

(b) Follows from $N^{(n;d,k)}_{m_1,m_2,\dots,m_n} \geq N^{(n;d+1,k)}_{m_1,m_2,\dots,m_n}$.

(c)

$$\begin{aligned}
C^{(n+1)}_{d,k} &= \lim_{m_1,m_2,\dots,m_{n+1}\to\infty} \frac{\log_2 N^{(n+1;d,k)}_{m_1,m_2,\dots,m_{n+1}}}{m_1 m_2 \dots m_{n+1}} \\
&\leq \lim_{m_1,m_2,\dots,m_{n+1}\to\infty} \frac{\log_2 (N^{(n;d,k)}_{m_1,m_2,\dots,m_n})^{m_{n+1}}}{m_1 m_2 \dots m_{n+1}} \\
&= \lim_{m_1,m_2,\dots,m_n\to\infty} \frac{\log_2 N^{(n;d,k)}_{m_1,m_2,\dots,m_n}}{m_1 m_2 \dots m_n} \\
&= C^{(n)}_{d,k}.
\end{aligned}$$

(d) $C^{(1)}_{d,d} = 0$ since $N^{(1;d,d)}_m \leq d+1$. The result then follows by induction and from the monotonicity in part (c).

(e) Let $T = \{1, 2, \ldots, m\}$, where m is a multiple of $2(d+1)$. Any mapping $f : T^n \to \{0,1\}$ satisfying $f(x_1, x_2, \ldots, x_n) = 1$ when $2(d+1)$ divides $\sum_{i=1}^n x_i$, and $f(x_1, x_2, \ldots, x_n) = 0$ when $d+1$ does not divide $\sum_{i=1}^n x_i$, induces a $(d, 2d+1)$-constrained pattern on T^n. Since the value of $f(x_1, x_2, \ldots, x_n)$ can be chosen arbitrarily when $\sum_{i=1}^n x_i \equiv (d+1) \bmod 2(d+1)$, the number of $(d, 2d+1)$-constrained patterns on T^n is at least $2^{m^n/(2(d+1))}$ and hence $N_{m,m,\ldots,m}^{(n;d,2d+1)} \geq 2^{m^n/(2(d+1))}$. Thus

$$C_{d,2d+1}^{(n)} \geq \lim_{m\to\infty} \frac{m^n/(2(d+1))}{m^n} = \frac{1}{2(d+1)}.$$

(f) Follows from (a) and (e).

(g) It is known [1] that $C_{d,\infty}^{(1)} = C_{d-1,2d-1}^{(1)}$ for $d \geq 1$, and also that for $0 \leq d < k < \infty$, the 1-dimensional capacity is the logarithm (base 2) of the largest real root of the equation $X^{k+1} - X^{k-d} - X^{k-d-1} - \cdots - X - 1 = 0$. The equation clearly has a root greater than 1, and thus the result follows.

(h) This was proven in [5]. □

2.2 Proof of Theorem 2.1

Proof. Lemma 2.3(c),(h) shows that $C_{d,d+1}^{(n)} = 0$ for all $d \geq 1$ and all $n \geq 2$. To prove $C_{d,k}^{(n)} > 0$ for $k \geq d+2$, it suffices by Lemma 2.3(a),(h) to prove $C_{d,d+2}^{(n)} > 0$ for all $d \geq 1$ and $n \geq 3$. This is shown below in Proposition 2.5 for even $d \geq 0$, and in Proposition 2.6 for odd $d \geq 3$. A special case of Lemma 2.3(e) shows the result for $d = 1$ and for all $n \geq 3$. This completes the proof of Theorem 2.1.

□

The following definitions are useful for proving Propositions 2.5 and 2.6. Let $S = \{0, 1, \ldots, d+1\}$. The set S^n is an *n-cube*, and any mapping $g : S^n \to \{0,1\}$ is a *binary n-cube*. A *row* of an n-cube is any set of the form $\{(c_1, \ldots, c_{l-1}, x, c_{l+1}, \ldots, c_n) :$

$x \in S\}$ for some fixed l, and some fixed $c_j \in S$ for $j = 1, \ldots, l-1, l+1, \ldots, n$. A binary n-cube g is a *permutation n-cube* if g equals 1 once per row of S^n.

A binary n-cube g is $(d, d+2)$-constrained unless g takes the value one twice on some consecutive d points in some row of S^n. It is clear that permutation n-cubes are $(d, d+2)$-constrained. A set of permutation n-cubes is $(d, d+2)$-*compatible* if the concatenation of any two of the cubes along a face (i.e. with translation but without rotation) is also $(d, d+2)$-constrained. If $S_1, \ldots, S_n$ are subsets of S, each consisting of two consecutive integers, the smaller of which is even, then $S_1 \times \cdots \times S_n$ is a *bi-subcube* of S^n. If a permutation n-cube g equals 1 exactly once per row in a bi-subcube, then the restriction of g to the bi-subcube is said to be a *permutation bi-subcube*.

A binary n-cube h is a *reversal* of a permutation n-cube g if h equals $1-g$ on the members of a (possibly empty) subset of all the bi-subcubes in S^n, on each of which g is a permutation bi-subcube, and h equals g elsewhere. A reversal h of any permutation cube g is also a permutation cube, and g and h together form a $(d, d+2)$-compatible set. More generally, any collection of reversals of a given permutation n-cube forms a $(d, d+2)$-compatible set (see Lemma 2.4). In Propositions 2.5 and 2.6, we construct a $(d, d+2)$-compatible family of reversals of a certain permutation n-cube, and then obtain a lower bound on the $(d, d+2)$-capacity from the cardinality of the family.

A mapping $\bar{f} : S^n \to S$ is a *latin n-cube* if on every row of S^n, $\bar{f}$ is a permutation of S. This definition is a generalization of a latin square, although alternate definitions have been given in [3]. For any permutation n-cube g, any $l < n$, and any $c_j \in S$ (for $j = 1, \ldots, l-1, l+1, \ldots, n-1$), the relation $x \mapsto y$ determined by $g(c_1, \ldots, c_{l-1}, x, c_{l+1}, \ldots, c_{n-1}, y) = 1$ is a permutation. This leads us to define a correspondence between permutation n-cubes and latin $(n-1)$-cubes as follows. Let $g : S^n \to \{0, 1\}$ be a permutation n-cube and for each $(x_1, x_2, \ldots, x_{n-1}) \in S^{n-1}$, let $y(x_1, \ldots, x_{n-1})$ be the unique element of S such that $g(x_1, x_2, \ldots, x_{n-1}, y(x_1, \ldots, x_{n-1})) = 1$. Then the mapping $\bar{g} : S^{n-1} \to S$ defined by $\bar{g}(x_1, x_2, \ldots,$

$x_{n-1}) = y(x_1, \ldots, x_{n-1})$ is a latin $(n-1)$-cube, and the correspondence $g \mapsto \bar{g}$ is bijective (see Lemma 2.4). The bar notation will be exclusively used for latin cubes. For any integers $a \geq 0$ and $b > 0$, we use the notation "a mod b" to mean the unique integer $a - \lfloor \frac{a}{b} \rfloor b$.

Lemma 2.4. *Let $\bar{e}_n : S^n \to S$ be a sequence of mappings defined recursively for $n \geq 3$ by*

$$\bar{e}_n(x_1, \ldots, x_n) = \bar{e}_2(\bar{e}_{n-1}(x_1, \ldots, x_{n-1}), x_n) \tag{2.1}$$

where $\bar{e}_2$ is a latin square. Then $\bar{e}_n$ is a latin n-cube for all $n \geq 2$, and the set of all reversals of the corresponding permutation $(n+1)$-cube e_n is $(d, d+2)$-compatible.

Proof. Use induction on n. Assume $\bar{e}_2, \ldots, \bar{e}_{n-1}$ are latin cubes (for $n \geq 3$) and fix all but one of the arguments $x_1, \ldots, x_n$ of $\bar{e}_n$. If $x_1, \ldots, x_{n-1}$ are fixed then $\bar{e}_n$ is a permutation of S since fixing the first argument of $\bar{e}_2$ yields a permutation of S. Likewise, if x_n and all but one of $x_1, \ldots, x_{n-1}$ are fixed, then by the induction hypothesis $\bar{e}_{n-1}(x_1, \ldots, x_{n-1})$ is a permutation of S and $\bar{e}_2$ is a permutation of S since its second argument x_n is fixed. Thus $\bar{e}_n$ is a latin n-cube.

Let h be a binary $(n+1)$-cube $h : S^{n+1} \to \{0, 1\}$ satisfying

$$h(x_1, \ldots, x_{n+1}) = \begin{cases} 1 & \text{if } x_{n+1} = \bar{e}_n(x_1, \ldots, x_n) \\ 0 & \text{otherwise.} \end{cases}$$

Then h is a permutation $(n+1)$-cube since $\bar{e}_n$ is a latin n-cube, and $\bar{h} = \bar{e}_n$ from the definition of the bar notation. This shows that there exists a unique permutation $(n+1)$-cube h (i.e. e_n) corresponding to the latin n-cube $\bar{e}_n$.

The permutation $(n+1)$-cube e_n has rows of length $d+2$, each containing a single one. For any collection of bi-subcubes, on each of which e_n is a permutation bi-subcube, any row of S^{n+1} can intersect at most one of these bi-subcubes. This implies that any facewise concatenation of any two reversals of e_n will only have pairs of ones at distances d, $d+1$, or $d+2$ apart, and thus any set of reversals of e_n is $(d, d+2)$-compatible.

$\square$

Proposition 2.5. *For every $n \geq 2$ and every even $d \geq 0$,*

$$C^{(n)}_{d,d+2} \geq \frac{1}{2^{n-1}(d+2)}.$$

Proof. Define a mapping $\bar{e}_2 : S^2 \to S$ such that

$$\bar{e}_2(x_1, x_2) = \begin{cases} (x_1 + x_2 - 2) \bmod (d+2) & \text{if } x_1 \text{ and } x_2 \text{ are odd} \\ (x_1 + x_2) \bmod (d+2) & \text{otherwise} \end{cases} \tag{2.2}$$

as in Figure 2.1. The mapping $\bar{e}_2$ is a latin square since $\bar{e}_2$ is a permutation of the set S when either the first or second component is held fixed. For each $n \geq 3$, use (2.1) to recursively define the latin n-cube $\bar{e}_n : S^n \to S$.

For each $n \geq 2$, let $x_1, \ldots, x_n$ be any set of even integers from S. We claim that for any $y_1, \ldots, y_n \in \{0, 1\}$,

$$\bar{e}_n(x_1 + y_1, \ldots, x_n + y_n) = \begin{cases} (x_1 + \cdots + x_n) \bmod (d+2) & \text{if } \sum_{i=1}^n y_i \text{ is even} \\ (1 + x_1 + \cdots + x_n) \bmod (d+2) & \text{if } \sum_{i=1}^n y_i \text{ is odd.} \end{cases}$$

To prove this claim, use induction on n. It is easy to see from (2.2) that the claim is true for $n = 2$. By (2.1) and the induction hypothesis,

$$\begin{aligned} &\bar{e}_n(x_1 + y_1, \ldots, x_n + y_n) \\ &\quad = \begin{cases} \bar{e}_2((x_1 + \cdots + x_{n-1}) \bmod (d+2), x_n + y_n) & \text{if } \sum_{i=1}^{n-1} y_i \text{ is even} \\ \bar{e}_2((1 + x_1 + \cdots + x_{n-1}) \bmod (d+2), x_n + y_n) & \text{if } \sum_{i=1}^{n-1} y_i \text{ is odd.} \end{cases} \end{aligned}$$

Equivalently, when $\sum_{i=1}^{n} y_i$ is even

$$\begin{aligned}
&\bar{e}_n(x_1+y_1,\ldots,x_n+y_n)\\
&\quad= \begin{cases} \bar{e}_2((x_1+\cdots+x_{n-1}) \bmod (d+2), \quad x_n) & \text{if } y_n=0\\ \bar{e}_2((1+x_1+\cdots+x_{n-1}) \bmod (d+2), \quad x_n+1) & \text{if } y_n=1 \end{cases}\\
&\quad= (x_1+\cdots+x_n) \bmod (d+2),
\end{aligned}$$

and when $\sum_{i=1}^{n} y_i$ is odd

$$\begin{aligned}
&\bar{e}_n(x_1+y_1,\ldots,x_n+y_n)\\
&\quad= \begin{cases} \bar{e}_2((x_1+\cdots+x_{n-1}) \bmod (d+2), \quad x_n+1) & \text{if } y_n=1\\ \bar{e}_2((1+x_1+\cdots+x_{n-1}) \bmod (d+2), \quad x_n) & \text{if } y_n=0 \end{cases}\\
&\quad= (1+x_1+\cdots+x_n) \bmod (d+2),
\end{aligned}$$

thus proving the claim.

The claim just proved implies that the corresponding permutation $(n+1)$-cube e_n satisfies

$$e_n(x_1+y_1,\ldots,x_{n+1}+y_{n+1}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n+1} y_i \text{ is even}\\ 0 & \text{if } \sum_{i=1}^{n+1} y_i \text{ is odd} \end{cases}$$

for any even integers $x_1,\ldots,x_{n+1} \in S$ such that $x_{n+1} = \sum_{i=1}^{n} x_i \bmod (d+2)$, and for any $y_1,\ldots,y_{n+1} \in \{0,1\}$. Thus the restriction of e_n to each bi-subcube $\{(x_1+y_1,\ldots,x_{n+1}+y_{n+1}) : y_1,\ldots,y_{n+1} \in \{0,1\}\}$ is a permutation bi-subcube. Then the cardinality of the set of all reversals of e_n is $2^{(\frac{d+2}{2})^n}$, and Lemma 2 gives the lower bound

$$C_{d,d+2}^{(n)} \geq \frac{\log_2 2^{\left(\frac{d+2}{2}\right)^{n-1}}}{(d+2)^n} = \frac{1}{2^{n-1}(d+2)}.$$

$\square$

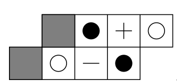
Proposition 2.6. *For every $n \geq 2$ and every odd $d \geq 3$,*

$$C_{d,d+2}^{(n)} \geq \frac{1}{(d+2)^n} \binom{n-1+\frac{d-3}{2}}{n-1}.$$

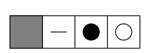
Proof. Define a mapping $\bar{e}_2 : S^2 \to S$ such that

$$\bar{e}_2(x_1, x_2) = \begin{cases} x_1 + x_2 - 2 & \text{if } x_1 \text{ and } x_2 \text{ are odd} \\ x_1 + x_2 & \text{otherwise} \end{cases} \tag{2.3}$$

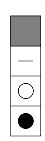
for $2\lfloor \frac{x_1}{2} \rfloor + 2\lfloor \frac{x_2}{2} \rfloor \leq d - 3$. The values of $\bar{e}_2$ for $2\lfloor \frac{x_1}{2} \rfloor + 2\lfloor \frac{x_2}{2} \rfloor > d - 3$ (i.e. below the bold 2-step staircase line in Figures 2.2 and 2.3) are defined as follows. The points on the diagonal line above the main diagonal have value d, as does the bottom right corner of the square. Thus, d appears once in each row and in each column in the square. The portion of the next higher diagonal that lies below the 2-step staircase line has value $d - 1$. The area below and including the main diagonal of the square, except the bottom row and the rightmost column, is partitioned into diagonal strips of width 4. Each diagonal strip is formed by repeating the staircase pattern shape of


.

The bottom row is formed by repeating the pattern



and the rightmost column is formed by repeating the pattern


.

(For the case $d \equiv 1 \bmod 4$ the bottom-rightmost diagonal strip is truncated at width 3, and the above patterns are cut off accordingly, as illustrated in Figure 2.2.) Within any given diagonal strip, all labels containing a particular symbol represent the same integer. In particular, in the jth diagonal, (for $j = 1, 2, \ldots, \lfloor \frac{d}{4} \rfloor + 1$), the square labels $\boxplus$, $\boxed{\bigcirc}$, $\boxed{\bullet}$, $\blacksquare$, and $\boxminus$ represent $4j-2$, $4j-3$, $4j-4$, $4j-5$, and $4j-6$ respectively (for $j = 1$, $\boxminus$ and $\blacksquare$ represent $d-1$ and $d+1$, respectively). For any $i \in \{0, 1, \ldots, d-2\}$ it can be seen that the value i appears once in every row and column of the top left $2\lfloor \frac{i}{2} \rfloor + 2$ rows and columns, and the value i appears once in every row and column of the bottom right $d - 2\lfloor \frac{i}{2} \rfloor$ rows and columns. Also, the main diagonal of S^2 contains only the value $d+1$, and the value $d-1$ appears in the rightmost column at $(x_1, x_2) = (1, d+2)$, in the bottom row at $(x_1, x_2) = (d+2, 1)$, and in alternating positions on the diagonals that lie two above and two below the main diagonal of S^2. The value $d-1$ appears in the rightmost column at $(x_1, x_2) = (1, d+2)$ and in the bottom row at $(x_1, x_2) = (d+2, 1)$, and these points do not lie on the diagonals two below nor two above the main diagonal. Consequently, every number $0, 1, \ldots, d+1$ appears exactly once in each row and in each column in the original $(d+2) \times (d+2)$ square S^2, showing that $\bar{e}_2$ is a latin square.

Using (2.1) and the definition of $\bar{e}_2$ just given, recursively define for each integer $n \geq 3$, the latin n-cube $\bar{e}_n : S^n \to S$. For any $n \geq 2$, if $x_1, \ldots, x_n$ are even integers from S such that $\sum_{i=1}^{n} x_i \leq d-3$, then for any $y_1, \ldots, y_n \in \{0, 1\}$,

$$\bar{e}_n(x_1 + y_1, \ldots, x_n + y_n) = \begin{cases} x_1 + \cdots + x_n & \text{if } \sum_{i=1}^{n} y_i \text{ is even} \\ 1 + x_1 + \cdots + x_n & \text{if } \sum_{i=1}^{n} y_i \text{ is odd} \end{cases}$$

from the same proof as in Proposition 2.5, but with the added constraint $\sum_{i=1}^{n} x_i \leq d-3$.

As in Proposition 2.5, the set of reversals of the permutation $(n+1)$-cube e_n is $(d, d+2)$-compatible. There are $\binom{n + \frac{d-3}{2}}{n}$ permutation bi-subcubes in this case and the volume of the $(n+1)$-cube S^{n+1} (i.e. the domain of e_n) is $(d+2)^{n+1}$. Hence

$$C_{d,d+2}^{(n)} \geq \frac{1}{(d+2)^n} \left(\begin{array}{c} n-1+\frac{d-3}{2} \\ n-1 \end{array} \right).$$

$\square$

2.3 Proof of Theorem 2.2

Proof. Lemma 2.3(e) gives $\lim_{n\to\infty} C_{d,2d+1}^{(n)} > 0$ for every $d \geq 0$, and thus $\lim_{n\to\infty} C_{d,k}^{(n)} > 0$ for every $k \geq 2d+1$ by Lemma 2.3(a). Lemma 2.7 below implies that $C_{d,k}^{(n)} \leq \left(\frac{k-d}{k-d+1}\right)^{n-1} C_{d,k}^{(1)}$ whenever $d < k \leq 2d$, and hence $\lim_{n\to\infty} C_{d,k}^{(n)} = 0$. This together with Lemma 2.3(d) completes the proof of Theorem 2.2.

$\square$

Lemma 2.7. *If* $n \geq 2$ *and* $1 \leq d < k \leq 2d$ *then*

$$C_{d,k}^{(n)} \leq \frac{k-d}{k-d+1} C_{d,k}^{(n-1)}.$$

Proof. Let l and m be positive integers and let $V = \{1, 2, \ldots, m\}$. Define the following n-dimensional hyper-rectangles (for $j = 1, 2, \ldots, l$):

$$\begin{aligned}
T &= \{(x_1, \ldots, x_{n-1}, x_n) : x_1, \ldots, x_{n-1} \in V,\ -d \leq x_n < (k-d+1)l\} \\
U_0 &= \{(x_1, \ldots, x_{n-1}, x_n) : x_1, \ldots, x_{n-1} \in V,\ -d \leq x_n < 0\} \\
U_j &= \{(x_1, \ldots, x_{n-1}, x_n) : \\
&\quad\ x_1, \ldots, x_{n-1} \in V,\ (k-d+1)(j-1) < x_n < (k-d+1)j\}
\end{aligned}$$

and let $U = \bigcup_{j=0}^{l} U_j$. Note that there is a gap of width one between consecutive sets U_j and U_{j+1} (To help visualize the proof, the case of $n = 3$ is illustrated in Figure 2.4). A binary mapping on U is said to be (d,k)-constrained if it induces a (d,k)-

constrained pattern on each U_j. Let N_T and N_{U_j} be the numbers of distinct (d,k)-constrained mappings on T and U_j (for $j = 0, 1, \ldots, l$), respectively. We show that $N_T \leq \prod_{j=0}^{l} N_{U_j}$.

To this end, it suffices to exhibit an injection from the set of all (d,k)-constrained mappings on T to those on U. Thus we demonstrate that every (d,k)-constrained mapping on T is completely determined by its restriction to U.

Assume the contrary. Then there exist two (d,k)-constrained mappings $f_0 : T \to \{0,1\}$ and $f_1 : T \to \{0,1\}$ that agree on U but differ on T. Let $(c_1, \ldots, c_{n-1}, c_n) \in T$ be such that $f_0(c_1, \ldots, c_{n-1}, c_n) \neq f_1(c_1, \ldots, c_{n-1}, c_n)$.

Since f_0 and f_1 agree on U, c_n must be a multiple of $k-d+1$. Let J be the smallest nonnegative integer j such that $f_0(c_1, \ldots, c_{n-1}, (k-d+1)j) \neq f_1(c_1, \ldots, c_{n-1}, (k-d+1)j)$. Without loss of generality assume $f_0(c_1, \ldots, c_{n-1}, (k-d+1)J) = 0$ and $f_1(c_1, \ldots, c_{n-1}, (k-d+1)J) = 1$. Note that $(k-d+1)(J+1)-1 \leq (k-d+1)J+d$ since $k \leq 2d$. Also, since $f_1(c_1, \ldots, c_{n-1}, (k-d+1)J) = 1$, f_1 must equal zero for at least d consecutive positions next to this point. Thus $f_1(c_1, \ldots, c_{n-1}, x) = 0$ for all x in the range $(k-d+1)J - d \leq x < (k-d+1)(J+1)$, excluding $x = (k-d+1)J$. Therefore $f_0(c_1, \ldots, c_{n-1}, x) = 0$ for this same set of x's, since either x is in U or else because of the choice of J. But by assumption $f_0(c_1, \ldots, c_{n-1}, (k-d+1)J) = 0$, so a string of $k+1$ zeros in a row occurs for f_0 (from $x = (k-d+1)J - d$ to $x = (k-1+d)(J+1)-1$) contradicting the (d,k) constraint. This proves that every (d,k)-constrained mapping on T is uniquely determined by its restriction to U. This establishes that $N_T \leq \prod_{j=0}^{l} N_{U_j}$.

Now, let M denote the number of distinct (d,k)-constrained mappings on an $(n-1)$-dimensional hypercube of side length m. Clearly, $\prod_{j=0}^{l} N_{U_j} \leq M^{(k-d)l+d}$, since $N_{U_0} \le M^d$ and $N_{U_j} \le M^{k-d}$ for $j = 1, 2, \ldots, l$. Thus,

$$\begin{aligned}
C_{d,k}^{(n)} &= \lim_{l,m\to\infty} \frac{\log_2 N_T}{\Big((k-d+1)l+d\Big)m^{n-1}} \le \lim_{l,m\to\infty} \frac{\log_2 \prod_{j=0}^{l} N_{U_j}}{\Big((k-d+1)l+d\Big)m^{n-1}} \\
&\le \lim_{l,m\to\infty} \frac{\log_2 M^{(k-d)l+d}}{\Big((k-d+1)l+d\Big)m^{n-1}} \\
&= \lim_{l\to\infty} \frac{(k-d)l+d}{(k-d+1)l+d} \cdot \lim_{m\to\infty} \frac{\log_2 M}{m^{n-1}} = \frac{k-d}{k-d+1} C_{d,k}^{(n-1)}.
\end{aligned}$$

□

2.4 Comments

For $d = 1$, Lemma 2.3(e) implies that $C_{1,3}^{(n)} \ge 1/4$ for $n \ge 3$. A more complicated proof can show that $C_{1,3}^{(n)} \ge C_{0,1}^{(n)}/2$ for all $n \ge 2$ (note that $C_{0,1}^{(n)} \ge 1/2$ by Lemma 2.3(e)). For odd $d \ge 3$ Proposition 2.6 gives $C_{d,d+2}^{(2)} \ge \frac{d-1}{2(d+2)^2}$ whereas a slightly better lower bound $C_{d,d+2}^{(2)} \ge \frac{d+1}{2(d+2)^2}$ was given in [5, Theorem 2]. Propositions 2.5 and 2.6 establish that $C_{d,d+2}^{(n)} > 0$. Alternatively it is possible to prove $C_{d,d+2}^{(n)} > 0$ in a simpler manner, but with weaker lower bounds on $C_{d,d+2}^{(n)}$ than those given in these propositions.

This chapter, in full, is a reprint of the material as it appears in: H. Ito, A. Kato, Zs. Nagy, and K. Zeger, Zero Capacity Region of Multidimensional Run Length Constraints, *Elec. J. Combinatorics*, 6(1)(R33), 1999.

References

[1] J. J. Ashley and P. H. Siegel. A Note on the Shannon Capacity of Run-Length-Limited Codes. *IEEE Trans. Information Theory*, 33(4):601–605, July 1987.

[2] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[3] J. Dénes and A. D. Keedwell. *Latin Squares: New Developments in the Theory and Applications*, Elsvier, Amsterdam, 1991.

[4] K. A. S. Immink, P. H. Siegel, and J. K. Wolf. Codes for Digital Recorders. *IEEE Trans. Inform. Theory*, 44:2260–2299, October 1998.

[5] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[6] D. Lind and B. H. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, New York, 1995.

[7] Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel. *IEEE Trans. Inform. Theory*, 46(3):1030–1033, May 2000.

x_2

x_1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	d-2	d-1	d	d+1
1	0	3	2	5	4	7	6	9	8	11	10	13	12	d-1	d-2	d+1	d
2	3	4	5	6	7	8	9	10	11	12	13	14	15	d	d+1	0	1
3	2	5	4	7	6	9	8	11	10	13	12	15	14	d+1	d	1	0
4	5	6	7	8	9	10	11	12	13	14	15	d	d+1	0	1	2	3
5	4	7	6	9	8	11	10	13	12	15	14	d+1	d	1	0	3	2
6	7	8	9	10	11	12	13	14	15	d	d+1	0	1	2	3	4	5
7	6	9	8	11	10	13	12	15	14	d+1	d	1	0	3	2	5	4
8	9	10	11	12	13	14	15	d	d+1	0	1	2	3	4	5	6	7
9	8	11	10	13	12	15	14	d+1	d	1	0	3	2	5	4	7	6
10	11	12	13	14	15	d	d+1	0	1	2	3	4	5	6	7	8	9
11	10	13	12	15	14	d+1	d	1	0	3	2	5	4	7	6	9	8
12	13	14	15	d	d+1	0	1	2	3	4	5	6	7	8	9	10	11
13	12	15	14	d+1	d	1	0	3	2	5	4	7	6	9	8	11	10
d-2	d-1	d	d+1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
d-1	d-2	d+1	d	1	0	3	2	5	4	7	6	9	8	11	10	13	12
d	d+1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	d-2	d-1
d+1	d	1	0	3	2	5	4	7	6	9	8	11	10	13	12	d-1	d-2

Figure 2.1: Latin square $\bar{e}_2$ for $d = 16$ (even d)

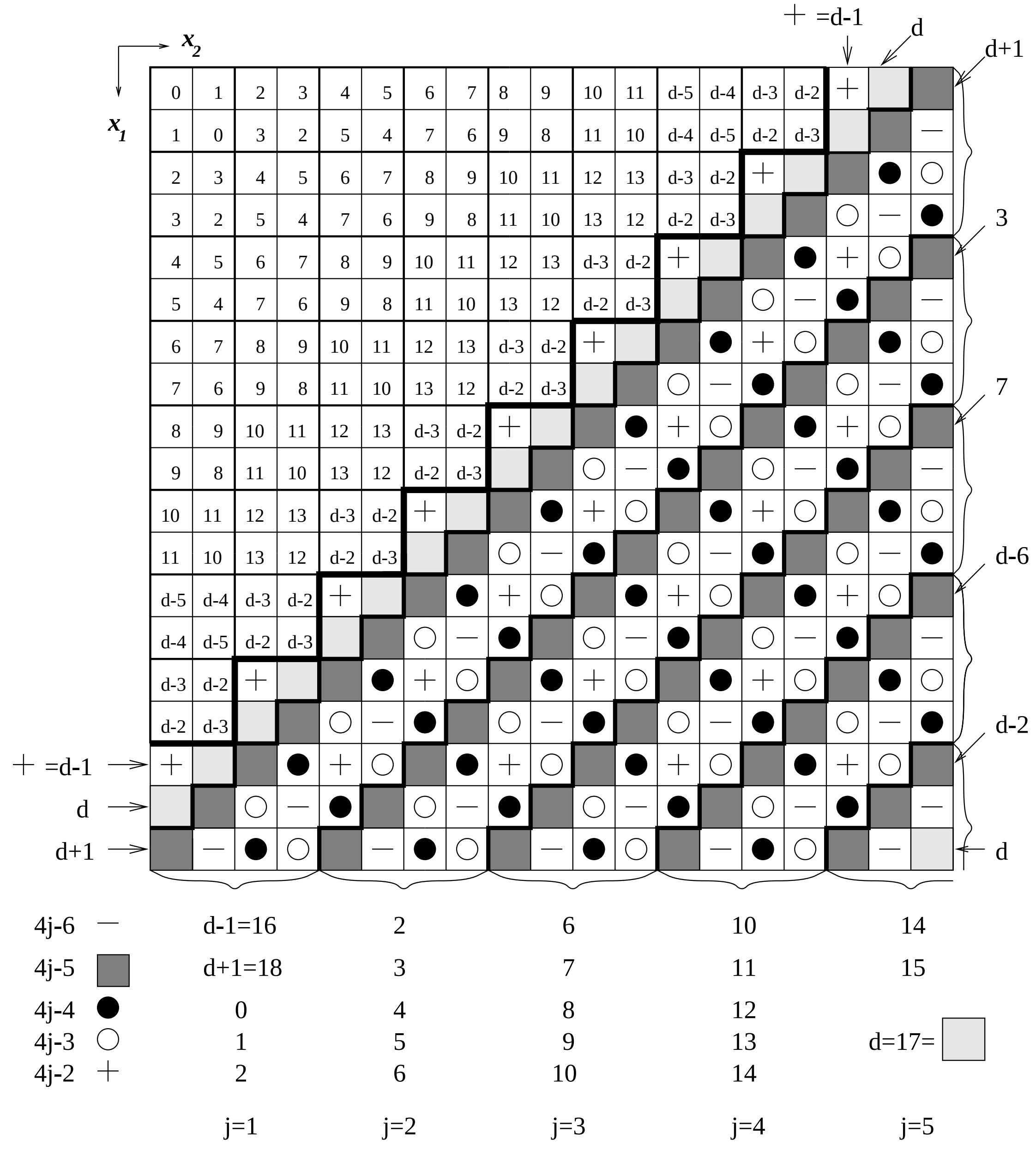


Figure 2.2: Latin square $\bar{e}_2$ for $d = 17$ ($d \equiv 1 \mod 4$)

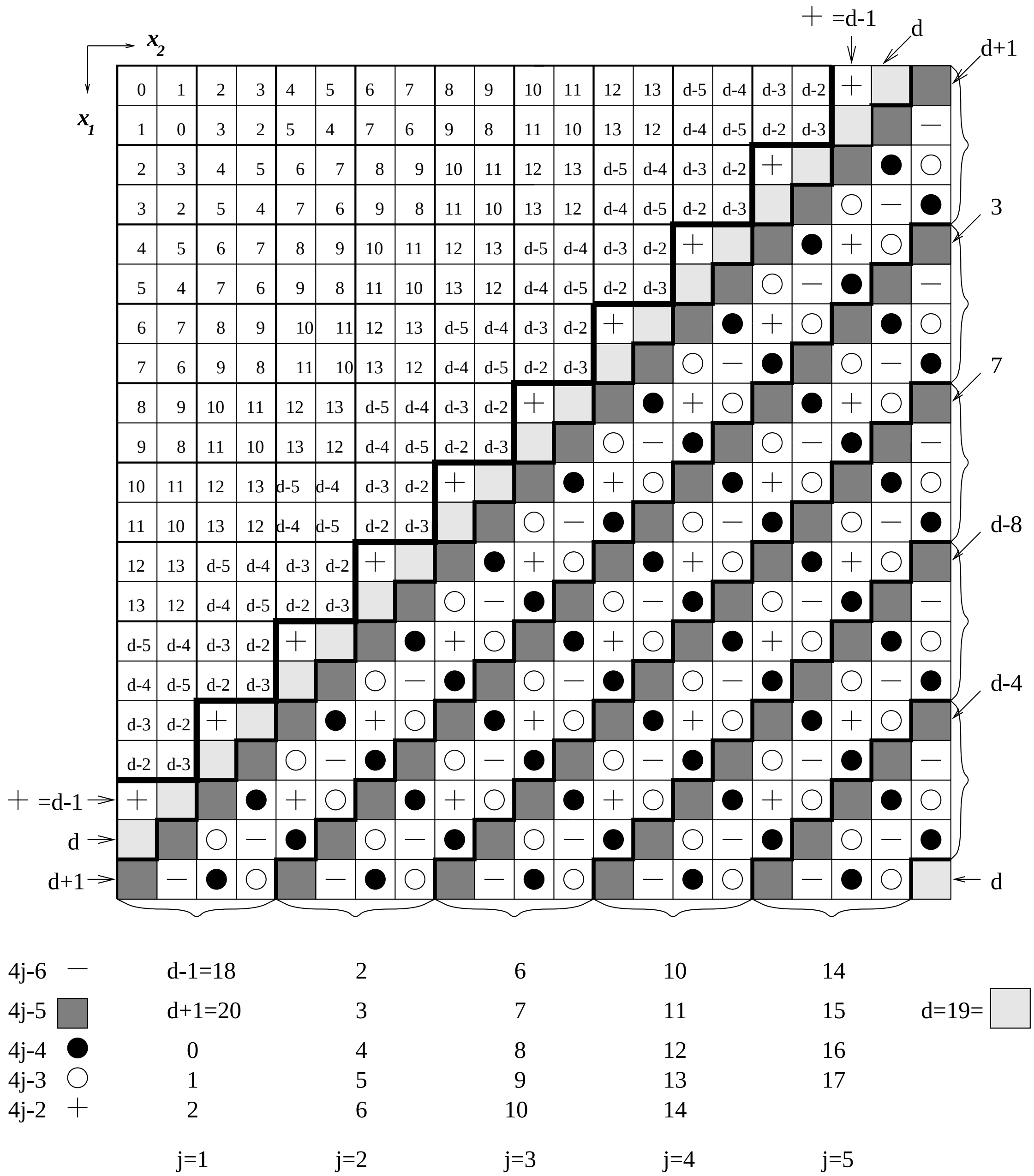


Figure 2.3: Latin square $\bar{e}_2$ for $d = 19$ ($d \equiv 3 \mod 4$)

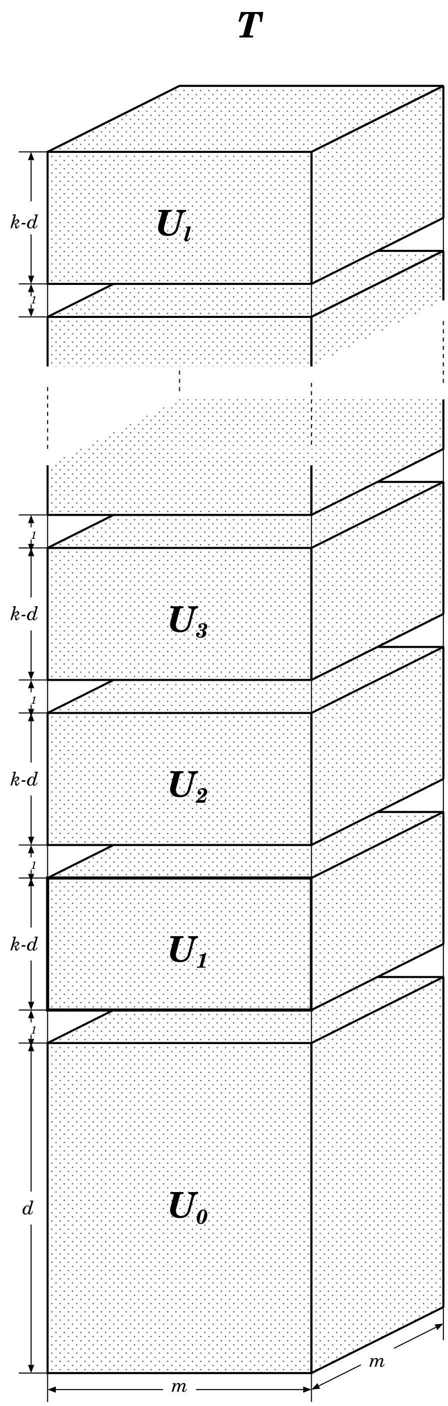


Figure 2.4: Illustration of the sets T and U_j for three dimensions in the proof of Lemma 2.7.

Chapter 3

Capacity Bounds for the Three-Dimensional $(0,1)$ Run Length Limited Channel

Abstract

The capacity $C_{0,1}^{(3)}$ of a 3-dimensional $(0,1)$ run length constrained channel is shown to satisfy $0.522501741838 \leq C_{0,1}^{(3)} \leq 0.526880847825$.

3.1 Introduction

A binary sequence satisfies a 1-dimensional (d,k) run length constraint if the number of consecutive 0s is at most k, and between any two 1s in the sequence are at least d 0s. An n-dimensional binary array is said to satisfy a (d,k) run length constraint, if it satisfies the 1-dimensional (d,k) run length constraint along every direction parallel to a coordinate axis. Such an array is called *valid*. The number of valid n-dimensional arrays of size $m_1 \times m_2 \times \ldots \times m_n$ is denoted by $N_{m_1,m_2,\ldots,m_n}^{(d,k)}$ and the corresponding *capacity* is defined as

$$C_{d,k}^{(n)} = \lim_{m_1,m_2,\ldots m_n \to \infty} \frac{\log_2 N_{m_1,m_2,\ldots m_n}^{(d,k)}}{m_1 m_2 \cdots m_n}.$$

By exchanging the roles of 0 and 1 it can be seen that $C_{0,1}^{(n)} = C_{1,\infty}^{(n)}$ for all $n \geq 1$.

A simple proof of the existence of the 2-dimensional (d,k) capacities can be found in [7], and the proof can be generalized to n-dimensions.

It is known (e.g. see [8]) that the 1-dimensional $(0,1)$-constrained capacity is the logarithm of the golden ratio, i.e.

$$C_{0,1}^{(1)} = \log_2 \frac{1+\sqrt{5}}{2} = 0.694242\ldots$$

and in [2] very close upper and lower bounds were given for the 2-dimensional $(0,1)$-constrained capacity. The bounds in [2] were calculated with greater precision in [10] and are slightly improved here (see Remark section at end for more details), now agreeing in 9 decimal positions:

$$0.587891161775 \le C_{0,1}^{(2)} \le 0.587891161868 \quad . \tag{3.1}$$

These bounds were also independently obtained to 8 decimal positions in [3]. A lower bound of $C_{0,1}^{(2)} \ge 0.5831$ was obtained in [9] by using an implementable encoding procedure known as "bit-stuffing". The known bounds on $C_{0,1}^{(2)}$ have played a useful role in [7] for obtaining bounds on other (d,k)-constraints in two dimensions. The 3-dimensional $(0,1)$-constrained bounds given in the present paper can play a similar role for obtaining different 3-dimensional bounds, and are also of theoretical interest. In fact, a recent tutorial paper [6] discusses an interesting connection between run length constrained capacities in more than one dimension and crossword puzzles (based on work of Shannon from 1948). In the present paper we consider the 3-dimensional $(0,1)$ constraint, and by extending ideas from [2] and using two new bounds, our main result is to derive (in Sections 3.2 and 3.3) the following bounds on the 3-dimensional $(0,1)$ capacity.

Theorem 3.1.

$$0.522501741838 \le C_{0,1}^{(3)} \le 0.526880847825$$

It is assumed henceforth in this paper that $d = 0$ and $k = 1$. Two valid $m_1 \times m_2$ rectangles can be put next to each other in 3 dimensions without violating the 3-dimensional $(0,1)$ constraint if they have no two zeros in the same positions. Define a *transfer matrix* T_{m_1,m_2} to be an $N_{m_1,m_2}^{(0,1)} \times N_{m_1,m_2}^{(0,1)}$ binary matrix, such that the rows and columns are indexed by the valid 2-dimensional $m_1 \times m_2$ patterns, and an entry of T_{m_1,m_2} is 1 if and only if the corresponding two rectangles can be placed next to each other in 3 dimensions without violating the $(0,1)$ constraint. Then,

$$N_{m_1,m_2,m_3}^{(0,1)} = \mathbf{1}' \cdot T_{m_1,m_2}^{m_3-1}\mathbf{1} = \mathbf{1}' \cdot T_{m_1,m_3}^{m_2-1}\mathbf{1} = \mathbf{1}' \cdot T_{m_2,m_3}^{m_1-1}\mathbf{1} \tag{3.2}$$

where **1** is the all ones column vector and prime denotes transpose. The matrix T_{m_1,m_2} meets the conditions of the Perron-Frobenius theorem [1], since it has nonnegative real elements and is irreducible (since the all ones rectangle can be placed next to any valid rectangle without violating the $(0,1)$ constraint). Therefore the largest magnitude eigenvalue Λ_{m_1,m_2}, of T_{m_1,m_2}, is positive, real, and has multiplicity one. This implies that

$$\lim_{m_3\to\infty} (N_{m_1,m_2,m_3}^{(0,1)})^{1/m_3} = \Lambda_{m_1,m_2},$$

and

$$\begin{aligned}
C_{0,1}^{(3)} &= \lim_{m_1,m_2,m_3\to\infty} \frac{\log_2 N_{m_1,m_2,m_3}^{(0,1)}}{m_1 m_2 m_3} \\
&= \lim_{m_1,m_2\to\infty} \frac{\log_2 \lim_{m_3\to\infty}(N_{m_1,m_2,m_3}^{(0,1)})^{1/m_3}}{m_1 m_2} \\
&= \lim_{m_1,m_2\to\infty} \frac{\log_2 \Lambda_{m_1,m_2}}{m_1 m_2} \\
&= \lim_{m_1\to\infty} \frac{\log_2 \lim_{m_2\to\infty} \Lambda_{m_1,m_2}^{1/m_2}}{m_1} \\
&= \lim_{m_1\to\infty} \frac{\log_2 \Lambda_{m_1}}{m_1}, \qquad (3.3)
\end{aligned}$$

where $\Lambda_{m_1} = \lim_{m_2\to\infty} \Lambda_{m_1,m_2}^{1/m_2}$. The quantities $\frac{\log_2 \Lambda_{m_1,m_2}}{m_1 m_2}$ and $\frac{\log_2 \Lambda_{m_1}}{m_1}$ can be viewed as capacities corresponding to 3-dimensional arrays with two fixed sides (lengths m_1 and m_2), and one fixed side (length m_1), respectively.

Upper and lower bounds on the 3-dimensional capacity can be computed directly from the inequalities (similar to the 2-dimensional case, as noted in [10])

$$\frac{\log_2 \Lambda_{m_1,m_2}}{(m_1+1)(m_2+1)} \le C_{0,1}^{(3)} \le \frac{\log_2 \Lambda_{m_1,m_2}}{m_1 m_2}$$

but these do not yield particularly tight bounds for values of m_1 and m_2 for which the corresponding value of Λ_{m_1,m_2} could be computed by us. (e.g. Table 3.1 shows that the eigenvalues Λ_{m_1,m_2} correspond to matrices with more than 40 million elements when roughly $m_1 m_2 \ge 20$). The upper and lower capacity bounds derived in this paper agree to within ± 0.002 and were computed using less than 100 Mbytes of computer memory.

3.2 Lower bound on $C_{0,1}^{(3)}$

To derive a lower bound on $C_{0,1}^{(3)}$ we generalize a method of Calkin and Wilf [2]. Since T_{m_1,m_2} is a symmetric matrix, the Courant-Fischer Minimax Theorem [4, pg. 394] implies that

$$\Lambda_{m_1,m_2}^p \geq \frac{\mathbf{x}' \cdot T_{m_1,m_2}^p \mathbf{x}}{\mathbf{x}' \cdot \mathbf{x}} \tag{3.4}$$

for any nonzero vector $\mathbf{x}$ and any integer $p \geq 0$. Choosing $\mathbf{x} = T_{m_1,m_2}^q \mathbf{1}$ for any integer $q \geq 0$, and using identity (3.2) gives

$$\Lambda_{m_1,m_2}^p \geq \frac{\mathbf{1}' \cdot T_{m_1,m_2}^{p+2q} \mathbf{1}}{\mathbf{1}' \cdot T_{m_1,m_2}^{2q} \mathbf{1}} = \frac{\mathbf{1}' \cdot T_{m_1,p+2q+1}^{m_2-1} \mathbf{1}}{\mathbf{1}' \cdot T_{m_1,2q+1}^{m_2-1} \mathbf{1}}. \tag{3.5}$$

Thus,

$$\begin{aligned} 2^{pC_{0,1}^{(3)}} &= \left(\lim_{m_1,m_2\to\infty} \Lambda_{m_1,m_2}^{1/(m_1 m_2)} \right)^p = \lim_{m_1\to\infty} \left(\lim_{m_2\to\infty} \Lambda_{m_1,m_2}^{p/m_2} \right)^{1/m_1} \\ &\geq \lim_{m_1\to\infty} \left(\frac{\Lambda_{m_1,p+2q+1}}{\Lambda_{m_1,2q+1}} \right)^{1/m_1} = \frac{\lim_{m_1\to\infty} \Lambda_{m_1,p+2q+1}^{1/m_1}}{\lim_{m_1\to\infty} \Lambda_{m_1,2q+1}^{1/m_1}} = \frac{\Lambda_{p+2q+1}}{\Lambda_{2q+1}} \end{aligned} \tag{3.6}$$

and therefore for any odd integer $r \geq 1$ and any integer $z > r$,

$$C_{0,1}^{(3)} \geq \frac{1}{z-r} \log_2 \left(\frac{\Lambda_z}{\Lambda_r} \right). \tag{3.7}$$

This lower bound on $C_{0,1}^{(3)}$ is analogous to a 2-dimensional bound in [2], but Λ_z and Λ_r are not eigenvalues associated with transfer matrices of 2-dimensional arrays here, and cannot easily be computed as in the 2-dimensional case. Instead, we obtain a lower bound on Λ_z and an upper bound on Λ_r. From (3.5) and (3.6) a lower bound on Λ_z is

$$\Lambda_z = \lim_{m_2\to\infty} \Lambda_{z,m_2}^{1/m_2} \geq \lim_{m_2\to\infty} \left(\frac{\mathbf{1}' \cdot T_{z,v}^{m_2-1} \mathbf{1}}{\mathbf{1}' \cdot T_{z,u}^{m_2-1} \mathbf{1}} \right)^{1/((v-u)m_2)} = \left(\frac{\Lambda_{z,v}}{\Lambda_{z,u}} \right)^{1/(v-u)},$$

where u is an arbitrary positive odd integer, $v > u$, and $\Lambda_{z,v}$ and $\Lambda_{z,u}$ are the largest eigenvalues of the transfer matrices $T_{z,v}$ and $T_{z,u}$, respectively.

To find an upper bound on the quantity Λ_r for a given r, we apply a modified version of a method in [2]. We say that a binary matrix satisfies the $(0,1)$ *cylindrical constraint* if it satisfies the usual 2-dimensional $(0,1)$ constraint after joining its leftmost column to its rightmost column (i.e. the left and right columns can be put next to each other without violating the $(0,1)$ constraint). A binary matrix satisfies the $(0,1)$ *toroidal constraint* if it satisfies the usual 2-dimensional $(0,1)$ constraint after both joining its leftmost column to its rightmost column, and its top row to its bottom row.

Proposition 3.2. *Let s be a positive integer and let T_{m_1,m_2} be the transfer matrix whose rows and columns are indexed by all $(0,1)$-constrained $m_1 \times m_2$ rectangles. Let $B_{m_1,s}$ denote the transfer matrix whose rows and columns are indexed by all cylindrically $(0,1)$-constrained $m_1 \times s$ rectangles. Then,*

$$\mathrm{Trace}[\mathrm{T}^{\mathrm{s}}_{\mathrm{m_1,m_2}}] = \boldsymbol{1}' \cdot \mathrm{B}^{\mathrm{m_2-1}}_{\mathrm{m_1,s}} \boldsymbol{1}.$$

Proof. Trace$[T^s_{m_1,m_2}]$ is the number of $m_1 \times m_2 \times (s+1)$ valid arrays, whose first and last $m_1 \times m_2$ rectangles are the same, or equivalently the number of 3-dimensional $m_1 \times m_2 \times s$ valid arrays, whose first $m_1 \times m_2$ rectangle can be put after the last one without violating the $(0,1)$ constraint. Viewing these 3-dimensional arrays along their side of length m_2, they can be described as a sequence of m_2 cylindrically $(0,1)$-constrained 2-dimensional rectangles of size $m_1 \times s$ (see Figure 3.1), and thus the number of arrays counting in this manner is the sum of the entries in $B^{m_2-1}_{m_1,s}$. $\square$

The proof above generalizes the 2-dimensional version in [2]. Let s be a positive even integer. Then for every positive integer m_1 and m_2, the matrix $T^s_{m_1,m_2}$ has non-negative eigenvalues and thus any one of its eigenvalues is upper bounded by its trace.

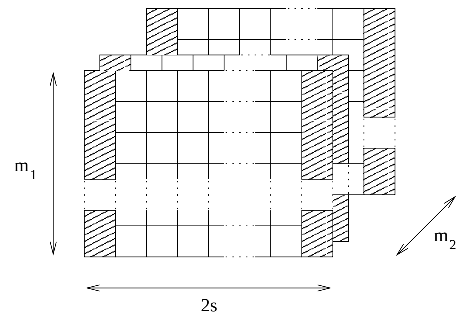


Figure 3.1: Cylindrically $(0,1)$-constrained $m_1 \times s$ rectangles used to build cylindric $m_1 \times m_2 \times s$ arrays.

Hence,

$$\Lambda_{m_1,m_2} \leq \text{Trace}\left[T^s_{m_1,m_2}\right]^{1/s} = \left(\mathbf{1}' \cdot B^{m_2-1}_{m_1,s}\mathbf{1}\right)^{1/s} \tag{3.8}$$

which gives the following upper bound on Λ_r:

$$\Lambda_r = \lim_{m_2\to\infty} \Lambda^{1/m_2}_{r,m_2} \leq \lim_{m_2\to\infty} \left(\mathbf{1}' \cdot B^{m_2-1}_{r,s}\mathbf{1}\right)^{\frac{1}{sm_2}} = \xi^{1/s}_{r,s}, \tag{3.9}$$

where $\xi_{r,s}$ is the largest eigenvalue of $B_{r,s}$ (note that $B_{r,s}$ satisfies the Perron-Frobenius theorem for the same reasons as for T_{m_1,m_2} in Section 3.1).

The lower bound on $C^{(3)}_{0,1}$ in (3.7) can now be written as

$$C^{(3)}_{0,1} \geq \frac{1}{z-r}\log_2\left(\frac{\left(\frac{\Lambda_{z,v}}{\Lambda_{z,u}}\right)^{1/(v-u)}}{\xi^{1/s}_{r,s}}\right) \qquad \begin{array}{l} r \text{ and } u \text{ odd, } s \text{ even} \\ z > r \geq 1 \\ v > u \geq 1 \\ s \geq 2 \end{array} \tag{3.10}$$

To obtain the best possible lower bound, the right hand side of (3.10) should be maximized over all acceptable choices of r, z, u, v, and s, subject to the numerical computability of the quantities $\Lambda_{z,v}$, $\Lambda_{z,u}$, and $\xi_{r,s}$. Table 3.1 shows the largest eigenvalues of various transfer matrices which were numerically computable. From this table, the best parameters we could find for the lower bound in (3.10) on the capacity were $r = 3$, $z = 4$, $u = 5$, $v = 6$, and $s = 10$, yielding

$$C_{0,1}^{(3)} \geq \frac{1}{4-3} \log_2 \frac{\frac{9346.35893701}{2102.73425568}}{(80481.0598379)^{1/10}} \geq 0.522501741838.$$

3.3 Upper bound on $C_{0,1}^{(3)}$

Proposition 3.3. *Let s_1 and s_2 be positive even integers and let $B^*_{s_1,s_2}$ denote the transfer matrix whose rows and columns are indexed by all toroidally $(0,1)$-constrained $s_1 \times s_2$ rectangles. If $\xi^*_{s_1,s_2}$ is the largest eigenvalue of $B^*_{s_1,s_2}$, then $C_{0,1}^{(3)} \leq \frac{1}{s_1 s_2} \log_2 \xi^*_{s_1,s_2}$.*

Proof. Let T_{m_1,m_2} and B_{m_1,s_1} be the same transfer matrices as defined in Section 3.2, and let ξ_{m_1,s_1} denote the largest eigenvalue of B_{m_1,s_1}. From Proposition 3.2 and the argument used to obtain inequality (3.9) we can also conclude that

$$\Lambda_{m_1} \leq \xi_{m_1,s_1}^{1/s_1}.$$

Also, the same argument used to obtain (3.8) gives

$$\xi_{m_1,s_1} \leq \left(\text{Trace}\left[B_{m_1,s_1}^{s_2}\right]\right)^{1/s_2} = \left(\mathbf{1}, (B^*_{s_1,s_2})^{m_1-1}\mathbf{1}\right)^{1/s_2}$$

and thus

$$\Lambda_{m_1}^{1/m_1} \leq \xi_{m_1,s_1}^{1/(m_1 s_1)} \leq \left(\mathbf{1}, (B^*_{s_1,s_2})^{m_1-1}\mathbf{1}\right)^{1/(m_1 s_1 s_2)}.$$

This uses the fact that $B^*_{s_1,s_2}$ satisfies the Perron-Frobenius theorem (for the same reasons as for T_{m_1,m_2} in Section 3.1). Since $C_{0,1}^{(3)} = \lim_{m_1\to\infty} \log_2 \Lambda_{m_1}^{1/m_1}$, we have

$$2^{C_{0,1}^{(3)}} = \lim_{m_1\to\infty} \Lambda_{m_1}^{1/m_1} \leq \left(\xi^*_{s_1,s_2}\right)^{1/(s_1 s_2)}.$$

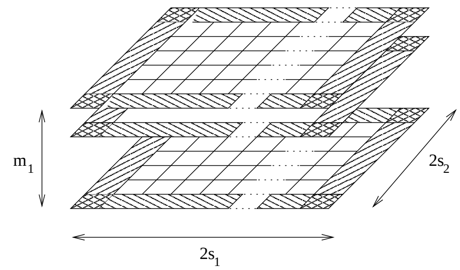


Figure 3.2: Toroidally $(0,1)$-constrained $s_1 \times s_2$ rectangles used to build doubly cylindric $m_1 \times s_1 \times s_2$ arrays.

□

Proposition 3.3 generalizes an upper bound in [2] and is illustrated in Figure 3.2. Note that $B_{2,s_2} = B^*_{2,s_2}$ and thus $\xi_{2,s_2} = \xi^*_{2,s_2}$. The best parameters we were able to find (from Table 3.1) were $s_1 = 4$ and $s_2 = 6$, and the resulting eigenvalue gave the following upper bound:

$$C^{(3)}_{0,1} \leq \frac{1}{24} \log_2 6405.69924332 \leq 0.526880847825.$$

3.4 Remark

Direct computation of eigenvalues using standard linear algebra algorithms generally requires the storage of an entire matrix. This severely restricts the matrix sizes allowable, due to memory constraints on computers. By exploiting the fact that our matrices are all binary, symmetric, and easily computable, we were able to obtain the largest eigenvalues of much larger matrices. Specifically, the eigenvalues used to obtain the capacity bounds in Theorem 3.1 were computed using the following result.

Lemma 3.4. ***[5, pg. 493]*** *Let A be an $n \times n$ matrix with nonnegative entries only. Then for any n-dimensional positive vector* ***x*** *we have*

$$\min_{1 \le i \le n} \frac{1}{x_i} \sum_{j=1}^{n} a_{ij} x_j \le \rho(A) \le \max_{1 \le i \le n} \frac{1}{x_i} \sum_{j=1}^{n} a_{ij} x_j ,$$

and

$$\min_{1 \le j \le n} x_j \sum_{i=1}^{n} \frac{a_{ij}}{x_i} \le \rho(A) \le \max_{1 \le j \le n} x_j \sum_{i=1}^{n} \frac{a_{ij}}{x_i} \quad ,$$

where $\rho(A)$ denotes the spectral radius of the matrix A.

The convergence rate of the power method depends on the relative size of the largest and second largest eigenvalues, but the second largest eigenvalues are generally unknown to us. Hence, we iterated the eigenvalue computation until the eigenvalues appeared to stabilize in the 14th significant decimal place (computing $\Lambda_{4,5}$, $\Lambda_{4,6}$, $\xi_{3,10}$, and $\xi^*_{4,6}$). The resulting eigenvector estimates were used as the values of **x** in Lemma 3.4 to obtain *exact* upper and lower bounds on the largest eigenvalues.

Similarly, we obtained the upper bound in (3.1) with the power method (computing $\Lambda_{1,21}$, $\Lambda_{1,23}$, and $\xi_{1,24}$). Originally these bounds were computed in [2] as $0.587891161 \le C^{(2)}_{0,1} \le 0.588339078$ (computing $\Lambda_{1,13}$, $\Lambda_{1,15}$, and $\xi_{1,6}$) and were later improved in [10] (computing $\Lambda_{1,13}$, $\Lambda_{1,14}$, and $\xi_{1,14}$) to $0.587891161775 \le C^{(2)}_{0,1} \le 0.587891494943$. The lower bound in (3.1) is from [10].

We expect the bounds in (3.10) and in Proposition 3.3 to improve in the future as increased computational speed and memory expand more of Table 3.1.

This chapter, in full, is a reprint of the material as it appears in: Zs. Nagy and K. Zeger, Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel, *IEEE Trans. Inform. Theory*, 46(3):1030–1033, May 2000. The dissertation author was the primary investigator of this paper.

References

[1] R. B. Bapat and T. E. S. Raghavan. *Nonnegative Matrices and Applications.* Cambridge University Press, Cambridge, U.K., 1997.

[2] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[3] S. Forchhammer and J. Justesen. Entropy Bounds for Constrained Two-Dimensional Random Fields. *IEEE Trans. Inform. Theory*, 45:118–127, January 1999.

[4] G. H. Golub and C. F. van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD and London, U.K., 1996.

[5] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, U.K., 1985.

[6] K. A. S. Immink, P. H. Siegel, and J. K. Wolf. Codes for Digital Recorders. *IEEE Trans. Inform. Theory*, 44:2260–2299, October 1998.

[7] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[8] D. Lind and B. H. Marcus. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, New York, 1995.

[9] P. H. Siegel and J. K. Wolf. Bit Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays. *Proc. 1998 IEEE Intl. Symp. on Inform. Theory*, page 323, MIT, Cambridge, MA, August 1998.

[10] W. Weeks and R. E. Blahut. The Capacity and Coding Gain of Certain Checkerboard Codes. *IEEE Trans. Inform. Theory*, 44(3):1193–1203, May 1998.

Table 3.1: The largest eigenvalues of the transfer matrices $T_{a,b}$, $B_{a,b}$, and $B^*_{a,b}$ are $\Lambda_{a,b}$, $\xi_{a,b}$, and $\xi^*_{a,b}$, respectively. The values for $B_{a,b}$ are only given when b is even, and for $B^*_{a,b}$ when both a and b are even. Eigenvalue entries in the table with a "*" next to them indicate that they were computed using the power method, instead of by direct computation (see Remark section). The eigenvalues $\Lambda_{a,b}$ and $\xi_{a,b}$ are symmetric in their indices.

a	b	$\Lambda_{a,b}$	rows of $T_{a,b}$	$\xi_{a,b}$	rows of $B_{a,b}$	$\xi^*_{a,b}$	rows of $B^*_{a,b}$
1	1	1.61803398875	2				
	2	2.41421356237	3	2.41421356237	3		
	3	3.63138126040	5				
	4	5.45770539597	8	5.15632517466	7		
	5	8.20325919376	13				
	6	12.3298822153	21	11.5517095660	18		
	7	18.5324073775	34				
	8	27.8550990963	55	26.0579860919	47		
	9	41.8675533183	89				
	10	62.9289457252	144	58.8519350815	123		
	11	94.5852312050	233				
	12	142.166150393	377	132.947794048	322		
	13	213.682559741	610				
	14	321.175161677	987	300.345852027	843		
	15	482.741710897	1597				
	16	725.584002895*	2584	678.525669346	2207		
	17	1090.58764423*	4181				
	18	1639.20566742*	6765	1532.89283597*	5778		
	19	2463.80493521*	10946				
	20	3703.21728345*	17711	3463.03987027*	15127		
	21	5566.11363689*	28657				
	22	8366.13642876*	46368	7823.53857819*	39603		
	23	12574.7053170*	75025				
	24	18900.3867144*	121393	17674.5747630*	103682		
2	2	5.15632517466	7	5.15632517466	7	5.15632517466	7
	3	11.1103016575	17				
	4	23.9250625386	41	21.9287654025	35	21.9287654025	35
	5	51.5229210280	99				
	6	110.954925971	239	100.236549238	199	100.236549239	199
	7	238.942175857	577				
	8	514.563569622	1393	463.203410887	1155	463.203410887	1155
	9	1108.11608218*	3363				
	10	2386.33538059*	8119	2146.04060032*	6727	2146.04060032*	6727
	11	5138.98917320*	19601				
	12	11066.8474924*	47312	9949.63685703*	39203	9949.63685703*	39203
3	3	34.4037405361	63				
	4	106.439377528	227	94.2548937790	181		
	5	329.331697608	827				
	6	1018.97101980*	2999	884.498791440	2309		
	7	3152.75734322*	10897				
	8	9754.81971205*	39561	8421.60680806*	30277		
	9	30181.9963196*	143677				
	10	93384.9044989*	521721	80481.0598378*	398857		
4	4	473.069084944	1234	404.943621498	933	355.525781764	743
	5	2102.73425567*	6743				
	6	9346.35893702*	36787	7799.87080772*	26660	6405.69924332*	18995

Chapter 4

Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions

Abstract

A rigorous derivation is given of the coding rate of a variable-to-variable length bit stuffing coder for a two-dimensional $(1, \infty)$-constrained channel. The coder studied is “nearly” a fixed-to-fixed length algorithm. Then an analogous variable-to-variable length bit stuffing algorithm for the three-dimensional $(1, \infty)$-constrained channel is presented, and its coding rate is analyzed using the two-dimensional method. The three-dimensional coding rate is demonstrated to be at least 0.502, which is proven to be within 4% of the capacity.

4.1 Introduction

A binary sequence satisfies the (d, k) *run length constraint* if the number of consecutive 0s is at most k, and between any two 1s in the sequence are at least d 0s. A subset of $\mathbf{Z}^n$ satisfies the n-dimensional (d, k) constraint if it satisfies the one-dimensional (d, k) constraint along directions parallel with every coordinate axis. Run length constrained binary sequences in one and more dimensions have applications in magnetic and optical data storage systems, and have been studied extensively [9]. Other two-dimensional constraints such as asymmetric run length constraints, run length constraints along diagonals, and constraints defined by two-dimensional sets are also of theoretical and practical interest [1], [6], [14], [20], [21], [23]. Three-dimensional constraints were studied in [7] and [13], and the positive capacity region of general n-dimensional run length constraints was determined in [10]. The mathematical analysis of high dimensional constraints often is more difficult than the one-dimensional case.

For practical applications, implementable and efficient coding schemes are needed, but only a few such algorithms exist for two and higher dimensional constraints. Some examples for conservative and weight-constrained arrays can be found in [15], [21], and [22].

An important special channel is when $d = 1$ and $k = \infty$ (or equivalently when $d = 0$ and $k = 1$) and this paper will concentrate exclusively on the $(1, \infty)$ run length constraint. In one dimension, the $(1, \infty)$ constrained channel capacity is known exactly. In two dimensions, the channel capacity has been studied by Calkin and Wilf [3] and Engel [5], and for three dimensions it was studied in [13]. The capacity of the $(1, \infty)$ constraint is not known exactly in two and higher dimensions but has been very accurately upper and lower bounded in two and three dimensions.

One particularly efficient algorithm for coding under a $(1, \infty)$ constraint is called "bit stuffing" and was first proposed in 1988 by Lee [12] for the one-dimensional $(0, k)$ constraint. Bit stuffing was then generalized in 1993 by Bender and Wolf [2] to the one-dimensional (d, k) constraint and in 1998 by Siegel and Wolf [19] to the two-dimensional (d, ∞) constraint. In 2002 Halevy et. al [8] generalized bit stuffing to hexagonal two-dimensional lattices for certain (d, ∞) constraints.

An analysis of a two-dimensional bit stuffing algorithm for the $(1, \infty)$ constraint was presented by Roth, Siegel, and Wolf [16]. The algorithm converts an infinite unbiased independent and identically distributed (i.i.d.) binary input sequence into a biased i.i.d. sequence, before mapping the bits into $\mathbf{Z}^2$. In a subsequent paper [17] they improved the bit stuffing encoder (i.e. increased the coding rate closer to the channel capacity) by converting the input into two biased i.i.d. sequences. They also use a randomized initial labeling of certain points in $\mathbf{Z}^2$ in order to facilitate analysis.

The coding rate calculations in [16], [17], and [19] were performed without a precisely defined mapping from unbiased input sequences to biased sequences, and without prescribing how the infinite biased sequence is encoded using finite size regions in $\mathbf{Z}^2$. One specific (and efficient) implementation of the Roth-Siegel-Wolf coding algorithm would be to transform the unbiased input sequence into a biased sequence using an ideal arithmetic decoder, and then encode the biased sequence using bit stuffing. However, a rigorous analysis of such implementation appears difficult because of the behavior of arithmetic coders on finite length input sequences.

In this paper, we first examine a close variant of the Roth-Siegel-Wolf two-dimensional algorithm using their same underlying bit stuffing building block. Our encoder maps an infinite binary sequence into a $(1, \infty)$-constrained labeling of $\mathbf{Z}^2$, by parsing the input using a prefix code. The encoder is variable-to-variable length and uses a deterministic initial labeling, in contrast to the encoders in [16], [17], and [19]. We give a rigorous derivation for the coding rate of our two-dimensional algorithm (our coding rate is exactly the same as theirs, as expected). We then modify the two-dimensional algorithm to create a three-dimensional algorithm based on bit stuffing that maps an input binary sequence into $\mathbf{Z}^3$ and satisfies the $(1, \infty)$ constraint. Finally, the two-dimensional coding rate analysis is used (in part) to rigorously derive the coding rate of the three-dimensional algorithm. We prove that the coding rate in three dimensions is within 4% of the three-dimensional channel capacity.

The paper is organized as follows. In Section 4.2 basic definitions and terminology are introduced. Two-dimensional bit stuffing is described in Section 4.3 and our variable-to-variable length algorithm and analysis are given in Section 4.4. The two-dimensional coding rate derivation is given in Theorem 4.2. Three-dimensional bit stuffing is described in Section 4.5 and our variable-to-variable length algorithm and analysis are given in Section 4.6. The three-dimensional coding rate result is given in Theorem 4.7 and its maximum value is given in Theorem 4.19. Various tedious calculations are relegated to the Appendix.

4.2 Preliminaries

For any binary string s let $l(s)$ denote its length, $|s|$ the number of 1s in the string, and s_i the ith bit in the string. Let $\mathbf{Z}$ denote the integers and $\mathbf{Z}^+$ the positive integers. For any $n \in \mathbf{Z}^+$, let $\mathbf{Z}^n$ be the n-dimensional integer lattice. Throughout the paper N will denote a positive integer and random variables will be denoted with "hat" notation. Let

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$$

denote the binary entropy function.

A sequence $\hat{u}_0,\ \hat{u}_1, \ldots$ of random variables taking on values from a set A is called a *Markov chain*, if for all $n \in \mathbf{Z}^+$ and $a_n \in A$,

$$\mathbf{P}(\hat{u}_n = a_n | \hat{u}_{n-1} = a_{n-1}, \ldots, \hat{u}_0 = a_0) = \mathbf{P}(\hat{u}_n = a_n | \hat{u}_{n-1} = a_{n-1}).$$

A Markov chain is *homogeneous* (or *time invariant*) if $\mathbf{P}(\hat{u}_n = a | \hat{u}_{n-1}) = \mathbf{P}(\hat{u}_1 = a | \hat{u}_0)$ for all $n \in \mathbf{Z}^+$ and $a \in A$. For every $a, b \in A$ the conditional probabilities $\mathbf{P}(\hat{u}_1 = a | \hat{u}_0 = b)$ of a homogeneous Markov chain are called the *transition probabilities*. A Markov chain is *stationary* if $\mathbf{P}(\hat{u}_n = a) = \mathbf{P}(\hat{u}_0 = a)$ for all $n \in \mathbf{Z}^+$ and $a \in A$. We say that two homogeneous Markov chains are *identical* if both Markov chains take on values from the same set A, and have the same transition probabilities and initial probabilities.

For any $S \subset \mathbf{Z}^n$, a function $f : S \longrightarrow \{0,1\}$ is a *labeling* of S. Let $\Lambda_{d,k}^{(n)}(S)$ denote the set of all labelings of S that satisfy the n-dimensional (d,k) constraint. Such labelings are called *valid*. The *capacity* $C_{d,k}^{(n)}$ of the n-dimensional (d,k) constraint (or of the constrained channel) is

$$C_{d,k}^{(n)} = \lim_{m \to \infty} \frac{\log_2 |\Lambda_{d,k}^{(n)}(R_m^{(n)})|}{m^n}$$

where $R_m^{(n)} = \{0, 1, \ldots, m-1\}^n$ (there are various other equivalent definitions). The exact value of the capacity is not known in general. If $d = k$ then $C_{d,k}^{(n)} = 0$, and it has been shown [10], [11] that if $k > d \geq 1$ and $n \geq 2$ then $C_{d,k}^{(n)} = 0 \Longleftrightarrow k = d+1$. Numerical upper and lower bounds on $C_{1,\infty}^{(2)}$ were established in [3], and these bounds were later improved in [23] and then in [13]. The best known bounds on $C_{1,\infty}^{(2)}$ agree in the first 9 decimal places as

$$0.587891161775 \leq C_{1,\infty}^{(2)} \leq 0.587891161868.$$

Numerical bounds on the three-dimensional capacity $C_{1,\infty}^{(3)}$ were calculated in [13] as

$$0.522501741838 \leq C_{1,\infty}^{(3)} \leq 0.526880847825. \tag{4.1}$$

The n-dimensional capacity associated with a constraint is a theoretical bound on the average number of information bits that can be stored per position in $\mathbf{Z}^n$. The lower bound in (4.1), however, was not derived using a constructive encoding technique.

A constrained coding algorithm serves as a method for mapping an input binary information source into the lattice $\mathbf{Z}^n$ such that the constraint is not violated and such that the information source can be perfectly recovered from the labeling of $\mathbf{Z}^n$. The quality (or efficiency) of a coding algorithm is generally described by its coding rate.

The coding rate of an algorithm is a measure of the average ratio between the length of the input and the number of points in $\mathbf{Z}^n$ that are labeled for a particular input, in the limit as the amount of source information grows to infinity. The coding rate of any coding algorithm provides a lower bound on the capacity of the constraint.

An n-dimensional $(1,\infty)$-constrained *encoder* is an injection

$$\mathcal{E}^{(n)} : \{0,1\}^\infty \longrightarrow \bigcup_{S \subset \mathbf{Z}^n} \Lambda^{(n)}_{1,\infty}(S)$$

and its inverse is called a *decoder*. The encoder $\mathcal{E}^{(n)}$ maps an infinite binary input sequence into a labeling of a subset of $\mathbf{Z}^n$. An encoder and decoder are together called a *coding algorithm*.

One way to implement an encoder is to first parse the infinite binary source and then independently map each resulting finite length binary string into disjoint regions of $\mathbf{Z}^n$, such that no two such regions have neighboring points. Then zero padding can be added between regions to assure the $(1,\infty)$ constraint is not violated, provided each parsed string is mapped into a region without locally violating the $(1,\infty)$ constraint. This is described formally below.

Let V be a finite complete prefix code[1], and for each $v \in V$ let $S_v \subset \mathbf{Z}^n$. A n-dimensional $(1,\infty)$-constrained *word encoder* is an injection

$$\mathcal{E}^{(n)}_V : V \longrightarrow \bigcup_{S \subset \mathbf{Z}^n} \Lambda^{(n)}_{1,\infty}(S) \tag{4.2}$$

that maps the elements of V into labelings of subsets of $\mathbf{Z}^n$. Let $z \in \{0,1\}^\infty$ be an arbitrary infinite binary sequence that is parsed by the prefix code V as $z = z^{(1)}z^{(2)}\ldots$, where $z^{(i)} \in V$ for all i. If two points in $\mathbf{Z}^n$ are a distance 1 apart, then we call them

[1]The code V is a *prefix* code if no codeword is a prefix of any other codeword. *Complete* means that in the decoding tree, every node is either a leaf or has two children.

neighbors. For any set $S \subset \mathbf{Z}^n$, the *closure of* S is denoted by $\bar{S}$, and it contains the points that are either in S or have at least one neighboring point in S. The elements of $\{t_i \in \mathbf{Z}^n : i \in \mathbf{Z}^+\}$ are called *translation vectors* if for all i, the sets $t_i + S_{z^{(i)}}$ are disjoint and no points in different sets are neighbors. An n-dimensional $(1,\infty)$-constrained *composite encoder* $\mathcal{E}^{(n)}$ (with respect to V) is defined by:

$$\mathcal{E}^{(n)}(z)(u) = \begin{cases} \mathcal{E}_V^{(n)}(z^{(i)})(u - t_i) & \text{for all } i = 1, 2, \dots, \text{ if } u \in t_i + S_{z^{(i)}} \\ 0 & \text{if } \exists i \text{ s.t. } u \in t_i + \bar{S}_{z^{(i)}} \setminus S_{z^{(i)}} \end{cases} .$$

That is, $\mathcal{E}^{(n)}(z)$ is a labeling of translates of the sets $S_{z^{(1)}}, S_{z^{(2)}}, \dots$ composed of the labelings $\mathcal{E}_V^{(n)}(z^{(1)}), \mathcal{E}_V^{(n)}(z^{(2)}), \dots$. The labeling of points in $\mathbf{Z}^n$ outside of any $t_i + S_{z^{(i)}}$ by 0 is called *zero padding*. It is possible to choose the word encoder $\mathcal{E}_V^{(n)}$ and translation vectors $t_1, t_2, \dots$ such that the composite encoder is injective (i.e. is an encoder).

Define the following quantities for a word encoder:

$$\begin{aligned} \bar{r}\left(\mathcal{E}_V^{(n)}\right) &= \sum_{v \in V} \mathbf{P}(v) \cdot \frac{l(v)}{|S_v|} \\ \underline{r}\left(\mathcal{E}_V^{(n)}\right) &= \sum_{v \in V} \mathbf{P}(v) \cdot \frac{l(v)}{|\bar{S}_v|}. \end{aligned}$$

These upper and lower bound, respectively, the average ratio between the input length and the number of points in $\mathbf{Z}^n$ that are labeled for a particular prefix code V. The probability $\mathbf{P}(v)$ is taken with respect to the distribution of an unbiased random source.

If $l(v)$ is a constant for all $v \in V$, then if $|S_v|$ is a constant, $\mathcal{E}_V^{(2)}$ is a *fixed-to-fixed length* encoder, and if $|S_v|$ is not a constant then $\mathcal{E}_V^{(2)}$ is a *fixed-to-variable length* encoder. Similarly, if $l(v)$ is not a constant, then if $|S_v|$ is a constant, $\mathcal{E}_V^{(2)}$ is a *variable-to-fixed length* encoder, and if $|S_v|$ is not a constant then $\mathcal{E}_V^{(2)}$ is a *variable-to-variable length* encoder.

If $\{V_i\}$ is a sequence of prefix codes with increasing cardinality, then the *coding* *rate* of a composite encoder $\mathcal{E}^{(n)}$ (with respect to V_i) is

$$r\left(\mathcal{E}^{(n)}\right) = \lim_{i\to\infty} \bar{r}\left(\mathcal{E}_{V_i}^{(n)}\right) = \lim_{i\to\infty} \underline{r}\left(\mathcal{E}_{V_i}^{(n)}\right)$$

provided that the limits exist and are equal. It is known [18] that the coding rate is upper bounded by the capacity.

In this paper we discuss specific coding algorithms related to the concept of "bit stuffing" for which the following particular parameter choices apply: For $n = 2$, the code V_i is a prefix code with at most two codeword lengths; the sets S_v are parallelograms with one fixed side length (the other side length depends on v); the translation vectors are such that the parallelograms S_v lie next to each other in parallel rows, with zero padding between the rows. For $n = 3$, the code V_i is a prefix code with at most two codeword lengths; the sets S_v are parallelepipeds with two fixed side lengths (the third side length depends on v); the translation vectors are such that the parallelepipeds S_v are next to each other in parallel rows in three-dimensional space, with zero padding between the rows.

For $n = 2$, the bit stuffing technique of [16] and [19] prescribes how the word encoder $\mathcal{E}_V^{(n)}$ operates, that is, how a word from a prefix code is mapped to a parallelogram in $\mathbf{Z}^2$. The main idea is that a string is copied directly into a parallelogram bit by bit but skipping over 0s which were added whenever a 1 appeared previously in $z^{(i)}$. For the encoder defined in Section 4.4, and a given prefix code V, the length of one side of the parallelograms is fixed and the other side length is a function of the parsed word $z^{(i)}$ being processed. Then, as the prefix code grows in size, so does the fixed side length of the parallelograms.

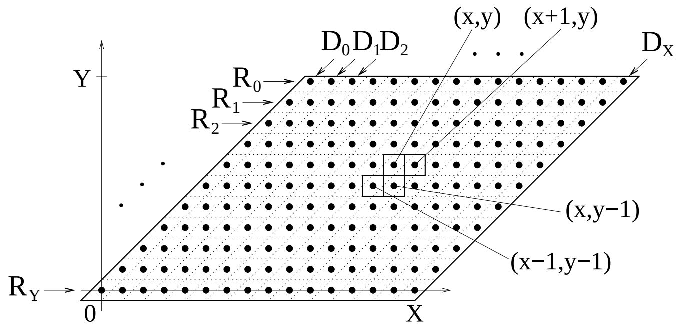


Figure 4.1: The parallelogram $\Delta_{X,Y}$ and its diagonals D_i and rows R_j.

4.3 Two-dimensional bit stuffing

A binary sequence is called a p-sequence if the bits are i.i.d. and if a 1 occurs with probability p. Throughout this paper let $\hat{w}$ be a $1/2$-sequence and $\hat{s}$ a p-sequence. For $X, Y \in \mathbf{Z}^+$, $i \in \{0, \ldots, X\}$, and $j \in \{0, \ldots, Y\}$ let

$$\begin{aligned} D_i &= \{(y+i, y) : 0 \le y \le Y\} \\ R_j &= \{(x+j, j) : 0 \le x \le X\} \\ \Delta_{X,Y} &= \bigcup_{i=0}^{X} D_i = \bigcup_{j=0}^{Y} R_j. \end{aligned}$$

The set $\Delta_{X,Y}$ is a parallelogram whose diagonals and rows are D_i and R_j, respectively, as shown in Figure 4.1. The set of points $D_0 \cup R_0$ is called the *boundary* of $\Delta_{X,Y}$.

One way to map a binary sequence w into a $(1, \infty)$-constrained labeling of $\Delta_{X,Y}$ is the following. The bits w_1, $w_2, \ldots$ are written into the diagonals of $\Delta_{X,Y}$ top to bottom, and left to right (i.e. along D_0, then $D_1, \ldots$, up to D_X). To ensure that the resulting labeling of $\Delta_{X,Y}$ is $(1, \infty)$-constrained, every time a 1 is written, extra 0s are written (said to be "stuffed") in the positions to the right and below the 1. These positions are skipped in the process of labeling the next diagonal. This procedure is continued until $\Delta_{X,Y}$ is filled up, i.e. until every element of $\Delta_{X,Y}$ is assigned a label.

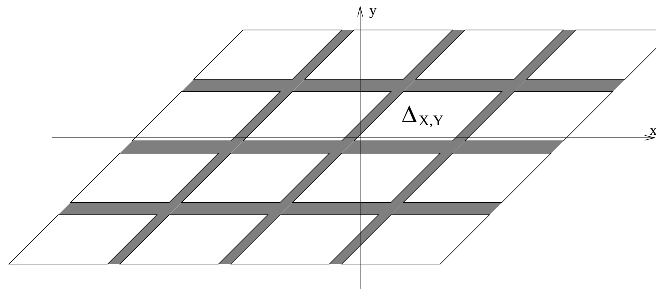


Figure 4.2: A method of mapping an infinite binary source into $\mathbf{Z}^2$ satisfying the $(1, \infty)$ constraint. The source is mapped into labelings of translates of $\Delta_{X,Y}$ separated by rows and diagonals of 0s. The shaded areas indicate the padding 0s.

Thus a finite number of input sequence bits are mapped into a $(1, \infty)$-constrained labeling of $\Delta_{X,Y}$. An arbitrary number of bits of the input can be encoded into $\Delta_{X,Y}$ by choosing X and Y large enough, or into a collection of translates of $\Delta_{X,Y}$ by using the same mapping on translates of $\Delta_{X,Y}$ with zero padding rows and diagonals between translates, as shown in Figure 4.2. Note that a bit sequence mapped into $\Delta_{X,Y}$ can never be a proper prefix of a different bit sequence mapped into $\Delta_{X,Y}$.

The bit stuffing method proposed by Siegel and Wolf [19] is based on the above encoding scheme with the following modifications. To increase the performance, the unbiased source $\hat{w}$ is transformed into a sequence $\hat{s}$, whose bits are independent, but whose 0s and 1s have unequal probabilities. The transformation increases the average length of a finite input sequence from $\hat{w}$, but the transformed bits of $\hat{s}$ more efficiently fit into $\Delta_{X,Y}$ if the bias is carefully chosen, since fewer 1s in $\hat{s}$ implies fewer stuffed 0s in $\Delta_{X,Y}$. To make the mathematical analysis of the algorithm simpler in [16] and [17], the boundary diagonal D_0 and boundary row R_0 are "initialized" with random labels independent of the sequence $\hat{s}$. Their initialization of the boundary points guarantees that for every $i \in \{0, \ldots, X\}$, the labels of D_i form a stationary Markov chain. The initialization degrades the performance of the algorithm, but the degradation is negligible as X and Y get large. Roth, Siegel, and Wolf [16] studied a certain two-dimensional

Table 4.1: Parameters used in Sections 4.3 and 4.4.

Parameter	Description
N	Positive integer parallelogram side length. Goes to ∞.
γ	Probability of unstuffed bit by $\tilde{\mathcal{E}}^{(2)}$. Function of p.
λ	Initial labeling of U_∞.
τ	Target number of translates of $\Delta_{N,N}$ in $\Delta_{X,N}$.
ϵ	Positive real typical sequence tolerance. Goes to 0.
p	Probability of 1 in transformed sequence. Optimized.
σ	Auxiliary binary string. p-sequence.
s	Input binary string.

bit stuffing algorithm, and computed that the expected coding rate is within 1% of the capacity $C_{1,\infty}^{(2)}$. The algorithm was later improved in [17] with an encoding rate within 0.1% of the capacity $C_{1,\infty}^{(2)}$.

A list of variables defined in Sections 4.3 and 4.4 and the parameters they depend on are given in Tables 4.1 and 4.2 as a reference.

4.3.1 A variable-to-fixed length bit stuffing encoder

Here we define a variable-to-fixed length encoder to label $\Delta_{N,N}$ and then use the encoder as a building block in a variable-to-variable length encoder to label larger portions of $\mathbf{Z}^2$. Then we take $N \to \infty$.

Define the following total ordering on the points of $\mathbf{Z}^2$:

$$(x_1, y_1) \prec (x_2, y_2) \iff \begin{cases} x_1 - y_1 < x_2 - y_2 \text{ or} \\ x_1 - y_1 = x_2 - y_2 \text{ and } x_1 > x_2 \end{cases}$$

for any $(x_1, y_1), (x_2, y_2) \in \mathbf{Z}^2$. That is $(x_1, y_1) \prec (x_2, y_2)$ if the diagonal that (x_1, y_1)

Table 4.2: Variables introduced in Sections 4.3 and 4.4 and the parameters they depend on.

Notation	Parameters	Description
$\tilde{\mathcal{E}}^{(2)}$	$N, \quad , \lambda$	Variable-to-fixed length encoder. Labels $\Delta_{N,N}$.
$\bar{\mathcal{E}}^{(2)}$	$N, \gamma, \lambda, \tau, \quad , \sigma$	Fixed-to-variable length encoder. Labels $\Delta_{X,N}$.
$Q^{(i)}$	$N, \gamma, \lambda, \tau, \quad , \sigma, s$	Number of bits $\bar{\mathcal{E}}^{(2)}$ maps into $\Delta^{(i)}_{N,N}$.
q	$N, \gamma, \lambda, \tau, \quad , \sigma$	Number of bits $\bar{\mathcal{E}}^{(2)}$ does not map into 1st τ translates of $\Delta_{N,N}$.
B	$N, \gamma, \lambda, \tau, \epsilon$	Set of strings that $\bar{\mathcal{E}}^{(2)}$ nearly maps into 1st τ translates of $\Delta_{N,N}$.
A	$N, \gamma, \quad , \tau, \epsilon, p$	Set of typical sequences.
T	$N, \gamma, \lambda, \tau, \epsilon, p$	Complete prefix code of size $\lvert A \cap B \rvert$.
t	$N, \gamma, \lambda, \tau, \epsilon, p$	Bijection from T to $A \cap B$.
$\mathcal{E}^{(2)}$	$N, \gamma, \lambda, \tau, \epsilon, p, \sigma$	Variable-to-variable length encoder. Labels $\Delta_{X,N}$.

lies on is above and to the left of the diagonal that (x_2, y_2) lies on, or if they lie on the same diagonal but with (x_1, y_1) above and to the right of (x_2, y_2).

For any $u = (u_1, u_2) \in \Delta_{N,N}$, let $u_l = (u_1 - 1, u_2)$ and $u_t = (u_1, u_2 + 1)$ be the left and top neighbors of u. Also let $u_b = (N-1, N)$ be the least upper bound of the points in $\Delta_{N,N}$ under the ordering $\prec$. Let $\lambda : R_0 \cup D_0 \longrightarrow \{0,1\}$ be an *initial labeling* of the boundary of $\Delta_{N,N}$. Then define a two-dimensional variable-to-fixed length $(1,\infty)$-constrained bit stuffing encoder $\tilde{\mathcal{E}}^{(2)}$ recursively, with input string $s \in \{0,1\}^*$, by

$$\begin{aligned}
\tilde{\mathcal{E}}^{(2)}(s)(u) &= \begin{cases} \lambda(u) & \text{if } u \in R_0 \cup D_0 \\ 0 & \text{if } \tilde{\mathcal{E}}^{(2)}(s)(u_l) = 1 \text{ or } \tilde{\mathcal{E}}^{(2)}(s)(u_t) = 1 \\ s_{\beta(u)} & \text{otherwise} \end{cases} \\
\beta(u) &= 1 + \left| \{ v \in \Delta_{N,N} \setminus (D_0 \cup R_0) : v \prec u, \tilde{\mathcal{E}}^{(2)}(s)(v_l) \neq 1, \tilde{\mathcal{E}}^{(2)}(s)(v_t) \neq 1 \} \right|.
\end{aligned}$$

The number $\beta(u)$ is one more than the number of previously nonstuffed bits in $\Delta_{N,N}$. Let

$$V_{N,\lambda} = \{ s \in \{0,1\}^* : l(s) = \beta(u_b) - 1 \} \tag{4.3}$$

be the set of all binary strings with length $\beta(u_b) - 1$, i.e. such a string perfectly fits into $\Delta_{N,N}$ under the bit stuffing mapping $\tilde{\mathcal{E}}^{(2)}$. Then $V_{N,\lambda}$ is a prefix code and the mapping

$$\tilde{\mathcal{E}}^{(2)} : V_{N,\lambda} \longrightarrow \Lambda^{(2)}_{1,\infty}(\Delta_{N,N})$$

is a word encoder, as defined in (4.2).

To encode a given binary input sequence, the encoder $\tilde{\mathcal{E}}^{(2)}$ first initializes the boundary of $\Delta_{N,N}$. Then it labels the points of $\Delta_{N,N}$ in increasing order with respect to the ordering $\prec$, such that every point of $\Delta_{N,N}$ is labeled either with a bit of the input sequence or with a "stuffed" 0 to ensure that the labeling is $(1,\infty)$-constrained. The

Table 4.3: The variable-to-fixed length $(1,\infty)$-constrained two-dimensional bit stuffing encoder $\tilde{\mathcal{E}}^{(2)}$. The algorithm maps a finite input string s from $V_{N,\lambda}$ into a $(1,\infty)$-constrained labeling of $\Delta_{N,N}$.

1.	Initialize the elements of $D_0 \cup R_0$ using λ. Let $i = 1$.
2.	Let $(u_1, u_2) = \min_{\prec}\{(v_1, v_2) \in \Delta_{N,N} : (v_1, v_2)$ is unlabeled$\}$.
3.	If $(u_1 - 1, u_2)$ is labeled with 1 or $(u_1, u_2 + 1)$ is labeled with 1
4.	Label (u_1, u_2) with 0.
5.	Else
6.	Label (u_1, u_2) with s_i. Let $i = i + 1$.
7.	If all of $\Delta_{N,N}$ is labeled then stop, else go to 2.

encoder $\tilde{\mathcal{E}}^{(2)}$ is invertible; the inverse mapping scans the diagonals D_1, $D_2, \ldots,$ D_N skipping over stuffed 0s to recover the input sequence. A pseudo-code description of the encoder is given in Table 4.3.

The encoder $\tilde{\mathcal{E}}^{(2)}$ is completely determined by the integer N and the initial labeling λ. If N is a fixed constant then $\tilde{\mathcal{E}}^{(2)}$ is a variable-to-fixed length encoder, as defined in [16] and [19] (they actually used a more general parallelogram $\Delta_{X,Y}$ instead of $\Delta_{N,N}$). In Section 4.4.1 we define a fixed-to-variable length encoder by letting a parameter X be a function of the input s. The set $\Delta_{X,N}$ is decomposed into multiple translates of $\Delta_{N,N}$, which allows $\Delta_{X,N}$ to grow large enough to accommodate certain long input strings. Then in Section 4.4.2 we use the fixed-to-variable length encoder to define a variable-to-variable length encoder. The variable-to-variable length encoder is “nearly” a fixed-to-fixed length encoder, which allows precise mathematical analysis of its coding rate.

4.4 A two-dimensional variable-to-variable length encoder

Using a finite complete prefix code defined in Section 4.4.2, a sequence w is parsed into finite variable length strings $w^{(1)}, w^{(2)}, \ldots$. Each string $w^{(i)}$ in the prefix code is mapped into a $(1,\infty)$-constrained labeling of the set $\Delta_{X(w^{(i)}),N}$, where $X(w^{(i)})$ is a positive integer chosen so that the mapped prefix code fits into $\Delta_{X(w^{(i)}),N}$ using bit stuffing. The infinite sequence of finite length strings $w^{(1)}, w^{(2)}, \ldots$ is mapped into labelings of translates of the parallelograms $\Delta_{X(w^{(1)}),N}, \Delta_{X(w^{(2)}),N}, \ldots$ that tile a quadrant of $\mathbf{Z}^2$. The translates are separated by one diagonal and one row of zero padding (see Figure 4.3). The tiling can be generalized to all of $\mathbf{Z}^2$ by alternately placing the parallelograms in

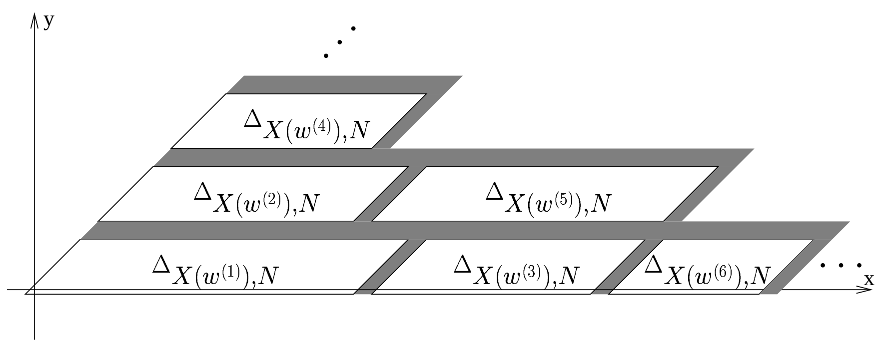


Figure 4.3: Translates of the parallelograms $\Delta_{X(w^{(1)}),N}, \ldots, \Delta_{X(w^{(6)}),N}, \ldots$ are used to encode the words $w^{(1)}, \ldots, w^{(6)}, \ldots$, respectively. The shaded areas indicate the padding 0s.

the four quadrants. Henceforth we abbreviate $X(w^{(i)})$ with X.

4.4.1 An intermediate fixed-to-variable length encoder

For each $i \geq 0$ define the following translations of $\Delta_{N,N}$, its boundary diagonal D_0, boundary row R_0, and an arbitrary $u \in \mathbf{Z}^2$:

$$\begin{aligned}
\Delta_{N,N}^{(i)} &= \Delta_{N,N} + i(N+2,0) \\
D_0^{(i)} &= D_0 + i(N+2,0) \\
R_0^{(i)} &= R_0 + i(N+2,0) \\
u^{(i)} &= u - i(N+2,0).
\end{aligned}$$

Let γ be a positive real number. Let $\tau \in \mathbf{Z}^+$, called the *target number of translates*, and let $s \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor}$ be an input string. For each $j \geq 1$ let

$$U_j = \bigcup_{i=0}^{j-1} (D_0^{(i)} \cup R_0^{(i)})$$

be a union of boundaries of translates of $\Delta_{N,N}$, and let $\lambda : U_\infty \longrightarrow \{0,1\}$ be an *initial labeling* of U_∞ satisfying $\lambda(u) = 0$ for all $u \notin U_\tau$. For each $i \geq 0$ let $\lambda_i : D_0^{(i)} \cup R_0^{(i)} \longrightarrow \{0,1\}$ be the restriction of λ to the set $D_0^{(i)} \cup R_0^{(i)}$; that is $\lambda_i(u) = \lambda(u)$ for all $u \in D_0^{(i)} \cup R_0^{(i)}$. Let σ be an infinite binary string called an *auxiliary sequence*.

For each i, the labeling λ_i induces a prefix code V_{N,λ_i} based on labeling $\Delta_{N,N}^{(i)}$, as in (4.3). The sequence of prefix codes $V_{N,\lambda_0}, V_{N,\lambda_1}, \ldots$ induces a partition of the concatenation of the input and auxiliary strings as

$$s\sigma = z^{(0)} z^{(1)} \ldots$$

Let

$$\tau_a = \max\left\{\tau, \min\{i : s \text{ is a prefix of } z^{(0)} \ldots z^{(i)}\}\right\}.$$

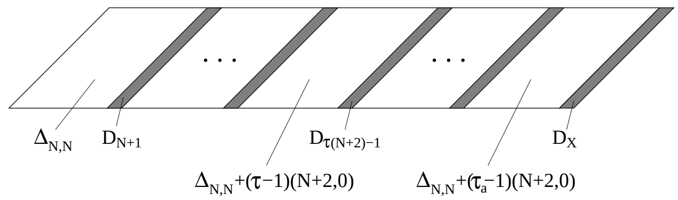


Figure 4.4: The set $\Delta_{X,N}$ consists of τ_a translates of $\Delta_{N,N}$, and a diagonal of zero padding after each translate.

be the number of translates of $\Delta_{N,N}$ used to perform the labeling. That is, s will be encoded into a labeling of $\Delta^{(0)}_{N,N}, \ldots, \Delta^{(\tau_a-1)}_{N,N}$. Later, we will force τ_a to be close to the target τ, with high probability. Let

$$\begin{aligned} X &= \tau_a(N+2)-1 \\ \Delta_{X,N} &= \left(\bigcup_{i=0}^{\tau_a-1} \Delta^{(i)}_{N,N}\right) \cup \left(\bigcup_{j=1}^{\tau_a} D_{j(N+2)-1}\right). \end{aligned}$$

The parallelogram $\Delta_{X,N}$ is decomposed into τ_a translates $\Delta^{(i)}_{N,N}$, to be filled with information bits and stuffed 0s, and diagonals $D_{j(N+2)-1}$, to be filled with zero padding (see Figure 4.4).

This is formalized by defining a fixed-to-variable length encoder

$$\bar{\mathcal{E}}^{(2)} : \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor} \longrightarrow \bigcup_{S\subset \mathbf{Z}^2} \Lambda^{(2)}_{1,\infty}(S)$$

by specifying the labeling $\bar{\mathcal{E}}^{(2)}(s) : \Delta_{X,N} \longrightarrow \{0,1\}$ as:

$$\bar{\mathcal{E}}^{(2)}(s)(u) = \begin{cases} \tilde{\mathcal{E}}^{(2)}(z^{(i)})(u^{(i)}) & \text{if } u \in \Delta^{(i)}_{N,N} \text{ for any } i = 0, \ldots, \tau_a - 1 \\ 0 & \text{if } u \in D_{j(N+2)-1} \text{ for any } j = 1, \ldots, \tau_a \end{cases}$$

where $\tilde{\mathcal{E}}^{(2)}$ is the encoder defined in Section 4.3.1. The encoder $\bar{\mathcal{E}}^{(2)}$ is completely determined by N, τ, σ, γ, λ and will serve as the second stage of a variable-to-variable length $(1,\infty)$-constrained bit stuffing encoder to be defined in Section 4.4.2. Note that s can be recovered from the labeling of $\Delta_{X,N}$.

The process of encoding the input string s is described in detail below. The points of U_τ are assigned a fixed initial labeling using λ. The translates $\Delta^{(0)}_{N,N}, \ldots, \Delta^{(\tau-1)}_{N,N}$ are labeled with the bits of s using the variable-to-fixed length encoder $\tilde{\mathcal{E}}^{(2)}$ and the fixed initial labelings $\lambda_0, \ldots \lambda_{\tau-1}$, respectively. The inter-translate diagonals $D_{j(N+2)-1}$, for $j = 1, 2, \ldots, \tau$, are filled with 0s.

Labeling all τ translates of $\Delta_{N,N}$ using the encoder $\tilde{\mathcal{E}}^{(2)}$ (with the initialization λ_i on the ith translate) and adding the padding diagonals after each translate, defines a labeling of the set $\Delta_{\tau(N+2)-1,N}$. Each $\tilde{\mathcal{E}}^{(2)}$ is a variable-to-fixed length encoder, so it is possible that to encode exactly $\lfloor \tau\gamma N^2 \rfloor$ input bits might require either more or less space in $\mathbf{Z}^2$ than just the set $\Delta_{\tau(N+2)-1,N}$. If s is too short to label all of $\Delta_{\tau(N+2)-1,N}$, then $\bar{\mathcal{E}}^{(2)}$ uses the auxiliary sequence σ as input to finish labeling $\Delta_{\tau(N+2)-1,N}$, and if s is too long to label $\Delta_{\tau(N+2)-1,N}$, then $\bar{\mathcal{E}}^{(2)}$ continues the encoding process and maps the remaining bits of s into the additional translates $\Delta^{(\tau)}_{N,N}, \ldots, \Delta^{(\tau_a-1)}_{N,N}$, using the auxiliary sequence σ to finish filling the last translate $\Delta^{(\tau_a-1)}_{N,N}$, and using the all zero initialization on the boundary elements of the additional translates. The inter-translate diagonals $D_{(\tau+1)(N+2)-1}, \ldots, D_{\tau_a(N+2)-1}$ are filled with padding 0s. In Section 4.4.3 we will choose τ and γ to guarantee that s fills up $\Delta_{\tau(N+2)-1,N}$ almost perfectly with high probability, and therefore the number of additional translates will typically be small.

For a given τ, γ, N, binary input string $s \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor}$, auxiliary binary sequence σ, and initial labeling λ of U_∞, and for each $i = 0, 1, \ldots, \tau - 1$ let

$$
\begin{aligned}
Q^{(i)} &= \text{number of bits of } s\sigma \text{ that } \bar{\mathcal{E}}^{(2)} \text{ maps into } \Delta^{(i)}_{N,N} \\
q(s) &= \lfloor \tau\gamma N^2 \rfloor - \sum_{i=0}^{\tau-1} Q^{(i)}.
\end{aligned}
$$

If positive, $q(s)$ is the number of bits of s that do not get mapped into $\Delta_{\tau(N+2)-1,N}$, and otherwise $q(s)$ is minus the number of bits of σ that get mapped into $\Delta_{\tau(N+2)-1,N}$. For any $\epsilon, \gamma > 0$ and $\lambda : U_\infty \longrightarrow \{0,1\}$, let

$$B = \left\{ s \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor} : q(s) < \tau\epsilon \right\}. \tag{4.4}$$

The set B represents the strings that "fit well" into $\Delta_{\tau(N+2)-1,N}$, i.e. for every $s \in B$ the fraction of bits of s that are not mapped into $\Delta_{\tau(N+2)-1,N}$ is smaller than about $\frac{\epsilon}{\gamma N^2}$. Note that even though B is a function of the encoder $\bar{\mathcal{E}}^{(2)}$ by way of $q(s)$, B is in fact independent of the auxiliary sequence σ, since $s \in B$ whenever $q(s) < 0$. The set B is determined by N, γ, λ, τ, ϵ.

4.4.2 Restriction to typical sequences

For any $\epsilon > 0$, the *typical set* A of blocklength $\lfloor \tau\gamma N^2 \rfloor$ with respect to p is defined as [4, p. 51]:

$$\begin{aligned} A \quad = \quad & \left\{ s \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor} : \right. \\ & \left. 2^{-\lfloor \tau\gamma N^2 \rfloor (H(p)+\epsilon)} \leq p^{|s|}(1-p)^{\lfloor \tau\gamma N^2 \rfloor - |s|} \leq 2^{-\lfloor \tau\gamma N^2 \rfloor (H(p)-\epsilon)} \right\}. \end{aligned}$$

The term $p^{|s|}(1-p)^{\lfloor \tau\gamma N^2 \rfloor - |s|}$ is the probability of a length $\lfloor \tau\gamma N^2 \rfloor$ p-sequence $\hat{s}$ being equal to s. The set A is determined by N, γ, τ, ϵ, p, and an element of A is called an *ϵ-typical sequence.*

Let T be a complete prefix code of cardinality $|A \cap B|$, whose codewords are one of two possible lengths[2], and let

$$t : T \longrightarrow A \cap B$$

[2]For any $i \geq 2$ there exists a complete prefix code with i codewords, all of length $\lfloor \log_2 i \rfloor$ or $\lfloor \log_2 i \rfloor + 1$.

be any bijection. Both t and T are determined by the parameters N, γ, λ, τ, ϵ, p.

The code T parses an infinite input sequence and t maps a finite parsed string to an ϵ-typical (with respect to p) sequence s that is likely to fit into the first τ translates of $\Delta_{N,N}$. Since T is a complete prefix code, a binary sequence w can uniquely be parsed into strings $w^{(1)}, w^{(2)}, \ldots \in T$. A *variable-to-variable length* $(1,\infty)$*-constrained bit stuffing encoder* $\mathcal{E}^{(2)}$ is defined as the composition

$$\mathcal{E}^{(2)} = \bar{\mathcal{E}}^{(2)} \circ t.$$

The encoder $\mathcal{E}^{(2)}$ is completely determined by the parameters N, γ, λ, τ, ϵ, σ, p.

That is, each string $w^{(i)} \in T$ of the parsed sequence w is transformed into the typical, well-fitting string $s^{(i)} \in A \cap B$ by the bijection t, and then $s^{(i)}$ is mapped into a $(1,\infty)$-constrained labeling of $\Delta_{X,N}$ using the encoder $\bar{\mathcal{E}}^{(2)}$. The transformation t approximates transforming an infinite $1/2$-sequence into a p-sequence with an arithmetic decoder. The *variable-to-variable length two-dimensional* $(1,\infty)$*-constrained bit stuffing algorithm* consists of the mapping $\mathcal{E}^{(2)}$ and its inverse. The mapping $\mathcal{E}^{(2)}$ is referred to as the algorithm's encoder, and the inverse is called the algorithm's decoder (see Figure 4.5).

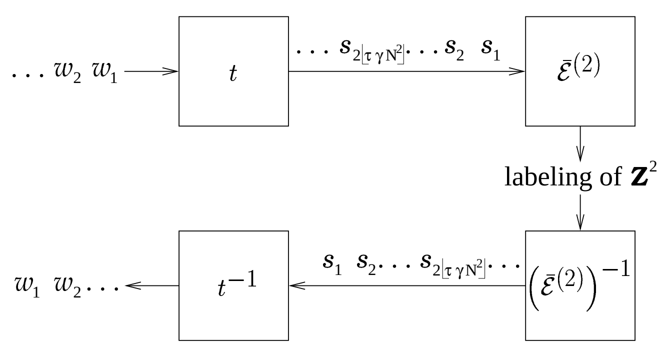


Figure 4.5: A two-dimensional $(1,\infty)$-constrained bit stuffing algorithm. The input bits $w_1, w_2, \ldots$ are mapped into the sequence $s_1, s_2, \ldots,$ which is encoded into a labeling of $\mathbf{Z}^2$ by $\bar{\mathcal{E}}^{(2)}$.

Note that it is guaranteed by the encoder $\mathcal{E}^{(2)}$ that the last diagonal of $\Delta_{X,N}$ is

(a): $\hat{\mu}^{(1)}$ (b): $\hat{\mu}^{(2)}$

Figure 4.6: The homogeneous Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ generating the labels of: (a) a boundary diagonal; and (b) a boundary row.

filled with 0s. An additional row R_{N+1} of padding 0s is added to $\Delta_{X,N}$ to ensure that a tiling by the parallelograms defines a valid labeling of $\mathbf{Z}^2$.

4.4.3 Coding rate analysis

Consider the encoder $\tilde{\mathcal{E}}^{(2)}$ with the boundary elements D_0 and R_0 assigned random initial labels independently of the p-sequence $\hat{s}$ by the stationary homogeneous Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$, respectively (see Figures 4.6a and 4.6b).

The transition probabilities π_1, π_2, π_3 are constrained such that the stationary distribution of the Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ are the same; thus the labeling of D_0 fixes the label of the origin, which is used to initiate the labeling of R_0. It follows from the results in [16] and [17] that for any p the parameters π_1, π_2, π_3 can be chosen to guarantee that the labels of each diagonal D_i ($1 \leq i \leq N$) form a stationary homogeneous Markov chain identical to the Markov chain $\hat{\mu}^{(1)}$ labeling D_0 (see Theorem 4.4). If π_1, π_2, π_3 are chosen such that the labels of the D_i's form identical Markov chains, then the initialization is called a *standard initialization corresponding to* p and the resulting labeling of $\Delta_{N,N}$ is a *standard labeling corresponding to* p.

Let

$$\gamma = \mathbf{P}\left((x, y+1) \text{ and } (x-1, y) \text{ both labeled with 0 by } \tilde{\mathcal{E}}^{(2)}\right).$$

For a standard initialization $\hat{\lambda}$, the probability γ depends only on the probability p (i.e. it is independent of N) and is the probability that any position in $\Delta_{N,N}$ is unstuffed by $\tilde{\mathcal{E}}^{(2)}$.

Henceforth we assume that the auxiliary sequence $\hat{\sigma}$ is a p-sequence. An initial labeling λ of U_∞ used by $\mathcal{E}^{(2)}$ is implied in the following lemma.

Lemma 4.1. *For the p-sequence $\hat{s} \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor}$, and any $N \in \mathbf{Z}^+$, and $\epsilon, \gamma > 0$, there exists $\tau_0 \in \mathbf{Z}^+$ such that for every $\tau \geq \tau_0$ there is an initial labeling $\lambda : U_\infty \longrightarrow \{0,1\}$ such that*

$$\mathbf{P}\left(\hat{s} \in B\right) > 1 - \epsilon. \tag{4.5}$$

Proof. Suppose that for some $\tau \in \mathbf{Z}^+$ the p-sequence $\hat{s}$ is encoded into $\Delta_{X,N}$ by $\bar{\mathcal{E}}^{(2)}$, with a random initial labeling $\hat{\lambda}$ of U_∞ that assigns labels using the Markov chain $\hat{\mu}^{(1)}$ for the translates $D_0^{(i)}$ $(0 \leq i < \tau)$, the Markov chain $\hat{\mu}^{(2)}$ for the translates $R_0^{(i)}$ $(0 \leq i < \tau)$, and initializes $D_0^{(i)}$ and $R_0^{(i)}$ with 0s for $i \geq \tau$. By using a random auxiliary sequence $\hat{\sigma}$, we will demonstrate that there is at least one initial labeling $\lambda : U_\infty \longrightarrow \{0,1\}$ such that for any auxiliary sequence σ, (4.5) holds.

Since each $\Delta_{N,N}^{(i)}$ is initialized by $\hat{\lambda}_i$, the labeling of each $\Delta_{N,N}^{(i)}$ is a standard labeling, and thus by the definition of γ, we have

$$E\left[\hat{Q}^{(i)}\right] = \gamma N^2$$

for every $i \in \{0,1,\ldots,\tau-1\}$. For any two distinct $i \in \{0,1,\ldots,\tau-1\}$, the random variables $\hat{Q}^{(i)}$ are independent and have finite variances (independent of τ). Therefore the weak law of large numbers implies that for every $\epsilon > 0$,

$$\lim_{\tau\to\infty} \mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1}\hat{Q}^{(i)} - \gamma N^2\right| < \epsilon\right) = 1. \tag{4.6}$$

The random variables $\hat{Q}^{(i)}$ in (4.6) are functions of the random input p-sequence $\hat{s}$, the random auxiliary sequence $\hat{\sigma}$, and the random initialization $\hat{\lambda}$ of U_τ. Then (4.6) and the inequalities

$$\begin{aligned}
\mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1}\hat{Q}^{(i)} - \gamma N^2\right| < \epsilon\right) &\leq \mathbf{P}\left(\tau\gamma N^2 - \sum_{i=0}^{\tau-1}\hat{Q}^{(i)} < \tau\epsilon\right) \\
&\leq \mathbf{P}\left(\lfloor \tau\gamma N^2 \rfloor - \sum_{i=0}^{\tau-1}\hat{Q}^{(i)} < \tau\epsilon\right) \\
&= \mathbf{P}\left(q(\hat{s}) < \tau\epsilon\right)
\end{aligned}$$

imply that

$$\lim_{\tau\to\infty} \mathbf{P}\left(q(\hat{s}) < \tau\epsilon\right) = 1. \tag{4.7}$$

It follows from (4.7) that there exists a τ_0 such that for all $\tau \geq \tau_0$,

$$\mathbf{P}\left(q(\hat{s}) < \tau\epsilon\right) > 1 - \epsilon.$$

Thus there must exist at least one initial labeling λ (depending on τ) such that

$$\mathbf{P}\left(q(\hat{s}) < \tau\epsilon \middle| \lambda\right) > 1 - \epsilon \tag{4.8}$$

where the conditioning in (4.8) is on the event that the random labeling $\hat{\lambda}$ equals the fixed labeling λ.

Equivalently, for every $\tau \geq \tau_0$,

$$\mathbf{P}\left(\hat{s} \in B\right) > 1 - \epsilon.$$

$\square$

A number $r^{(2)}$ is said to be an *achievable coding rate* of a two-dimensional $(1,\infty)$-constrained bit stuffing algorithm $\mathcal{E}^{(2)}$ if

$$r^{(2)} = \lim_{\epsilon \to 0} \lim_{N \to \infty} r\left(\mathcal{E}^{(2)}\right).$$

Theorem 4.2. *The two-dimensional bit stuffing algorithm achieves a coding rate of*

$$r^{(2)} = \gamma H(p).$$

Proof. For $N \in \mathbf{Z}^+$ and $\epsilon > 0$, let τ_0 be defined as in Lemma 4.1. It is known [4, pp. 51-52] that $\tau \geq \tau_0$ can be chosen large enough such that the p-sequence $\hat{s} \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor}$ satisfies

$$\mathbf{P}\left(\hat{s} \in A\right) > 1 - \epsilon.$$

Therefore, using Lemma 4.1,

$$\begin{aligned} 1 - 2\epsilon &< \mathbf{P}\left(\hat{s} \in A \cap B\right) \\ &\leq \sum_{s \in A \cap B} 2^{-\lfloor \tau\gamma N^2 \rfloor (H(p) - \epsilon)} \\ &= |A \cap B| \cdot 2^{-\lfloor \tau\gamma N^2 \rfloor (H(p) - \epsilon)} \end{aligned}$$

which implies

$$|A \cap B| \geq (1 - 2\epsilon) \cdot 2^{\lfloor \tau\gamma N^2 \rfloor (H(p) - \epsilon)}. \tag{4.9}$$

Similarly,

$$\begin{aligned} 1 &\geq \mathbf{P}\left(\hat{s} \in A \cap B\right) \\ &\geq \sum_{s \in A \cap B} 2^{-\left\lfloor \tau\gamma N^2 \right\rfloor (H(p)+\epsilon)} \\ &= |A \cap B| \cdot 2^{\left\lfloor \tau\gamma N^2 \right\rfloor (H(p)+\epsilon)} \end{aligned}$$

which implies

$$|A \cap B| \leq 2^{\left\lfloor \tau\gamma N^2 \right\rfloor (H(p)+\epsilon)}. \tag{4.10}$$

Any string $z \in T$ has length either $\lfloor \log_2 |A \cap B| \rfloor$ or $\lfloor \log_2 |A \cap B| \rfloor + 1$. Note that $\Delta_{X,N}$ together with one row of zero padding occupies $(X+1)(N+2)$ points in $\mathbf{Z}^2$. Therefore (4.9), the definition of the set B, and $\sum_{z \in T} \mathbf{P}(z) = 1$ imply that the coding rate is lower bounded as

$$\begin{aligned} r\left(\mathcal{E}^{(2)}\right) &= \sum_{z \in T} \mathbf{P}(z) \frac{l(z)}{(X+1)(N+2)} \\ &\geq \frac{\lfloor \log_2 |A \cap B| \rfloor}{\tau(N+2)^2 + \left\lceil \frac{3\tau\epsilon}{N^2} \right\rceil (N+2)^2} \sum_{z \in T} \mathbf{P}(z) \\ &\geq \frac{\log_2(1-2\epsilon) + \lfloor \tau\gamma N^2 \rfloor (H(p)-\epsilon) - 1}{\tau(N+2)^2 + \left\lceil \frac{3\tau\epsilon}{N^2} \right\rceil (N+2)^2}. \end{aligned} \tag{4.11}$$

The term $\left\lceil \frac{3\tau\epsilon}{N^2} \right\rceil$ in the denominator is an upper bound on the number of additional translates of $\Delta_{N,N}$ needed after the first τ translates, since every input bit is mapped into at most three bits by $\bar{\mathcal{E}}^{(2)}$, and each translate can be labeled by at most N^2 bits. Using (4.10) and the fact that $X \geq \tau(N+2) - 1$ for any input word $z \in T$, the coding rate is upper bounded as

$$\begin{aligned} r\left(\mathcal{E}^{(2)}\right) &= E\left[\frac{l(z)}{(X+1)(N+2)}\right] \\ &\le \frac{\lceil \log_2 |A \cap B| \rceil}{\tau(N+2)^2} \\ &\le \frac{\tau\gamma N^2 (H(p)+\epsilon)+1}{\tau(N+2)^2}. \end{aligned} \tag{4.12}$$

Taking limits as $N \to \infty$ and $\epsilon \to 0$, the theorem follows from (4.11) and (4.12). $\square$

For the parameters N, p, ϵ, the value of τ needed to make $r\left(\mathcal{E}^{(2)}\right)$ as close to $\gamma H(p)$ as desired is implied in Lemma 4.1 and Theorem 4.2. For a fixed value of τ, the existence of the initial labeling λ used by the encoder $\mathcal{E}^{(2)}$ is given in Lemma 4.1. The parameter γ is a function of p (see (4.20)).

Note that if p is close to 0, then the p-sequence $\hat{s}$ contains fewer 1s, and fewer stuffed 0s are forced into the labeling of $\Delta_{X,N}$ making the encoder more efficient (i.e. increasing γ). This in turn makes X smaller. However, small values of p have the disadvantage of decreasing $H(p)$. Thus to maximize the coding rate $r^{(2)} = \gamma H(p)$, there is a tradeoff between increasing and decreasing p in the range $[0, \frac{1}{2}]$. (The maximum of $r^{(2)}$ is attained for p in the range $[0, \frac{1}{2}]$, since both γ and $H(p)$ decrease as p goes above $1/2$.)

4.4.4 Coding rate maximization

Many of the results in this paper are based on the following property of Markov chains, whose proof is included here for completeness.

Lemma 4.3. *Let $\hat{u}_0, \hat{u}_1, \ldots$ be a Markov chain that takes on values from a set A. Let $\hat{u}^*$ be a random variable that takes on values from the set A^*, and for some $n \in \mathbf{Z}^+$, let $\hat{u}^*$ be conditionally independent of $\hat{u}_n$ if $\hat{u}_{n-1}, \ldots, \hat{u}_0$ are given. Then $\hat{u}^*$ is conditionally* *independent of* $\hat{u}_n$ *if* $\hat{u}_{n-1}$ *is given.*

Proof. Let $a^* \in A^*$.

$$\begin{aligned}
&\mathbf{P}(\hat{u}^* = a^* | \hat{u}_n, \hat{u}_{n-1}) \\
&= \sum_{a_0,\ldots,a_{n-2} \in A} \mathbf{P}(\hat{u}^* = a^* | \hat{u}_n, \hat{u}_{n-1}, \hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0) \\
&\qquad \cdot \mathbf{P}(\hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0 | \hat{u}_n, \hat{u}_{n-1})
\end{aligned}$$

where the summation is taken over all values $a_0, \ldots, a_{n-2}$ such that the event we condition on has positive probability. By the assumption of the lemma,

$$\begin{aligned}
&\mathbf{P}(\hat{u}^* = a^* | \hat{u}_n, \hat{u}_{n-1}, \hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0) \\
&= \mathbf{P}(\hat{u}^* = a^* | \hat{u}_{n-1}, \hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0).
\end{aligned}$$

Furthermore, since the reverse sequence $\{\hat{u}_i\}_{i=n}^{0}$ is a Markov chain, it follows that

$$\mathbf{P}(\hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0 | \hat{u}_n, \hat{u}_{n-1}) = \mathbf{P}(\hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0 | \hat{u}_{n-1}).$$

Hence,

$$\begin{aligned}
&\mathbf{P}(\hat{u}^* = a^* | \hat{u}_n, \hat{u}_{n-1}) \\
&= \sum_{a_0,\ldots,a_{n-2} \in A} \mathbf{P}(\hat{u}^* = a^* | \hat{u}_{n-1}, \hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0) \\
&\qquad \cdot \mathbf{P}(\hat{u}_{n-2} = a_{n-2}, \ldots, \hat{u}_0 = a_0 | \hat{u}_{n-1}) \\
&= \mathbf{P}(\hat{u}^* = a^* | \hat{u}_{n-1}).
\end{aligned}$$

□

Let the p-sequence $\hat{s}$ be encoded into a labeling of $\Delta_{N,N}$ using the encoder $\tilde{\mathcal{E}}^{(2)}$ with a random initial labeling $\hat{\lambda}$ assigned to the boundary elements $D_0 \cup R_0$ as defined in Section 4.4.3. Let $\hat{F}(v)$ denote the random label assigned to the point $v \in \mathbf{Z}^2$. To simplify the notation, we will use $\hat{F}(v_1, \ldots, v_j)$ to denote the joint random variables $(\hat{F}(v_1), \ldots, \hat{F}(v_j))$ for any integer j and for $v_1, \ldots, v_j \in \Delta_{N,N}$. Necessary and sufficient conditions for the labels on each diagonal $D_1, \ldots, D_X$ to form a Markov chain identical to the labels of D_0 are given in Theorem 4.4 below. In the theorem, the random initial labeling assigned to the boundary elements $D_0 \cup R_0$ is that defined in Section 4.4.3, and the parameters x and y are given in Figure 4.1.

Theorem 4.4. *Let the p-sequence $\hat{s}$ be encoded into a labeling of $\Delta_{N,N}$ using the encoder $\tilde{\mathcal{E}}^{(2)}$ with the random initial labeling assigned to the boundary elements $D_0 \cup R_0$. The following statements are equivalent.*

1. *The labels assigned to the elements of D_i, for $i \in \{1, \ldots, N\}$, form a stationary homogeneous Markov chain identical to the labels of D_0 (i.e. the labeling of $\Delta_{N,N}$ is a standard labeling).*

2. *The labels assigned to the elements of R_j, for $j \in \{1, \ldots, N\}$, form a stationary homogeneous Markov chain identical to the labels of R_0.*

3. *The transition probabilities of the Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ satisfy*

$$\pi_1 = \frac{2}{1+p+\sqrt{(1+3p)(1-p)}}, \quad \pi_2 = \frac{2p}{1+p+\sqrt{(1+3p)(1-p)}}, \quad \pi_3 = \frac{2(1-p)}{1-p+\sqrt{(1+3p)(1-p)}}.$$

4. *The joint distribution of the random variables $\hat{F}((x,y))$, $\hat{F}((x+1,y))$, $\hat{F}((x-1,y-1))$, $\hat{F}((x,y-1))$, for $x, y \in \{0, \ldots, N-1\}$ is independent of the choice of x and y.*

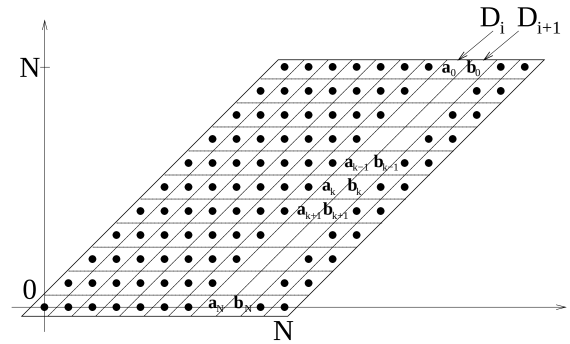


Figure 4.7: The diagonals D_i and D_{i+1} of $\Delta_{N,N}$.

Proof. The equivalence of Conditions 1, 2, and 3 follows from the results in [16][3] and [17]. We show that Conditions 1, 2, and 3 are together equivalent to Condition 4. For a fixed $i \in \{0, \ldots, N-1\}$, consider the points $a_j = (i+j, j)$ and $b_j = (i+j+1, j)$, for $0 \leq j \leq N$, of the diagonals D_i and D_{i+1} (see Figure 4.7). Let v_1, v_2, v_3, v_4 be a valid labeling of the points a_{k-1}, b_{k-1}, a_k, b_k for $0 < k \leq N$. Then

$$
\begin{aligned}
&\mathbf{P}\big(\hat{F}(a_{k-1}, b_{k-1}, a_k, b_k) = (v_1, v_2, v_3, v_4)\big) \\
&= \mathbf{P}\big(\hat{F}(a_k) = v_3\big) \cdot \mathbf{P}\big(\hat{F}(a_{k-1}, b_k) = (v_1, v_4)\big|\hat{F}(a_k) = v_3\big) \\
&\quad \cdot \mathbf{P}\big(\hat{F}(b_{k-1}) = v_2\big|\hat{F}(a_{k-1}, a_k, b_k) = (v_1, v_3, v_4)\big) \\
&= \mathbf{P}\big(\hat{F}(a_k) = v_3\big) \cdot \mathbf{P}\big(\hat{F}(a_{k-1}, b_k) = (v_1, v_4)\big|\hat{F}(a_k) = v_3\big) \\
&\quad \cdot \mathbf{P}\big(\hat{F}(b_{k-1}) = v_2\big|\hat{F}(a_{k-1}, a_k) = (v_1, v_3)\big) \qquad (4.13) \\
&= \mathbf{P}\big(\hat{F}(a_k) = v_3\big) \cdot \mathbf{P}\big(\hat{F}(a_{k-1}) = v_1\big|\hat{F}(a_k) = v_3\big) \cdot \mathbf{P}\big(\hat{F}(b_k) = v_4\big|\hat{F}(a_k) = v_3\big) \\
&\quad \cdot \mathbf{P}\big(\hat{F}(b_{k-1}) = v_2\big|\hat{F}(a_{k-1}, a_k) = (v_1, v_3)\big) \qquad (4.14)
\end{aligned}
$$

where (4.13) follows from the definition of the bit stuffing encoder; and (4.14) follows from Lemma 4.3 with $\hat{u}_j = \hat{F}(a_j)$ and $\hat{u}^* = \hat{F}(b_k)$, since the labels $\big\{\hat{F}(a_j)\big\}_{j=N}^{0}$ form a Markov chain, and $\hat{F}(b_k)$ is independent of $\hat{F}(a_{k-1})$ if $\hat{F}(a_k), \hat{F}(a_{k+1}), \ldots, \hat{F}(a_0)$ are

[3]Equation (24) in [16] is incorrect. It should read $z = \frac{(4-3q)-\sqrt{(4-3q)^2-4(1-q)(4-3q)}}{2(1-q)(4-3q)}$.

given. Conditions 1 and 2 imply that the first three terms of (4.14) are independent of the diagonal i and the position k, and the last term of (4.14) is independent of i and k by the definition of the bit stuffing encoder. Hence, Conditions 1 and 2 together imply Condition 4.

To prove the converse, first note that $D_0 \cap R_0 = \{(0,0)\}$, and therefore the Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ must have the same stationary probabilities. Let α denote the stationary probability of the state 0. Then,

$$\alpha = \alpha\pi_1 + (1-\alpha)(1-\pi_2) \tag{4.15}$$

$$\alpha = \alpha\pi_3 + (1-\alpha). \tag{4.16}$$

Furthermore, by Condition 4 we have $\mathbf{P}\big(\hat{F}((0,0)) = \hat{F}((-1,-1)) = 0\big)$ $= \mathbf{P}\big(\hat{F}((1,0)) = \hat{F}((0,-1)) = 0\big)$ which implies

$$\begin{aligned}
\alpha\pi_1 &= \mathbf{P}\left(\hat{F}((0,0)) = 0, \hat{F}((-1,-1)) = 0, \hat{F}((1,0)) = 0, \hat{F}((0,-1)) = 0\right) \\
&\quad + \mathbf{P}\left(\hat{F}((0,0)) = 0, \hat{F}((-1,-1)) = 1, \hat{F}((1,0)) = 0, \hat{F}((0,-1)) = 0\right) \\
&\quad + \mathbf{P}\left(\hat{F}((0,0)) = 1, \hat{F}((-1,-1)) = 0, \hat{F}((1,0)) = 0, \hat{F}((0,-1)) = 0\right) \\
&\quad + \mathbf{P}\left(\hat{F}((0,0)) = 1, \hat{F}((-1,-1)) = 1, \hat{F}((1,0)) = 0, \hat{F}((0,-1)) = 0\right) \\
&= \alpha\pi_1\pi_3(1-p) + \alpha\pi_3(1-\pi_1) + (1-\alpha)(1-\pi_2) + \pi_2(1-\alpha) \\
&= \alpha\pi_3(\pi_1(1-p) + 1 - \pi_1) + 1 - \alpha
\end{aligned} \tag{4.17}$$

and $\mathbf{P}\big(\hat{F}((0,0)) = \hat{F}((1,0)) = 0\big) = \mathbf{P}\big(\hat{F}((-1,-1)) = \hat{F}((0,-1)) = 0\big)$ which implies

$$\begin{aligned}
\alpha\pi_3 &= \mathbf{P}\left(\hat{F}((0,0))=0, \hat{F}((1,0))=0, \hat{F}((-1,-1))=0, \hat{F}((0,-1))=0\right) \\
&\quad +\mathbf{P}\left(\hat{F}((0,0))=0, \hat{F}((1,0))=1, \hat{F}((-1,-1))=0, \hat{F}((0,-1))=0\right) \\
&\quad +\mathbf{P}\left(\hat{F}((0,0))=1, \hat{F}((1,0))=0, \hat{F}((-1,-1))=0, \hat{F}((0,-1))=0\right) \\
&= \alpha\pi_1\pi_3(1-p) + \alpha\pi_1(1-p)(1-\pi_3) + (1-\alpha)(1-\pi_2) \\
&= \alpha\pi_1(1-p) + (1-\alpha)(1-\pi_2). \qquad (4.18)
\end{aligned}$$

The solution of (4.15)-(4.18) for π_1, π_2, π_3 in terms of p gives the formulas in Condition 3. □

Remark 4.5. *Equations (4.15)-(4.18) imply that*

$$\alpha = \frac{1}{2}\left(1 + \frac{(1-p)}{\sqrt{(1+3p)(1-p)}}\right) \qquad (4.19)$$

where $\alpha = \mathbf{P}\big(\hat{F}((0,0)) = 0\big)$. The conditions of Theorem 4.4 imply that for a bit stuffing encoder $\tilde{\mathcal{E}}^{(2)}$ with standard initialization, the probability that the label of any point $(x,y) \in \Delta_{N,N}$ is 0 equals α (independent of x and y).

Using Theorems 4.2, 4.4, Remark 4.5, and the fact that

$$\gamma = \alpha\pi_1 \qquad (4.20)$$

the achievable coding rate $r^{(2)}$ can be written as a function of p as

$$r^{(2)} = \gamma \cdot H(p) = \alpha\pi_1 \cdot H(p) \qquad (4.21)$$

since α and π_1 are implicit functions of p. The largest coding rate is found by maximizing (4.21) over the parameter p. In [16] this maximization was computed approximately as

$$\max_{p\in[0,\frac{1}{2}]} r^{(2)} = 0.58305621.$$

and occurred at $p = 0.3556$. The performance of the bit stuffing algorithm was later improved in [17]. The authors implicitly split a source into two subsources and apply different transformers to each subsource to create two different biased sources for stuffing. The authors obtained the approximate expected coding rate of their encoder $\mathcal{E}^{(2)\prime}$ as

$$r\left(\mathcal{E}^{(2)\prime}\right) = 0.587277.$$

In the present paper we generalize our previously described two-dimensional variable-to-variable length $(1,\infty)$-constrained bit stuffing encoder $\mathcal{E}^{(2)}$ to three dimensions. We show (in Theorem 4.19) that the three-dimensional algorithm achieves the approximate coding rate of

$$r^{(3)} = 0.502005.$$

4.5 Three-dimensional bit stuffing

In this section we describe a generalization of the two-dimensional bit stuffing algorithm to three dimensions. Often, identical notation to that used in earlier sections for two-dimensional bit stuffing will be redefined for three dimensions in an analogous way.

For $X, Y, Z \in \mathbf{Z}^+$ and $i \in \{0, \ldots, Z\}$, define the set

$$L^{(i)} = \Delta_{X,Y,0} + (0, i, i),$$

a translate of the parallelogram $\Delta_{X,Y,0} = \Delta_{X,Y}$, defined in Section 4.3. Let

$$\Delta_{X,Y,Z} = \bigcup_{i=0}^{Z} L^{(i)}$$

as shown in Figure 4.8. The set $L^{(i)}$ is called the ith layer of $\Delta_{X,Y,Z}$. For $j \in \{0, \ldots, X\}$ and $m \in \{0, \ldots, Y\}$ let

$$\begin{aligned} D_j^{(0)} &= \{(y + j, y, 0) : 0 \leq y \leq Y\} \\ R_m^{(0)} &= \{(x + m, m, 0) : 0 \leq x \leq X\} \end{aligned}$$

be the same subsets of $L^{(0)} = \Delta_{X,Y,0}$ as in Section 4.3. For $i \in \{1, \ldots, Z\}$, define similar subsets on each layer $L^{(i)}$, namely

$$\begin{aligned} D_j^{(i)} &= D_j^{(0)} + (0, i, i) = \{(y + j, y + i, i) : 0 \leq y \leq Y\} \\ R_m^{(i)} &= R_m^{(0)} + (0, i, i) = \{(x + m, m + i, i) : 0 \leq x \leq X\} \end{aligned}$$

for $j \in \{0, \ldots, X\}$ and $m \in \{0, \ldots, Y\}$ (see Figure 4.8). The points in $L^{(0)}$, $D_0^{(i)}$, $R_0^{(i)}$, for $i \in \{1, \ldots, Z\}$, are called the *boundary* of $\Delta_{X,Y,Z}$. That is, the boundary points consist of the entire first layer $L^{(0)}$ and the first diagonal and first row on every other layer. The non-boundary points of $\Delta_{X,Y,Z}$ are the *internal points*.

Notation: Every point of $\Delta_{X,Y,Z}$ can be determined by the layer, the diagonal on a given layer, and the relative position within the diagonal where the point lies. Therefore, to simplify the notation, the elements of $\Delta_{X,Y,Z}$ will be addressed by a three-tuple

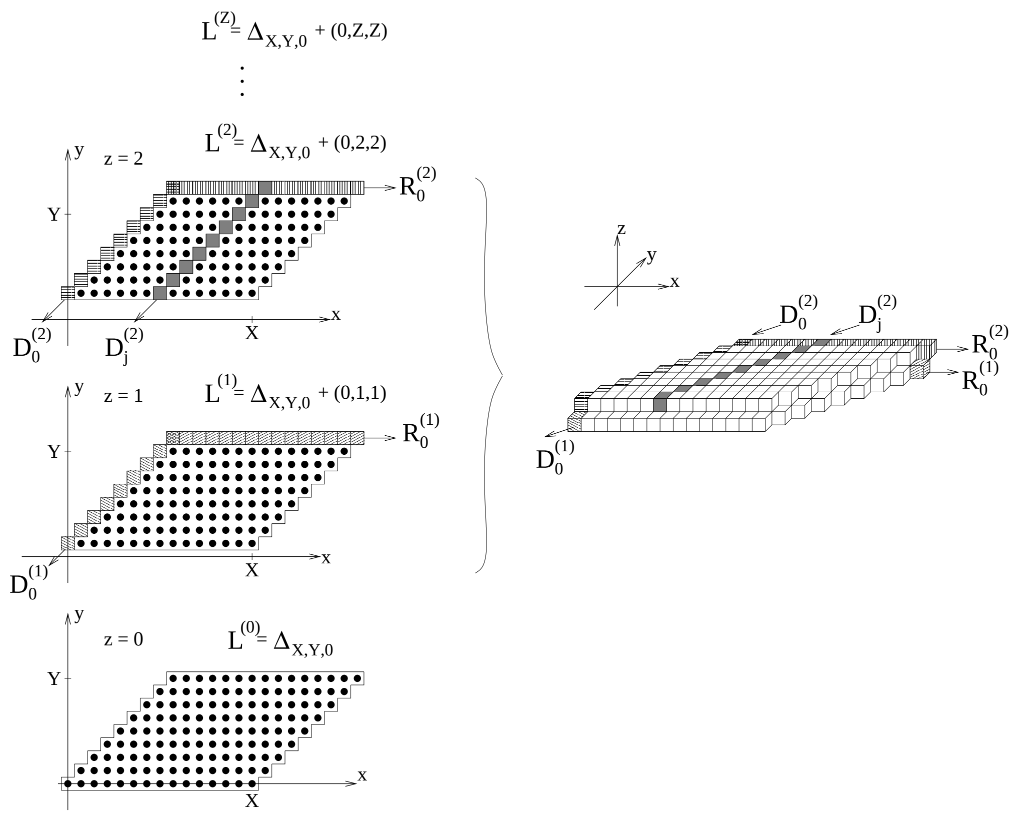


Figure 4.8: Layers of $\Delta_{X,Y,Z}$ are shown on the left hand side. The relative position of layers $L^{(1)}$ and $L^{(2)}$ is illustrated on the right hand side. The diagonals $D_0^{(1)}$, $D_0^{(2)}$, $D_j^{(2)}$ and the rows $R_0^{(1)}$, $R_0^{(2)}$ are shaded.

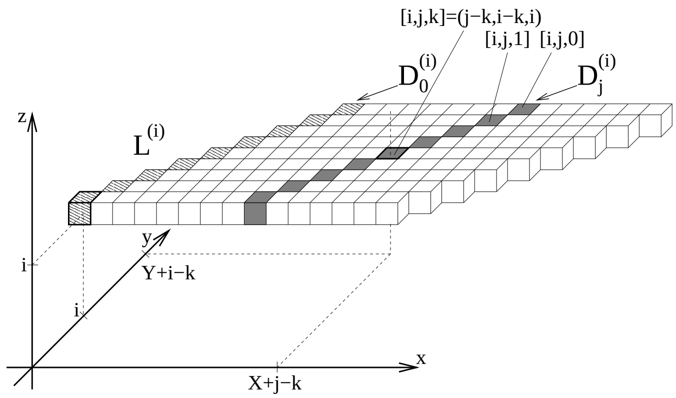


Figure 4.9: The coordinates $[i, j, k]$ denote the point on the ith layer, jth diagonal, kth position. In the Cartesian coordinate system this point is $[i, j, k] = (X + j - k, Y + i - k, i)$.

$[i, j, k]$, where $i \in \{0, \ldots, Z\}$ determines the layer $L^{(i)}$, $j \in \{0, \ldots, X\}$ determines the diagonal $D_j^{(i)}$ on layer $L^{(i)}$, and $k \in \{0, \ldots, Y\}$ is the position within the diagonal $D_j^{(i)}$. The point $[i, j, k]$ has coordinates $(X + j - k, Y + i - k, i)$ in the Cartesian coordinate system (see Figure 4.9).

An efficient three-dimensional coding algorithm would be to transform the unbiased input sequence into a biased sequence, and use a three-dimensional generalization of the bit stuffing encoder to map the biased sequence into a $(1, \infty)$-constrained labeling of $\mathbf{Z}^3$. To perform a rigorous analysis of the coding rate, we introduce a close variant of this implementation. We present a three-dimensional bit stuffing algorithm similar to the two-dimensional one defined in Section 4.4.

The three-dimensional algorithm's encoder is denoted by $\mathcal{E}^{(3)}$, and works as follows. As in two dimensions, a sequence w is parsed into the sequence of strings $w^{(1)}, w^{(2)}, \ldots$ using a complete prefix code. Then, the string $w^{(i)}$ is mapped into $\big(s^{(i,1)}, s^{(i,2)}\big)$, where $s^{(i,1)}$ is an ϵ-typical string with respect to p_1 of length $\lfloor \tau\gamma_1 N^3 \rfloor$ and $s^{(i,2)}$ is an ϵ-typical string with respect to p_2 of length $\lfloor \tau\gamma_2 N^3 \rfloor$. The value of τ is defined similarly as in the two-dimensional case; γ_1, γ_2 are defined in Section 4.6.3; and p_1

Table 4.4: Parameters used in Sections 4.5 and 4.6.

Parameter	Description
N	Positive integer parallelepiped side length. Goes to ∞.
γ_i	Probability of a bit copied from the ith ($i = 1, 2$) input string by $\tilde{\mathcal{E}}^{(3)}$.
λ	Initial labeling of U_τ.
τ	Number of translates of $\Delta_{N,N,N}$ in $\Delta_{N,N,Z}$.
ϵ	Positive real. Goes to 0.
p_i	Probability of 1 in ith ($i = 1, 2$) transformed sequence.
$\sigma^{(1)}, \sigma^{(2)}$	Auxiliary binary strings.
$s^{(1)}, s^{(2)}$	Input binary strings.

and p_2 are calculated in Section 4.6.4. Then, the three-dimensional fixed-to-variable length $(1, \infty)$-constrained bit stuffing encoder $\bar{\mathcal{E}}^{(3)}$ maps $\left(s^{(i,1)}, s^{(i,2)}\right)$ into a $(1, \infty)$-constrained labeling of $\Delta_{N,N,Z}$. The exact definitions of $\bar{\mathcal{E}}^{(3)}$ and $\mathcal{E}^{(3)}$ are given in Section 4.6.

A list of variables defined in Sections 4.5 and 4.6 and the parameters they depend on are given in Tables 4.4 and 4.5 as a reference.

4.5.1 A variable-to-fixed length encoder

The ordering $\prec$ defined in Section 4.3.1 is extended to $\mathbf{Z}^3$ in the following way. For any (x_1, y_1, z_1), $(x_2, y_2, z_2) \in \mathbf{Z}^3$, let

$$(x_1, y_1, z_1) \prec (x_2, y_2, z_2) \Longleftrightarrow \begin{cases} z_1 < z_2 \text{ or} \\ z_1 = z_2 \text{ and } (x_1, y_1) \prec (x_2, y_2) \end{cases} .$$

Note that the ordering of the elements in $\Delta_{X,Y,Z}$ with respect to $\prec$ is equivalent to lexicographic ordering if we use the coordinates $[i, j, k]$ to represent the points of $\Delta_{X,Y,Z}$.

Table 4.5: Variables introduced in Sections 4.5 and 4.6 and the parameters they depend on.

Notation	Parameters	Description
$\tilde{\mathcal{E}}^{(3)}$	$N,\quad,\lambda$	Variable-to-fixed length encoder. Labels $\Delta_{N,N,N}$.
$\bar{\mathcal{E}}^{(3)}$	$N,\gamma_1,\gamma_2,\lambda,\tau,\quad,\sigma^{(1)},\sigma^{(2)}$	Fixed-to-variable length encoder. Labels $\Delta_{N,N,Z}$.
$Q_i^{(j)}$	$N,\gamma_1,\gamma_2,\lambda,\tau,\quad,\sigma^{(1)},\sigma^{(2)},s^{(1)},s^{(2)}$	Number of bits $\bar{\mathcal{E}}^{(3)}$ maps into ith translate of $\Delta_{N,N,N}$ from jth input string.
q	$N,\gamma_1,\gamma_2,\lambda,\tau,\quad,\sigma^{(1)},\sigma^{(2)}$	Number of bits $\bar{\mathcal{E}}^{(3)}$ does not map into 1st τ translates of $\Delta_{N,N,N}$.
B	$N,\gamma_1,\gamma_2,\lambda,\tau,\epsilon,\quad,\sigma^{(1)},\sigma^{(2)}$	Set of strings that $\bar{\mathcal{E}}^{(3)}$ nearly maps into 1st τ translates of $\Delta_{N,N,N}$.
A	$N,\gamma_1,\gamma_2,\quad,\tau,\epsilon,p_1,p_2$	Set of typical sequences.
T	$N,\gamma_1,\gamma_2,\lambda,\tau,\epsilon,p_1,p_2,\sigma^{(1)},\sigma^{(2)}$	Complete prefix code of size $\lvert A\cap B\rvert$.
t	$N,\gamma_1,\gamma_2,\lambda,\tau,\epsilon,p_1,p_2,\sigma^{(1)},\sigma^{(2)}$	Bijection from T to $A\cap B$.
$\mathcal{E}^{(3)}$	$N,\gamma_1,\gamma_2,\lambda,\tau,\epsilon,p_1,p_2,\sigma^{(1)},\sigma^{(2)}$	Variable-to-variable length encoder. Labels $\Delta_{N,N,Z}$.

Table 4.6: The variable-to-fixed length $(1,\infty)$-constrained three-dimensional bit stuffing encoder $\tilde{\mathcal{E}}^{(3)}$. The algorithm maps $s^{(1)}$ and $s^{(2)}$ into a $(1,\infty)$-constrained labeling of $\Delta_{N,N,N}$.

1.	Initialize the elements of $L^{(0)}$ and $\bigcup_{i=1}^{N}\left(D_0^{(i)} \cup R_0^{(i)}\right)$ using λ. Let $m_1 = 1$ and $m_2 = 1$.
2.	Let $(x_0, y_0, z_0) = \min_{\prec}\{(x,y,z) \in \Delta_{N,N,N} : (x,y,z)$ is unlabeled$\}$.
3.	If $(x_0 - 1, y_0, z_0)$, $(x_0, y_0 + 1, z_0)$, or $(x_0, y_0, z_0 - 1)$ is labeled with 1
4.	Label (x_0, y_0, z_0) with 0.
5.	Else
6.	If $(x_0 + 1, y_0, z_0 - 1)$ and $(x_0, y_0 - 1, z_0 - 1)$ are labeled with 0
7.	Label (x_0, y_0, z_0) with $s_{m_1}^{(1)}$. Let $m_1 = m_1 + 1$.
8.	Else
9.	Label (x_0, y_0, z_0) with $s_{m_2}^{(2)}$. Let $m_2 = m_2 + 1$.
10.	If all of $\Delta_{N,N,N}$ is labeled then stop, else go to 2.

Let

$$\lambda : L^{(0)} \cup \left(\bigcup_{i=1}^{N} \left(D_0^{(i)} \cup R_0^{(i)} \right) \right) \longrightarrow \{0,1\}$$

be an initial labeling of the boundary of $\Delta_{N,N,N}$. The three-dimensional variable-to-fixed length bit stuffing encoder

$$\tilde{\mathcal{E}}^{(3)} : V_{N,\lambda} \longrightarrow \Lambda_{1,\infty}^{(3)}(\Delta_{N,N,N})$$

labels the points of $\Delta_{N,N,N}$ in increasing order with respect to the ordering $\prec$. The set $V_{N,\lambda}$ consists of pairs of strings $\left(s^{(1)}, s^{(2)}\right)$ that perfectly fit into $\Delta_{N,N,N}$ under the mapping $\tilde{\mathcal{E}}^{(3)}$ (analogous to the two-dimensional case in (4.3)). For $\left(s^{(1)}, s^{(2)}\right) \in V_{N,\lambda}$, every point of $\Delta_{N,N,N}$ is labeled either with a bit of $s^{(1)}$ or $s^{(2)}$ or with a stuffed 0. A pseudo-code description of $\tilde{\mathcal{E}}^{(3)}$ is given in Table 4.6. Note that in Step 6 the encoder makes a decision whether the first or the second input string is used to label the current position. This selection process ensures that the encoder is invertible (see Remark 4.18). The inverse mapping scans the elements of $\Delta_{N,N,N}$ in increasing order with respect to the ordering $\prec$, skipping over stuffed 0s to recover the input sequences.

In the following section $\tilde{\mathcal{E}}^{(3)}$ is used to define a three-dimensional variable-to-variable length $(1,\infty)$-constrained bit stuffing encoder.

4.6 A three-dimensional variable-to-variable length encoder

A three-dimensional variable-to-variable length $(1,\infty)$-constrained bit stuffing encoder $\mathcal{E}^{(3)}$ is defined analogously to the two-dimensional encoder $\mathcal{E}^{(2)}$. Using a finite complete prefix code defined in Section 4.6.2, an input sequence w is parsed into variable length strings $w^{(1)}, w^{(2)}, \ldots$. Each string $w^{(i)}$ is mapped into a $(1,\infty)$-constrained labeling of a translate of the set $\Delta_{N,N,Z(w^{(i)})}$, where N is a parameter of the encoder, and where $Z(w^{(i)})$ is a positive integer chosen so that the mapped prefix code fits into $\Delta_{N,N,Z(w^{(i)})}$ using bit stuffing. Henceforth we abbreviate $Z(w^{(i)})$ with Z. An analysis of the coding rate of $\mathcal{E}^{(3)}$ is given in Section 4.6.3 when the input is the $1/2$-sequence $\hat{w}$.

4.6.1 An intermediate fixed-to-variable length encoder

The set $\Delta_{N,N,Z}$ can be decomposed as

$$\Delta_{N,N,Z} = \left(\bigcup_{i=0}^{\tau-1} (\Delta_{N,N,N} + i[N+2,0,0])\right) \cup \left(\bigcup_{j=1}^{\tau} L^{(j(N+2)-1)}\right) \cup \left(\bigcup_{k=\tau(N+2)}^{Z} L^{(k)}\right).$$

The translates $\Delta_{N,N,N} + i[N+2,0,0]$ are labeled with information bits and stuffed 0s, the layers $L^{(j(N+2)-1)}$ are padded with 0s, and some additional “overflow” layers $L^{(\tau(N+2))}, \ldots, L^{(Z)}$ are filled randomly.

We next define a fixed-to-variable length encoder

$$\bar{\mathcal{E}}^{(3)} : \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor} \times \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor} \longrightarrow \bigcup_{S \in \mathbf{Z}^3} \Lambda^{(3)}_{1,\infty}(S)$$

by specifying the labeling $\bar{\mathcal{E}}^{(3)}\left(s^{(1)}, s^{(2)}\right) : \Delta_{N,N,Z} \longrightarrow \{0,1\}$ as:

$$\bar{\mathcal{E}}^{(3)}\left(s^{(1)}, s^{(2)}\right)(u) = \begin{cases} \tilde{\mathcal{E}}^{(3)}\left(z^{(i,1)}, z^{(i,2)}\right)\left(u^{(i)}\right) & \text{if } u \in \Delta_{N,N,N} + i[N+2,0,0] \\ & \text{for some } i = 0, \ldots, \tau - 1 \\ \left(\bar{s}^{(1)}\bar{s}^{(2)}\right)_{\delta(u)} & \text{if } u = [u_1,u_2,u_3] \in \Delta_{N,N,N}, \\ & u_1 \in \{\tau(N+2), \ldots, Z\}, \\ & u_2 \text{ even}, \\ & \delta(u) \le l\left(\bar{s}^{(1)}\bar{s}^{(2)}\right) \\ 0 & \text{otherwise .} \end{cases}$$

where

$$\delta(u) = \frac{u_2}{2}(N+1) + u_3 + 1 + (u_1 - \tau(N+2))(N+1)\left\lceil \frac{(N+1)}{2} \right\rceil;$$

$\tilde{\mathcal{E}}^{(3)}$ is the encoder defined in Section 4.3.1; $\left(z^{(i,1)}, z^{(i,2)}\right)$ is a parsing of $\left(s^{(1)}\sigma^{(1)}, s^{(2)}\sigma^{(2)}\right)$ for some auxiliary sequences $\sigma^{(1)}$, $\sigma^{(2)}$; and $\bar{s}^{(i)}$ is the suffix of $s^{(i)}$ (for $i = 1, 2$) that does not get encoded into the first τ translates of $\Delta_{N,N,N}$. The default case when $\bar{\mathcal{E}}^{(3)}\left(s^{(1)}, s^{(2)}\right)(u) = 0$ includes those u which lie in the inter-translate layers $L^{(j(N+2)-1)}$ for $j = 1, 2, \ldots, \tau$, as well as those $u = [u_1, u_2, u_3] \in \Delta_{N,N,N}$ for which $u_1 \in \{\tau(N+2), \ldots, Z\}$, and either u_2 is odd or else $\delta(u) > l\left(\bar{s}^{(1)}\bar{s}^{(2)}\right)$.

The encoder $\bar{\mathcal{E}}^{(3)}$ will serve as the second stage of a variable-to-variable length encoder and is defined with respect to the fixed and finite auxiliary binary sequences

$\sigma^{(1)} \in \{0,1\}^{\tau N^3}$ and $\sigma^{(2)} \in \{0,1\}^{\tau N^3}$ which are described in Lemma 4.6. For any input strings $s^{(1)} \in \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor}$ and $s^{(2)} \in \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor}$, the labeling $\bar{\mathcal{E}}^{(3)}(s^{(1)}, s^{(2)})$: $\Delta_{N,N,Z} \longrightarrow \{0,1\}$ labels the set $\Delta_{N,N,Z}$ in the following way. Let

$$U_\tau = \bigcup_{j=0}^{\tau-1} \left(\left(\bigcup_{i=1}^{N} \left(D_0^{(i)} \cup R_0^{(i)} \right) \cup L^{(0)} \right) + [j(N+2), 0, 0] \right)$$

be a union of boundaries of the first τ translates of $\Delta_{N,N,N}$.

The elements of U_τ are assigned a fixed initial labeling λ (to be determined from Lemma 4.6). The translates of $\Delta_{N,N,N}$ are labeled with $s^{(1)}$ and $s^{(2)}$ using the variable-to-fixed length bit stuffing encoder $\tilde{\mathcal{E}}^{(3)}$ for each translate with the fixed initial labeling λ on the boundary points.

Labeling all τ translates of $\Delta_{N,N,N}$ using the encoder $\tilde{\mathcal{E}}^{(3)}$, and adding the padding layers after each translate defines a labeling of the set $\Delta_{N,N,\tau(N+2)-1}$. In a similar manner as in two dimensions, if the sequence $s^{(1)}$ is shorter than the necessary bits to label $\Delta_{N,N,\tau(N+2)-1}$, the auxiliary sequence $\sigma^{(1)}$ is appended as a suffix to the string $s^{(1)}$. Likewise, the auxiliary sequence $\sigma^{(2)}$ is used if all bits of $s^{(2)}$ are encoded before the labeling of $\Delta_{N,N,\tau(N+2)-1}$ is complete. Note that τN^3 is an upper bound on the number of auxiliary bits needed.

It is also possible that some bits of $s^{(1)}$ or $s^{(2)}$ do not get encoded into $\Delta_{N,N,\tau(N+2)-1}$. In this case, first the unencoded bits of $s^{(1)}$ are copied into the even numbered diagonals $D_0^{(\tau(N+2))}$, $D_2^{(\tau(N+2))}, \ldots$ of the first overflow layer $L^{(\tau(N+2))}$ with a padding diagonal of 0s separating them. Unlike in two dimensions, the input bits are copied bit by bit into these diagonals, i.e. without using bit stuffing. The encoder continues this process on consecutive layers until all bits of $s^{(1)}$ are encoded. After that, the remaining bits of $s^{(2)}$ are encoded using a similar method. The last layer used to encode the last input bits may contain unlabeled points which are labeled with 0s. Finally, an additional layer of padding 0s is added, whose index is defined to be Z.

The padding diagonals on each overflow layer guarantee that the labeling of the layers $L^{(\tau(N+2))}, \ldots, L^{(Z)}$ is $(1, \infty)$-constrained. By choosing τ appropriately, and by adjusting the parameters γ_1 and γ_2, it will be guaranteed that $s^{(1)}$ and $s^{(2)}$ fill up $\Delta_{N,N,\tau(N+2)-1}$ almost perfectly (for large N and small ϵ), and therefore the number of overflow layers added will be small.

For given τ, γ_1, γ_2, N, binary input strings $s^{(1)}$ and $s^{(2)}$, auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$, and initial labeling λ of U_τ, we define the quantities:

$$\begin{aligned}
Q_i^{(1)} &= \text{number of bits of } s^{(1)}\sigma^{(1)} \text{ that } \bar{\mathcal{E}}^{(3)} \text{ maps into } \Delta_{N,N,N} + i[N+2,0,0] \\
Q_i^{(2)} &= \text{number of bits of } s^{(2)}\sigma^{(2)} \text{ that } \bar{\mathcal{E}}^{(3)} \text{ maps into } \Delta_{N,N,N} + i[N+2,0,0] \\
q_1(s^{(1)}, s^{(2)}) &= \text{length of the longest suffix of } s^{(1)} \text{ not mapped into } \Delta_{N,N,\tau(N+2)-1} \\
q_2(s^{(1)}, s^{(2)}) &= \text{length of the longest suffix of } s^{(2)} \text{ not mapped into } \Delta_{N,N,\tau(N+2)-1}.
\end{aligned}$$

If the auxiliary sequence $\sigma^{(i)}$ (for $i = 1, 2$) is used during encoding, then $q_i(s^{(1)}, s^{(2)})$ is minus the number of bits of $\sigma^{(i)}$ that get mapped into $\Delta_{N,N,\tau(N+2)-1}$. For any $\epsilon, \gamma_1, \gamma_2 > 0$ and for fixed λ, $\sigma^{(1)}$, $\sigma^{(2)}$, let

$$\begin{aligned}
B \;=\; & \Big\{ (s^{(1)}, s^{(2)}) \in \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor} \times \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor} : \\
& \qquad n_1(s^{(1)}, s^{(2)}) < \tau\epsilon,\ n_2(s^{(1)}, s^{(2)}) < \tau\epsilon \Big\}. \qquad (4.22)
\end{aligned}$$

The set B represents the pairs of strings that “fit well” into $\Delta_{N,N,\tau(N+2)-1}$.

4.6.2 Restriction to typical sequences

Let

$$
\begin{aligned}
A = \Big\{ & (s^{(1)}, s^{(2)}) \in \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor} \times \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor} : \\
& 2^{-\lfloor \tau\gamma_1 N^3 \rfloor (H(p_1)+\epsilon)} \leq p_1^{|s|}(1-p_1)^{\lfloor \tau\gamma_1 N^3 \rfloor - |s|} \leq 2^{-\lfloor \tau\gamma_1 N^3 \rfloor (H(p_1)-\epsilon)}, \\
& 2^{-\lfloor \tau\gamma_2 N^3 \rfloor (H(p_2)+\epsilon)} \leq p_2^{|s|}(1-p_2)^{\lfloor \tau\gamma_2 N^3 \rfloor - |s|} \leq 2^{-\lfloor \tau\gamma_2 N^3 \rfloor (H(p_2)-\epsilon)} \Big\}
\end{aligned}
$$

be ϵ-typical sequence pairs $s^{(1)}$ and $s^{(2)}$ with respect to p_1 and p_2, of length $\tau\gamma_1 N^3$ and $\tau\gamma_2 N^3$, respectively.

Let T be a finite complete prefix code of cardinality $|A \cap B|$ whose codewords are one of two possible lengths. Let

$$t : T \longrightarrow A \cap B$$

be any bijection. The code T parses an infinite input sequence, and t maps a finite parsed string to a pair of ϵ-typical (with respect to p_1 and p_2, respectively) sequences $\left(s^{(1)}, s^{(2)}\right)$ that are likely to fit into the first τ translates of $\Delta_{N,N,N}$. Since T is a complete prefix code, a binary sequence w can uniquely be parsed into a sequence of words $w^{(1)}, w^{(2)}, \ldots$ such that $w^{(i)} \in T$. We define a *three-dimensional variable-to-variable length* $(1,\infty)$*-constrained bit stuffing encoder* $\mathcal{E}^{(3)}$ is defined as the composition

$$\mathcal{E}^{(3)} = \bar{\mathcal{E}}^{(3)} \circ t.$$

That is, a string $w^{(i)} \in T$ of the parsed sequence w is transformed into the typical, well-fitting string $\left(s^{(i,1)}, s^{(i,2)}\right) \in A \cap B$ by the bijection t, and then $\left(s^{(i,1)}, s^{(i,2)}\right)$ is mapped into a $(1,\infty)$-constrained labeling of $\Delta_{N,N,Z}$ using the bit stuffing encoder $\bar{\mathcal{E}}^{(3)}$. The *variable-to-variable length three-dimensional* $(1,\infty)$*-constrained bit stuffing algorithm* consists of the mapping t, a bit stuffing encoder, a bit-stuffing decoder, and the inverse mapping of t. The mapping $\mathcal{E}^{(3)}$ is referred to as the algorithm's encoder, and the inverse is called the algorithm's decoder. An arbitrary number of words $w^{(i)}$ can be transformed into $\left(s^{(i,1)}, s^{(i,2)}\right)$, and mapped into labelings of translates of Δ_{N,N,Z_i}. The translates are separated by padding 0s in three dimensions similarly as in two dimensions.

4.6.3 Coding rate analysis

Let $\hat{s}^{(1)}$ be a p_1-sequence and $\hat{s}^{(2)}$ be a p_2-sequence. Let the variable-to-fixed length encoder $\tilde{\mathcal{E}}^{(3)}$ map $\hat{s}^{(1)}$ and $\hat{s}^{(2)}$ into $\Delta_{N,N,N}$. Before the encoding, let the boundary elements $L^{(0)} \cup \left(\bigcup_{i=1}^{N} \left(D_0^{(i)} \cup R_0^{(i)}\right)\right)$ be randomly assigned initial labels by $\hat{\lambda}$. For every internal point $[i, j, k]$ define

$$\begin{aligned}
\gamma_1(i,j,k) &= \text{probability that } [i,j,k] \text{ is labeled with a bit from } \hat{s}^{(1)} \\
\gamma_2(i,j,k) &= \text{probability that } [i,j,k] \text{ is labeled with a bit from } \hat{s}^{(2)}
\end{aligned}$$

as in Steps 7 and 9 of Table 4.6.

The random initial labeling $\hat{\lambda}$ is called a *standard three-dimensional initialization corresponding to* p_1 *and* p_2 if for $l = 1, 2$ and for every internal point $[i, j, k]$ (where $i, j, k \in \{1, \ldots, N\}$) the quantity $\gamma_l(i, j, k) = \gamma_l$ is independent of i, j, k, N. The corresponding labeling of $\Delta_{N,N,N}$ is called a *standard three-dimensional labeling corresponding to* p_1 *and* p_2. Let

$$\begin{aligned}
\Omega = \Big\{ (p_1, p_2) :\ & \text{there exists a random initialization} \\
& \hat{\lambda} : L^{(0)} \cup \left(\textstyle\bigcup_{i=1}^{N} \left(D_0^{(i)} \cup R_0^{(i)}\right)\right) \longrightarrow \{0,1\} \text{ such that} \\
& \text{the labeling of } \Delta_{N,N,N} \text{ by } \tilde{\mathcal{E}}^{(3)} \text{ is a standard labeling} \Big\}.
\end{aligned}$$

It is shown in Section 4.6.4 that Ω is nonempty (in the paragraph preceding Remark 4.18).

A fixed initial labeling λ of U_τ, and auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$ used by $\mathcal{E}^{(3)}$, are implied in the following lemma. The set B in the lemma is defined in (4.22).

Lemma 4.6. *Consider the p_1-sequence $\hat{s}^{(1)} \in \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor}$ and the p_2-sequence $\hat{s}^{(2)} \in \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor}$ for some $(p_1, p_2) \in \Omega$. For any $N \in \mathbf{Z}^+$ and any $\epsilon > 0$, there exists $\tau_0 \in \mathbf{Z}^+$ such that for any $\tau \geq \tau_0$ there is an initial labeling $\lambda : U_\tau \longrightarrow \{0,1\}$ and auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$ such that*

$$\mathbf{P}\left(\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) \in B\right) > 1 - \epsilon.$$

Proof. Suppose that $\hat{s}^{(1)}$ and $\hat{s}^{(2)}$ are encoded into $\Delta_{N,N,Z}$ by $\bar{\mathcal{E}}^{(3)}$ such that the random initial labels defined by $\hat{\lambda} : U_\tau \longrightarrow \{0,1\}$ constitute a standard three-dimensional initialization corresponding to p_1 and p_2 on each translate of $\Delta_{N,N,N}$. Furthermore, let the auxiliary sequence $\hat{\sigma}^{(1)} \in \{0,1\}^{\gamma_1 N^3}$ be a p_1-sequence, and let the auxiliary sequence $\hat{\sigma}^{(2)} \in \{0,1\}^{\gamma_2 N^3}$ be a p_2-sequence.

Since the labeling of each translate of $\Delta_{N,N,N}$ is a standard labeling, by the definition of γ_1 and γ_2, for every $i \in \{0, 1, \ldots, \tau - 1\}$ we have

$$\begin{aligned} E\left[\hat{Q}_i^{(1)}\right] &= \gamma_1 N^3 \\ E\left[\hat{Q}_i^{(2)}\right] &= \gamma_2 N^3. \end{aligned}$$

For $l = 1, 2$ and any two distinct $i \in \{0, 1, \ldots, \tau - 1\}$, the random variables $\hat{Q}_i^{(l)}$ are independent with finite variances (independent of τ). Thus, the weak law of large numbers implies that for every $\epsilon > 0$,

$$\lim_{\tau\to\infty} \mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}_i^{(1)} - \gamma_1 N^3\right| < \epsilon\right) = 1 \tag{4.23}$$

$$\lim_{\tau\to\infty} \mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}_i^{(2)} - \gamma_2 N^3\right| < \epsilon\right) = 1. \tag{4.24}$$

The random variables $\hat{Q}_i^{(l)}$ in (4.23) and (4.24) are functions of the random input p_1-sequence $\hat{s}^{(1)}$ and the p_2-sequence $\hat{s}^{(2)}$, the random auxiliary sequences $\hat{\sigma}^{(1)}$ and $\hat{\sigma}^{(2)}$, and the random initialization $\hat{\lambda}$ of U_τ. It follows from (4.23) and (4.24) that

$$\lim_{\tau\to\infty} \mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1}\hat{Q}_i^{(1)} - \gamma_1 N^3\right| < \epsilon, \left|\frac{1}{\tau}\sum_{j=0}^{\tau-1}\hat{Q}_j^{(2)} - \gamma_2 N^3\right| < \epsilon\right) = 1. \tag{4.25}$$

Then (4.25) and the inequalities

$$\begin{aligned}
&\mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1}\hat{Q}_i^{(1)} - \gamma_1 N^3\right| < \epsilon, \left|\frac{1}{\tau}\sum_{j=0}^{\tau-1}\hat{Q}_j^{(2)} - \gamma_2 N^3\right| < \epsilon\right) \\
\leq\ & \mathbf{P}\left(\tau\gamma_1 N^3 - \sum_{i=0}^{\tau-1}\hat{Q}_i^{(1)} < \tau\epsilon, \tau\gamma_2 N^3 - \sum_{j=0}^{\tau-1}\hat{Q}_j^{(2)} < \tau\epsilon\right) \\
\leq\ & \mathbf{P}\left(\lfloor\tau\gamma_1 N^3\rfloor - \sum_{i=0}^{\tau-1}\hat{Q}_i^{(1)} < \tau\epsilon, \lfloor\tau\gamma_2 N^3\rfloor - \sum_{j=0}^{\tau-1}\hat{Q}_j^{(2)} < \tau\epsilon\right) \\
=\ & \mathbf{P}\left(q_1\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon, q_2\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon\right)
\end{aligned}$$

imply that

$$\lim_{\tau\to\infty} \mathbf{P}\left(q_1\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon, q_2\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon\right) = 1. \tag{4.26}$$

It follows from (4.26) that there exists a τ_0 such that for all $\tau \geq \tau_0$,

$$\mathbf{P}\left(q_1\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon, q_2\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon\right) > 1 - \epsilon.$$

Thus there must exist at least one initial labeling λ and auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$ (all three depending on τ) such that

$$\mathbf{P}\left(q_1\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon, q_2\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) < \tau\epsilon \middle| \lambda, \sigma^{(1)}, \sigma^{(2)}\right) > 1 - \epsilon \tag{4.27}$$

where the conditioning in (4.27) is on the event that the random labeling $\hat{\lambda}$ equals the fixed labeling λ, and the random auxiliary sequences $\hat{\sigma}^{(1)}$ and $\hat{\sigma}^{(2)}$ equal $\sigma^{(1)}$ and $\sigma^{(2)}$, respectively. Equivalently, for every $\tau \geq \tau_0$,

$$\mathbf{P}\left(\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) \in B\right) > 1 - \epsilon.$$

□

A number $r^{(3)}$ is said to be an *achievable coding rate* of a three-dimensional $(1, \infty)$-constrained bit stuffing algorithm $\mathcal{E}^{(3)}$ if

$$r^{(3)} = \lim_{\epsilon \to 0} \lim_{N \to \infty} r\left(\mathcal{E}^{(3)}\right).$$

Theorem 4.7. *For any $(p_1, p_2) \in \Omega$, the three-dimensional bit stuffing algorithm achieves a coding rate of*

$$r^{(3)} = \gamma_1 H(p_1) + \gamma_2 H(p_2).$$

Proof. Let $(p_1, p_2) \in \Omega$, and let the p_1-sequence $\hat{s}^{(1)}$ and the p_2-sequence $\hat{s}^{(2)}$ be independent. We have

$$\mathbf{P}\left(\left(\hat{s}^{(1)}, \hat{s}^{(2)}\right) \in A\right) > (1 - \epsilon)^2$$

for $\tau \geq \tau_0$ large enough (see [4, pp. 51-52]). The rest of the proof is similar to the proof of Theorem 4.2 and is therefore omitted. □

As in two dimensions, for given parameters N, p_1, p_2, ϵ, the value of τ is induced by Lemma 4.6 and Theorem 4.7. For a fixed τ, the initial labeling λ and the auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$ are implied in Lemma 4.6. The parameters γ_1 and γ_2 are given in (4.40) and (4.41) of Appendix 4.8.

4.6.4 Coding rate maximization

The following lemmas about the two-dimensional standard labeling of $\Delta_{N,N}$ follow from Theorem 4.4 and the definition of the two-dimensional bit stuffing encoder. These results are necessary for the analysis of the three-dimensional algorithm. Let $\hat{\lambda}'$ denote a standard two-dimensional random initialization of the boundary of $\Delta_{N,N}$ by the stationary homogeneous Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$. The diagonals D_i and D_{i+1} in the following lemma are illustrated in Figure 4.7.

Lemma 4.8. *Let the p-sequence $\hat{s}$ be encoded into the parallelogram $\Delta_{N,N}$ by $\tilde{\mathcal{E}}^{(2)}$ using a standard initial labeling $\hat{\lambda}'$. For a fixed $i \in \{0, \ldots, N-1\}$, let $a_j = (i+j, j)$ and $b_j = (i+j+1, j)$ denote the elements of the diagonals D_i and D_{i+1}, respectively. The sequence $\left\{\hat{F}(a_j, b_j)\right\}_{j=0}^{N}$ forms a stationary homogeneous Markov chain.*

Lemma 4.9. *Let the p-sequence $\hat{s}$ be encoded into the parallelogram $\Delta_{N,N}$ by $\tilde{\mathcal{E}}^{(2)}$ using a standard initial labeling $\hat{\lambda}'$. For $i \in \{0, \ldots, N\}$, let the random vector $\hat{F}(D_i)$ denote the labels of the diagonal D_i. The sequence $\left\{\hat{F}(D_i)\right\}_{i=0}^{N}$ forms a stationary homogeneous Markov chain.*

Henceforth in this section we consider the labeling of the set $\Delta_{N,N,N}$ by the encoder $\tilde{\mathcal{E}}^{(3)}$, where the input sequences are the p_1-sequence $\hat{s}^{(1)}$ and the p_2-sequence $\hat{s}^{(2)}$, and where the boundary elements of $\Delta_{N,N,N}$ are assigned random initial labels by $\hat{\lambda} : L^{(0)} \cup \left(\bigcup_{i=1}^{N} \left(D_0^{(i)} \cup R_0^{(i)}\right)\right) \longrightarrow \{0,1\}$ described in what follows.

The initial labels of $L^{(0)}$ are chosen independent of the input sequences and such that the labels of $L^{(0)}$ constitute a standard two-dimensional labeling. More precisely, let $p \in [0,1]$, and let π_1, π_2, π_3 be defined as in Theorem 4.4. The diagonal $D_0^{(0)}$ and the row $R_0^{(0)}$ are initialized by the stationary homogeneous Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$, respectively. Let $\hat{\sigma}'$ be an auxiliary p-sequence independent of $\hat{s}^{(1)}$ and $\hat{s}^{(2)}$, used only to initialize $L^{(0)}$. Using $\hat{\sigma}'$ and the two-dimensional bit stuffing encoder $\tilde{\mathcal{E}}^{(2)}$, we label the remainder of the elements of $L^{(0)}$. Thus, the resulting labeling of $L^{(0)}$ is a standard two-dimensional labeling. The points of $D_0^{(i)}$ and $R_0^{(i)}$ on each layer $L^{(i)}$ (for $i \in \{1, \ldots, N\}$) are initialized independently for each i by the Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ used to initialize $D_0^{(0)}$ and $R_0^{(0)}$, respectively.

Note that p is a parameter of the above random initialization of $\Delta_{N,N,N}$. We show that p_1 and p_2 can also be expressed in terms of p such that the resulting labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ is a standard three-dimensional labeling corresponding to p_1 and p_2.

Let the random label of the point $[i, j, k] \in \Delta_{N,N,N}$ be denoted by $\hat{F}(i, j, k)$. Our goal is to prove that if $\Delta_{N,N,N}$ is labeled by $\tilde{\mathcal{E}}^{(3)}$, then Conditions (**a**)-(**d**) below are sufficient for the labels of $\Delta_{N,N,N}$ to be a standard labeling. In Appendix 4.8 we show that there exist parameters $\pi_1, \pi_2, \pi_3, p, p_1, p_2$, such that Conditions (**a**)-(**d**) are satisfied.

Let $\mathcal{A} = [0, 0, 0]$, $\mathcal{B} = [0, 1, 0]$, $\mathcal{C} = [1, 0, 0]$, $\mathcal{D} = [1, 1, 0]$, $\mathcal{E} = [0, 0, 1]$, $\mathcal{F} = [0, 1, 1]$, $\mathcal{G} = [1, 0, 1]$, $\mathcal{H} = [1, 1, 1]$, as shown in Figure 4.10.

Condition (**a**) The joint distribution of $\hat{F}(\mathcal{B}, \mathcal{D}, \mathcal{F}, \mathcal{H})$ is identical to the joint distribution of $\hat{F}(\mathcal{A}, \mathcal{C}, \mathcal{E}, \mathcal{G})$.

Condition (**b**) The joint distribution of $\hat{F}(\mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H})$ is identical to the joint distribution of $\hat{F}(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$.

Condition (**c**) The joint distribution of $\hat{F}(\mathcal{C}, \mathcal{D}, \mathcal{G}, \mathcal{H})$ is identical to the joint distribution of $\hat{F}(\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F})$.

Condition (**d**) The joint random variables $\hat{F}(\mathcal{E}, \mathcal{G})$ are conditionally independent of $\hat{F}(\mathcal{B}, \mathcal{D})$ given $\hat{F}(\mathcal{F}, \mathcal{H})$.

In the following we show that Conditions (**a**)-(**d**) imply that for every $x \in \Delta_{N,N,N}$ if $\mathcal{A} + x, \mathcal{B} + x, \ldots, \mathcal{H} + x \in \Delta_{N,N,N}$, then the labels of the translates $\hat{F}(\mathcal{A}+x, \mathcal{B}+x, \mathcal{C}+x, \mathcal{D}+x, \mathcal{E}+x, \mathcal{F}+x, \mathcal{G}+x, \mathcal{H}+x)$ have the same joint probability distribution as the labels $\hat{F}(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{H})$. First, we show that it holds for the translates on the diagonals $D_0^{(0)}, D_1^{(0)}, D_0^{(1)}, D_1^{(1)}$. In the following lemma, the elements a_l, b_l, c_l, d_l are shown in Figure 4.11.

Lemma 4.10. *Let* $a_l = [0, 0, l]$, $b_l = [0, 1, l]$, $c_l = [1, 0, l]$, $d_l = [1, 1, l]$ *denote the*

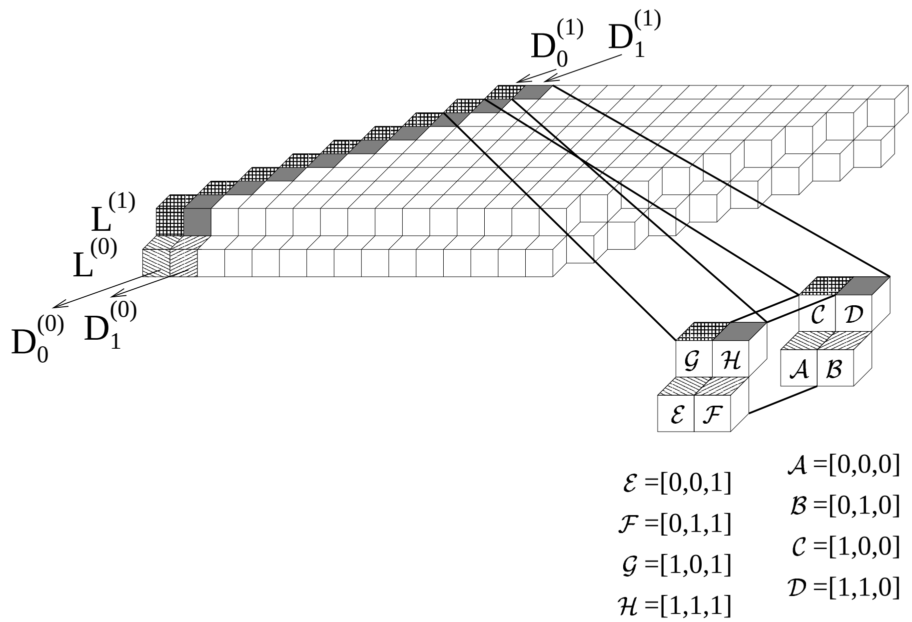


Figure 4.10: The elements of $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$ that appear in Conditions (**a**)-(**d**). The labels represent the centers of the cubes.

*elements of $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$, respectively. If the labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ satisfies Condition (**b**), then for all $l \in \{1, \dots, N\}$, the joint distribution of $\hat{F}(a_{l-1}, b_{l-1}, c_{l-1}, d_{l-1}, a_l, b_l, c_l, d_l)$ is identical to the joint distribution of $\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1)$.*

Proof. The lemma trivially holds for $l = 1$. Suppose the lemma is true for $l = k$, where $1 \leq k < N$. We will show that the joint distribution of $\hat{F}(a_k, b_k, c_k, d_k, a_{k+1}, b_{k+1}, c_{k+1}, d_{k+1})$ is identical to the joint distribution of $\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1)$. Let $v_1, v_2, \dots, v_8 \in \{0, 1\}$ be a valid labeling of the points $a_k, b_k, c_k, d_k, a_{k+1}, b_{k+1}, c_{k+1}, d_{k+1}$, respectively. Then,

$$\begin{aligned}
&\mathbf{P}\big(\hat{F}(a_k, b_k, c_k, d_k, a_{k+1}, b_{k+1}, c_{k+1}, d_{k+1}) = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)\big) \\
&\quad = \mathbf{P}\big(\hat{F}(a_k, b_k, c_k, d_k) = (v_1, v_2, v_3, v_4)\big) \\
&\qquad \cdot \mathbf{P}\big(\hat{F}(a_{k+1}, b_{k+1}) = (v_5, v_6) \big| \hat{F}(a_k, b_k, c_k, d_k) = (v_1, v_2, v_3, v_4)\big) \\
&\qquad \cdot \mathbf{P}\big(\hat{F}(c_{k+1}, d_{k+1}) = (v_7, v_8) \big| \hat{F}(a_k, b_k, c_k, d_k, a_{k+1}, b_{k+1}) = (v_1, v_2, v_3, v_4, v_5, v_6)\big).
\end{aligned} \tag{4.28}$$

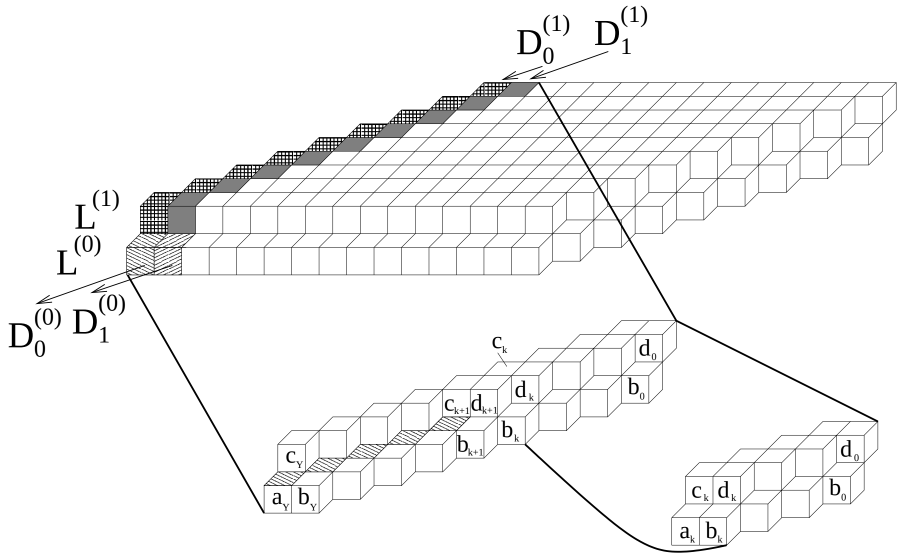


Figure 4.11: The diagonals $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$.

In what follows, we rewrite each of the three terms in (4.28). First we have

$$\begin{aligned}
&\mathbf{P}\big(\hat{F}(a_k, b_k, c_k, d_k) = (v_1, v_2, v_3, v_4)\big) \\
&\quad = \mathbf{P}\big(\hat{F}(a_1, b_1, c_1, d_1) = (v_1, v_2, v_3, v_4)\big) && (4.29) \\
&\quad = \mathbf{P}\big(\hat{F}(a_0, b_0, c_0, d_0) = (v_1, v_2, v_3, v_4)\big) && (4.30)
\end{aligned}$$

where (4.29) follows from the induction hypothesis (summing out four terms), and (4.30) follows from Condition (**b**). Furthermore,

$$\begin{aligned}
&\mathbf{P}\big(\hat{F}(a_{k+1}, b_{k+1}) = (v_5, v_6) \big| \hat{F}(a_k, b_k, c_k, d_k) = (v_1, v_2, v_3, v_4)\big) \\
&\quad = \mathbf{P}\big(\hat{F}(a_{k+1}, b_{k+1}) = (v_5, v_6) \big| \hat{F}(a_k, b_k) = (v_1, v_2)\big) && (4.31) \\
&\quad = \mathbf{P}\big(\hat{F}(a_1, b_1) = (v_5, v_6) \big| \hat{F}(a_0, b_0) = (v_1, v_2)\big) && (4.32) \\
&\quad = \mathbf{P}\big(\hat{F}(a_1, b_1) = (v_5, v_6) \big| \hat{F}(a_0, b_0, c_0, d_0) = (v_1, v_2, v_3, v_4)\big) && (4.33)
\end{aligned}$$

where (4.31) follows from Lemma 4.3 with $\hat{u}_i = \hat{F}(a_i, b_i)$ and $\hat{u}^* = \hat{F}(c_k, d_k)$, since the labels $\{\hat{F}(a_i, b_i)\}_{i=0}^{N}$ form a stationary homogeneous Markov chain by Lemma 4.8; (4.32) follows from Lemma 4.8; and (4.33) holds since the initial labels of c_0 and d_0 are chosen independently of the initial labels of a_0, b_0, a_1, b_1. Finally, since the algorithm uses the same procedure to label c_{k+1} and d_{k+1} as it did to label c_1 and d_1, we have

$$\begin{aligned} &\mathbf{P}\big(\hat{F}(c_{k+1}, d_{k+1}) = (v_7, v_8)\big|\hat{F}(a_k, b_k, c_k, d_k, a_{k+1}, b_{k+1}) = (v_1, v_2, v_3, v_4, v_5, v_6)\big) \\ &= \mathbf{P}\big(\hat{F}(c_1, d_1) = (v_7, v_8)\big|\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1) = (v_1, v_2, v_3, v_4, v_5, v_6)\big). \end{aligned} \tag{4.34}$$

Combining (4.28) with (4.30), (4.33), and (4.34) gives

$$\begin{aligned} &\mathbf{P}\big(\hat{F}(a_k, b_k, c_k, d_k, a_{k+1}, b_{k+1}, c_{k+1}, d_{k+1}) = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)\big) \\ &= \mathbf{P}\big(\hat{F}(a_0, b_0, c_0, d_0) = (v_1, v_2, v_3, v_4)\big) \\ &\quad \cdot\mathbf{P}\big(\hat{F}(a_1, b_1) = (v_5, v_6)\big|\hat{F}(a_0, b_0, c_0, d_0) = (v_1, v_2, v_3, v_4)\big) \\ &\quad \cdot\mathbf{P}\big(\hat{F}(c_1, d_1) = (v_7, v_8)\big|\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1) = (v_1, v_2, v_3, v_4, v_5, v_6)\big) \\ &= \mathbf{P}\big(\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1) = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)\big). \end{aligned}$$

This completes the induction argument. □

The elements a_l, b_l, c_l, d_l in the following corollary are illustrated in Figure 4.11.

Corollary 4.11. *Let* $a_l = [0, 0, l]$, $b_l = [0, 1, l]$, $c_l = [1, 0, l]$, $d_l = [1, 1, l]$ *denote the elements of* $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$, *respectively. If the labeling of* $\Delta_{N,N,N}$ *by* $\tilde{\mathcal{E}}^{(3)}$ *satisfies Condition (**b**), then the sequence of labels* $\{\hat{F}(a_l, b_l, c_l, d_l)\}_{l=0}^{N}$ *forms a stationary homogeneous Markov chain.*

Proof. The fact that the sequence $\{\hat{F}(a_l, b_l, c_l, d_l)\}_{l=0}^{N}$ is a Markov chain follows from the definition of the bit stuffing encoder. Lemma 4.10 implies that this Markov chain is stationary and homogeneous. □

Lemma 4.12. *Let $\hat{u}_1, \hat{u}_2, \ldots \hat{u}_m$ and $\hat{u}'_1, \hat{u}'_2, \ldots \hat{u}'_m$ be finite sequences of random variables. If $\left(\hat{u}_i, \hat{u}'_i\right)$ is a Markov chain, and $\hat{u}_i$ is conditionally independent of $\hat{u}'_{i-1}$ given $\hat{u}'_i$, for $i = 1, 2, \ldots, m$, then $\hat{u}'_i$ is a Markov chain.*

Proof. To prove the lemma it suffices to show that the reverse chain $\hat{u}'_m, \ldots, \hat{u}'_1$ is a Markov chain. Let $\hat{u}_i$ take on values from the alphabet A. Let $v \in A$ and $k \in \{2, \ldots, m\}$. Then,

$$\begin{aligned}
&\mathbf{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right) \\
&= \sum_{u_k, \ldots, u_m \in A} \mathbf{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k, \ldots, \hat{u}'_m, \hat{u}_k = u_k, \ldots, \hat{u}_m = u_m\right) \\
&\qquad \cdot \mathbf{P}\left(\hat{u}_k = u_k, \ldots, \hat{u}_m = u_m \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right) && (4.35) \\
&= \sum_{u_k, \ldots, u_m \in A} \mathbf{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k, \hat{u}_k = u_k\right) \cdot \mathbf{P}\left(\hat{u}_k = u_k, \ldots, \hat{u}_m = u_m \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right) && (4.36) \\
&= \sum_{u_k, \ldots, u_m \in A} \mathbf{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k\right) \cdot \mathbf{P}\left(\hat{u}_k = u_k, \ldots, \hat{u}_m = u_m \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right) && (4.37) \\
&= \mathbf{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k\right)
\end{aligned}$$

where the summation in (4.35) is taken over all values $u_k, \ldots, u_m$ such that the conditioning event has positive probability; (4.36) follows since $\left(\hat{u}_i, \hat{u}'_i\right)$ is a Markov chain; and (4.37) follows from the assumption that $\hat{u}_i$ is conditionally independent of $\hat{u}'_{i-1}$ given $\hat{u}_i$, for every $i = 1, 2, \ldots, m$. □

The elements a_l, b_l, c_l, d_l in the following lemma are illustrated in Figure 4.11.

Lemma 4.13. *Let $a_l = [0, 0, l]$, $b_l = [0, 1, l]$, $c_l = [1, 0, l]$, $d_l = [1, 1, l]$ denote the elements of $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$, respectively. If a three-dimensional bit stuffing algorithm $\tilde{\mathcal{E}}^{(3)}$ labeling $\Delta_{N,N,N}$ satisfies Conditions **(a)**, **(b)**, and **(d)**, then the sequence of labels $\left\{\hat{F}(b_l, d_l)\right\}_{l=0}^{N}$ forms a stationary homogeneous Markov chain identical to $\left\{\hat{F}(a_l, c_l)\right\}_{l=0}^{N}$.*

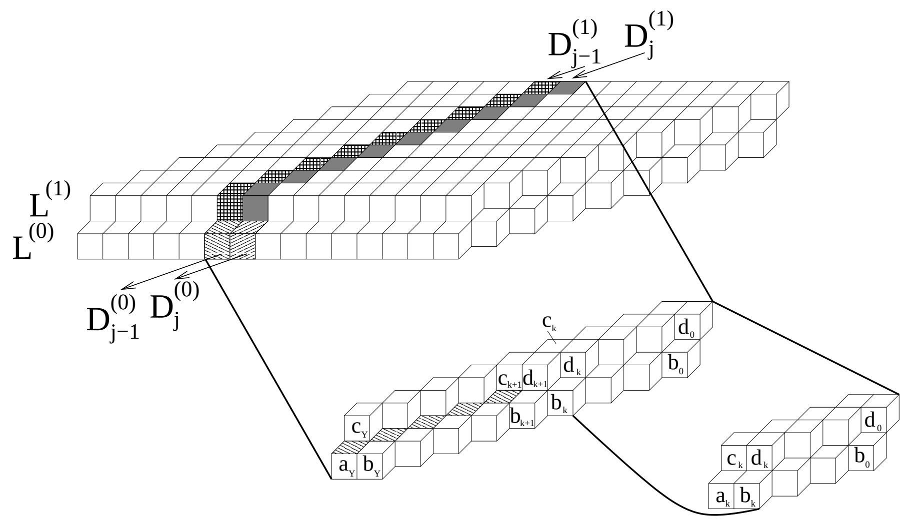


Figure 4.12: The diagonals $D^{(0)}_{j-1}$, $D^{(0)}_{j}$, $D^{(1)}_{j-1}$, $D^{(1)}_{j}$.

Proof. Let $\hat{u}_l = \hat{F}(a_l, c_l)$ and $\hat{u}'_l = \hat{F}(b_l, d_l)$. Corollary 4.11 implies that $\left(\hat{u}_l, \hat{u}'_l\right)$ is a Markov chain. Furthermore, Condition (**d**) and Lemma 4.10 imply that $\hat{u}_l$ is conditionally independent of $\hat{u}'_{l-1}$ given $\hat{u}'_l$. Hence, the conditions of Lemma 4.12 hold for $\hat{u}_l$ and $\hat{u}'_l$, and therefore $\hat{u}'_l$ is a Markov chain. Condition (**a**) and Lemma 4.10 imply that the Markov chain $\left\{\hat{F}(b_l, d_l)\right\}_{l=0}^{N}$ is identical to the Markov chain $\left\{\hat{F}(a_l, c_l)\right\}_{l=0}^{N}$. □

Lemma 4.14 below generalizes Lemma 4.10 to the points of $L^{(0)} \cup L^{(1)}$. The elements a_l, b_l, c_l, d_l in the following lemma are illustrated in Figure 4.12.

Lemma 4.14. *Let $j \in \{1, \ldots, N\}$. Let $a_l = [0, j-1, l]$, $b_l = [0, j, l]$, $c_l = [1, j-1, l]$, $d_l = [1, j, l]$ denote the elements of $D^{(0)}_{j-1}$, $D^{(0)}_{j}$, $D^{(1)}_{j-1}$, $D^{(1)}_{j}$, respectively. If the labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ satisfies Conditions (**a**), (**b**), and (**d**), then for all $l \in \{1, \ldots, N\}$, the joint distribution of $\hat{F}(a_{l-1}, b_{l-1}, c_{l-1}, d_{l-1}, a_l, b_l, c_l, d_l)$ is identical to the joint distribution of $\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1)$.*

Proof. Recall that the random vector $\hat{F}(D^{(i)}_j)$ denotes the labels of the diagonal $D^{(i)}_j$ for $i, j \in \{0, \ldots, N\}$. Since the labels of $L^{(0)}$ form a standard two-dimensional labeling, $\left\{\hat{F}(D^{(0)}_j)\right\}_{j=0}^{N}$ is a stationary homogeneous Markov chain by Lemma 4.9. Using Lemma 4.3 with $\hat{u}_j = \hat{F}(D_j^{(0)})$ and $\hat{u}^* = \hat{F}(D_1^{(1)})$, it follows that $\hat{F}(D_2^{(0)})$ is conditionally independent of $\hat{F}(D_1^{(1)})$ given $\hat{F}(D_1^{(0)})$. Moreover, the joint distribution of $(\hat{F}(D_1^{(0)}), \hat{F}(D_1^{(1)}))$ is identical to the joint distribution of $(\hat{F}(D_0^{(0)}), \hat{F}(D_0^{(1)}))$ by Lemma 4.13. Similarly, the joint distribution of $(\hat{F}(D_1^{(0)}), \hat{F}(D_2^{(0)}))$ is identical to the joint distribution of $(\hat{F}(D_0^{(0)}), \hat{F}(D_1^{(0)}))$ by Lemma 4.9. The above argument implies that the joint distribution of $(\hat{F}(D_1^{(1)}), \hat{F}(D_1^{(0)}), \hat{F}(D_2^{(0)}))$ is identical to the joint distribution of $(\hat{F}(D_0^{(1)}), \hat{F}(D_0^{(0)}), \hat{F}(D_1^{(0)}))$. In other words, when the algorithm labels the diagonal $D_2^{(1)}$, it encounters the same probability distribution on the neighboring diagonals $D_1^{(1)}, D_1^{(0)}, D_2^{(0)}$ as it did on the diagonals $D_0^{(1)}, D_0^{(0)}, D_1^{(0)}$ when labeling $D_1^{(1)}$. Therefore, Lemma 4.10, Corollary 4.11, and Lemma 4.13 hold for the elements of the diagonals $D_1^{(0)}, D_2^{(0)}, D_1^{(1)}, D_2^{(1)}$. Repeating this argument for consecutive diagonals $D_i^{(0)}$, $D_{i+1}^{(0)}$, $D_i^{(1)}$, $D_{i+1}^{(1)}$ (for $i = 2, 3, \ldots, N-1$) generalizes Lemma 4.10 to the elements of $L^{(0)} \cup L^{(1)}$. $\square$

Lemma 4.15 below generalizes Lemma 4.14 to the points of $\Delta_{N,N,N}$. The elements a_l, b_l, c_l, d_l in the following lemma are illustrated in Figure 4.13.

Lemma 4.15. *Let $i, j \in \{1, \ldots, N\}$. Let $a_l = [i-1, j-1, l]$, $b_l = [i-1, j, l]$, $c_l = [i, j-1, l]$, $d_l = [i, j, l]$ denote the elements of $D_{j-1}^{(i-1)}$, $D_j^{(i-1)}$, $D_{j-1}^{(i)}$, $D_j^{(i)}$, respectively. If the labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ satisfies Conditions **(a)**-**(d)**, then for all $l \in \{1, \ldots, N\}$, the joint distribution of $\hat{F}(a_{l-1}, b_{l-1}, c_{l-1}, d_{l-1}, a_l, b_l, c_l, d_l)$ is identical to the joint distribution of $\hat{F}(a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1)$.*

Proof. Let $j \in \{1, \ldots, N\}$. Let $c_k = [1, j-1, k]$ and $d_k = [1, j, k]$ (where $k \in \{0, \ldots, N\}$) denote the elements of $D_{j-1}^{(1)}$ and $D_j^{(1)}$, respectively (see Figure 4.12). It follows from Lemma 4.14 and Condition (**c**) that the labels $\hat{F}(c_{k-1}, d_{k-1}, c_k, d_k)$ have the same probability distribution as the labels $\hat{F}(\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F})$ (where $\mathcal{A} = [0,0,0], \mathcal{B} = [0,1,0], \mathcal{E} = [0,0,1], \mathcal{F} = [0,1,1]$, as in Figure 4.10). Thus the labels of $L^{(1)}$ satisfy Condition 4 of Theorem 4.4. This implies that the probability distribution of the label-

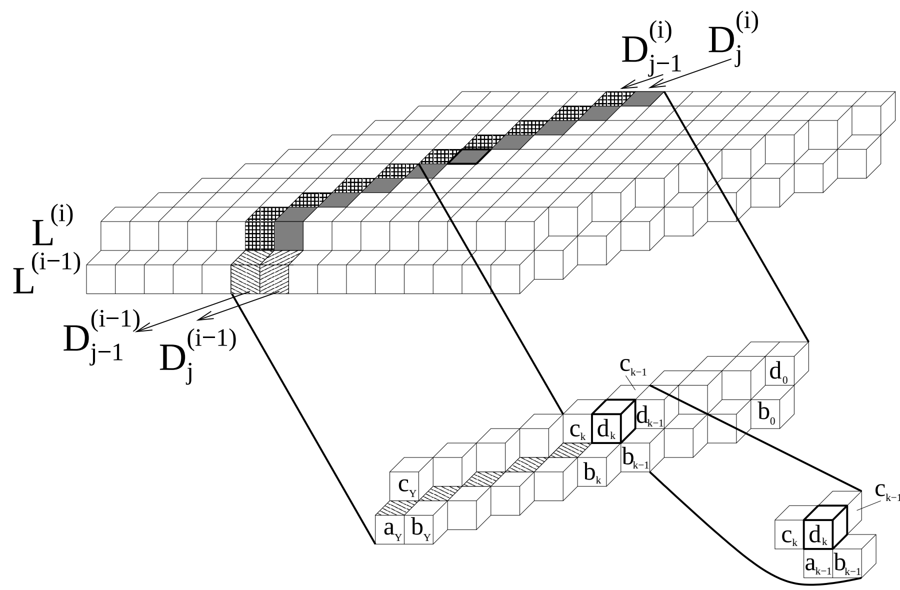


Figure 4.13: The diagonals $D^{(i-1)}_{j-1}$, $D^{(i-1)}_{j}$, $D^{(i)}_{j-1}$, $D^{(i)}_{j}$, the internal point d_k, and its neighbors.

ings of $L^{(1)}$ by the three-dimensional bit stuffing encoder is the same as the probability distribution of a labeling of $L^{(1)}$ generated by $\tilde{\mathcal{E}}^{(3)}$ using a standard three-dimensional initialization and an auxiliary p-sequence $\hat{\sigma}'$ as input (i.e. identical to the probability distribution of the initial labelings of $L^{(0)}$). Therefore, the same arguments used to show Lemma 4.10, Corollary 4.11, Lemma 4.13, and Lemma 4.14 for the elements of $L^{(0)} \cup L^{(1)}$ can be used to prove the same results for the layers $L^{(1)} \cup L^{(2)}, \ldots,$ $L^{(N-1)} \cup L^{(N)}$. $\square$

Remark 4.16. *As noted in the proof of Theorem 4.4, since $D^{(i)}_0 \cap R^{(i)}_0 = \{[i,0,0]\}$, for $i \in \{0,\ldots,N\}$, the stationary probabilities of the Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ must be identical. Let α denote the stationary probability of the state 0. If the labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ satisfies Conditions **(a)**-**(d)**, it follows from Lemma 4.15 and Condition **(c)** that the probability distribution of the labelings of $L^{(i)}$ (for $i \in \{0,\ldots,N\}$) is identical to*

Table 4.7: The numerical values of the parameters that maximize $r^{(3)}$.

p	π_1	π_2	π_3	α
0.255838	0.832393	0.212958	0.787042	0.824431

p_1	p_2	γ_1	γ_2
0.343793	0.255838	0.350456	0.21531

the probability distribution of the labelings of $L^{(0)}$. Hence

$$\mathbf{P}\big(\hat{F}(i,j,k)=0\big)=\alpha=\frac{1}{2-\pi_3}=\frac{1}{2}\left(1+\frac{1-p}{\sqrt{(1+3p)(1-p)}}\right)$$

for every $[i,j,k]\in\Delta_{N,N,N}$.

Theorem 4.17. *If the labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ satisfies Conditions (**a**)-(**d**), then the labeling of $\Delta_{N,N,N}$ is a standard three-dimensional labeling.*

Proof. Consider the points in Figure 4.13 for $i,j,k\in\{1,\ldots,N\}$. Lemma 4.15 implies that the joint probability distribution of $\hat{F}(a_{k-1},c_{k-1},c_k,b_{k-1},b_k)$ is independent of the layer i, the diagonal j, and the position k, which proves the theorem. $\square$

Conditions (**a**)-(**d**) translate into a set of equations for the parameters π_1, π_2, π_3, α, p_1, p_2, p (see Appendix 4.8). Using Condition 3 in Theorem 4.4, (4.19), and (4.38)-(4.41), we expressed the parameters π_1, π_2, π_3, α, p_1, p_2, γ_1, γ_2 in terms of p. Therefore, for π_1, π_2, π_3, α, p_1, p_2, p, γ_1, $\gamma_2\in[0,1]$, that also satisfy Condition 3 in Theorem 4.4, (4.19), and (4.38)-(4.41), it follows that $(p_1,p_2)\in\Omega$, i.e. the corresponding labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ is a standard labeling. Optimization for the achievable coding rate $r^{(3)}=\gamma_1H(p_1)+\gamma_2H(p_2)$ over $p\in\left[\frac{1}{2},1\right]$ gives the numerical results in Table 4.7.

Remark 4.18. *Conditions (**a**)-(**d**) with the additional requirement that $p_1=p_2$ yield the trivial solution $p=p_1=p_2=0$. Thus, to obtain a nontrivial solution, it was necessary* *to use two sequences* $\hat{s}^{(1)}$ *and* $\hat{s}^{(2)}$ *with parameters* $p_1 \neq p_2$.

Theorem 4.19. *The three-dimensional bit stuffing encoder* $\mathcal{E}^{(3)}$ *achieves a coding rate of*

$$r^{(3)} = 0.50200500727$$

which is within 4% of the capacity.

Proof. Substituting the numerical values given in Table 4.7 into the formula $\gamma_1 H(p_1) + \gamma_2 H(p_2)$ determined in Theorem 4.7 gives $r^{(3)} = 0.50200500727$. Using the bounds in (4.1) we get

$$\frac{C_{1,\infty}^{(3)} - r^{(3)}}{C_{1,\infty}^{(3)}} \leq 1 - \frac{0.502005}{0.52250174} < 0.03923 < 4\%.$$

□

4.7 Acknowledgments

The authors thank Ron Roth, Paul Siegel, and Jack Wolf for helpful discussions.

This chapter, in full, has been submitted for publication as: Zs. Nagy and K. Zeger, Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions, *IEEE Trans. Inform. Theory*, November 2002. The dissertation author was the primary investigator of this paper.

Appendix

4.8 Equations corresponding to the three-dimensional bit stuffing algorithm used to obtain Table 4.7

In this appendix we list the equations used to obtain Table 4.7. The equations are given in their initial form without any cancellation of variables. Recall that $\mathcal{A} = [0,0,0]$, $\mathcal{B} = [0,1,0]$, $\mathcal{C} = [1,0,0]$, $\mathcal{D} = [1,1,0]$, $\mathcal{E} = [0,0,1]$, $\mathcal{F} = [0,1,1]$, $\mathcal{G} = [1,0,1]$, $\mathcal{H} = [1,1,1]$ as shown in Figure 4.10.

Corresponding to each valid labeling of $\mathcal{A},\mathcal{C},\mathcal{E},\mathcal{G}$ and $\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H}$ there are 16 equations implied by Condition (**a**). These equations, generated by $\mathbf{P}\big(\hat{F}(\mathcal{A},\mathcal{C},\mathcal{E},\mathcal{G}) = \hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H}) = (v_1,v_2,v_3,v_4)\big)$ are given below for $(v_1,v_2,v_3,v_4) =$

$$
\begin{aligned}
0\,0\,0\,0: \quad & \alpha^2\pi_1^2 = \; 1+\alpha(\pi_3(2-\pi_1 p)-2 \\
& \qquad +\alpha(1+\pi_3(\pi_1-\pi_1(1-p)-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1))) \\
0\,0\,0\,1: \quad & \alpha^2\pi_1(1-\pi_1) = \alpha^2\pi_1\pi_3^2(1-\pi_1 p)p_1 \\
0\,0\,1\,0: \quad & \alpha^2(1-\pi_1)\pi_1 = \alpha\pi_1\pi_3 p(1-\alpha(1-\pi_3(1-\pi_1 p_2))) \\
0\,0\,1\,1: \quad & \alpha^2(1-\pi_1)^2 = \alpha^2\pi_1^2\pi_3^2 p p_2 \\
0\,1\,0\,0: \quad & \alpha\pi_1(1-\alpha)(1-\pi_2) = \alpha(1-\pi_3)(1-\alpha(1-\pi_3(1-\pi_1 p)(1-\pi_1 p_1))) \\
0\,1\,0\,1: \quad & \alpha\pi_1\pi_2(1-\alpha) = \alpha^2\pi_1\pi_3(1-\pi_3)(1-\pi_1 p)p_1 \\
0\,1\,1\,0: \quad & \alpha(1-\alpha)(1-\pi_1)(1-\pi_2) = \alpha^2\pi_1\pi_3(1-\pi_3)p(1-\pi_1 p_2) \\
0\,1\,1\,1: \quad & \alpha\pi_2(1-\alpha)(1-\pi_1) = \alpha^2\pi_1^2\pi_3(1-\pi_3)pp_2 \\
1\,0\,0\,0: \quad & \alpha\pi_1(1-\alpha)(1-\pi_2) = \alpha(1-\pi_3)(1-\pi_1 p)(1-\alpha(1-\pi_3+\pi_1\pi_3 p_2)) \\
1\,0\,0\,1: \quad & \alpha(1-\alpha)(1-\pi_1)(1-\pi_2) = \alpha^2\pi_1\pi_3(1-\pi_3)(1-\pi_1 p)p_2 \\
1\,0\,1\,0: \quad & \alpha\pi_1\pi_2(1-\alpha) = \alpha\pi_1(1-\pi_3)p(1-\alpha(1-\pi_3+\pi_1\pi_3 p_2)) \\
1\,0\,1\,1: \quad & \alpha\pi_2(1-\alpha)(1-\pi_1) = \alpha^2\pi_1^2\pi_3(1-\pi_3)pp_2
\end{aligned}
$$

$$1\,1\,0\,0: \quad (1-\alpha)^2(1-\pi_2)^2 = \alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)$$
$$1\,1\,0\,1: \quad \pi_2(1-\alpha)^2(1-\pi_2) = \alpha^2\pi_1(1-\pi_3)^2(1-\pi_1 p)p_2$$
$$1\,1\,1\,0: \quad \pi_2(1-\alpha)^2(1-\pi_2) = \alpha^2\pi_1(1-\pi_3)^2 p(1-\pi_1 p_2)$$
$$1\,1\,1\,1: \quad \pi_2^2(1-\alpha)^2 = \alpha^2\pi_1^2(1-\pi_3)^2 p p_2$$

Corresponding to each valid labeling of $\mathcal{B}, \mathcal{D}, \mathcal{A}, \mathcal{C}$ and $\mathcal{F}, \mathcal{H}, \mathcal{E}, \mathcal{G}$ there are 9 equations implied by Condition (**b**). These equations, generated by $\mathbf{P}\big(\hat{F}(\mathcal{B}, \mathcal{D}, \mathcal{A}, \mathcal{C}) = \hat{F}(\mathcal{F}, \mathcal{H}, \mathcal{E}, \mathcal{G}) = (v_1, v_2, v_3, v_4)\big)$ are given below for $(v_1, v_2, v_3, v_4) =$

$$0\,0\,0\,0: \quad \alpha^2\pi_3^2 = (1-\pi_2)^2 - \alpha(1-\pi_2)(2-\pi_1-2\pi_2-\pi_1(1-p)) \\ +\alpha^2((1-\pi_2)^2 - \pi_1(1-\pi_2)(2-p) + \pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2)))$$
$$0\,0\,0\,1: \quad \alpha\pi_3(1-\alpha) = (\pi_2+\alpha(1-\pi_1-\pi_2))((1-\alpha)(1-\pi_2)+\alpha\pi_1(1-p))$$
$$0\,0\,1\,0: \quad \alpha\pi_3(1-\alpha) = \pi_2 - \pi_2^2 + \alpha(1-\pi_1-\pi_2)(1-2\pi_2) - \alpha^2(1-\pi_1-2\pi_2+\pi_2^2 \\ +2\pi_1\pi_2 - \pi_1\pi_3(1-p_1) + \pi_1^2\pi_3(1-p_1) - \pi_1(1-\pi_1)(1-\pi_3)(1-p_2))$$
$$0\,0\,1\,1: \quad (1-\alpha)^2 = (\pi_2+\alpha(1-\pi_1-\pi_2))^2$$
$$0\,1\,0\,0: \quad \alpha^2\pi_3(1-\pi_3) = \alpha^2\pi_1^2(1-p)(p_2-\pi_3(p_2-p_1))$$
$$0\,1\,1\,0: \quad \alpha(1-\alpha)(1-\pi_3) = \alpha^2\pi_1(1-\pi_1)(p_2-\pi_3(p_2-p_1))$$
$$1\,0\,0\,0: \quad \alpha^2\pi_3(1-\pi_3) = \alpha\pi_1 p((1-\alpha)(1-\pi_2)+\alpha\pi_1(1-p_2))$$
$$1\,0\,0\,1: \quad \alpha(1-\alpha)(1-\pi_3) = \alpha\pi_1(\pi_2+\alpha(1-\pi_1-\pi_2))p$$
$$1\,1\,0\,0: \quad \alpha^2(1-\pi_3)^2 = \alpha^2\pi_1^2 p p_2$$

Corresponding to the valid labelings of $\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}$ and $\mathcal{C}, \mathcal{D}, \mathcal{G}, \mathcal{H}$ there are 8 equations implied by Condition (**c**). Some equations are tautologies - these are omitted. These equations, generated by $\mathbf{P}\big(\hat{F}(\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}) = \hat{F}(\mathcal{C}, \mathcal{D}, \mathcal{G}, \mathcal{H}) = (v_1, v_2, v_3, v_4)\big)$ are given below for $(v_1, v_2, v_3, v_4) =$

$$0\,0\,0\,0: \quad \alpha\pi_1\pi_3(1-p) = \alpha\pi_1\pi_3(1-\alpha(p_2-\pi_3(1-\pi_1 p)(p_2-p_1)))$$

0 0 0 1 : $\alpha\pi_1\pi_3 p = \alpha^2\pi_1\pi_3(p_2 - \pi_3(1-\pi_1 p)(p_2 - p_1))$

0 0 1 0 :

0 1 0 0 : $\alpha\pi_1(1-\pi_3)(1-p) = \alpha\pi_1(1-\pi_3)(1-\alpha(p_2 - \pi_3(1-\pi_1 p)(p_2-p_1)))$

0 1 0 1 : $\alpha\pi_1(1-\pi_3)p = \alpha^2\pi_1(1-\pi_3)(p_2 - \pi_3(1-\pi_1 p)(p_2-p_1))$

0 1 1 0 :

1 0 0 0 :

1 0 1 0 :

The equations corresponding to Condition (**d**) are of the form

$$\frac{\mathbf{P}\big(\hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H},\mathcal{E},\mathcal{G})=(v_1,v_2,v_3,v_4,v_5,v_6)\big)}{\mathbf{P}\big(\hat{F}(\mathcal{F},\mathcal{H},\mathcal{E},\mathcal{G})=(v_3,v_4,v_5,v_6)\big)}=\frac{\mathbf{P}\big(\hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H})=(v_1,v_2,v_3,v_4)\big)}{\mathbf{P}\big(\hat{F}(\mathcal{F},\mathcal{H})=(v_3,v_4)\big)}$$

where (v_3, v_4, v_5, v_6) is a valid labeling of the points $(\mathcal{F},\mathcal{H},\mathcal{E},\mathcal{G})$. Some equations are tautologies - these are omitted. The list of equations is given below for $(v_1,\, v_2,\, v_3,\, v_4,\, v_5,\, v_6) =$

0 0 0 0 0 0 : $((1-\alpha)(1-\pi_2)(1-\pi_2-\alpha(1-\pi_2-\pi_1\pi_3(2-p)))$

$+\alpha^2\pi_1^2\pi_3^2(1-p)(1-p_1))$

$((1-\pi_2)^2-\alpha(1-\pi_2)(2-\pi_1-2\pi_2-\pi_1(1-p))+\alpha^2((1-\pi_2)^2$

$-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2))))^{-1}$

$$= \frac{1+\alpha(\pi_3(2-\pi_1 p)-2+\alpha(1+\pi_3(\pi_1 p-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1)))}{1+\alpha\pi_1(-p+\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2))}$$

0 0 0 0 0 1 : $\frac{((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3)(-(\alpha-1)(1-\pi_2)+\alpha\pi_1\pi_3(1-p))}{(\alpha(\pi_1+\pi_2-1)-\pi_2)(-(\alpha-1)(1-\pi_2)+\alpha\pi_1(1-p))}$

$$= \frac{1+\alpha(\pi_3(2-\pi_1 p-2)+\alpha(1+\pi_3(\pi_1 p-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1)))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$$

0 0 0 0 1 0 : $$((\alpha-1)\pi_2(\alpha(1+\pi_3-2\pi_1\pi_3)-1)-((\alpha-1)^2\pi_2^2)+$$
$$\alpha(1-\pi_1)\pi_3(1+\alpha(\pi_1\pi_3(1-p_1)-1)))$$
$$(\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2$$
$$+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+\pi_1(1-p_2)(1-\pi_1)(1-\pi_3)-1))^{-1}$$
$$=\frac{1+\alpha(\pi_3(2-\pi_1 p)+\alpha(1+\pi_3(\pi_1 p-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1))-2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$$

0 0 0 0 1 1 : $\frac{(\pi_2(\alpha-1)-\alpha\pi_3(1-\pi_1))^2}{(\pi_2-\alpha(\pi_1+\pi_2-1))^2}=\frac{1+\alpha(\pi_3(2-\pi_1 p)+\alpha(1+\pi_3(\pi_1 p-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1))-2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

0 0 0 1 0 0 : $\frac{-\alpha^2\pi_1^2\pi_3^2(1-p)p_1}{\alpha^2\pi_1^2(1-p)(\pi_3(p_2-p_1)-p_2)}=\frac{-\alpha^2\pi_1\pi_3^2(1-\pi_1 p)p_1}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

0 0 0 1 1 0 : $\frac{-\alpha^2(1-\pi_1)\pi_1\pi_3^2 p_1}{\alpha^2(1-\pi_1)\pi_1(\pi_3(p_2-p_1)-p_2)}=\frac{-\alpha^2\pi_1\pi_3^2(1-\pi_1 p)p_1}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

0 0 1 0 0 0 : $\frac{\alpha\pi_1\pi_3 p((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p_2))}{\alpha\pi_1 p((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p_2))}=\frac{\alpha\pi_1\pi_3 p(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$

0 0 1 0 0 1 : $\frac{\alpha\pi_1\pi_3((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3)p}{\alpha\pi_1(\alpha(\pi_1+\pi_2-1)-\pi_2)p}=\frac{\alpha\pi_1\pi_3 p(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$

0 0 1 1 0 0 :

0 1 0 0 0 0 : $$(\alpha\pi_1(1-\pi_3)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p)(1-p_1)))$$
$$((1-\pi_2)^2+\alpha(1-\pi_2)(2\pi_2+\pi_1(2-p)-2)+\alpha^2((1-\pi_2)^2$$
$$-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2))))^{-1}$$
$$=\frac{-\alpha(1-\pi_3)(1+\alpha(\pi_3(1-\pi_1 p)(1-\pi_1 p_1)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$$

0 1 0 0 0 1 : $\frac{\alpha(1-\pi_1)(1-\pi_3)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p))}{(\alpha(\pi_1+\pi_2-1)-\pi_2)((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p))}=\frac{-\alpha(1-\pi_3)(1+\alpha(\pi_3(1-\pi_1 p)(1-\pi_1 p_1)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

0 1 0 0 1 0 : $$(-\alpha\pi_1(1-\pi_3)((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3(1-p_1)))$$
$$(\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2$$
$$+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+(1-\pi_1)\pi_1(1-\pi_3)(1-p_2)-1))^{-1}$$
$$=\frac{-\alpha(1-\pi_3)(1+\alpha(\pi_3(1-\pi_1 p)(1-\pi_1 p_1)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$$

0 1 0 0 1 1 : $\frac{\alpha(1-\pi_1)(1-\pi_3)((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3)}{(\pi_2-\alpha(\pi_1+\pi_2-1))^2}=\frac{-\alpha(1-\pi_3)(1+\alpha(\pi_3(1-\pi_1 p)(1-\pi_1 p_1)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

0 1 0 1 0 0 : $\frac{\alpha^2\pi_1^2(1-\pi_3)\pi_3(1-p)p_1}{\alpha^2\pi_1^2(1-p)(\pi_3(p_2-p_1)-p_2)}=\frac{\alpha^2\pi_1(1-\pi_3)\pi_3(1-\pi_1 p)p_1}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

0 1 0 1 1 0 : $\frac{\alpha^2(1-\pi_1)\pi_1(1-\pi_3)\pi_3 p_1}{\alpha^2(1-\pi_1)\pi_1(\pi_3(p_2-p_1)-p_2)}=\frac{\alpha^2\pi_1(1-\pi_3)\pi_3(1-\pi_1 p)p_1}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

0 1 1 0 0 0 : $\frac{\alpha^2\pi_1^2(1-\pi_3)\pi_3 p(1-p_2)}{\alpha\pi_1 p((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p_2))}=\frac{-\alpha^2\pi_1(1-\pi_3)\pi_3 p(1-\pi_1 p_2)}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$

0 1 1 0 0 1 : $\frac{\alpha^2(1-\pi_1)\pi_1(1-\pi_3)\pi_3 p}{\alpha\pi_1(\alpha(\pi_1+\pi_2-1)-\pi_2)p}=\frac{-\alpha^2\pi_1(1-\pi_3)\pi_3 p(1-\pi_1 p_2)}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$

0 1 1 1 0 0 :

1 0 0 0 0 0 :
$$
\begin{aligned}
&(\alpha\pi_1(1-\pi_3)(1-p)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p_2))) \\
&((1-\pi_2)^2+\alpha(1-\pi_2)(2\pi_2+\pi_1(2-p)-2)+\alpha^2((1-\pi_2)^2 \\
&-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2))))^{-1} \\
&\qquad = \frac{-\alpha(1-\pi_3)(1-\pi_1 p)(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}
\end{aligned}
$$

1 0 0 0 0 1 : $\frac{\alpha\pi_1(1-\pi_3)((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3)(1-p)}{(\alpha(\pi_1+\pi_2-1)-\pi_2)((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p))} = \frac{-\alpha(1-\pi_3)(1-\pi_1 p)(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

1 0 0 0 1 0 :
$$
\begin{aligned}
&(-\alpha(1-\pi_1)(1-\pi_3)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p_2))) \\
&(\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2 \\
&+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+(1-\pi_1)\pi_1(1-\pi_3)(1-p_2)-1))^{-1} \\
&\qquad = \frac{-\alpha(1-\pi_3)(1-\pi_1 p)(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}
\end{aligned}
$$

1 0 0 0 1 1 : $\frac{\alpha(1-\pi_1)(1-\pi_3)((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3)}{(\pi_2-\alpha(\pi_1+\pi_2-1))^2} = \frac{-\alpha(1-\pi_3)(1-\pi_1 p)(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

1 0 0 1 0 0 : $\frac{\alpha^2\pi_1^2\pi_3(1-p)(1-\pi_3)p_2}{\alpha^2\pi_1^2(1-p)(\pi_3(p_2-p_1)-p_2)} = \frac{\alpha^2\pi_1\pi_3(1-\pi_3)(1-\pi_1 p)p_2}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

1 0 0 1 1 0 : $\frac{\alpha^2\pi_1(1-\pi_1)(1-\pi_3)\pi_3 p_2}{\alpha^2\pi_1(1-\pi_1)(\pi_3(p_2-p_1)-p_2)} = \frac{\alpha^2\pi_1\pi_3(1-\pi_3)(1-\pi_1 p)p_2}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

1 0 1 0 0 0 : $\frac{-\alpha\pi_1(1-\pi_3)p((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p_2))}{\alpha\pi_1 p((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p_2))} = \frac{-\alpha\pi_1(1-\pi_3)p(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$

1 0 1 0 0 1 : $\frac{-\alpha\pi_1(1-\pi_3)((\alpha-1)\pi_2-\alpha\pi_3(1-\pi_1))p}{\alpha\pi_1(\alpha(\pi_1+\pi_2-1)-\pi_2)p} = \frac{-\alpha\pi_1(1-\pi_3)p(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$

1 0 1 1 0 0 :

1 1 0 0 0 0 :
$$
\begin{aligned}
&(\alpha^2\pi_1^2(1-p)(1-p_2)(1-\pi_3)^2) \\
&((1-\pi_2)^2+\alpha(1-\pi_2)(2\pi_2+\pi_1(2-p)-2)+\alpha^2((1-\pi_2)^2 \\
&-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2)))^{-1} \\
&\qquad = \frac{\alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}
\end{aligned}
$$

1 1 0 0 0 1 : $\frac{\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2(1-p)}{(\alpha(\pi_1+\pi_2-1)-\pi_2)((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p))} = \frac{\alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

1 1 0 0 1 0 :
$$
\begin{aligned}
&(\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2(1-p_2)) \\
&(\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2 \\
&+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+(1-\pi_1)\pi_1(1-\pi_3)(1-p_2)-1))^{-1} \\
&\qquad = \frac{\alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}
\end{aligned}
$$

1 1 0 0 1 1 : $\frac{\alpha^2(1-\pi_1)^2(1-\pi_3)^2}{(\pi_2-\alpha(\pi_1+\pi_2-1))^2} = \frac{\alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$

1 1 0 1 0 0 : $\frac{-\alpha^2\pi_1^2(1-\pi_3)^2(1-p)p_2}{\alpha^2\pi_1^2(1-p)(\pi_3(p_2-p_1)-p_2)} = \frac{-\alpha^2\pi_1(1-\pi_3)^2(1-\pi_1 p)p_2}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$

$$\begin{array}{ll}
1\,1\,0\,1\,1\,0: & \frac{-\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2p_2}{\alpha^2\pi_1(1-\pi_1)(\pi_3(p_2-p_1)-p_2)} = \frac{-\alpha^2\pi_1(1-\pi_3)^2(1-\pi_1p)p_2}{\alpha^2\pi_1(1-\pi_1p)(\pi_3(p_2-p_1)-p_2)} \\
1\,1\,1\,0\,0\,0: & \frac{-\alpha^2\pi_1^2(1-\pi_3)^2p(1-p_2)}{\alpha\pi_1p((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p_2))} = \frac{\alpha^2\pi_1(1-\pi_3)^2p(1-\pi_1p_2)}{\alpha\pi_1p(1-\alpha\pi_1p_2)} \\
1\,1\,1\,0\,0\,1: & \frac{-\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2p}{\alpha\pi_1(\alpha(\pi_1+\pi_2-1)-\pi_2)p} = \frac{\alpha^2\pi_1(1-\pi_3)^2p(1-\pi_1p_2)}{\alpha\pi_1p(1-\alpha\pi_1p_2)} \\
1\,1\,1\,1\,0\,0: &
\end{array}$$

Substituting from Theorem 4.4 and from (4.19) the quantities

$$\pi_1 = \frac{2}{1+p+\sqrt{(1+3p)(1-p)}}, \qquad \pi_2 = \frac{2p}{1+p+\sqrt{(1+3p)(1-p)}}, \qquad \pi_3 = \frac{2(1-p)}{1-p+\sqrt{(1+3p)(1-p)}},$$
$$\alpha = \frac{1}{2}\left(1+\frac{1-p}{\sqrt{(1+3p)(1-p)}}\right).$$

The above equations corresponding to Conditions (**a**)-(**d**) reduce to at most two independent equations. A set of independent equations we used to express p_1 and p_2 is

$$\frac{2-(1-p)^2(1+p_1)+(2+(1-p)(1+p_1))\sqrt{(1+3p)(1-p)}}{1+p+\sqrt{(1+3p)(1-p)}} = 1 \quad (4.38)$$
$$\frac{1-p-2(1-p)^2+2(1-p)(1-p_2)+\sqrt{(1+3p)(1-p)}}{1-p+\sqrt{(1+3p)(1-p)}} = 1. \quad (4.39)$$

The parameters γ_1 and γ_2 are given in terms of the other parameters as

$$\gamma_1 = \alpha^2\pi_1\pi_3(1-\pi_1p) \quad (4.40)$$
$$\gamma_2 = \alpha^2\pi_1(1-\pi_3(1-\pi_1p)). \quad (4.41)$$

References

[1] J. J. Ashley and B. H. Marcus. Two-Dimensional Low-Pass Filtering Codes. *IEEE Trans. on Communications*, 46:724–727, June 1998.

[2] P. Bender and J. K. Wolf. A Universal Algorithm for Generating Optimal and Nearly Optimal Run-Length-Limited, Charge Constrained Binary Sequences. *Proc. 1993 IEEE Intl. Symp. on Inform. Theory*, page 6, San Antonio, TX, January 1993.

[3] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York, 1991.

[5] K. Engel. On the Fibonacci Number of an $M \times N$ Lattice. *Fibonacci Quarterly*, 28:72–78, 1990.

[6] T. Etzion. Cascading methods for runlength-limited arrays. *IEEE Trans. Inform. Theory*, 43:319–324, January 1997.

[7] S. Forchhammer. Bounds on the Capacity of Constrained Three-Dimensional Codes. *Preprint*, 2002.

[8] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf. Improved Bit-Stuffing Bounds on 2-Dimensional Constraints. *Proc. 2002 IEEE Intl. Symp. on Inform. Theory*, page 385, Lausanne, Switzerland, June-July 2002.

[9] K. A. S. Immink, P. H. Siegel, and J. K. Wolf. Codes for Digital Recorders. *IEEE Trans. Inform. Theory*, 44:2260–2299, October 1998.

[10] H. Ito, A. Kato, Zs. Nagy, and K. Zeger. Zero Capacity Region of Multidimensional Run Length Constraints. *Elec. J. Combinatorics*, 6(1)(R33), 1999.

[11] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[12] P. Lee. Combined Error-Correcting/Modulation Recording Codes. *Ph.D. dissertation*, University of California, San Diego, 1988.

[13] Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel. *IEEE Trans. Inform. Theory*, 46(3):1030–1033, May 2000.

[14] Zs. Nagy and K. Zeger. Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints. *IEEE Trans. Inform. Theory* (submitted July 2002).

[15] E. Ordentlich and R. M. Roth. Two-Dimensional Weight-Constrained Codes through Enumeration Bounds. *IEEE Trans. Inform. Theory*, 46(4):1292–1301, July 2000.

[16] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding for a Two-Dimensional Runlength-Limited Constraint. *Proc. of the SPIE - The Intl. Soc. for Optical Engineering*, Denver, CO, 3802:8–17, July 1999.

[17] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding Schemes for the Hard-Square Model. *IEEE Trans. Inform. Theory*, 47:1166–1176, March 2001.

[18] C. E. Shannon. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27(3 and 4):379–423 and 623–656, July and October 1948.

[19] P. H. Siegel and J. K. Wolf. Bit Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays. *Proc. 1998 IEEE Intl. Symp. on Inform. Theory*, page 323, MIT, Cambridge, MA, August 1998.

[20] R. E. Swanson and J. K. Wolf. A New Class of Two-Dimensional Runlength-Limited Recording Codes. *IEEE Trans. on Mag. Recording*, 28:3407–3416, November 1992.

[21] R. Talyansky, T. Etzion, and R. M. Roth. Efficient Code Construction for Certain Two-Dimensional Constraints. *IEEE Trans. Inform. Theory*, 45(2):794–799, March 1999.

[22] A. Vardy, M. Blaum, P. H. Siegel, and G. T. Sincerbox. Conservative Arrays: Multidimensional Modulation Codes for Holographic Recording. *IEEE Trans. Inform. Theory*, 42(1):227–230, January 1996.

[23] W. Weeks and R. E. Blahut. The Capacity and Coding Gain of Certain Checkerboard Codes. *IEEE Trans. Inform. Theory*, 44(3):1193–1203, May 1998.

Chapter 5

Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints

Abstract

A checkerboard constraint is a bounded measurable set $S \subset \mathbf{R}^2$, containing the origin. A binary labeling of the $\mathbf{Z}^2$ lattice satisfies the checkerboard constraint S if whenever $t \in \mathbf{Z}^2$ is labeled 1, all of the other $\mathbf{Z}^2$-lattice points in the translate $t + S$ are labeled 0. Two-dimensional channels that only allow labelings of $\mathbf{Z}^2$ satisfying checkerboard constraints are studied. Let $A(S)$ be the area of S, and let $A(S) \to \infty$ mean that S retains its shape but is inflated in size in the form αS as $\alpha \to \infty$. It is shown that for any open checkerboard constraint S, there exist positive reals K_1 and K_2 such that as $A(S) \to \infty$, the channel capacity C_S decays to zero at least as fast as $(K_1 \log_2 A(S))/A(S)$ and at most as fast as $(K_2 \log_2 A(S))/A(S)$. It is also shown that if S is an open convex and symmetric checkerboard constraint, then as $A(S) \to \infty$, the capacity decays exactly at the rate $4\delta(S)(\log_2 A(S))/A(S)$, where $\delta(S)$ is the packing density of the set S. An implication is that the capacity of such checkerboard constrained channels is asymptotically determined only by the areas of the constraint and

the smallest hexagon that can be circumscribed about the constraint. In particular, this establishes that channels with square, diamond, or hexagonal checkerboard constraints all asymptotically have the same capacity, since $\delta(S) = 1$ for such constraints.

5.1 Introduction

One-dimensional channels satisfying run length constraints are important in magnetic recording applications and two-dimensional channels satisfying run length constraints have been considered in relation to optical recording applications (see the references in [14]). One-dimensional (d, k) run length constraints require that in any binary sequence, the number of consecutive 0s be at most k, and between any two 1s in the sequence be at least d 0s. Two-dimensional run length constraints require that one-dimensional run length constraints be satisfied both horizontally and vertically in a two-dimensional rectangular binary array.

An important special two-dimensional channel is one satisfying the (d, ∞) run length constraint. In two dimensions, the $(1, \infty)$ constraint, for example, has been studied in terms of computing the channel capacity [4], [7] and for efficient coding algorithms [21], [22]. The capacity of the $(1, \infty)$ constrained channel is not known exactly, but has been very accurately upper and lower bounded.

If a two-dimensional (d, ∞) run length constraint is further constrained along one diagonal direction to similarly only allow (d, ∞) constrained sequences (e.g. in the northwest-southeast direction as shown in Figure 5.1f), then this is equivalent to a channel that allows binary labeled patterns of a hexagonal grid (as opposed to a rectangular grid) such that a (d, ∞) run length constraint must be met along the three natural axes of the hexagonal grid. A complicated non-rigorous[1] derivation of the capacity for the case

[1] Baxter comments on his derivation[2, p. 409]: "It is not mathematically rigorous, in that certain

$d = 1$ (known as the ”hard hexagon model”) was given in [3], from which an analytic expression for the capacity was presented in [15], [19], and [26].

Various interpretations of two-dimensional run length constrained capacities appear in other fields of study. For example, the two-dimensional $(1, \infty)$ capacity is equal to the growth rate (as $N \to \infty$) of the number of configurations of mutually non-attacking princes on an $N \times N$ chessboard, where a “prince” acts as a chess piece that can move to any square that shares an edge with its current location. Likewise, the analytic capacity in [15], [19], [26] gives the growth rate of the number of configurations of mutually non-attacking princes on a hexagonal chessboard. The growth rates of the number of certain configurations of mutually non-attacking chess pieces on an $N \times N$ chessboard have been extensively studied (e.g. for kings, in [18], [30]). The capacity calculations in [4] were formulated in terms of counting independent sets of vertices in graphs. The capacities are also closely related to gases, lattices, and Ising model entropies in statistical mechanics [2].

In addition to run length constraints, other types of constraints can be used to model two-dimensional channels for certain applications [1], [8], [9], [10], [12], [13], [23], [24], [25], [27], [28]. For example, run length constraints along diagonals in both directions (northwest-southeast and northeast-southwest) can be imposed, in addition to horizontal and vertical constraints. An example of a circularly symmetric two-dimensional constraint occurs by requiring that any point in the two-dimensional $\mathbf{Z}^2$ lattice be labeled 0 if it is within a prescribed distance from a lattice point with label 1. In other words, each 1 must be surrounded by a certain circle of 0s.

One could alternatively require that every 1 be surrounded by 0s falling in a given sized hexagon, square, or more generally any other shape of interest. In general, a large class of such two-dimensional constraints can be characterized by some bounded

---

analyticity properties of κ are assumed, and the results of Chapter 13 (which depend on assuming that various large-lattice limits can be interchanged) are used. However, I believe that these assumptions, and therefore (14.1.18)-(14.1.24), are in fact correct.”

measurable two-dimensional set S, and the requirement that for every 1 stored in the plane, it must at least be surrounded by a set of 0s arranged in the shape of S. Such a code is said to satisfy the constraint S. These constraints are known as checkerboard constraints [29]. Two-dimensional (d, ∞) constraints are examples of checkerboard constraints, in which case the set S is the union of the intervals $[-d, d]$ on both the horizontal and vertical axes in the plane (i.e. a "+" shape). Likewise the hexagonal-grid constraint studied in [2] is a checkerboard constraint.

In this paper we focus on the asymptotic behavior of the capacity of two-dimensional channels satisfying checkerboard constraints. In the special case of the two-dimensional (d, ∞) run length constrained channel, the asymptotic behavior of the capacity is well understood. It was shown in [16] that the capacity decays to zero at the exact rate $(\log_2 d)/d$ as $d \to \infty$. For a general checkerboard constraint, the asymptotics analogous to run length constraints are when the constraint S retains its shape but is inflated in size in the form αS as $\alpha \to \infty$.

As α goes to infinity, the amount of information that can be stored per unit area shrinks to zero. In other words the capacity decays to zero. We determine the rate at which the capacity goes to zero, as a function of the area $A(S)$ of the constraint, for certain classes of checkerboard constraints. If the checkerboard constraint S is assumed to be open, then we show (Theorem 5.18) that as $A(S) \to \infty$, the capacity decays to zero at a rate bounded between $(K_1 \log_2 A(S))/A(S)$ and $(K_2 \log_2 A(S))/A(S)$, for some positive finite constants K_1 and K_2. If the checkerboard constraint S is additionally assumed to be convex and symmetric, then we show (Theorem 5.16) that as $A(S) \to \infty$, the capacity decays to zero at the rate $4\delta(S)(\log_2 A(S))/A(S)$, where $\delta(S)$ is the packing density of the set S. Thus, for example, since the packing density (in the plane) of squares or hexagons is $\delta(S) = 1$, this implies that the capacity of two-dimensional channels satisfying square or hexagon checkerboard constraints is asymptotically equal to $4(\log_2 A(S))/A(S)$ as the area grows without bound. Similarly, if S is a circular constraint, then the asymptotic capacity is $\frac{2\pi}{\sqrt{3}}(\log_2 A(S))/A(S)$ since $\delta(S) = \pi/(2\sqrt{3})$.

Since the constraint S corresponding to a two-dimensional (d, ∞) run length constraint is neither open nor convex, the results in this paper do not specialize to the (d, ∞) constraint case, but they do provide an interesting related checkerboard constraint result.

5.2 Preliminaries

Let $\mathbf{R}^2$ denote the two-dimensional plane. A two-dimensional *lattice* is a set $T \subset \mathbf{R}^2$ of the form $T = \{\kappa u + \lambda v : \kappa, \lambda \in \mathbf{Z}\}$ where $u, v \in \mathbf{R}^2$ are independent. In particular, $\mathbf{Z}^2$ denotes the two-dimensional integer lattice. Given a set $S \subset \mathbf{R}^2$, any function $f : S \cap \mathbf{Z}^2 \to \{0, 1\}$ is called a *labeling* of S. For any set $S \subset \mathbf{R}^2$, let $A(S)$ be the area of S and let $\Lambda(S) = |S \cap \mathbf{Z}^2|$ be the number of $\mathbf{Z}^2$-lattice points contained in S.

A set $S \subset \mathbf{R}^2$ is *symmetric* if $x \in S \Leftrightarrow -x \in S$. For any $S \subset \mathbf{R}^2$, $y \in \mathbf{R}^2$, and $\alpha \in \mathbf{R}$ let $S + y = \{x + y : x \in S\}$ and $\alpha S = \{\alpha x : x \in S\}$. Also, for sets $S, T \subset \mathbf{R}^2$ let $S + T = \{x + y : x \in S, y \in T\}$. The closure of S is denoted by $\bar{S}$.

For any $a \in \mathbf{R}^2$ and $b \in \mathbf{R}$, the set $l = \{x \in \mathbf{R}^2 : (a \cdot x) + b = 0\}$ is a *line*, where $a \cdot x$ is the dot product of a and x. A line l is a *supporting line* to the set $S \subset \mathbf{R}^2$ if $l \cap \bar{S} \neq \emptyset$ and one of the two closed halfplanes determined by l contains S.

Let $R, S, T \subset \mathbf{R}^2$. For each $t \in T$, the set $S + t$ is called a T-*translate* of S. The set $T \subset \mathbf{R}^2$ is called an

- S-*packing* of R if the interiors of the T-translates are disjoint and are contained in R;
- S-*covering* of R if the union of the closures of the T-translates contains R;
- S-*tiling* of R if it is both an S-packing and an S-covering of R.

A *rectangle* is any set $R_{(\kappa,\lambda)}^{(\mu,\nu)} = \{(x,y) \in \mathbf{R}^2 : \kappa \le x \le \mu, \lambda \le y \le \nu\}$ for some $\kappa, \lambda, \mu, \nu \in \mathbf{R}$.

The following definitions are from [20]. Let $S, T \subset \mathbf{R}^2$ and define

$$\begin{aligned} \rho_+(S,T) &= \limsup_{\kappa,\lambda,\mu,\nu\to\infty} \rho(S,T,\kappa,\lambda,\mu,\nu) \\ \rho_-(S,T) &= \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \rho(S,T,\kappa,\lambda,\mu,\nu) \end{aligned}$$

where

$$\rho(S,T,\kappa,\lambda,\mu,\nu) = \frac{\sum_{t\in T} A\left((S+t)\cap R_{(-\kappa,-\lambda)}^{(\mu,\nu)}\right)}{A\left(R_{(-\kappa,-\lambda)}^{(\mu,\nu)}\right)}.$$

The *packing density* of S is

$$\delta(S) = \sup_T \rho_+(S,T) \tag{5.1}$$

where the supremum is taken over all S-packings T of $\mathbf{R}^2$, and the *covering density* of S is

$$\theta(S) = \inf_T \rho_-(S,T) \tag{5.2}$$

where the infimum is taken over all S-coverings T of $\mathbf{R}^2$.

The following lemma states that the densest packing and the sparsest covering using convex symmetric sets, are attained by a lattice packing and lattice covering, respectively.

Lemma 5.1. *[20, pp. 12,17] For every convex symmetric set $S \subset \mathbf{R}^2$ there exist lattices L_1 and L_2 such that $\delta(S) = \rho_+(S, L_1)$ and $\theta(S) = \rho_-(S, L_2)$.*

A two-dimensional *constrained channel* is a set of labelings of $\mathbf{R}^2$. Such labelings are called *valid*. A constraint is a description of which labelings are valid for a particular constrained channel. A *checkerboard constraint* is a bounded measurable set $S \subset \mathbf{R}^2$ that contains the origin. The terminology "checkerboard constraint" was introduced in [29] to mean a "two-dimensional arrangement of *zeros* that must surround every *one* in a two-dimensional binary code", which is consistent with the present definition. It was noted in [29]:

> "For example, in two-dimensional optical recording systems bits may be stored on media in the form of dark or bright patterns. As the storage "disk" is read, these patterns pass through various lenses and other image-forming devices, thus producing intersymbol interference (ISI). Checkerboard constraints will reduce this ISI, so naturally we wish to analyze such constraints."

Given a set $V \subset \mathbf{R}^2$ and a checkerboard constraint S, a labeling f of V is said to be S-*valid* on V if $f(y) = 0$ whenever $f(x) = 1$, for all $x \in V \cap \mathbf{Z}^2$ and $y \in (x + S) \cap (V \setminus \{x\}) \cap \mathbf{Z}^2$. That is, f satisfies the checkerboard constraint S on the set $V \subset \mathbf{R}^2$. Note that any S-valid labeling of a subset of $\mathbf{R}^2$ can be extended to an S-valid labeling of $\mathbf{R}^2$ by making the labeling equal 0 outside of the subset. The number of S-valid labelings of a set $V \subset \mathbf{R}^2$ is denoted by $N_S(V)$. The *capacity* C_S corresponding to the checkerboard constraint S is

$$C_S = \lim_{\kappa,\lambda,\mu,\nu \to \infty} \frac{\log_2 N_S\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)}. \tag{5.3}$$

A proof given in [16] shows that the above limit exists.

An example of a checkerboard constraint is a run length constraint. For each non-negative integer d, the two-dimensional (d, ∞) *run length constraint* is defined as

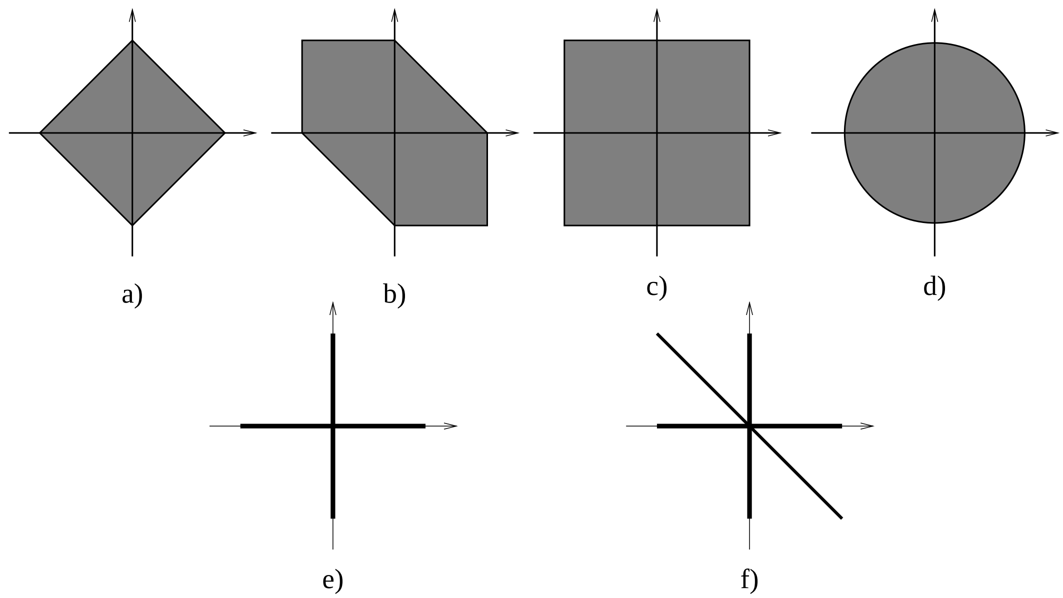


Figure 5.1: Various checkerboard constraints: a) diamond; b) hexagon; c) square; d) circle; e) (d, ∞) run length; and e) (d, ∞) hexagonal-grid run length.

the following subset of $\mathbf{R}^2$:

$$S_{d,\infty} = \{(0, x) : -d \le x \le d\} \cup \{(x, 0) : -d \le x \le d\} \tag{5.4}$$

The capacities of various channels satisfying convex checkerboard constraints were studied in [29]. These included the diamond, hexagonal, square, and (d, ∞) run length checkerboard constraints, and are shown in Figure 5.1.

Lemma 5.2. *Let $V \subset \mathbf{R}^2$, let S be a checkerboard constraint, and let f be a labeling of V. If f is S-valid then f is $-S$-valid.*

Proof. Suppose f is not $-S$-valid. Then there exist $x, y \in V$ such that $f(x) = f(y) = 1$ and $y \in x + (-S)$. This implies that $x \in y + S$ and therefore f is not S-valid. $\square$

Corollary 5.3. *Let $V \subset \mathbf{R}^2$, S be a checkerboard constraint, and f be a labeling of V. Then f is S-valid if and only if f is $(S \cup -S)$-valid.*

Corollary 5.3 follows immediately from Lemma 5.2. It follows from Corollary 5.3 that every checkerboard constraint S is equivalent to the symmetric checkerboard constraint $S \cup -S$ in the sense that the sets of S-valid labelings and $(S \cup -S)$-valid labelings of any set $V \subset \mathbf{R}^2$ are identical. That is, any non-symmetric checkerboard constraint is also a symmetric checkerboard constraint. Therefore $N_S(V) = N_{S\cup -S}(V)$ for any set $V \subset \mathbf{R}^2$, establishing Corollary 5.4 below. Thus no generality is lost if we restrict attention to symmetric checkerboard constraints when computing capacities.

Corollary 5.4. *If S is a checkerboard constraint then $C_S = C_{S\cup -S}$.*

Lemma 5.5. *If S is a convex symmetric checkerboard constraint which is either open or closed, and k is a positive integer, then $\frac{1}{2}S \subset \frac{k+1}{2}S + u_1 + \ldots + u_k$ for any $u_1, \ldots, u_k \in \frac{1}{2}\bar{S}$.*

Proof. Let $y \in \frac{1}{2}S$. Then $(k+1)y \in \frac{k+1}{2}S$ and $(k+1)u_i \in \frac{k+1}{2}\bar{S}$ for $i = 1, \ldots, k$. Since S is symmetric, $-(k+1)u_i \in \frac{k+1}{2}\bar{S}$. The quantity $y - \sum_{i=1}^{k} u_i$ is a convex combination of the points $(k+1)y, -(k+1)u_1, \ldots, -(k+1)u_k$ and $\frac{k+1}{2}S$ is a convex set. If S is open then y lies in the interior of $\frac{1}{2}S$ and thus also in the interior of $\frac{k+1}{2}S$. Therefore, by convexity, $y - \sum_{i=1}^{k} u_i \in \frac{k+1}{2}S$ (see [17, p. 111, Theorem 5]). If S is closed then $y - \sum_{i=1}^{k} u_i \in \frac{k+1}{2}\bar{S} = \frac{k+1}{2}S$. In both cases, $y = (y - \sum_{i=1}^{k} u_i) + \sum_{i=1}^{k} u_i \in \frac{k+1}{2}S + \sum_{i=1}^{k} u_i$. $\square$

Lemma 5.6. *Let S be a convex symmetric checkerboard constraint which is either open or closed. For any S-valid labeling f of $\mathbf{R}^2$, any set $Q \subset \frac{1}{2}S$, and every $w \in \mathbf{R}^2$, the set $Q + w$ can not contain more than one $\mathbf{Z}^2$-lattice point with label 1.*

Proof. Suppose to the contrary that there exist $\mathbf{Z}^2$-lattice points $x, y \in Q + w \subset \frac{1}{2}S + w$ such that $f(x) = f(y) = 1$. Then $x - w, y - w \in \frac{1}{2}S$. Taking $k = 1$ in Lemma 5.5 implies that $x - w \in S + y - w$, and therefore $x \in S + y$, which contradicts the assumption that f is S-valid. $\square$

Remark 5.7. *Suppose f is a valid labeling. In the special case where the set of $\mathbf{Z}^2$-lattice points with label 1 forms a lattice, Lemma 5.6 follows from Minkowski's Convex Body Theorem [5, pp. 71-72.]* [2].

Lemma 5.8. *Let S be an open convex symmetric checkerboard constraint, and let f be a labeling of $\mathbf{R}^2$. Then f is S-valid if and only if the set $f^{-1}(1)$ is a $\frac{1}{2}S$-packing of $\mathbf{R}^2$.*

Proof. Suppose f is not S-valid. Then there exist distinct $x, y \in f^{-1}(1)$ such that $y \in S+x$. Since S contains the origin, $y \in \frac{1}{2}S+y$. If $y \in \frac{1}{2}S+x$, then $(\frac{1}{2}S+x)\cap(\frac{1}{2}S+y) \neq \emptyset$ which implies $f^{-1}(1)$ is not a $\frac{1}{2}S$-packing, since S is open. So assume $y \notin \frac{1}{2}S + x$ and likewise $x \notin \frac{1}{2}S + y$. Let $l_0 = \{tx + (1-t)y : t \in [0,1]\}$ denote the line segment between the points x and y.

Since S is convex and x lies in the interior of $\frac{1}{2}S+x$, the line segment l_0 intersects the boundary of $\frac{1}{2}S + x$ in exactly one point, say r_1 (see [17, p. 112, Theorem 9]). Similarly, let r_2 be the point where l_0 intersects the boundary of $\frac{1}{2}S+y$. By the symmetry of S, one gets $r_2 = -r_1 + x + y$, and therefore $|r_1 - x| = |r_2 - y|$. Since $x, y \in S + x$ and S is convex, we have $l_0 \subset S + x$. Let r_3 be the unique point on the extension of l_0 beyond y, that intersects the boundary of the set $S + x$. Since S is symmetric and since the line segment connecting x to r_3 is contained in $S + x$, we have $|l_0| < |x - r_3| = 2|x - r_1| = |r_1 - x| + |r_2 - y|$. Consequently, r_1 is between the points r_2 and y on l_0, and hence all points of l_0 between r_1 and r_2 are contained in $(\frac{1}{2}S+x)\cap(\frac{1}{2}S+y)$. Thus, $f^{-1}(1)$ is not a $\frac{1}{2}S$-packing.

Now suppose that $f^{-1}(1)$ is not a $\frac{1}{2}S$-packing of $\mathbf{R}^2$. Then there exist $x, y \in f^{-1}(1)$ such that $(\frac{1}{2}S + x) \cap (\frac{1}{2}S + y) \neq \emptyset$. If $y \in \frac{1}{2}S + x$, then f is not S-valid, so assume $y \notin \frac{1}{2}S + x$ (and likewise $x \notin \frac{1}{2}S + y$) and let l_0, r_1, and r_2 be defined as above. Since $\frac{1}{2}S + x$ is convex, there exists a supporting line l_1 at the point r_1 to the set $\frac{1}{2}S + x$

[2] *There is a typographical error in the last line of the statement of the corollary in [5]. It should read "whose difference $\frac{1}{2}x_1 - \frac{1}{2}x_2$ is in Λ."*

(see [17, p. 143, Corollary 6]). Similarly, by symmetry, $\frac{1}{2}S + y$ has a supporting line $l_2 = -l_1 + x + y$ at the point r_2. The lines l_1 and l_2 are parallel. Let P_1 denote the closed halfplane defined by l_1 that contains $\frac{1}{2}S + x$, and let P_2 denote the closed halfplane defined by l_2 that contains $\frac{1}{2}S + y$. Then $l_1 \subset P_2$, $l_2 \subset P_1$, and $l_1 \neq l_2$ for otherwise $\frac{1}{2}S + x$ and $\frac{1}{2}S + y$ would be disjoint. Since $r_1 \in l_1$ and $r_2 \in l_2$, it follows that r_1 is between the points r_2 and y on l_0, and therefore $|l_0| < |r_1 - x| + |r_2 - y| = 2|r_1 - x|$. This implies that $l_0 \subset S + x$, and hence $y \in S + x$. Thus f is not S-valid. □

5.3 Hexagonal checkerboard constraints

By a *hexagon* we mean any convex 6-sided polygon, where it is possible that more than two vertices are colinear. A checkerboard constraint is *hexagonal* if it is an open, convex, symmetric hexagon. An open regular hexagon is an example of a hexagonal checkerboard constraint. By the definition of a hexagon, the diamond and square constraints shown in Figure 5.1a,c are also considered hexagonal checkerboard constraints.

Notation: Let U be the set of all checkerboard constraints and let $f : U \to \mathbf{R}$. For any $S \in U$ and $L \in \mathbf{R}$, we write $\lim_{A(S)\to\infty} f(S) = L$ to mean that $\lim_{\alpha\to\infty} f(\alpha S) = L$. That is, the set S is inflated without bound by the factor α but retains the same shape.

Theorem 5.9. *If H is a hexagonal checkerboard constraint with capacity C_H and area $A(H)$, then*

$$\lim_{A(H)\to\infty} C_H \cdot \frac{A(H)}{\log_2 A(H)} = 4.$$

Proof. It follows immediately from Lemma 5.11 and Lemma 5.12 below. □

The proof of following lemma is an easy exercise left to the reader.

Lemma 5.10. *If H is a hexagonal checkerboard constraint, then there is a lattice H-*

tiling of the plane.

Lemma 5.11. *If H is a hexagonal checkerboard constraint, then*

$$\limsup_{A(H)\to\infty} C_H \cdot \frac{A\left(\frac{1}{2}H\right)}{\log_2 A\left(\frac{1}{2}H\right)} \leq 1.$$

Proof. Let $\beta \in (0,1)$, and for each $\alpha > 0$ let $H_\alpha = \alpha H$. By Lemma 5.10, there exists a $\frac{1}{2}(1-\beta)\bar{H}_\alpha$-tiling T of $\mathbf{R}^2$. The set T depends on α and β. Since $\frac{1}{2}(1-\beta)\bar{H}_\alpha \subset \frac{1}{2}H_\alpha$ and H is open, Lemma 5.6 implies that for all $t \in T$ and for each H_α-valid labeling of $\mathbf{R}^2$, at most one $\mathbf{Z}^2$-lattice point in $t + \frac{1}{2}(1-\beta)\bar{H}_\alpha$ has label 1. The number of H_α-valid labelings of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ is upper bounded if we independently assign an H_α-valid labeling to the $\mathbf{Z}^2$-lattice points in each of the closed translates $t + \frac{1}{2}(1-\beta)\bar{H}_\alpha$ that intersects $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ (labelings of boundary points of translates may be over counted). For each $\alpha > 0$, let

$$\epsilon_\alpha \quad = \quad \sup_{t\in T} \frac{\Lambda\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha + t\right)}{A\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha\right)} - 1.$$

Different translates of $\frac{1}{2}(1-\beta)\bar{H}_\alpha$ from the tiling T may contain different numbers of $\mathbf{Z}^2$-lattice points, but $\epsilon_\alpha \to 0$ as $\alpha \to \infty$. From the definition of ϵ_α, we have for all $t \in T$,

$$\Lambda\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha + t\right) \quad \leq \quad A\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha\right)(1+\epsilon_\alpha). \tag{5.5}$$

Define the sets

$$\begin{aligned} T_i &= \left\{t \in T : \frac{1}{2}(1-\beta)\bar{H}_\alpha + t \subset R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right\} \\ T_b &= \left\{t \in T : \emptyset \neq \left(\frac{1}{2}(1-\beta)\bar{H}_\alpha + t\right) \cap R^{(\mu,\nu)}_{(-\kappa,-\lambda)} \neq \frac{1}{2}(1-\beta)\bar{H}_\alpha + t\right\} \end{aligned}$$

and denote their cardinalities as

$$\begin{aligned} n &= |T_i| \\ m &= |T_b|. \end{aligned}$$

The integers n and m count the number of translates of $\frac{1}{2}(1-\beta)\bar{H}_\alpha$ from the tiling T that are contained in the rectangle or partially intersect the rectangle, respectively.

Since for any distinct $t_1, t_2 \in T_i$, the sets $\frac{1}{2}(1-\beta)H_\alpha + t_1$ and $\frac{1}{2}(1-\beta)H_\alpha + t_2$ are disjoint, we get the lower bound

$$\begin{aligned} A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right) &\geq \sum_{t\in T_i} A\left(\frac{1}{2}(1-\beta)H_\alpha + t\right) \\ &= n \cdot A\left(\frac{1}{2}(1-\beta)H_\alpha\right) \\ &= n \cdot A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2. \end{aligned} \tag{5.6}$$

Since H is open, Lemma 5.6 implies that at most one $\mathbf{Z}^2$-lattice point in $\frac{1}{2}(1-\beta)\bar{H}_\alpha$ can be labeled 1 in an H_α-valid labeling. By independently choosing at most one $\mathbf{Z}^2$-lattice point to be labeled with a 1 in each of the $m+n$ T-translates of $\frac{1}{2}(1-\beta)\bar{H}_\alpha$ that intersect $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$, we obtain an upper bound on the number of H_α-valid labelings of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$, namely

$$\begin{aligned} N_{H_\alpha}\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right) &\leq \prod_{t\in T_i\cup T_b}\left(\Lambda\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha + t\right)+1\right) \\ &\leq \sup_{t\in T}\left(\Lambda\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha + t\right)+1\right)^{|T_i\cup T_b|} \\ &\leq \left(A\left(\frac{1}{2}(1-\beta)\bar{H}_\alpha\right)(1+\epsilon_\alpha)+1\right)^{m+n} \qquad (5.7) \\ &= \left(A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)+1\right)^{m+n} \qquad (5.8) \end{aligned}$$

where (5.7) follows from (5.5); and (5.8) follows from $A\left(\frac{1}{2}H\right) = A\left(\frac{1}{2}\bar{H}\right)$. Using (5.3), the lower bound in (5.6), and the upper bound in (5.8), the capacity of the checkerboard constraint H_α is upper bounded as

$$
\begin{aligned}
C_{H_\alpha} &\le \lim_{\kappa,\lambda,\mu,\nu\to\infty} \frac{\log_2\left(A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)+1\right)^{m+n}}{nA\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2} && (5.9)\\
&= \left(\frac{\log_2 A\left(\frac{1}{2}H_\alpha\right)}{A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2} + \frac{\log_2\left((1-\beta)^2(1+\epsilon_\alpha)+\frac{1}{A\left(\frac{1}{2}H_\alpha\right)}\right)}{A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2}\right)\\
&\quad \cdot \lim_{\kappa,\lambda,\mu,\nu\to\infty}\left(1+\frac{m}{n}\right) && (5.10)\\
&= \frac{\log_2 A\left(\frac{1}{2}H_\alpha\right)}{A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2} + \frac{\log_2\left((1-\beta)^2(1+\epsilon_\alpha)+\frac{1}{A\left(\frac{1}{2}H_\alpha\right)}\right)}{A\left(\frac{1}{2}H_\alpha\right)(1-\beta)^2} && (5.11)
\end{aligned}
$$

where the existence of the limit in (5.9) follows from the existence of the limit in (5.10); and (5.11) follows from the fact that $m/n \to 0$ as $\kappa, \lambda, \mu, \nu \to \infty$. Since $\epsilon_\alpha \to 0$ and $A\left(\frac{1}{2}H_\alpha\right) \to \infty$ as $\alpha \to \infty$, it follows that

$$\limsup_{\alpha\to\infty} C_{H_\alpha} \cdot \frac{A\left(\frac{1}{2}H_\alpha\right)}{\log_2 A\left(\frac{1}{2}H_\alpha\right)} \le \frac{1}{1-\beta^2}.$$

Since β was chosen arbitrarily from the interval $(0,1)$ we have

$$\limsup_{\alpha\to\infty} C_{H_\alpha} \cdot \frac{A\left(\frac{1}{2}H_\alpha\right)}{\log_2 A\left(\frac{1}{2}H_\alpha\right)} \le \inf_{\beta\in(0,1)} \frac{1}{1-\beta^2} = 1.$$

$\square$

In order to establish a lower bound on C_H, we design a labeling algorithm for $\mathbf{R}^2$. We again consider translates of inflated copies of $\frac{1}{2}H$ that tile $\mathbf{R}^2$, and assign labels to the $\mathbf{Z}^2$-lattice points in a block of $\gamma \times \gamma$ neighboring scaled copies of $\frac{1}{2}H$ in the tiling.

Lemma 5.12. *If H is a hexagonal checkerboard constraint, then*

$$\liminf_{A(H)\to\infty} C_H \cdot \frac{A\left(\frac{1}{2}H\right)}{\log_2 A\left(\frac{1}{2}H\right)} \geq 1.$$

Proof. For each $\alpha > 0$ let $H_\alpha = \alpha H$, and define

$$\epsilon_\alpha = \inf_{t\in\mathbf{R}^2} \frac{\Lambda\left(\frac{1}{2}H_\alpha + t\right)}{A\left(\frac{1}{2}H_\alpha\right)} - 1.$$

By Lemma 5.10, one can tile $\mathbf{R}^2$ with copies of $\frac{1}{2}H_\alpha$ on a lattice. Let $x, y \in \mathbf{R}^2$ be independent vectors such that the lattice $T = \{ix + jy : i, j \in \mathbf{Z}\}$ is a $\frac{1}{2}H_\alpha$-tiling. The lattice T depends on α. For each positive odd integer γ, define the sets

$$\begin{aligned} T_\gamma &= \left\{ix + jy : -\frac{\gamma-1}{2} \leq i, j \leq \frac{\gamma-1}{2}\right\} \\ B_\gamma &= \bigcup_{z\in T_\gamma} \left(z + \frac{1}{2}H_\alpha\right) \end{aligned}$$

and define $\sigma = 1/(2\gamma)$.

Note that each translate $\frac{1}{2}H_\alpha + z$ (where $z \in T_\gamma$) can be written in the form

$$\frac{1}{2}H_\alpha + z = \frac{1}{2}H_\alpha + ix + jy = \frac{1}{2}H_\alpha + \sum_{l=1}^{2\gamma-2} u_l,$$

where $-\frac{\gamma-1}{2} \leq i, j \leq \frac{\gamma-1}{2}$, $u_1, \ldots, u_{2i} = \frac{x}{2}$, $u_{2i+1}, \ldots, u_{2i+2j} = \frac{y}{2}$, and $u_l = 0$ for all $l > 2i + 2j$. Thus $u_l \in \{\pm\frac{1}{2}x, \pm\frac{1}{2}y, 0\}$ for $l = 1, \ldots, 2\gamma - 2$. Since $\pm\frac{1}{2}x$ and $\pm\frac{1}{2}y$ lie on the boundary of $\frac{1}{2}H_\alpha$, we have $\pm\frac{1}{2}x, \pm\frac{1}{2}y \in \frac{1}{2}\bar{H}_\alpha$ (and $0 \in \frac{1}{2}\bar{H}_\alpha$). Therefore by Lemma 5.5, we have $\frac{1}{2}H_\alpha \subset \frac{2\gamma-1}{2}H_\alpha - \sum_{l=1}^{2\gamma-2} u_l$, and therefore $\frac{1}{2}H_\alpha + \sum_{l=1}^{2\gamma-2} u_l \subset \frac{2\gamma-1}{2}H_\alpha \subset \frac{2\gamma}{2}H_\alpha$. Thus $B_\gamma \subset \frac{2\gamma}{2}H_\alpha$ and hence $\sigma B_\gamma \subset \frac{1}{2}H_\alpha$.

For all $z, w \in T_\gamma$, define the hexagonal *minicell* $D_{z,w} = z + \sigma\left(w + \frac{1}{2}H_\alpha\right)$. The union $\cup_{w\in T_\gamma}\sigma\left(w + \frac{1}{2}H_\alpha\right)$ is equal to σB_γ which is contained in $\frac{1}{2}H_\alpha$. Thus, for each

$z \in T_\gamma$, the minicell $D_{z,z}$ lies inside the hexagonal *cell* $z + \frac{1}{2}H_\alpha$, and is in the same relative position within the $\gamma \times \gamma$ block $z + \sigma B_\gamma$ of minicells in the cell $z + \frac{1}{2}H_\alpha$, as is the position of the cell $z + \frac{1}{2}H_\alpha$ within the $\gamma \times \gamma$ block of cells B_γ. The vector w determines the position within $z + \frac{1}{2}H_\alpha$ that the minicell lies.

Let f be a labeling of B_γ defined as follows. For each $z \in T_\gamma$, label exactly one $\mathbf{Z}^2$-lattice point in the minicell $D_{z,z}$ with a 1 and label all other $\mathbf{Z}^2$-lattice points in $D_{z,z}$ with a 0. For each $w, z \in T_\gamma$, if $w \neq z$, then label all the $\mathbf{Z}^2$-lattice points in the minicell $D_{z,w}$ with a 0. Label all other $\mathbf{Z}^2$-lattice points with a 0, if they are not in a minicell (i.e. all $\mathbf{Z}^2$-lattice points in $(z + \frac{1}{2}H_\alpha) \setminus \bigcup_{w \in T_\gamma} D_{z,w}$, for each $z \in T_\gamma$).

So exactly one $\mathbf{Z}^2$-lattice point in each of the γ^2 cells is labeled 1 and all others are labeled 0. Each such labeling is an H_α-valid labeling of B_γ. In addition, f can be extended to $B_{\gamma+2}$ by labeling every $\mathbf{Z}^2$-lattice point in $B_{\gamma+2} \setminus B_\gamma$ with a 0. Then f is an H_α-valid labeling of $B_{\gamma+2}$. Figure 5.3 illustrates an example labeling.

The total number of such labelings f of B_γ is a lower bound to the total number of H_α-valid labelings of $B_{\gamma+2}$. That is,

$$\begin{aligned}
N_{H_\alpha}(B_{\gamma+2}) &\geq N_{H_\alpha}(B_\gamma) \\
&\geq \prod_{z \in T_\gamma} \Lambda\left(D_{z,z}\right) \\
&\geq \left(\inf_{z \in T} \Lambda\left(D_{z,z}\right)\right)^{|T_\gamma|} \\
&\geq \left(\inf_{z,w \in T} \Lambda\left((z + \sigma w) + \frac{\sigma}{2}H_\alpha\right)\right)^{\gamma^2} \\
&\geq \left(A\left(\frac{\sigma}{2}H_\alpha\right)(1 + \epsilon_{\alpha\sigma})\right)^{\gamma^2} \\
&= \left(A\left(\frac{1}{2}H_\alpha\right)\sigma^2(1 + \epsilon_{\alpha\sigma})\right)^{\gamma^2} \qquad (5.12)
\end{aligned}$$

where we used that fact that $\frac{\sigma}{2}H_\alpha = \frac{1}{2}H_{\alpha\sigma}$. For every $w \in (\gamma + 2)T$, let f_w be any such H_α-valid labeling of $B_{\gamma+2} \cap \mathbf{Z}^2$ and assume its value is 0 elsewhere on $\mathbf{Z}^2$. Then

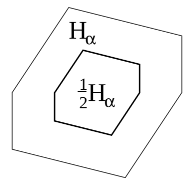


Figure 5.2: The checkerboard constraint H_α and its scaled version $\frac{1}{2}H_\alpha$.

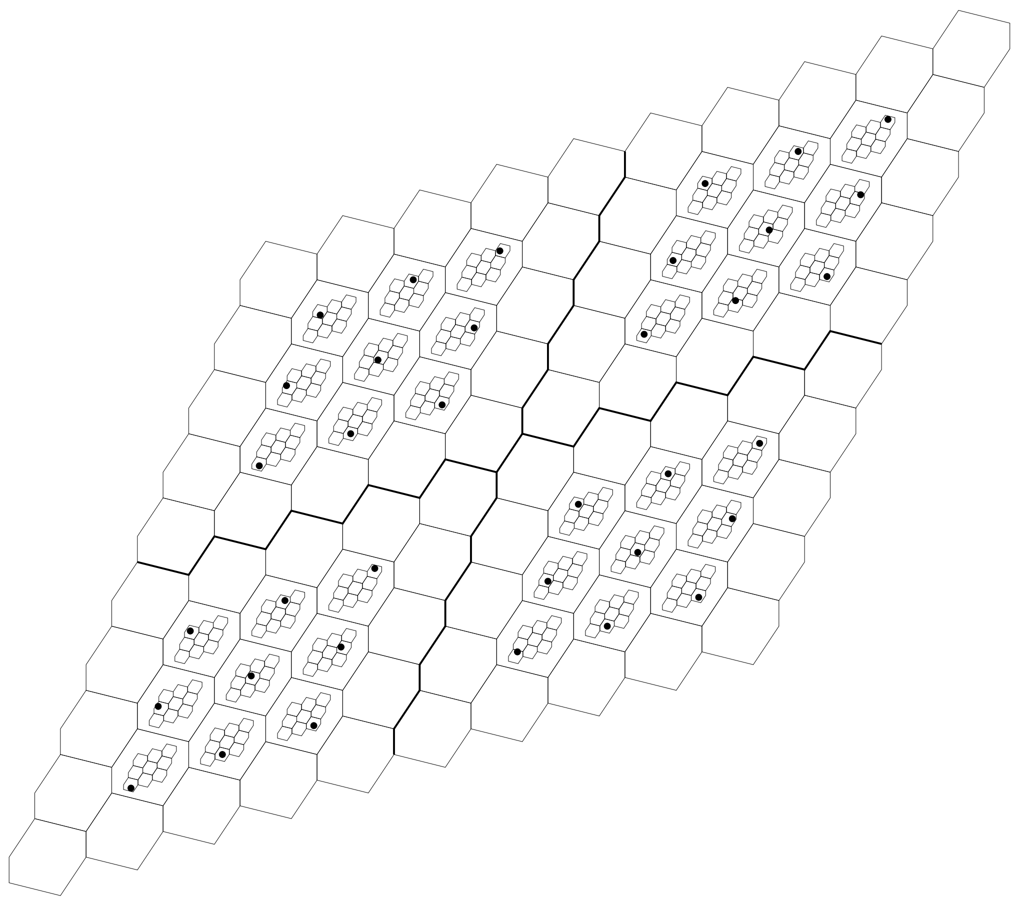


Figure 5.3: Four $(\gamma + 2) \times (\gamma + 2)$ blocks of translates (cells) of $\frac{1}{2}H_\alpha$ for $\gamma = 3$. In each block, outermost row on each of the four sides has padding cells filled with only 0s. Each non-padding cell has a $\gamma \times \gamma$ block of minicells in it. Of all the $\mathbf{Z}^2$-lattice points in each minicell, only the darkened $\mathbf{Z}^2$-lattice points have label 1 in the illustrated labeling. Repeating this construction gives an H_α-valid labeling of $\mathbf{Z}^2$.

an extension to an H_α-valid labeling f of all of $\mathbf{Z}^2$ can be defined by

$$f(z) = \sum_{w \in (\gamma+2)T} f_w(z - (\gamma + 2)w).$$

Although the capacity of a checkerboard constraint is defined in (5.3) as a limit as a rectangle grows in size, it is straightforward to show that the limit can also be taken over a set such as $B_{\gamma+2}$, as γ grows without bound. Thus, since

$$A\left(B_{\gamma+2}\right) = (\gamma+2)^2 A\left(\frac{1}{2}H_\alpha\right) \tag{5.13}$$

the capacity can be lower bounded using (5.12) and (5.13) as

$$\begin{aligned} C_{H_\alpha} &\geq \frac{\log_2 N_{H_\alpha}(B_{\gamma+2})}{A\left(B_{\gamma+2}\right)} \\ &\geq \frac{\log_2 \left(A\left(\frac{1}{2}H_\alpha\right)\sigma^2(1+\epsilon_{\alpha\sigma})\right)^{\gamma^2}}{(\gamma+2)^2 A\left(\frac{1}{2}H_\alpha\right)} \\ &= \left(\frac{\gamma}{\gamma+2}\right)^2 \left(\frac{\log_2 A\left(\frac{1}{2}H_\alpha\right)}{A\left(\frac{1}{2}H_\alpha\right)} + \frac{\log_2(\sigma^2(1+\epsilon_{\alpha\sigma}))}{A\left(\frac{1}{2}H_\alpha\right)}\right). \end{aligned}$$

For each α, choose $\gamma = \lfloor \log_2 \alpha \rfloor$. Then as $\alpha \to \infty$, both $\frac{\gamma}{\gamma+2} \to 1$ and $\alpha\sigma = \alpha/(2\gamma) \geq \alpha/(2\log_2 \alpha) \to \infty$. Thus $\epsilon_{\alpha\sigma} \to 0$ as $\alpha \to \infty$. Since $\frac{-\log_2 \sigma^2}{\log_2 A\left(\frac{1}{2}H_\alpha\right)} = \frac{-2\log_2 \sigma}{\log_2(\alpha^2 A\left(\frac{1}{2}H\right))} = \frac{2\log_2(2\gamma)}{2\log_2 \alpha + \log_2 A\left(\frac{1}{2}H\right)} \leq \frac{2+2\log_2\log_2 \alpha}{2\log_2 \alpha + \log_2 A\left(\frac{1}{2}H\right)} \to 0$ as $\alpha \to \infty$, we get

$$\liminf_{\alpha\to\infty} C_{H_\alpha} \cdot \frac{A\left(\frac{1}{2}H_\alpha\right)}{\log_2 A\left(\frac{1}{2}H_\alpha\right)} \quad \geq \quad 1.$$

$\square$

5.4 Open convex symmetric checkerboard constraints

In this section we generalize Theorem 5.9 to any open convex symmetric checkerboard constraint. The following lemma guarantees that among all minimal area hexagons containing a given convex symmetric set, at least one is itself also convex and symmetric.

Lemma 5.13. *[6, p. 122] Let $S \subset \mathbf{R}^2$ be a convex symmetric set. Then there exists a hexagon containing S that is of minimal area, symmetric, and convex.*

The following lemma shows that the packing density of a convex symmetric set is achieved by a symmetric circumscribed hexagon of minimal area.

Lemma 5.14. *[20, p. 12] Let $S \subset \mathbf{R}^2$ be a convex symmetric set and let H be a minimal area symmetric hexagon that contains S. Then $\delta(S) = A\left(S\right)/A\left(H\right)$.*

Lemma 5.15. *[11, p. 163] Let R be a convex hexagon and $S \subset \mathbf{R}^2$ a convex set. The cardinality of any S-packing of R is at most $A\left(R\right)/A\left(H\right)$, where H is a hexagon of least possible area containing S.*

Note that for $\alpha > 0$, if H_α is a minimal area symmetric hexagon that contains αS, then the ratio $A\left(\alpha S\right)/A\left(H_\alpha\right)$ is a constant, independent of α. Thus if the term $\delta(S)$ appears inside a limit as $A\left(S\right) \to \infty$, then the $\delta(S)$ can be brought outside the limit. This fact is used in the proof of Theorem 5.16 below.

Theorem 5.16. *If S is an open convex symmetric checkerboard constraint with area $A\left(S\right)$, capacity C_S, and packing density $\delta(S)$, then*

$$\lim_{A(S)\to\infty} C_S \cdot \frac{A\left(S\right)}{\log_2 A\left(S\right)} = 4\delta(S). \tag{5.14}$$

Proof. Let H be a symmetric (by Lemma 5.13) hexagon containing S, of minimal area

$A(H)$. Then

$$\frac{1}{\delta(S)} \cdot \lim_{A(S)\to\infty} C_H \cdot \frac{A(S)}{\log_2 A(S)} = \lim_{A(H)\to\infty} C_H \cdot \frac{A(H)}{\log_2 A(H)} = 4 \tag{5.15}$$

where the second limit exists by Theorem 5.9 and therefore the first limit exists by Lemma 5.14. Since $S \subset H$, we have $C_S \geq C_H$ and therefore

$$\frac{1}{4\delta(S)} \cdot \liminf_{A(S)\to\infty} C_S \cdot \frac{A(S)}{\log_2 A(S)} \geq 1.$$

Since $A\left(\frac{1}{2}S\right) = \frac{1}{4}A(S)$, in order to prove the theorem it suffices to show that

$$\frac{1}{\delta(S)} \cdot \limsup_{A(S)\to\infty} C_S \cdot \frac{A\left(\frac{1}{2}S\right)}{\log_2 A\left(\frac{1}{2}S\right)} \leq 1. \tag{5.16}$$

Let $\beta \in (0,1)$, and for each $\alpha > 0$, let $S_\alpha = \alpha S$. We prove (5.16) by upper bounding the number of S_α-valid labelings of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$. Let p be the maximum number of $\mathbf{Z}^2$-lattice points that can be labeled 1 on $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ without violating the checkerboard constraint S_α. By Lemma 5.1, there exists a $\frac{1}{2}(1-\beta)\bar{S}_\alpha$-covering T of $\mathbf{R}^2$ that attains $\theta(S)$. Let $T' = \{t \in T : (\frac{1}{2}(1-\beta)\bar{S}_\alpha + t) \cap R^{(\mu,\nu)}_{(-\kappa,-\lambda)} \neq \emptyset\}$ and $q = |T'|$. The sets T and T' depend on both α and β, and the quantities p and q are both functions of κ, λ, μ, ν, and S_α. For every $\alpha > 0$, define

$$\epsilon_\alpha \;=\; \sup_{t\in\mathbf{R}^2} \frac{\Lambda\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha + t\right)}{A\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha\right)} - 1$$

and note that $\epsilon_\alpha \to 0$ as $\alpha \to \infty$. Also, for all $\alpha > 0$ and $t \in T$,

$$\begin{aligned}\Lambda\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha + t\right) &\leq (1+\epsilon_\alpha)A\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha\right)\\ &= (1+\epsilon_\alpha)(1-\beta)^2 A\left(\frac{1}{2}S_\alpha\right).\end{aligned} \tag{5.17}$$

Since S is open and $\frac{1}{2}(1-\beta)\bar{S}_\alpha \subset \frac{1}{2}S_\alpha$, Lemma 5.6 implies that each of the q copies of $\frac{1}{2}(1-\beta)\bar{S}_\alpha$ intersecting $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ can contain at most one $\mathbf{Z}^2$-lattice point with label 1 in any S_α-valid labeling of $\mathbf{R}^2$. Thus, $p \le q$.

The number of S_α-valid labelings of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ can be upper bounded by considering all possible collections of i of the T'-translates of $\frac{1}{2}(1-\beta)\bar{S}_\alpha$, for $i = 0, \ldots, p$, and assuming that each such translate has exactly one point labeled 1 and no other translate has any points labeled 1. This counts every S_α-valid labeling at least once. Since different collections of i of the T'-translates might yield the same set of i points being labeled 1, some S_α-valid labelings may be counted more than once in this manner. Thus,

$$\begin{aligned} N_{S_\alpha}\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right) &\le \sum_{i=0}^{p} \sum_{\substack{\hat{T}\subset T' \\ |\hat{T}|=i}} \prod_{t\in\hat{T}} \Lambda\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha + t\right) \\ &\le \sum_{i=0}^{p} \binom{q}{i} \left(\sup_{t\in\mathbf{R}^2} \Lambda\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha + t\right)\right)^i \\ &\le \sum_{i=0}^{p} \binom{q}{i} \left(A\left(\frac{1}{2}S_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)\right)^i \qquad (5.18) \\ &\le \left(A\left(\frac{1}{2}S_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)\right)^p \sum_{i=0}^{p} \binom{q}{i} \\ &\le \left(A\left(\frac{1}{2}S_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)\right)^p 2^q \qquad (5.19) \end{aligned}$$

where (5.18) follows from (5.17).

Lemma 5.8 implies that in any S_α-valid labeling, the $\mathbf{Z}^2$-lattice points with label 1 in $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ are a $\frac{1}{2}S_\alpha$-packing of $\mathbf{R}^2$. By the definition of p, there exists an S_α-valid labeling of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ with exactly p points labeled 1. For the $\frac{1}{2}S_\alpha$-packing of $\mathbf{R}^2$ determined by the points labeled 1 in this particular labeling, let p_i denote the number of translates of $\frac{1}{2}S_\alpha$ that lie inside the boundary of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$. Then,

$$\begin{aligned}
\liminf_{\kappa,\lambda,\mu,\nu\to\infty} \frac{pA\left(\frac{1}{2}S_\alpha\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} &= \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \left(\frac{p}{p_i}\right) p_i \cdot \frac{A\left(\frac{1}{2}S_\alpha\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \\
&\leq \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \frac{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)}{A\left(\frac{1}{2}H_\alpha\right)} \cdot \frac{A\left(\frac{1}{2}S_\alpha\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \qquad (5.20) \\
&= \frac{A\left(\frac{1}{2}S_\alpha\right)}{A\left(\frac{1}{2}H_\alpha\right)} = \delta(S_\alpha) = \delta(S), \qquad (5.21)
\end{aligned}$$

where (5.20) follows from Lemma 5.15 (since the rectangle $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$ is a convex hexagon), and the fact that $p/p_i \to 1$ as $\kappa, \lambda, \mu, \nu \to \infty$; and (5.21) follows from Lemma 5.14.

Let $T'' = \{t \in T' : \frac{1}{2}(1-\beta)\bar{S}_\alpha + t \subset R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\}$ and $q_i = |T''|$. The quantity q_i denotes the number of T'-translates of $\frac{1}{2}(1-\beta)\bar{S}_\alpha$ that lie inside the boundary of $R^{(\mu,\nu)}_{(-\kappa,-\lambda)}$. The $\frac{1}{2}(1-\beta)\bar{S}_\alpha$-covering T of $\mathbf{R}^2$ satisfies

$$\begin{aligned}
\liminf_{\kappa,\lambda,\mu,\nu\to\infty} \frac{qA\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} &= \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \left(\frac{q}{q_i}\right) \frac{q_i A\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \\
&= \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \frac{\sum_{t\in T''} A\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha + t\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \qquad (5.22) \\
&\leq \lim_{\kappa,\lambda,\mu,\nu\to\infty} \frac{\sum_{t\in T} A\left(\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha + t\right) \cap R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \\
&= \theta\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha\right) \qquad (5.23) \\
&= \theta(S) \qquad (5.24)
\end{aligned}$$

where (5.22) follows from the fact that $q/q_i \to 1$ as $\kappa, \lambda, \mu, \nu \to \infty$; and (5.23) follows from (5.2). Combining (5.3),(5.19), (5.21), and (5.24), the capacity is bounded as

$$\begin{aligned}
C_{S_\alpha} &\leq \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \frac{\log_2\left(\left(A\left(\frac{1}{2}S_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)\right)^p 2^q\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \\
&= \liminf_{\kappa,\lambda,\mu,\nu\to\infty} \frac{p\log_2\left(A\left(\frac{1}{2}S_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)\right)}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} + \frac{q}{A\left(R^{(\mu,\nu)}_{(-\kappa,-\lambda)}\right)} \\
&\leq \frac{\log_2\left(A\left(\frac{1}{2}S_\alpha\right)(1-\beta)^2(1+\epsilon_\alpha)\right)}{A\left(\frac{1}{2}S_\alpha\right)} \cdot \delta(S) + \frac{1}{A\left(\frac{1}{2}(1-\beta)\bar{S}_\alpha\right)} \cdot \theta(S).
\end{aligned}$$

Thus,

$$\begin{aligned}
\limsup_{\alpha\to\infty} C_{S_\alpha} \cdot \frac{A\left(\frac{1}{2}S_\alpha\right)}{\log_2 A\left(\frac{1}{2}S_\alpha\right)} &\leq \delta(S)\cdot\limsup_{\alpha\to\infty}\left(1+\frac{\log_2\left((1-\beta)^2(1+\epsilon_\alpha)\right)}{\log_2 A\left(\frac{1}{2}S_\alpha\right)}\right) \\
&\quad +\theta(S)\cdot\limsup_{\alpha\to\infty}\frac{1}{(1-\beta)^2\log_2 A\left(\frac{1}{2}S_\alpha\right)} \\
&= \delta(S).
\end{aligned}$$

□

5.5 Arbitrary checkerboard constraints

For a given checkerboard constraint S, the area $A(S)$ was grown without bound in Theorem 5.16 to obtain convergence rates for the capacity of channels constrained by S. As S grows, the area of S becomes approximately equal to the number of $\mathbf{Z}^2$-lattice points in S, in the sense that their ratio approaches 1. A larger class of constrained channels may be examined by relaxing the requirement that a constraining set be open and have nonempty interior. However, the area of such a set may be zero, in which case it is more useful to identify the number of internal $\mathbf{Z}^2$-lattice points.

The following corollary restates Theorem 5.16 in terms of the number of $\mathbf{Z}^2$-lattice points in a constraint instead of the area of a constraint, since both are equal asymptotically as the constraint grows in size.

Corollary 5.17. *If S is an open convex symmetric checkerboard constraint, then*

$$\lim_{\Lambda(S)\to\infty} C_S \cdot \frac{\Lambda(S)}{\log_2 \Lambda(S)} = 4\delta(S)$$

where $\delta(S)$ is the packing density of S.

Proof. It follows immediately from Theorem 5.16 and the fact that $\lim_{A(S)\to\infty} A(S)/\Lambda(S) = 1$. □

The (d,∞) constraint $S_{d,\infty}$ defined in (5.4) is a checkerboard constraint but it is neither convex nor open, two properties which were used to obtain Corollary 5.17. Furthermore, $\delta(S_{d,\infty}) = 0$. However a similar result is still true. It is known [16] that the capacity[3] $C_{d,\infty}$ of the two-dimensional (d,∞) run length constrained channel asymptotically decays to zero at the rate $(\log_2 d)/d$. That is,

$$\lim_{d\to\infty} C_{d,\infty} \cdot \frac{d}{\log_2 d} = 1. \tag{5.25}$$

Since $\Lambda(S_{d,\infty}) = 4d+1$ for all d, the asymptotic capacity in (5.25) can be written as

$$\lim_{\Lambda(S_{d,\infty})\to\infty} C_{d,\infty} \cdot \frac{\Lambda(S_{d,\infty})}{\log_2 \Lambda(S_{d,\infty})} = 4$$

which is similar in form to Corollary 5.17, but is for the non-convex and non-open constraint $S_{d,\infty}$.

In fact, a more general rate of convergence can be obtained for the capacity of two-dimensional channels with checkerboard constraints whose interior contains the origin, but without exactly identifying the convergence constant. Such constraints are not necessarily convex. The capacity is shown in Theorem 5.18 below to still decay

---

[3] The more common notation $C_{d,\infty}$ is used here, instead of the more cumbersome $C_{S_{d,\infty}}$.

asymptotically at the rate $(\log A(S))/A(S)$ in these cases. Theorem 5.18 makes precise a prediction given in [29]: "Intuitively, we expect that the capacity of a given constraint will be inversely proportional to the number of *zeros* in the constraint."

Theorem 5.18. *If S is a checkerboard constraint whose interior contains the origin, then*

$$0 < \liminf_{A(S)\to\infty} C_S \cdot \frac{A(S)}{\log_2 A(S)} \le \limsup_{A(S)\to\infty} C_S \cdot \frac{A(S)}{\log_2 A(S)} < \infty.$$

Proof. Since the origin lies in the interior of S, there is an open regular hexagon R contained in S and whose center is the origin. Since S is bounded it is contained in an open regular hexagon Q whose center is the origin. R and Q are hexagonal checkerboard constraints with packing densities $\delta(R) = \delta(Q) = 1$. Since $R \subset S \subset Q$, we have $1 \le A(R) \le A(S) \le A(Q) < \infty$ and $C_Q \le C_S \le C_R$. Thus, by Theorem 5.9,

$$\begin{aligned} 4 &= \lim_{\alpha\to\infty} C_Q \cdot \frac{A(\alpha Q)}{\log_2 A(\alpha Q)} \le \liminf_{\alpha\to\infty} C_S \cdot \frac{A(\alpha Q)}{\log_2 A(\alpha Q)} \\ &= A(Q) \cdot \liminf_{\alpha\to\infty} C_S \cdot \frac{\alpha^2}{\log_2 \alpha^2} = \frac{A(Q)}{A(S)} \cdot \liminf_{\alpha\to\infty} C_S \cdot \frac{A(\alpha S)}{\log_2 A(\alpha S)} \end{aligned}$$

so that

$$0 < \frac{4A(S)}{A(Q)} \le \liminf_{A(S)\to\infty} C_S \cdot \frac{A(S)}{\log_2 A(S)}.$$

Also, by Theorem 5.9,

$$\begin{aligned} 4 &= \lim_{\alpha\to\infty} C_R \cdot \frac{A(\alpha R)}{\log_2 A(\alpha R)} \ge \limsup_{\alpha\to\infty} C_S \cdot \frac{A(\alpha R)}{\log_2 A(\alpha R)} \\ &= A(R) \cdot \limsup_{\alpha\to\infty} C_S \cdot \frac{\alpha^2}{\log_2 \alpha^2} = \frac{A(R)}{A(S)} \cdot \limsup_{\alpha\to\infty} C_S \cdot \frac{A(\alpha S)}{\log_2 A(\alpha S)} \end{aligned}$$

so that

$$\infty > \frac{4A(S)}{A(R)} \ge \limsup_{A(S)\to\infty} C_S \cdot \frac{A(S)}{\log_2 A(S)}.$$

□

Note that special cases of Theorem 5.18 include when S is an open checkerboard constraint or when S is the closure of an open checkerboard constraint.

5.6 Capacity relative to a scaled lattice

The results obtained in this paper have indicated the asymptotic capacities of certain two-dimensional checkerboard constrained channels. The capacities are given in terms of the "area" of the constraint S. The quantity $A(S)$ was defined as the two-dimensional Lebesgue measure of the set S. The units of capacity were given as bits per lattice point location on the $\mathbf{Z}^2$-lattice. It is reasonable to ask what happens to the results if the lattice itself is scaled. For example, suppose we ask how many bits of information can be stored on a lattice $\beta\mathbf{Z}^2$ subject to a constraint S. This is identical to determining how many bits can be stored on the usual $\mathbf{Z}^2$-lattice using a constraint $(1/\beta)S$.

Let $\Lambda_\beta(S)$ be the number of $\beta\mathbf{Z}^2$-lattice points in S. Then the area $A(S)$ of an open set S is related to $\Lambda_\beta(S)$ by the estimate $A(S)/\Lambda_\beta(S) \approx \beta^2$, where the approximation becomes equality in the limit as $A(S) \to \infty$. Thus, using Corollary 5.17, if the checkerboard constraint S is open, convex, and symmetric then the asymptotic number of bits that can be stored per lattice point on $\beta\mathbf{Z}^2$ is

$$\frac{\log_2 \Lambda_\beta(S)}{\Lambda_\beta(S)} = \frac{\log_2 \Lambda\left(\frac{1}{\beta}S\right)}{\Lambda\left(\frac{1}{\beta}S\right)} \approx \frac{\log_2 A(S) - 2\log_2 \beta}{A(S)/\beta^2}.$$

The capacity per unit area in the plane is therefore asymptotically equal to the capacity per lattice point multiplied by the number of lattice points per unit area, that is

$$\frac{\log_2 A(S) - 2\log_2 \beta}{A(S)/\beta^2} \cdot \frac{1}{\beta^2} = \frac{\log_2 A(S) - 2\log_2 \beta}{A(S)}.$$

Thus for any fixed β, in the limit as $A(S) \to \infty$, the capacity still decays at the rate $\frac{\log_2 A(S)}{A(S)}$, even though for any fixed $\beta < 1$, the capacity is larger than for $\beta = 1$. In summary, the asymptotic results presented are independent of the scaling of the underlying lattice, although for finite constraint areas there may be a difference.

5.7 Acknowledgments

The authors would like to thank Pierre Loyer for pointing out the connection between Lemma 5.6 and Minkowski's classical results, and Károly Böröczky for some helpful references.

This chapter, in full, has been submitted for publication as: Zs. Nagy and K. Zeger, Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints, *IEEE Trans. Inform. Theory*, July 2002. The dissertation author was the primary investigator of this paper.

References

[1] J. J. Ashley and B. H. Marcus. Two-Dimensional Low-Pass Filtering Codes. *IEEE Trans. on Communications*, 46:724–727, June 1998.

[2] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics*. Academic Press, 1982.

[3] R. J. Baxter and S. K. Tsang. Entropy of Hard Hexagons. *J. Physics A*, 13(3):1023–1030, March 1980.

[4] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[5] J. W. S. Cassels. *An Introduction to the Geometry of Numbers*. Springer-Verlag, Berlin, Göttingen, Heidelberg, 1959.

[6] C. H. Dowker. On Minimum Circumscribed Polygons. *Bulletin Amer. Math. Soc.*, 50:120–122, 1944.

[7] K. Engel. On the Fibonacci Number of an $M \times N$ Lattice. *Fibonacci Quarterly*, 28:72–78, 1990.

[8] W. H. Erxleben and M. W. Marcellin. Error-Correcting Two-Dimensional Modulation Codes. *IEEE Trans. Inform. Theory*, 41:1116–1126, July 1995.

[9] T. Etzion. Cascading methods for runlength-limited arrays. *IEEE Trans. Inform. Theory*, 43:319–324, January 1997.

[10] T. Etzion and K. G. Paterson. Zero/positive Capacities of Two-Dimensional Runlength Constrained Arrays. *Proc. 2001 IEEE Intl. Symp. on Inform. Theory*, page 269, Washington, DC, USA, June 2001.

[11] L. Fejes Tóth. *Regular Figures*. Pergamonn Press, 1964.

[12] S. Forchhammer and J. Justesen. Entropy Bounds for Constrained Two-Dimensional Random Fields. *IEEE Trans. Inform. Theory*, 45:118–127, January 1999.

[13] S. Halevy and R. M. Roth. Parallel Constrained Coding with Application to Two-Dimensional Constraints. *IEEE Trans. Inform. Theory*, 48:1009–1020, May 2002.

[14] K. A. S. Immink, P. H. Siegel, and J. K. Wolf. Codes for Digital Recorders. *IEEE Trans. Inform. Theory*, 44:2260–2299, October 1998.

[15] G. S. Joyce. On the Hard-hexagon Model and the Theory of Modular Functions. *Philos. Trans. Roy. Soc. London Ser. A*, 325(1588):643–702, 1988.

[16] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[17] P. J. Kelly and M. L. Weiss. *Geometry and Convexity: A Study in Mathematical Methods*. John Wiley & Sons, New York, 1979.

[18] M. Larsen. The Problem of the Kings. *Electronic Journal of Combinatorics*, 2, 1995.

[19] M. P. Richey and C. A. Tracy. Equation of State and Isothermal Compressibility for the Hard Hexagon Model in the Disordered Regime. *J. Phys. A*, 20(16):L1121–L1126, November 1987.

[20] C. A. Rogers. *Packing and Covering*. Cambridge University Press, Cambridge, U.K., 1974.

[21] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding Schemes for the Hard-Square Model. *IEEE Trans. Inform. Theory*, 47:1166–1176, March 2001.

[22] P. H. Siegel and J. K. Wolf. Bit Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays. *Proc. 1998 IEEE Intl. Symp. on Inform. Theory*, page 323, MIT, Cambridge, MA, August 1998.

[23] E. Soljanin. Writing Sequences on the Plane. *IEEE Trans. Inform. Theory*, 48:1344–1354, June 2002.

[24] R. E. Swanson and J. K. Wolf. A New Class of Two-Dimensional Runlength-Limited Recording Codes. *IEEE Trans. on Mag. Recording*, 28:3407–3416, November 1992.

[25] R. Talyansky, T. Etzion, and R. M. Roth. Efficient Code Construction for Certain Two-Dimensional Constraints. *IEEE Trans. Inform. Theory*, 45(2):794–799, March 1999.

[26] C. A. Tracy, L. Grove, and M. F. Newman. Modular Properties of the Hard-hexagon Model. *J. Stat. Phys.*, 48(3 and 4):477–502, August 1987.

[27] B. V. Vasic. Capacity of Channels with Redundant Multitrack (d, k) Constraints: the $k < d$ Case. *IEEE Trans. Inform. Theory*, 42:1546–1548, September 1996.

[28] B. V. Vasic, S. W. McLaughlin, and O. Milenkovic. Shannon Capacity of M-ary Redundant Multitrack Runlength Limited Codes. *IEEE Trans. Inform. Theory*, 44:766–774, March 1998.

[29] W. Weeks and R. E. Blahut. The Capacity and Coding Gain of Certain Checkerboard Codes. *IEEE Trans. Inform. Theory*, 44(3):1193–1203, May 1998.

[30] H. S. Wilf. The Problem of the Kings. *Electronic Journal of Combinatorics*, 2, 1995.

Chapter 6

Capacity Bounds for the Hard-Triangle Model

Abstract

A binary labeling of the triangles in a regular tiling of the two-dimensional plane satisfies the hard-triangle constraint if every triangle labeled with 1 has its three neighbors labeled with 0s. We show that the capacity associated with this constraint lies in the interval $[0.628831217, 0.634775895]$. The upper bound is obtained by bounding the largest magnitude eigenvalue of a certain transfer matrix and the lower bound is established by constructing an encoding algorithm whose coding rate is within 1% of the capacity.

6.1 Introduction

Various constraints imposed on one-dimensional binary sequences are important in magnetic recording applications. Likewise, two-dimensional constraints play a role in optical storage devices. In particular, run length constraints in one and higher dimensions have been a subject of intense research. In two dimensions, such constraints have primarily been studied for rectangular and hexagonal lattices. We examine such constraints for an equilateral triangular non-lattice tiling of the two-dimensional plane.

A binary sequence satisfies a *one-dimensional* (d, k) *run length constraint* if there are at most k zeros in a row, and between every two consecutive ones there are at least d zeros. A two-dimensional binary rectangular array is said to satisfy a *two-dimensional* (d, k) *run length constraint*, if it satisfies the one-dimensional (d, k) run length constraint along the directions parallel to the coordinate axes. Such an array is called (d, k)-*valid*. A physical medium with a two-dimensional rectangular grid which accepts only (d, k)-valid two-dimensional labelings of the grid points is called a (d, k)-*constrained channel*. The number of (d, k)-valid two-dimensional arrays of size $m \times n$ is denoted by $\nu_{d,k}(m, n)$ and the corresponding *channel capacity* is defined as

$$C_{d,k}^{(2)} = \lim_{m,n \to \infty} \frac{\log_2 \nu_{d,k}(m, n)}{mn}.$$

It is known for all d and $k \geq d$ that the two-dimensional (d, k) capacities exist [18].

A coding technique, called bit stuffing, was first proposed by Lee [21] for one-dimensional $(0, k)$ constraints, then generalized by Bender and Wolf [5] to one-dimensional (d, k) constraints and then extended by Siegel and Wolf [30] to two-dimensional (d, ∞) constraints. Halevy et. al [14] generalized bit stuffing to hexagonal two-dimensional lattices for certain (d, ∞) constraints. The two-dimensional bit stuffing algorithm in [30] was studied by Roth, Siegel, and Wolf [27] and enhanced later in [28]. Encoders for other two-dimensional run length limited constraints can be found in [15], and other examples of two-dimensional constrained codes are weight-constrained codes [25] and burst error correcting codes using interleaving schemes [12].

Of particular interest has been the special case of two-dimensional (d, k)-constrained channels when $d = 1$ and $k = \infty$, which is known as the "hard-square model". By exchanging the roles of 0 and 1 one can easily verify that $C_{0,1}^{(2)} = C_{1,\infty}^{(2)}$. Upper and lower bounds on the capacity of the two-dimensional $(1, \infty)$-constrained channel were given by Engel [8] and by Calkin and Wilf [6]. The best known bounds on

$C^{(2)}_{1,\infty}$ [22] are presently $0.587891161775 \leq C^{(2)}_{1,\infty} \leq 0.587891161868$. The bit stuffing encoder for the $(1,\infty)$ constraint has been shown [27], [28], [30] to achieve an expected coding rate of 0.587277, which is within 0.1% of the capacity $C^{(2)}_{1,\infty}$.

Run length constraints on a hexagonal lattice have been studied in [3], [4], [14], [17], [19], and [34]. In particular, a derivation of the capacity for the case $d = 1$ (known as the ”hard hexagon model”) was given in [4], from which an analytic expression for the capacity was presented in [17], [26], and [34]. The capacities are also closely related to gases, lattices, and Ising model entropies in statistical mechanics [3]. Slightly different two-dimensional constraints were studied for the purpose of determining the growth rates of the number of certain chess configurations (e.g. [20], [38]). In addition to run length constraints, other types of constraints have been studied as well [1], [9], [10], [11], [13], [23], [31], [32], [33], [35], [36], [37].

In the present paper we consider a non-lattice tiling of the two-dimensional plane by equilateral triangles and use the center of each triangle to store a bit. Analogous to the square and hexagonal cases, we study a “hard triangle” constraint on the triangular tiling. Every triangle with a 1 in it must have all three of its neighboring triangles have 0s in them. We analyze the capacity by deriving an upper bound analytically and obtain a lower bound by exhibiting a bit stuffing algorithm for encoding arbitrary input binary sequences into the triangular tiling without violating the constraint.

In Section 6.2 we define terminology. In Section 6.3 we give an upper bound on the capacity using transfer matrices. To establish a lower bound, we first introduce a general hard-triangle constrained encoder in Section 6.4, and then use a specific variable-to-variable length encoder to analyze the coding rate in Section 6.5. We thus obtain a lower bound on the capacity. Our variable-to-variable length hard-triangle constrained encoder is based on ideas similar to the bit stuffing encoders of [24], [27], [28], and [30]. Our main results, the hard-triangle constrained capacity bounds, are summarized in Corollary 6.11.

6.2 Definitions

Let $\mathbf{Z}$ denote the integers, $\mathbf{Z}^+$ the positive integers, and $\mathbf{R}^2$ the two-dimensional plane. For any $S \subset \mathbf{R}^2$ and $u \in \mathbf{R}^2$, let $S + u = \{s + u : s \in S\}$. A two-dimensional *lattice* is a set $\{iu + jv : i, j \in \mathbf{Z}\}$ where $u, v \in \mathbf{R}^2$ are linearly independent.

For any $i, j \in \mathbf{Z}$, let

$$\begin{aligned}
[i,j] &= i(\sqrt{3},0) + j\left(\tfrac{\sqrt{3}}{2}, \tfrac{3}{2}\right) \\
[i,j]' &= [i,j] + (0,1) \\
T_1 &= \{[i,j] : i,j \in \mathbf{Z}\} \\
T_2 &= T_1 + (0,1) \\
T &= T_1 \cup T_2
\end{aligned}$$

(see Figure 6.1). The notation $[i,j]$ represents a point in T_1 with respect to the basis $\left\{(\sqrt{3},0), \left(\frac{\sqrt{3}}{2}, \frac{3}{2}\right)\right\}$. We say that two points (or their corresponding triangles) in T are *neigh-*

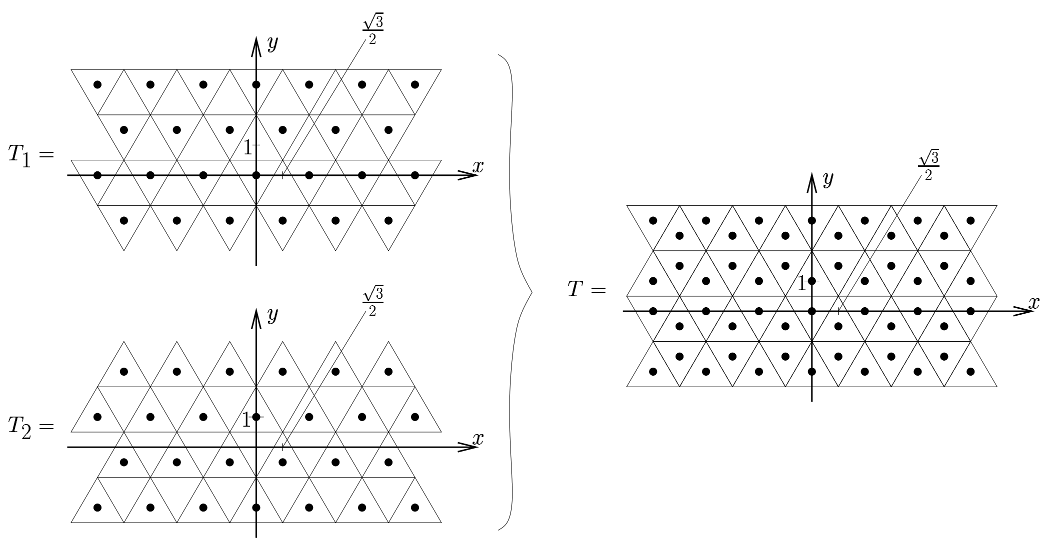


Figure 6.1: A tiling $T = T_1 \cup T_2$ of $\mathbf{R}^2$ with equilateral triangles of side length $\sqrt{3}$.

bors if the distance between them is 1. For any set $S \subset T$, the *closure of* S is denoted by $\bar{S}$, and it contains the points that are either in S or have at least one neighboring point in

S. Note that $T_2 = T_1 - \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$, $[i,j]' \in T_2$, the set T_1 is a lattice, T_2 and T are not lattices, and every point in T has three nearest neighbors at a distance 1.

For any $S \subset T$, a function $f : S \longrightarrow \{0,1\}$ is called a *labeling* of S. A labeling f of S satisfies the *hard-triangle constraint* if for every $t \in S$, the three nearest neighbors of t are labeled with 0s whenever $f(t) = 1$. In the rest of the paper we will call a labeling of S *valid* if it satisfies the hard-triangle constraint, and will denote the set of valid labelings by $L(S)$. The capacity of the hard-triangle constraint will be defined analogously to the hard-square constrained capacity.

For any string s on an alphabet $\mathcal{A}$ let $l(s)$ denote its length and $s_i \in \mathcal{A}$ denote the ith symbol in the string. Throughout the paper N will denote a positive integer, and random variables will be denoted with "hat" notation.

A sequence $\hat{u}_0, \hat{u}_1, \dots$ of random variables taking on values from an alphabet $\mathcal{A}$ is called a *Markov chain*, if for all $n \in \mathbf{Z}^+$ and $u_n \in \mathcal{A}$,

$$\mathbf{P}(\hat{u}_n = u_n | \hat{u}_{n-1} = u_{n-1}, \dots, \hat{u}_0 = u_0) = \mathbf{P}(\hat{u}_n = u_n | \hat{u}_{n-1} = u_{n-1}).$$

A Markov chain is *homogeneous* (or *time invariant*) if $\mathbf{P}(\hat{u}_n = u | \hat{u}_{n-1}) = \mathbf{P}(\hat{u}_1 = u | \hat{u}_0)$ for all $n \in \mathbf{Z}^+$ and $u \in \mathcal{A}$. For every $u_0, u_1 \in \mathcal{A}$, the conditional probabilities $\mathbf{P}(\hat{u}_1 = u_1 | \hat{u}_0 = u_0)$ of a homogeneous Markov chain are called the *transition probabilities*. A Markov chain is *stationary* if $\mathbf{P}(\hat{u}_n = u) = \mathbf{P}(\hat{u}_0 = u)$ for all $n \in \mathbf{Z}^+$ and $u \in \mathcal{A}$. We say that two homogeneous Markov chains are *identical* if both Markov chains take on values from the same set $\mathcal{A}$, and have the same transition probabilities and initial probabilities.

A binary sequence with independent identically distributed (i.i.d.) symbols from the alphabet $\{u_0, u_1\}$ is called a ψ*-sequence* if u_0 occurs with probability ψ. Similarly, a ternary i.i.d. sequence with symbol alphabet $\{u_0, u_1, u_2\}$ is called a (ϕ_0, ϕ_1)*-sequence* if u_0 occurs with probability ϕ_0 and u_1 occurs with probability ϕ_1. Throughout the paper

$\hat{w}$ will denote a $1/2$-sequence. Denote the binary and ternary entropy functions as

$$\begin{aligned} H_2(x) &= -x\log_2 x - (1-x)\log_2(1-x) \\ H_3(x,y) &= -x\log_2 x - y\log_2 y - (1-x-y)\log_2(1-x-y). \end{aligned}$$

For $X, Y \in \mathbf{Z}^+$, define the array $A_{X,Y} \subset T$ as (see Figure 6.3)

$$A_{X,Y} = \{[i,j] : 0 \le i \le X, 0 \le j \le Y\} \cup \{[i,j]' : 0 \le i \le X, -1 \le j \le Y-1\}$$

and let $\nu(X,Y)$ denote the number of valid labelings of $A_{X,Y}$. The capacity C_T corre-

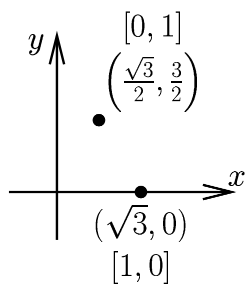


Figure 6.2: Basis vectors for T_1.

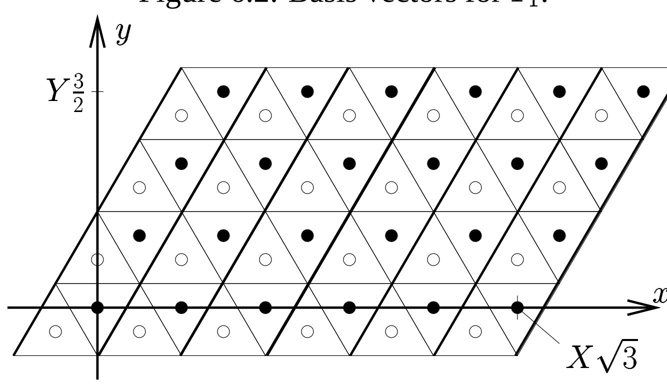


Figure 6.3: The array $A_{X,Y}$. Solid circles represent points lying in the lattice T_1, and hollow circles represent points lying in the set T_2.

sponding to the hard-triangle constraint is defined as

$$C_T = \lim_{X,Y\to\infty} \frac{\log_2 \nu(X,Y)}{|A_{X,Y}|} = \lim_{X,Y\to\infty} \frac{\log_2 \nu(X,Y)}{2(X+1)(Y+1)}$$

where the right hand side follows since the cardinality of $A_{X,Y}$ is $2(X+1)(Y+1)$. The existence of C_T can be shown using a similar proof as in [18]. Let $U_{X,Y} \subset A_{X,Y}$ be defined as:

$$U_{X,Y} = \big\{[i,j] : 0 \le i \le X, 0 \le j \le Y\big\} \cup \big\{[i,j]' : 1 \le i \le X, 0 \le j \le Y-1\big\}$$

(see Figure 6.4). Furthermore, define the following subsets of $U_{X,Y}$:

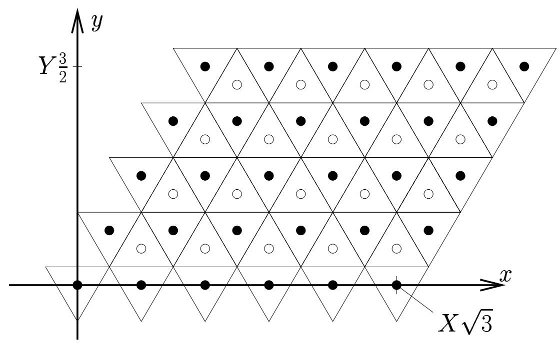


Figure 6.4: The set $U_{X,Y}$ is comprised of $(X+1)(Y+1)$ points of T_1 (solid circles) and XY points of T_2 (hollow circles).

$$\begin{aligned} D_l &= \{[l,j] : 0 \le j \le Y\} & (0 \le l \le X) \\ D'_l &= \{[l,j]' : 0 \le j \le Y-1\} & (1 \le l \le X) \\ R_0 &= \{[i,Y] : 0 \le i \le X\} & \end{aligned}$$

(see Figure 6.5). We call D_l and D'_l a *solid diagonal* and a *hollow diagonal*, respectively, of $U_{X,Y}$, and we call D_0 and R_0 a *boundary diagonal* and a *boundary row*, respectively, of $U_{X,Y}$. Note that $D_l \subset T_1$, $R_0 \subset T_1$, and $D'_l \subset T_2$. The elements of D_0 and R_0 are called the *boundary points of* $U_{X,Y}$, and the remainder of the points of $U_{X,Y}$ are called the *internal points of* $U_{X,Y}$.

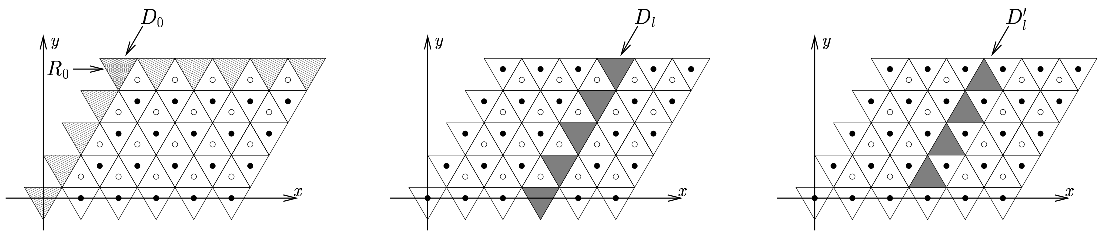


Figure 6.5: The boundary diagonal D_0, boundary row R_0, solid diagonals D_l and hollow diagonals D'_l of $U_{X,Y}$.

6.3 Hard-triangle capacity upper bound

Note that $\nu(0, Y)$ is the number of one-dimensional binary sequences of length $2Y + 2$ satisfying the $(1, \infty)$ constraint. Let $f_1,\ f_2, \dots,\ f_{\nu(0,Y)}$ denote the valid labelings of $A_{0,Y}$. Define the *transfer matrix* M_Y to be a $\nu(0, Y) \times \nu(0, Y)$ binary matrix, such that the rows and columns of M_Y are indexed by the valid labelings of $A_{0,Y}$, and the $(i, j)^{th}$ entry of M_Y is 1 if and only if the labeling

$$f(u) = \begin{cases} f_i(u) & \text{if } u \in A_{0,Y} \\ f_j(u - [1, 0]) & \text{if } u \in A_{0,Y} + [1, 0] \end{cases}$$

is valid on $A_{0,Y} \cup (A_{0,Y} + [1, 0])$. Then,

$$\nu(X, Y) = \mathbf{1}' \cdot M_Y^{X-1} \cdot \mathbf{1} = \mathbf{1}' \cdot M_X^{Y-1} \cdot \mathbf{1}$$

where **1** is the all-ones column vector of the appropriate dimension and prime denotes transpose. The matrix M_Y meets the conditions of the Perron-Frobenius theorem [2, p. 17], since it has nonnegative elements and is irreducible (since the all-zeros labeling of $A_{0,Y}$ can be placed next to any valid labeling of $A_{0,Y}$ without violating the hard-triangle constraint). Thus, the largest magnitude eigenvalue Λ_Y of M_Y is positive, real, and has multiplicity one. It follows that

$$\lim_{X\to\infty} (\nu(X,Y))^{1/X} = \Lambda_Y$$

and therefore

$$C_T = \lim_{X,Y\to\infty} \frac{\log_2 \nu(X,Y)}{2(X+1)(Y+1)} = \lim_{Y\to\infty} \frac{\log_2 \lim_{X\to\infty} (\nu(X,Y))^{1/(X+1)}}{2(Y+1)} = \frac{1}{2} \lim_{Y\to\infty} \frac{\log_2 \Lambda_Y}{Y+1}.$$

On the other hand, any valid labeling of $A_{X,k(Y+1)-1}$ defines a valid labeling of $A_{X,Y} + i(Y+1)[0,1]$ whenever $0 \le i < k$, and therefore $\nu(X,kY) \le (\nu(X,Y))^k$. Hence, for any $Y \ge 0$,

$$\begin{aligned} C_T &= \lim_{X,k\to\infty} \frac{\log_2 \nu(X,kY)}{2k(X+1)(Y+1)} \le \frac{1}{2(Y+1)} \lim_{X\to\infty} \log_2 (\nu(X,Y))^{1/(X+1)} \\ &= \frac{1}{2(Y+1)} \log_2 \Lambda_Y. \end{aligned} \quad (6.1)$$

Thus the sequence $\left\{\frac{\log_2 \Lambda_i}{2(i+1)}\right\}$ converges to C_T from above. Table 6.1 shows the largest eigenvalues Λ_i for $i \le 13$. Using Λ_{13} in (6.1) gives the bound

$$C_T \le 0.634775895. \quad (6.2)$$

6.3.1 Remark

Direct computation of eigenvalues using standard linear algebra algorithms generally requires the storage of an entire matrix. This restricts the matrix sizes allowable, due to memory constraints on computers. By exploiting the fact that the matrix M_Y is binary and the $(i,j)^{th}$ value is easily computable, we were able to obtain the largest eigenvalues of very large matrices. The eigenvalues listed in Table6.1 were computed using the power method together with the following result.

Table 6.1: Twelve digits of accuracy of the largest eigenvalue Λ_i of the transfer matrix M_i, and the capacity upper bound $\left\{\frac{\log_2 \Lambda_i}{2(i+1)}\right\}$ for $i \leq 13$.

i	Λ_i	$\left\{\frac{\log_2 \Lambda_i}{2(i+1)}\right\}$	number of rows in M_i
0	2.61803398875	0.694242	3
1	6.37228132327	0.667953	8
2	15.2974271890	0.655870	21
3	36.6144845358	0.649293	55
4	87.5793507833	0.645252	144
5	209.453046768	0.642541	377
6	500.907291067	0.640600	987
7	1197.91174803	0.639145	2584
8	2864.78199321	0.638012	6765
9	6851.06632288	0.637106	17711
10	16384.1807109	0.636365	46368
11	39182.4221132	0.635747	121393
12	93703.9345069	0.635224	317811
13	224090.979075	0.634776	832040

Lemma 6.1. ***[16, pg. 493]*** *Let M be an $n \times n$ matrix with nonnegative real entries. Then for any n-dimensional positive vector x, the magnitude of the largest eigenvalue ρ of M is bounded as*

$$\min_{1 \leq i \leq n} \frac{1}{x_i} \sum_{j=1}^{n} M_{ij} x_j \leq \rho \leq \max_{1 \leq i \leq n} \frac{1}{x_i} \sum_{j=1}^{n} M_{ij} x_j$$
$$\min_{1 \leq j \leq n} x_j \sum_{i=1}^{n} \frac{M_{ij}}{x_i} \leq \rho \leq \max_{1 \leq j \leq n} x_j \sum_{i=1}^{n} \frac{M_{ij}}{x_i}.$$

The convergence rate of the power method depends on the relative size of the largest and second largest eigenvalues, but the second largest eigenvalue is generally unknown. Hence, we iterated the power method computation until the eigenvalues appeared to stabilize in the 12th significant decimal place. The resulting eigenvector estimates were used as the values of x in Lemma 6.1 to obtain *exact* upper bounds on the largest eigenvalues, thus making (6.2) an exact inequality instead of an approximation.

6.4 Hard-triangle constrained encoding

To obtain a lower bound on the capacity C_T, we introduce the notion of a hard-triangle constrained encoder that maps $\hat{w}$ into a hard-triangle constrained labeling of T, and calculate the coding rate of the encoder. The coding rate is a quantity that measures the efficiency of the encoder, and is upper-bounded by the capacity C_T.

A hard-triangle constrained *encoder* is an injection

$$\mathcal{E} : \{0,1\}^{\infty} \longrightarrow \bigcup_{S \subset T} L(S)$$

and its inverse is called a *decoder*. The encoder $\mathcal{E}$ maps an infinite binary input sequence into a labeling of a subset of T. An encoder and decoder are together called a *coding algorithm*.

One way to implement an encoder is to first parse an infinite binary source and then independently map the resulting finite length binary strings into disjoint regions of T, such that no two such regions have neighboring triangles. Then zero padding can be added between regions to assure the hard-triangle constraint is not violated, provided each parsed string is mapped into a region without locally violating the hard-triangle constraint. This is described formally below.

Let V be a finite complete prefix code[1], and for each $v \in V$ let $S_v \subset T$. A hard-triangle constrained *word encoder* is an injection

$$\mathcal{E}_V : V \longrightarrow \bigcup_{S \subset T} L(S) \tag{6.3}$$

that maps each element of V into a labeling of some subset of T. Let $z \in \{0,1\}^\infty$ be an arbitrary infinite sequence that is parsed by V as $z = z^{(1)} z^{(2)} \ldots$, where $z^{(i)} \in V$ for all i. The elements of $\{\theta_i \in T_1 : i \in \mathbf{Z}^+\}$ are *translation vectors* if, for all i, the sets $\theta_i + S_{z^{(i)}}$ are disjoint and no points in different sets are neighbors. A hard-triangle constrained *composite encoder* $\mathcal{E}$ is defined by:

$$\mathcal{E}(z)(u) = \begin{cases} \mathcal{E}_V(z^{(i)})(u - \theta_i) & \text{for all } i = 1, 2, \ldots, \text{ if } u \in \theta_i + S_{z^{(i)}} \\ 0 & \text{if } \exists i \text{ s.t. } u \in \theta_i + \bar{S}_{z^{(i)}} \setminus S_{z^{(i)}} \end{cases}.$$

That is, $\mathcal{E}(z)$ is a labeling of translates of the sets $S_{z^{(1)}}, S_{z^{(2)}}, \ldots$ composed of the labelings $\mathcal{E}_V(z^{(1)}), \mathcal{E}_V(z^{(2)}), \ldots$. The labeling of points in T outside of any translate $\theta_i + S_{z^{(i)}}$ by 0 is called *zero padding*. It is possible to choose the word encoder $\mathcal{E}_V$ and translation vectors $\theta_1, \theta_2, \ldots$ such that the composite encoder is injective (i.e. is an encoder).

---

[1]The code V is a *prefix* code if no codeword is a prefix of any other codeword. *Complete* means that in the decoding tree, every node is either a leaf or has two children.

Define the following quantities for a word encoder:

$$\begin{aligned}\bar{r}\left(\mathcal{E}_V\right) &= \sum_{v \in V} \mathbf{P}(v) \frac{l(v)}{|S_v|} \\ \underline{r}\left(\mathcal{E}_V\right) &= \sum_{v \in V} \mathbf{P}(v) \frac{l(v)}{|\bar{S}_v|}.\end{aligned}$$

These upper and lower bound, respectively, the average ratio between the input length and the number of points in T that are labeled, for a particular prefix code V. The probability $\mathbf{P}(v)$ is taken with respect to the distribution of an unbiased random source.

If $l(v)$ is a constant for all $v \in V$, then if $|S_v|$ is a constant, $\mathcal{E}_V$ is a *fixed-to-fixed length* encoder, and if $|S_v|$ is not a constant then $\mathcal{E}_V$ is a *fixed-to-variable length* encoder. Similarly, if $l(v)$ is not a constant, then if $|S_v|$ is a constant, $\mathcal{E}_V$ is a *variable-to-fixed length* encoder, and if $|S_v|$ is not a constant then $\mathcal{E}_V$ is a *variable-to-variable length* encoder.

If $\{V_i\}$ is a sequence of prefix codes with increasing cardinality, then the *coding rate* of a composite encoder $\mathcal{E}$ (with respect to V_i) is

$$r\left(\mathcal{E}\right) = \lim_{i \to \infty} \bar{r}\left(\mathcal{E}_{V_i}\right) = \lim_{i \to \infty} \underline{r}\left(\mathcal{E}_{V_i}\right) \tag{6.4}$$

provided that the limits exist and are equal. It is known [29, p. 27] that the coding rate is upper-bounded by the capacity. (Although Shannon's theorem applies to one-dimensional channels, a replica of his proof can be used to show the same result for the two-dimensional hard-triangle constrained channel.) Therefore, by determining the numerical value of the limit $r\left(\mathcal{E}\right)$, we obtain the lower bound $r\left(\mathcal{E}\right) \leq C_T$ for the capacity of the hard-triangle constraint.

In Section 6.4.1 we present an intermediate variable-to-fixed length encoder with four input sequences. Based on this encoder, in Section 6.5 we introduce a variable-to-variable length encoding algorithm that maps an infinite binary input sequence w into a valid labeling of T. The variable-to-variable length encoder parses the input into strings $w^{(1)}, w^{(2)}, \ldots$, and then bijectively maps each string $w^{(i)}$ into four new strings that can be more easily mapped into a hard-triangle constrained labeling of the array $A_{X(w^{(i)}),N}$ by the variable-to-fixed length encoder. The quantity N is a parameter of the encoder and the number of solid diagonals in the array that $w^{(i)}$ is mapped into is $X(w^{(i)}) + 1$.

The resulting labeled arrays $A_{X(w^{(i)}),N}$ are valid, and in addition they are designed such that any two of them can be placed next to each other without violating the hard-triangle constraint. Thus a slanted quadrant of the two-dimensional plane can be labeled by the translates of the arrays $A_{X(w^{(i)}),N}$ (see Figure 6.6). The tiling can be generalized to all of T by alternately placing the arrays in the four quadrants. The definition of the encoder and a rigorous proof that shows the limit in (6.4) exists for the encoder are presented in Section 6.5. Our variable-to-variable length encoder can be viewed as "nearly" a fixed-to-fixed length encoder since the lengths of both the input and output are limited to a short range.

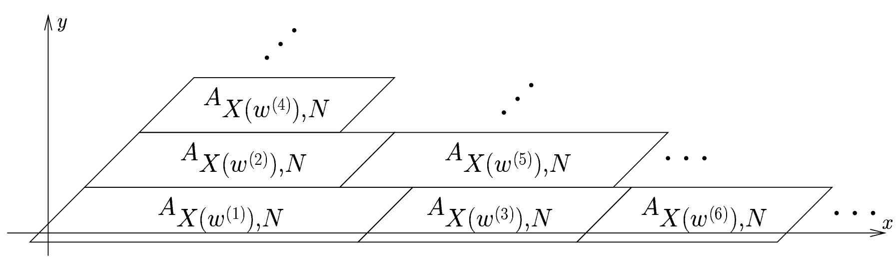


Figure 6.6: The parsed input sequence $w = w^{(1)}w^{(2)}\ldots$ is mapped into labelings of translates of $A_{X(w^{(1)}),N}, A_{X(w^{(2)}),N}, \ldots$.

We define the vectors $t_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $t_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $t_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and interpret them as the symbols

$$t_0 = \begin{array}{c} 0 \\ \hline 0 \end{array}, \quad t_1 = \begin{array}{c} 0 \\ \hline 1 \end{array}, \quad t_2 = \begin{array}{c} 1 \\ \hline 0 \end{array}$$

in a triangular labeling. In the following sections we define an encoder that maps an arbitrary binary input sequence into a sequence of 4-tuples of strings over $\{t_0, t_1, t_2\}$, and then maps sequences on $\{t_0, t_1, t_2\}$ into labelings of subsets of T by mapping the

Table 6.2: Parameters used in Sections 6.4 and 6.5.

Parameter	Description
N	Positive integer array side length. Goes to ∞.
γ_i	Probability that the ith ($i = 1, 2, 3, 4$) input string is used by $\tilde{\mathcal{E}}$.
λ	Initial labeling of J_τ.
τ	Number of translates of $U_{N,N}$ in $U_{X,N}$.
ϵ	Positive real. Goes to 0.
$\phi_0, \phi_1, \phi_2, \phi_3, \psi_0, \psi_1$	Symbol probabilities in transformed sequences.
$\sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \sigma^{(4)}$	Auxiliary strings.
$s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}$	Input strings.

lower and upper bits in t_0, t_1, t_2 into the positions $[i, j]$ and $[i, j]'$, respectively, for certain $[i, j]$.

A list of variables defined in Sections 6.4 and 6.5 and the parameters they depend on are given in Tables 6.2 and 6.3 as a reference.

6.4.1 A variable-to-fixed length encoder

We define the following total ordering on the points of T_1:

$$[x_1, y_1] \prec [x_2, y_2] \Longleftrightarrow \begin{cases} y_1 < y_2 \text{ or} \\ y_1 = y_2 \text{ and } x_1 > x_2 \end{cases} .$$

That is, $[x_1, y_1] \prec [x_2, y_2]$ if the diagonal, whose direction goes from bottom-left to top-right, that $[x_1, y_1]$ lies on is above and to the left of the diagonal that $[x_2, y_2]$ lies on, or if they lie on the same diagonal but with $[x_1, y_1]$ above and to the right of $[x_2, y_2]$.

Next we define a variable-to-fixed length encoder to label $U_{N,N}$ and then use the encoder as a building block in a variable-to-variable length encoder to label larger

Table 6.3: Variables introduced in Sections 6.4 and 6.5 and the parameters they depend on.

Notation	Parameters	Description
$\tilde{\mathcal{E}}$	N, λ	Variable-to-fixed length encoder. Labels $U_{N,N}$.
$\bar{\mathcal{E}}$	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \sigma^{(1)}, \ldots, \sigma^{(4)}$	Fixed-to-variable length encoder. Labels $U_{X,N}$.
$Q_i^{(j)}$	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \sigma^{(1)}, \ldots, \sigma^{(4)}$, $s^{(1)}, \ldots, s^{(4)}$	Number of symbols $\bar{\mathcal{E}}$ maps into $U_{N,N}^{(i)}$ from jth input string.
q	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \sigma^{(1)}, \ldots, \sigma^{(4)}$	Number of symbols $\bar{\mathcal{E}}$ does not map into 1st τ translates of $U_{N,N}$.
B	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \epsilon, \sigma^{(1)}, \ldots, \sigma^{(4)}$	Set of strings that $\bar{\mathcal{E}}$ nearly maps into 1st τ translates of $U_{N,N}$.
A	$N, \gamma_1, \ldots, \gamma_4, \tau, \epsilon, \phi_0, \ldots, \phi_3, \psi_0, \psi_1$	Set of typical sequences.
G	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \epsilon, \phi_0, \ldots, \phi_3, \psi_0, \psi_1$, $\sigma^{(1)}, \ldots, \sigma^{(4)}$	Complete prefix code of size $\lvert A \cap B \rvert$.
g	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \epsilon, \phi_0, \ldots, \phi_3, \psi_0, \psi_1$, $\sigma^{(1)}, \ldots, \sigma^{(4)}$	Bijection from G to $A \cap B$.
$\mathcal{E}$	$N, \gamma_1, \ldots, \gamma_4, \lambda, \tau, \epsilon, \phi_0, \ldots, \phi_3, \psi_0, \psi_1$, $\sigma^{(1)}, \ldots, \sigma^{(4)}$	Variable-to-variable length encoder. Labels $U_{X,N}$.

portions of T. Then we take $N \to \infty$. For any internal point $a = [i, j] \in U_{N,N} \cap T_1$, define the notation (see Figure 6.7)

$$\begin{aligned} a' &= [i, j]' \\ a_1 &= [i-1, j] \\ a_2 &= [i-1, j+1] \\ a_3 &= [i, j+1] \\ a_b &= [N, -1] \end{aligned}$$

and note that a_b is the least upper bound of the points in $U_{N,N} \cap T_1$ under the ordering $\prec$.

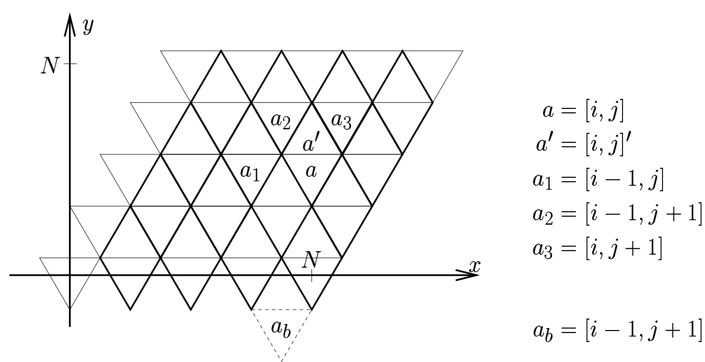


Figure 6.7: The internal point $a \in U_{N,N} \cap T_1$ and the neighborhood a', a_1, a_2, a_3.

Let $\lambda : D_0 \cup R_0 \longrightarrow \{0, 1\}$ be an *initial labeling* of the boundary of $U_{N,N}$. Then define (with respect to λ) a hard-triangle constrained encoder $\tilde{\mathcal{E}}$ with input strings

$s^{(1)}, s^{(2)} \in \{t_0, t_1, t_2\}^*$ and $s^{(3)}, s^{(4)} \in \{t_0, t_1\}^*$ recursively by

$$\begin{aligned}
\tilde{\lambda} &= \tilde{\mathcal{E}}\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right) \\
\tilde{\lambda}(a) &= \lambda(a) \text{ if } a \in D_0 \cup R_0 \\
\begin{pmatrix} \tilde{\lambda}(a') \\ \tilde{\lambda}(a) \end{pmatrix} &= \begin{cases}
s^{(1)}_{\beta_1(a)} & \text{if } a, a' \in U_{N,N} \setminus (D_0 \cup R_0) \text{ and} \\
& \tilde{\lambda}(a_1) = \tilde{\lambda}(a_2) = \tilde{\lambda}(a_3) = 0 \\
s^{(2)}_{\beta_2(a)} & \text{if } a, a' \in U_{N,N} \setminus (D_0 \cup R_0) \text{ and} \\
& \tilde{\lambda}(a_1) = 1 \text{ and } \tilde{\lambda}(a_2) = \tilde{\lambda}(a_3) = 0 \\
s^{(3)}_{\beta_3(a)} & \text{if } a, a' \in U_{N,N} \setminus (D_0 \cup R_0) \text{ and} \\
& \tilde{\lambda}(a_1) = 0 \text{ and } (\tilde{\lambda}(a_2) = 1 \text{ or } \tilde{\lambda}(a_3) = 1) \\
s^{(4)}_{\beta_4(a)} & \text{if } a, a' \in U_{N,N} \setminus (D_0 \cup R_0) \text{ and} \\
& \tilde{\lambda}(a_1) = 1 \text{ and } (\tilde{\lambda}(a_2) = 1 \text{ or } \tilde{\lambda}(a_3) = 1)
\end{cases}
\end{aligned}$$

$$\begin{aligned}
\beta_1(a) &= 1 + \Big|\Big\{b \in (U_{N,N} \cap T_1) \setminus (D_0 \cup R_0) : \\
&\qquad b \prec a, \tilde{\lambda}(b_1) = \tilde{\lambda}(b_2) = \tilde{\lambda}(b_3) = 0\Big\}\Big| \\
\beta_2(a) &= 1 + \Big|\Big\{b \in (U_{N,N} \cap T_1) \setminus (D_0 \cup R_0) : \\
&\qquad b \prec a, \tilde{\lambda}(b_1) = 1, \tilde{\lambda}(b_2) = \tilde{\lambda}(b_3) = 0\Big\}\Big| \\
\beta_3(a) &= 1 + \Big|\Big\{b \in (U_{N,N} \cap T_1) \setminus (D_0 \cup R_0) : \\
&\qquad b \prec a, \tilde{\lambda}(b_1) = 0, \Big(\tilde{\lambda}(b_2) = 1 \text{ or } \tilde{\lambda}(b_3) = 1\Big)\Big\}\Big| \\
\beta_4(a) &= 1 + \Big|\Big\{b \in (U_{N,N} \cap T_1) \setminus (D_0 \cup R_0) : \\
&\qquad b \prec a, \tilde{\lambda}(b_1) = 1, \Big(\tilde{\lambda}(b_2) = 1 \text{ or } \tilde{\lambda}(b_3) = 1\Big)\Big\}\Big| .
\end{aligned}$$

If $\begin{pmatrix} \tilde{\lambda}(a') \\ \tilde{\lambda}(a) \end{pmatrix} = t_i$, then the point a is labeled with the lower bit of t_i and a' is labeled with the upper bit of t_i. The number $\beta_k(a)$ is one more than the number of symbols previously copied from the string $s^{(k)}$ into $U_{N,N}$. Let

$$\begin{aligned} V_{N,\lambda} \;\; = \;\; & \Big\{ \left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right) \in \{t_0, t_1, t_2\}^* \times \{t_0, t_1, t_2\}^* \times \{t_0, t_1\}^* \times \{t_0, t_1\}^* : \\ & \; l(s^{(k)}) = \beta_k(a_b) - 1, \text{ for } k = 1, 2, 3, 4 \Big\} \end{aligned}$$

be the set of strings $\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$ that together fit perfectly into $U_{N,N}$ under the mapping $\tilde{\mathcal{E}}$. Then $V_{N,\lambda}$ is a prefix code and the mapping

$$\tilde{\mathcal{E}} : V_{N,\lambda} \longrightarrow L(U_{N,N}) \tag{6.5}$$

is a word encoder as defined in (6.3).

To encode an input, $\tilde{\mathcal{E}}$ first initializes the boundary elements of $U_{N,N}$, and then in each iteration it labels an internal point-pair (a, a') of $U_{N,N}$. The points (a, a') are labeled before the points (b, b') if and only if $a \prec b$. The input string used to label a particular internal point-pair (a, a') of $U_{N,N}$ is chosen to guarantee that the labeling of $U_{N,N}$ satisfies the hard-triangle constraint.

The encoder $\tilde{\mathcal{E}}$ is invertible. The inverse mapping scans the elements of $U_{N,N}$ in increasing order with respect to $\prec$ to recover the input sequences. A pseudo-code description of the encoder is given in Table 6.4. Note that $\tilde{\mathcal{E}}$ is completely determined by N and the initial labeling λ.

In Section 6.5.1 we define a fixed-to-variable length encoder that labels the set $U_{X,N}$, where the parameter X is a function of the input $\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$. The set $U_{X,N}$ is decomposed into multiple translates of $U_{N,N}$ and some additional solid diagonals D_k and hollow diagonals D'_k, which allows $U_{X,N}$ to grow large enough to accom-

Table 6.4: The variable-to-fixed length hard-triangle constrained encoder $\tilde{\mathcal{E}}$. The algorithm maps the finite input strings $s^{(1)}, \ldots , s^{(4)}$ from $V_{N,\lambda}$ into a hard-triangle constrained labeling of $U_{N,N}$.

1.	Initialize the elements of $D_0 \cup R_0$ using λ. Let $m_1 = 1$, $m_2 = 1$, $m_3 = 1$, $m_4 = 1$.
2.	Let $a = \min_{\prec}\{b \in U_{N,N} \cap T_1 : b$ is unlabeled$\}$.
3.	If both a_2 and a_3 are labeled with 0
4.	If a_1 is labeled with 0
5.	Label (a, a') with $s^{(1)}_{m_1}$. Let $m_1 = m_1 + 1$.
6.	Else
7.	Label (a, a') with $s^{(2)}_{m_2}$. Let $m_2 = m_2 + 1$.
8.	Else
9.	If a_1 is labeled with 0
10.	Label (a, a') with $s^{(3)}_{m_3}$. Let $m_3 = m_3 + 1$.
11.	Else
12.	Label (a, a') with $s^{(4)}_{m_4}$. Let $m_4 = m_4 + 1$.
13.	If all of $U_{N,N}$ is labeled then stop, else go to 2.

modate certain long input strings. Then in Section 6.5.2 we use the fixed-to-variable length encoder to define a variable-to-variable length encoder. The variable-to-variable length encoder maps binary sequences to $U_{X,N}$ and is nearly a fixed-to-fixed length encoder. This allows precise mathematical analysis of its coding rate.

6.5 Hard-triangle capacity lower bound

Using a finite complete binary prefix code defined in Section 6.5.2, a binary sequence w is parsed into a sequence of variable length strings $w^{(1)}, w^{(2)}, \ldots$. Each string $w^{(i)}$ in the prefix code is mapped into a hard-triangle constrained labeling of the set $U_{X(w^{(i)}),N}$. Then, the points of $A_{X(w^{(i)}),N} \setminus U_{X(w^{(i)}),N}$ are filled with 0s to ensure that any two arrays $A_{X(w^{(i)}),N}$ can be placed next to each other (as in Figure 6.6) without violating the hard-triangle constraint. Henceforth we abbreviate $X(w^{(i)})$ with X. An analysis of the algorithm's coding rate is given in Section 6.5.3 by taking the input to be the $1/2$-sequence $\hat{w}$.

6.5.1 An intermediate fixed-to-variable length encoder

For $i \in \{0, \ldots, \tau-1\}$ define the following translations of $U_{N,N}$, its boundary diagonal D_0, boundary row R_0, and an arbitrary $u \in T$:

$$\begin{aligned} U_{N,N}^{(i)} &= U_{N,N} + i[N+1, 0] \\ D_0^{(i)} &= D_0 + i[N+1, 0] \\ R_0^{(i)} &= R_0 + i[N+1, 0] \\ u^{(i)} &= u - i[N+1, 0]. \end{aligned}$$

Let γ_1, γ_2, γ_3, γ_4 be positive reals. Let $\tau \in \mathbf{Z}^+$, called the *number of translates*, and let $s^{(1)} \in \{t_0, t_1, t_2\}^{\lfloor \tau\gamma_1 N^2 \rfloor}$, $s^{(2)} \in \{t_0, t_1, t_2\}^{\lfloor \tau\gamma_2 N^2 \rfloor}$, $s^{(3)} \in \{t_0, t_1\}^{\lfloor \tau\gamma_3 N^2 \rfloor}$, $s^{(4)} \in \{t_0, t_1\}^{\lfloor \tau\gamma_4 N^2 \rfloor}$. Let

$$J_\tau = \bigcup_{i=0}^{\tau-1} (D_0^{(i)} \cup R_0^{(i)})$$

be a union of boundaries of the first τ translates of $U_{N,N}$, and let $\lambda : J_\tau \longrightarrow \{0,1\}$ be an *initial labeling* of J_τ. For each $i \in \{0, 1, \ldots, \tau-1\}$ let $\lambda_i : D_0^{(i)} \cup R_0^{(i)} \longrightarrow \{0,1\}$ be the restriction of λ to the set $D_0^{(i)} \cup R_0^{(i)}$; that is $\lambda_i(u) = \lambda(u)$ for all $u \in D_0^{(i)} \cup R_0^{(i)}$. Let $\sigma^{(1)} \in \{t_0, t_1, t_2\}^{\tau N^2}$, $\sigma^{(2)} \in \{t_0, t_1, t_2\}^{\tau N^2}$, $\sigma^{(3)} \in \{t_0, t_1\}^{\tau N^2}$, $\sigma^{(4)} \in \{t_0, t_1\}^{\tau N^2}$, which are called *auxiliary sequences*.

For each i, the labeling λ_i induces a prefix code V_{N,λ_i} based on the labeling of $U_{N,N}^{(i)}$ as in (6.5). The sequence of prefix codes $V_{N,\lambda_0}, \ldots, V_{N,\lambda_{\tau-1}}$ induces a partition of the concatenation of the input and auxiliary strings as

$$\begin{aligned}
&\left(s^{(1)}\sigma^{(1)}, s^{(2)}\sigma^{(2)}, s^{(3)}\sigma^{(3)}, s^{(4)}\sigma^{(4)}\right) \\
&\quad = \left(z^{(1,1)}, z^{(1,2)}, z^{(1,3)}, z^{(1,4)}\right)\left(z^{(2,1)}, z^{(2,2)}, z^{(2,3)}, z^{(2,4)}\right)\ldots \\
&\qquad \left(z^{(\tau-1,1)}, z^{(\tau-1,2)}, z^{(\tau-1,3)}, z^{(\tau-1,4)}\right)\left(\tilde{z}^{(1)}, \tilde{z}^{(2)}, \tilde{z}^{(3)}, \tilde{z}^{(4)}\right)\left(\bar{z}^{(1)}, \bar{z}^{(2)}, \bar{z}^{(3)}, \bar{z}^{(4)}\right)
\end{aligned} \tag{6.6}$$

where $\left(z^{(i,1)}, z^{(i,2)}, z^{(i,3)}, z^{(i,4)}\right) \in V_{N,\lambda_i}$, $\tilde{z}^{(k)}$ is a suffix of $s^{(k)}$, and $\bar{z}^{(k)}$ is a suffix of $\sigma^{(k)}$ for $k = 1, 2, 3, 4$. Let

$$X = \tau(N+1) + \left\lceil \frac{1}{\lceil N/2 \rceil} \sum_{k=1}^{4} l(\tilde{z}^{(k)}) \right\rceil$$

and note that $U_{X,N}$ can be decomposed as (see Figure 6.8)

$$U_{X,N} = \left(\bigcup_{i=0}^{\tau-1} U_{N,N}^{(i)} \right) \cup \left(\bigcup_{j=1}^{\tau} D'_{j(N+1)} \right) \cup D_{\tau(N+1)} \cup \left(\bigcup_{k=\tau(N+1)+1}^{X} (D_k \cup D'_k) \right).$$

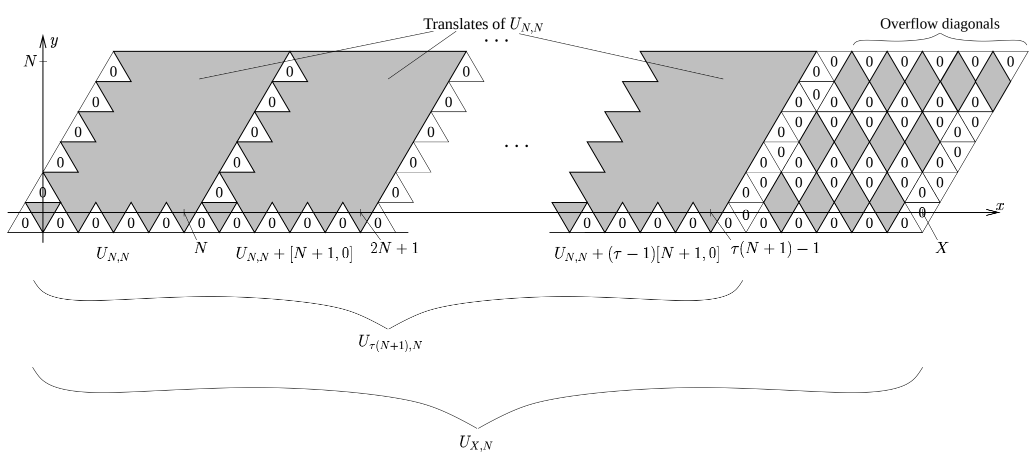


Figure 6.8: The set $U_{X,N}$ and the zero padding $A_{X,N} \setminus U_{X,N}$ is shown. The set $U_{X,N}$ consists of τ translates of $U_{N,N}$, a hollow diagonal of padding 0s after each translate, and some (possible) overflow solid and hollow diagonals.

The translates $U_{N,N}^{(i)}$ are filled with the ith 4-tuple in the parsing (6.6); the hollow diagonals $D'_{j(N+1)}$ and the solid diagonal $D_{\tau(N+1)}$ are padded with 0s; and some additional "overflow" solid diagonals D_k and hollow diagonals D'_k are filled with the $\tilde{z}_k$'s and padding 0s. The string $\tilde{z}^{(k)}$ is the suffix of $s^{(k)}$ that does not get encoded into the translates $U^{(i)}_{N,N}$.

This is formalized by defining a fixed-to-variable length encoder

$$\bar{\mathcal{E}} : \{t_0,t_1,t_2\}^{\lfloor \tau\gamma_1 N^2 \rfloor} \times \{t_0,t_1,t_2\}^{\lfloor \tau\gamma_2 N^2 \rfloor} \times \{t_0,t_1\}^{\lfloor \tau\gamma_3 N^2 \rfloor} \times \{t_0,t_1\}^{\lfloor \tau\gamma_4 N^2 \rfloor} \longrightarrow \bigcup_{S \subset T} L(S)$$

for which we specify the labeling $\bar{\mathcal{E}}\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right) : U_{X,N} \longrightarrow \{0,1\}$ by:

$$\begin{aligned}
\bar{\lambda} &= \bar{\mathcal{E}}\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right) \\
\bar{\lambda}(u) &= \begin{cases} \lambda(u) & \text{if } u \in J_\tau \\ 0 & \text{if } u \in \{[\tau(N+1),N],\ldots,[X,N]\} \cup \bigcup_{j=1}^{\tau-1} D'_{j(N+1)} \end{cases}
\end{aligned}$$

$$\begin{pmatrix} \bar{\lambda}(u') \\ \bar{\lambda}(u) \end{pmatrix} = \begin{cases} \begin{pmatrix} \tilde{\mathcal{E}}\left(z^{(i,1)},z^{(i,2)},z^{(i,3)},z^{(i,4)}\right)(u^{(i)\prime}) \\ \tilde{\mathcal{E}}\left(z^{(i,1)},z^{(i,2)},z^{(i,3)},z^{(i,4)}\right)(u^{(i)}) \end{pmatrix} & \begin{array}{l}\text{if } u \text{ is an internal point} \\ \text{of } U^{(0)}_{N,N} \cup \cdots \cup U^{(\tau-1)}_{N,N}\end{array} \\[2ex] \left(\tilde{z}^{(1)}\tilde{z}^{(2)}\tilde{z}^{(3)}\tilde{z}^{(4)}\right)_{\delta(u)} & \begin{array}{r}\text{if } u = [x,y] \in U_{X,N}, \\ x \in \{\tau(N+1)+1,\ldots,X\}, \\ N-1-y \text{ is even, and} \\ \delta(u) \le l\left(\tilde{z}^{(1)}\tilde{z}^{(2)}\tilde{z}^{(3)}\tilde{z}^{(4)}\right)\end{array} \\[2ex] t_0 & \begin{array}{l}\text{if } u \text{ is any other} \\ \text{internal point of } U_{X,N}\end{array} \end{cases}$$

where $\tilde{\mathcal{E}}$ is the encoder defined in Section 6.4.1 with the initial labeling λ_i if $u \in U^{(i)}_{N,N}$; the string $\tilde{z}^{(1)}\tilde{z}^{(2)}\tilde{z}^{(3)}\tilde{z}^{(4)}$ is the concatenation of the suffixes of $s^{(1)},\ldots,s^{(4)}$ that did not get encoded into the first τ translates of $U_{N,N}$; $\begin{pmatrix} \bar{\lambda}(u') \\ \bar{\lambda}(u) \end{pmatrix} = t_i$ means that the points u and u' are labeled with the lower and upper bits of the symbol t_i, respectively; and

$$\delta(u) = (x - (\tau(N+1)+1)) \lceil N/2 \rceil + \frac{N-1-y}{2} + 1.$$

The default case when $\begin{pmatrix} \bar{\lambda}(u') \\ \bar{\lambda}(u) \end{pmatrix} = t_0$ (i.e. zero padding) includes the internal points which lie on the solid diagonal $D_{\tau(N+1)}$ and hollow diagonal $D'_{\tau(N+1)}$ that separate $U^{(\tau-1)}_{N,N}$ and the overflow diagonals, as well as those internal points $u = [x, y] \in U_{X,N}$ and u' for which $x \in \{\tau(N+1)+1, \ldots, X\}$ and either $N-1-y$ is odd or else $\delta(u) > l\left(\tilde{z}^{(1)}\tilde{z}^{(2)}\tilde{z}^{(3)}\tilde{z}^{(4)}\right)$.

The encoder $\bar{\mathcal{E}}$ is completely determined by N, $\gamma_1, \ldots, \gamma_4$, λ, τ, and $\sigma^{(1)}, \ldots,$ $\sigma^{(4)}$ and will serve as the second stage of a variable-to-variable length hard-triangle constrained encoder to be defined in Section 6.5.2. Note that $\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$ can be recovered from the labeling of $U_{X,N}$.

The process of encoding the input $\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$ is described in detail below. The points of J_τ are assigned a fixed initial labeling λ (to be determined from Lemma 6.2). The translates $U^{(i)}_{N,N}$, for $i = 0, 1, \ldots, \tau-1$, are labeled with the symbols of $s^{(1)}, \ldots, s^{(4)}$ using the variable-to-fixed length encoder $\tilde{\mathcal{E}}$ for each translate with the fixed initial labeling defined by λ_i. The hollow diagonals $D'_{j(N+1)}$ for $j = 1, 2, \ldots, \tau$ and the solid diagonal $D_{\tau(N+1)}$ are filled with 0s.

Labeling all τ translates of $U_{N,N}$ using the encoder $\tilde{\mathcal{E}}$, and adding the padding hollow diagonals after each translate defines a labeling of the set $U_{\tau(N+1),N}$. For each i, $\tilde{\mathcal{E}}$ is a variable-to-fixed length encoder, so it is possible that to encode the strings $s^{(1)}, \ldots,$ $s^{(4)}$ might require either more or less space in T than just the set $U_{\tau(N+1),N}$. Suppose all $\lfloor \tau\gamma_1 N^2 \rfloor$ symbols of the string $s^{(1)}$ have already been encoded. In this case the auxiliary sequence $\sigma^{(1)}$ is appended as a suffix to the string $s^{(1)}$ to be encoded. Similarly, the auxiliary sequences $\sigma^{(2)}$, $\sigma^{(3)}$, $\sigma^{(4)}$ are added as suffixes to the strings $s^{(2)}$, $s^{(3)}$, $s^{(4)}$, if necessary, to complete the labeling of $U_{\tau(N+1),N}$. Note that τN^2 is an upper bound on the number of auxiliary symbols used.

On the other hand, it is possible that suffixes of the input strings $s^{(1)}$, $s^{(2)}$, $s^{(3)}$, or $s^{(4)}$ do not get encoded into $U_{\tau(N+1),N}$. In this case any remaining symbols of $s^{(1)}, \ldots,$ $s^{(4)}$ (in this order) are copied into the overflow diagonals until all symbols of $s^{(1)}, \ldots,$ $s^{(4)}$ are encoded.

To ensure that the labeling of the overflow diagonals is valid, the symbols are written into every second position, and are separated by 0s (see Figure 6.8). The points $[\tau(N+1)+1, N], \ldots, [X, N]$ (i.e. the remaining elements of the boundary row of $U_{X,N}$) are also filled with 0s. The last solid and hollow diagonals used to encode the last input symbols may contain unlabeled points which are then labeled with 0s. The parameter X is defined to be the index of the last solid diagonal generated by the encoding procedure.

In Section 6.5.3 we will choose $\gamma_1, \ldots, \gamma_4$ and τ to guarantee that the sequences $s^{(1)}, \ldots, s^{(4)}$ fill up $U_{\tau(N+1),N}$ almost perfectly (for large N and small ϵ), and therefore the number of overflow diagonals added will be small.

For given τ, $\gamma_1, \ldots, \gamma_4$, N, input strings $s^{(1)}, \ldots, s^{(4)}$, auxiliary sequences $\sigma^{(1)}, \ldots, \sigma^{(4)}$, and initial labeling λ of J_τ, and for each $i = 0, 1, \ldots, \tau - 1$, let

$$\begin{aligned} Q_i^{(k)} &= \text{number of symbols of } s^{(k)}\sigma^{(k)} \text{ that } \bar{\mathcal{E}} \text{ maps into } U_{N,N}^{(i)} \\ q_k\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right) &= \left\lfloor \tau\gamma_k N^2 \right\rfloor - \sum_{i=0}^{\tau-1} Q_i^{(k)}. \end{aligned}$$

The quantities $Q_i^{(k)}$ and $q_k\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$ are determined by the parameters N, $\gamma_1, \ldots, \gamma_4$, λ, τ, $\sigma^{(1)}, \ldots, \sigma^{(4)}$ and $s^{(1)}, \ldots, s^{(4)}$. If positive, $q_k\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$ is the number of symbols of $s^{(k)}$ that do not get mapped into $U_{\tau(N+1)-1,N}$, and otherwise

$q_k\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right)$ is minus the number of symbols of $\sigma^{(k)}$ that get mapped into $U_{\tau(N+1)-1,N}$. For the fixed initial labeling λ, fixed auxiliary sequences $\sigma^{(1)},\ldots,\sigma^{(4)}$, and for any $\epsilon,\gamma_1,\ldots,\gamma_4>0$, let

$$\begin{aligned} B &= \Big\{\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right)\in \\ &\quad \{t_0,t_1,t_2\}^{\lfloor\tau\gamma_1N^2\rfloor}\times\{t_0,t_1,t_2\}^{\lfloor\tau\gamma_2N^2\rfloor}\times\{t_0,t_1\}^{\lfloor\tau\gamma_3N^2\rfloor}\times\{t_0,t_1\}^{\lfloor\tau\gamma_4N^2\rfloor}: \\ &\quad q_i\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right)<\tau\epsilon,\ i=1,2,3,4\Big\}. \end{aligned} \tag{6.7}$$

The set B represents the 4-tuples $\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right)$ that "fit well" into $U_{\tau(N+1),N}$. For each k the fraction of symbols of $s^{(k)}$ that are not mapped into $U_{\tau(N+1),N}$ is smaller than about $\frac{\epsilon}{\gamma_k N^2}$. The set B is determined by N, $\gamma_1,\ldots,\gamma_4$, λ, τ, ϵ, $\sigma^{(1)},\ldots,\sigma^{(4)}$.

6.5.2 A variable-to-variable length encoder

For any sequence $s\in\{t_0,t_1,t_2\}^*$ and $k=0,1,2$, let $|s|_k$ denote the number of occurrences of t_k in s. The following definition extends those in [7, p. 51]. For any ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, $\psi_1\in[0,1]$ such that $1-\phi_0-\phi_1$, $1-\phi_2-\phi_3\in[0,1]$, and for any $\epsilon>0$, an *ϵ-typical set* (with blocklengths $\lfloor\tau\gamma_1N^2\rfloor$, $\lfloor\tau\gamma_2N^2\rfloor$, $\lfloor\tau\gamma_3N^2\rfloor$, $\lfloor\tau\gamma_4N^2\rfloor$) is defined as

$$\begin{aligned} A=\Big\{ &\left(s^{(1)},s^{(2)},s^{(3)},s^{(4)}\right)\in \\ &\{t_0,t_1,t_2\}^{\lfloor\tau\gamma_1N^2\rfloor}\times\{t_0,t_1,t_2\}^{\lfloor\tau\gamma_2N^2\rfloor}\times\{t_0,t_1\}^{\lfloor\tau\gamma_3N^2\rfloor}\times\{t_0,t_1\}^{\lfloor\tau\gamma_4N^2\rfloor}: \\ &2^{-\lfloor\tau\gamma_1N^2\rfloor(H_3(\phi_0,\phi_1)+\epsilon)}\le\phi_0^{|s^{(1)}|_0}\cdot\phi_1^{|s^{(1)}|_1}\cdot(1-\phi_0-\phi_1)^{|s^{(1)}|_2}\le 2^{-\tau\gamma_1N^2(H_3(\phi_0,\phi_1)-\epsilon)} \\ &2^{-\lfloor\tau\gamma_2N^2\rfloor(H_3(\phi_2,\phi_3)+\epsilon)}\le\phi_2^{|s^{(2)}|_0}\cdot\phi_3^{|s^{(2)}|_1}\cdot(1-\phi_2-\phi_3)^{|s^{(2)}|_2}\le 2^{-\tau\gamma_2N^2(H_3(\phi_2,\phi_3)-\epsilon)} \\ &2^{-\lfloor\tau\gamma_3N^2\rfloor(H_2(\psi_0)+\epsilon)}\le\psi_0^{|s^{(3)}|_0}\cdot(1-\psi_0)^{|s^{(3)}|_1}\le 2^{-\tau\gamma_3N^2(H_2(\psi_0)-\epsilon)} \\ &2^{-\lfloor\tau\gamma_4N^2\rfloor(H_2(\psi_1)+\epsilon)}\le\psi_1^{|s^{(4)}|_0}\cdot(1-\psi_1)^{|s^{(4)}|_1}\le 2^{-\tau\gamma_4N^2(H_3(\psi_1)-\epsilon)}\Big\}. \end{aligned}$$

The term $\phi_0^{|s^{(1)}|_0} \cdot \phi_1^{|s^{(1)}|_1} \cdot (1-\phi_0-\phi_1)^{|s^{(1)}|_2}$ is the probability of a (ϕ_0, ϕ_1)-sequence $\hat{s}^{(1)}$ of length $\lfloor \tau\gamma_1 N^2 \rfloor$ being equal to $s^{(1)} \in \{t_0, t_1, t_2\}^{\lfloor \tau\gamma_1 N^2 \rfloor}$. Similarly, the terms $\phi_2^{|s^{(2)}|_0} \cdot \phi_3^{|s^{(2)}|_1} \cdot (1-\phi_2-\phi_3)^{|s^{(2)}|_2}$, $\psi_0^{|s^{(3)}|_0} \cdot (1-\psi_0)^{|s^{(3)}|_1}$, $\psi_1^{|s^{(4)}|_0} \cdot (1-\psi_1)^{|s^{(4)}|_1}$ correspond to the probability of a (ϕ_2, ϕ_3)-sequence, a τ_0-sequence, and a τ_1-sequence of lengths $\lfloor \tau\gamma_2 N^2 \rfloor$, $\lfloor \tau\gamma_3 N^2 \rfloor$, $\lfloor \tau\gamma_4 N^2 \rfloor$, respectively. The set A is determined by the parameters N, $\gamma_1, \ldots, \gamma_4$, τ, $\phi_0, \ldots, \phi_3$, ψ_0, ψ_1.

Let G be a binary complete prefix code of cardinality $|A \cap B|$, whose codewords are of one of two possible lengths.[2] Let

$$g : G \to A \cap B$$

be any bijection. Both G and g are determined by the parameters N, $\gamma_1, \ldots, \gamma_4$, λ, τ, ϵ, $\phi_0, \ldots, \phi_3$, ψ_0, ψ_1, $\sigma^{(1)}, \ldots, \sigma^{(4)}$.

The code G parses an infinite binary input sequence, and g maps a finite parsed string to a 4-tuple of ϵ-typical sequences $\left(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}\right)$ that is likely to fit into τ translates of $U_{N,N}$.

Since G is a complete prefix code, a binary sequence w can uniquely be parsed into strings $w^{(1)}, w^{(2)}, \ldots$ such that $w^{(i)} \in G$ for all i. The *variable-to-variable length hard-triangle constrained encoder* $\mathcal{E}$ is defined as the composition

$$\mathcal{E} = \bar{\mathcal{E}} \circ g.$$

The encoder $\mathcal{E}$ is completely determined by the parameters N, $\gamma_1, \ldots, \gamma_4$, λ, τ, ϵ, $\phi_0, \ldots, \phi_3$, ψ_0, ψ_1, $\sigma^{(1)}, \ldots, \sigma^{(4)}$.

Each string $w^{(i)} \in G$ of the parsed sequence w is transformed into the typical, well-fitting strings $\left(s^{(i,1)},\ s^{(i,2)}, s^{(i,3)},\ s^{(i,4)}\right) \in A \cap B$ by the bijection g, and then $\left(s^{(i,1)},\right.$

[2] For any $i \geq 2$ there exists a complete prefix code with i codewords, all of length $\lfloor \log_2 i \rfloor$ or $\lfloor \log_2 i \rfloor + 1$.

$s^{(i,2)},\ s^{(i,3)},\ s^{(i,4)}\big)$ is mapped into a hard-triangle constrained labeling of $U_{X,N}$ using the encoder $\bar{\mathcal{E}}$. The *variable-to-variable length hard-triangle constrained algorithm* consists of the mapping $\mathcal{E}$ and its inverse. The mapping $\mathcal{E}$ is referred to as the algorithm's encoder, and the inverse is called the algorithm's decoder. The decoding part of the algorithm consists of the inverse mappings.

6.5.3 Coding rate analysis

Let $\hat{s}^{(1)}$ be a (ϕ_0, ϕ_1)-sequence and $\hat{s}^{(2)}$ a (ϕ_2, ϕ_3)-sequence on the alphabet $\{t_0,\ t_1,\ t_2\}$. Let $\hat{s}^{(3)}$ be a ψ_0-sequence and $\hat{s}^{(4)}$ a ψ_1-sequence on the alphabet $\{t_0, t_1\}$ (see Figure 6.9).

Sequence	$\hat{s}^{(1)}$			$\hat{s}^{(2)}$			$\hat{s}^{(3)}$		$\hat{s}^{(4)}$	
Symbols	$\frac{0}{0}$ t_0	$\frac{0}{1}$ t_1	$\frac{1}{0}$ t_2	$\frac{0}{0}$ t_0	$\frac{0}{1}$ t_1	$\frac{1}{0}$ t_2	$\frac{0}{0}$ t_0	$\frac{0}{1}$ t_1	$\frac{0}{0}$ t_0	$\frac{0}{1}$ t_1
Probabilities	ϕ_0	ϕ_1	$1-\phi_0-\phi_1$	ϕ_2	ϕ_3	$1-\phi_2-\phi_3$	ψ_0	$1-\psi_0$	ψ_1	$1-\psi_1$

Figure 6.9: The symbol alphabets of the sequences $\hat{s}^{(1)}$, $\hat{s}^{(2)}$, $\hat{s}^{(3)}$, $\hat{s}^{(4)}$ and the corresponding probability of each symbol.

Let the variable-to-fixed length encoder $\tilde{\mathcal{E}}$ map $\hat{s}^{(1)}, \ldots, \hat{s}^{(4)}$ into a labeling of $U_{N,N}$ using a random initial labeling $\hat{\lambda} : D_0 \cup R_0 \longrightarrow \{0,1\}$. For $k = 1,2,3,4$ let

$$\begin{aligned}\gamma_k \quad &= \quad \mathbf{P}\Big(\text{the internal point-pair } (a, a') \text{ is labeled with a symbol}\\ &\qquad\quad \text{from the sequence } \hat{s}^{(k)} \text{ by } \tilde{\mathcal{E}}\Big)\,.\end{aligned}$$

The random initial labeling $\hat{\lambda}$ is called a *standard initialization corresponding to* ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, ψ_1 if for $k = 1,2,3,4$ and for every internal point-pair (a, a'), the quantity

γ_k is independent of a and N. The corresponding labeling of $U_{N,N}$ is called a *standard labeling corresponding to* ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, ψ_1. Let

$$\Omega = \Big\{ (\phi_0, \phi_1, \phi_2, \phi_3, \psi_0, \psi_1) : \text{there exists an initialization of } D_0 \cup R_0 \\ \text{such that the labeling of } U_{N,N} \text{ by } \tilde{\mathcal{E}} \text{ is a standard labeling} \Big\} .$$

It is shown in Section 6.5.4 that Ω is nonempty.

A fixed initial labeling λ of J_τ and auxiliary sequences $\sigma^{(1)}, \ldots, \sigma^{(4)}$ used by $\mathcal{E}$ are implied in the following lemma. The set B in the lemma is defined in (6.7).

Lemma 6.2. *Let $\hat{s}^{(1)} \in \{t_0, t_1, t_2\}^{\lfloor \tau\gamma_1 N^2 \rfloor}$ be a (ϕ_0, ϕ_1)-sequence, $\hat{s}^{(2)} \in \{t_0, t_1, t_2\}^{\lfloor \tau\gamma_2 N^2 \rfloor}$ a (ϕ_2, ϕ_3)-sequence, $\hat{s}^{(3)} \in \{t_0, t_1\}^{\lfloor \tau\gamma_3 N^2 \rfloor}$ a ψ_0-sequence, and $\hat{s}^{(4)} \in \{t_0, t_1\}^{\lfloor \tau\gamma_4 N^2 \rfloor}$ a ψ_1-sequence, for some $(\phi_0,\ \phi_1,\ \phi_2,\ \phi_3,\ \psi_0,\ \psi_1) \in \Omega$. For any $N \in \mathbf{Z}^+$ and any $\epsilon > 0$ there exists $\tau_0 \in \mathbf{Z}^+$ such that for every $\tau \geq \tau_0$ there is an initial labeling $\lambda : J_\tau \longrightarrow \{0, 1\}$ and auxiliary sequences $\sigma^{(1)}, \ldots, \sigma^{(4)}$ whose corresponding set B satisfies*

$$\mathbf{P}\left(\left(\hat{s}^{(1)}, \hat{s}^{(2)}, \hat{s}^{(3)}, \hat{s}^{(4)} \right) \in B \right) > 1 - \epsilon. \tag{6.8}$$

Proof. Suppose that $\hat{s}^{(1)}, \ldots, \hat{s}^{(4)}$ are encoded into $U_{X,N}$ by $\bar{\mathcal{E}}$ where initial labels are defined by a random labeling $\hat{\lambda} : J_\tau \longrightarrow \{0, 1\}$, which is a standard initialization corresponding to ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, ψ_1 on each translate of $U_{N,N}$. Let $\hat{\sigma}^{(1)} \in \{t_0, t_1, t_2\}^{\tau N^2}$ be an auxiliary (ϕ_0, ϕ_1)-sequence, $\hat{\sigma}^{(2)} \in \{t_0, t_1, t_2\}^{\tau N^2}$ an auxiliary (ϕ_2, ϕ_3)-sequence, $\hat{\sigma}^{(3)} \in \{t_0, t_1\}^{\tau N^2}$ an auxiliary ψ_0-sequence, and $\hat{\sigma}^{(4)} \in \{t_0, t_2\}^{\tau N^2}$ an auxiliary ψ_1-sequence. We will demonstrate that there is at least one fixed initial labeling $\lambda : J_\tau \longrightarrow \{0, 1\}$ and fixed auxiliary sequences $\sigma^{(1)}, \ldots, \sigma^{(4)}$ such that (6.8) holds.

Since each $U^{(i)}_{N,N}$ is randomly initialized by $\hat{\lambda}$, the labeling of each $U^{(i)}_{N,N}$ is a standard labeling, and thus by the definition of γ_k, we have

$$E\left[\hat{Q}_i^{(k)}\right] = \gamma_k N^2$$

for every $i \in \{0, 1, \ldots, \tau - 1\}$ and $k = 1, 2, 3, 4$. For every $k \in \{1, 2, 3, 4\}$ and any two distinct $i \in \{0, \ldots, \tau - 1\}$, the random variables $\hat{Q}_i^{(k)}$ are independent and have finite variances (independent of τ). Therefore the weak law of large numbers implies that for every $\epsilon > 0$ and $j = 1, 2, 3, 4$,

$$\lim_{\tau \to \infty} \mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}_i^{(j)} - \gamma_j N^2\right| < \epsilon\right) = 1. \tag{6.9}$$

The random variables $\hat{Q}_i^{(j)}$ in (6.9) are functions of the random input sequences $\hat{s}^{(1)}, \ldots,$ $\hat{s}^{(4)}$, the random auxiliary sequences $\hat{\sigma}^{(1)}, \ldots, \hat{\sigma}^{(4)}$, and the random initialization $\hat{\lambda}$. It follows from (6.9) that

$$\lim_{\tau \to \infty} \mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}_i^{(j)} - \gamma_j N^2\right| < \epsilon, \text{ for all } j = 1, 2, 3, 4\right) = 1. \tag{6.10}$$

Then (6.10) and the inequalities

$$\begin{aligned}
&\mathbf{P}\left(\left|\frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}_i^{(j)} - \gamma_j N^2\right| < \epsilon, \text{ for all } j = 1, 2, 3, 4\right) \\
&\quad \leq \quad \mathbf{P}\left(\tau\gamma_j N^2 - \sum_{i=0}^{\tau-1} \hat{Q}_i^{(j)} < \tau\epsilon, \text{ for all } j = 1, 2, 3, 4\right) \\
&\quad \leq \quad \mathbf{P}\left(\left\lfloor \tau\gamma_j N^2 \right\rfloor - \sum_{i=0}^{\tau-1} \hat{Q}_i^{(j)} < \tau\epsilon, \text{ for all } j = 1, 2, 3, 4\right) \\
&\quad = \quad \mathbf{P}\left(q_j\left(\hat{s}^{(1)}, \hat{s}^{(2)}, \hat{s}^{(3)}, \hat{s}^{(4)}\right) < \tau\epsilon, \text{ for all } j = 1, 2, 3, 4\right)
\end{aligned}$$

imply that

$$\lim_{\tau\to\infty} \mathbf{P}\left(q_j\left(\hat{s}^{(1)}, \hat{s}^{(2)}, \hat{s}^{(3)}, \hat{s}^{(4)}\right) < \tau\epsilon, \text{ for all } j = 1,2,3,4\right) = 1. \tag{6.11}$$

It follows from (6.11) that there exists τ_0 such that for all $\tau \geq \tau_0$,

$$\mathbf{P}\left(q_j\left(\hat{s}^{(1)}, \hat{s}^{(2)}, \hat{s}^{(3)}, \hat{s}^{(4)}\right) < \tau\epsilon, \text{ for all } j = 1,2,3,4\right) > 1-\epsilon.$$

Thus for every τ there must exist at least one initial labeling λ and auxiliary sequences $\sigma^{(1)}, \ldots, \sigma^{(4)}$ (all five depending on τ) such that

$$\mathbf{P}\left(q_j\left(\hat{s}^{(1)}, \hat{s}^{(2)}, \hat{s}^{(3)}, \hat{s}^{(4)}\right) < \tau\epsilon, \text{ for all } j = 1,2,3,4 \middle| \lambda, \sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \sigma^{(4)}\right) > 1-\epsilon \tag{6.12}$$

where the conditioning in (6.12) is on the event that the random initial labeling $\hat{\lambda}$ equals λ, and the random auxiliary sequences $\hat{\sigma}^{(1)}, \ldots, \hat{\sigma}^{(4)}$ equal $\sigma^{(1)}, \ldots, \sigma^{(4)}$, respectively. Equivalently, this gives

$$\mathbf{P}\left(\left(\hat{s}^{(1)}, \hat{s}^{(2)}, \hat{s}^{(3)}, \hat{s}^{(4)}\right) \in B\right) > 1-\epsilon$$

for every $\tau \geq \tau_0$. $\square$

A number r_0 is said to be an *achievable coding rate* of a hard-triangle constrained encoder $\mathcal{E}$ if

$$r_0 = \lim_{\epsilon\to 0} \lim_{N\to\infty} r\left(\mathcal{E}\right).$$

Theorem 6.3. *The hard-triangle constrained algorithm achieves a coding rate of*

$$r_0 = \frac{1}{2}\left(\gamma_1 H_3(\phi_0,\phi_1) + \gamma_2 H_3(\phi_2,\phi_3) + \gamma_3 H_2(\psi_0) + \gamma_4 H_2(\psi_1)\right). \tag{6.13}$$

Proof. The (ϕ_0,ϕ_1)-sequence $\hat{s}^{(1)} \in \{t_0,t_1,t_2\}^{\lfloor \tau\gamma_1 N^2\rfloor}$, the (ϕ_2,ϕ_3)-sequence $\hat{s}^{(2)} \in \{t_0,t_1,t_2\}^{\lfloor \tau\gamma_2 N^2\rfloor}$, the ψ_0-sequence $\hat{s}^{(3)} \in \{t_0,t_1\}^{\lfloor \tau\gamma_3 N^2\rfloor}$, and the ψ_1-sequence $\hat{s}^{(4)} \in \{t_0,t_1\}^{\lfloor \tau\gamma_4 N^2\rfloor}$ are independent, and therefore

$$\mathbf{P}\left(\left(\hat{s}^{(1)},\hat{s}^{(2)},\hat{s}^{(3)},\hat{s}^{(4)}\right)\in A\right) > (1-\epsilon)^4$$

for $\tau \geq \tau_0$ large enough (see [7, pp. 51-52]). Hence, using Lemma 6.2,

$$\begin{aligned}
&(1-\epsilon)^4-\epsilon\\
&\quad\leq \mathbf{P}\left(\left(\hat{s}^{(1)},\hat{s}^{(2)},\hat{s}^{(3)},\hat{s}^{(4)}\right)\in A\cap B\right)\\
&\quad\leq \sum_{(s^{(1)},s^{(2)},s^{(3)},s^{(4)})\in A\cap B} 2^{-\tau N^2(\gamma_1(H_3(\phi_0,\phi_1)-\epsilon)+\gamma_2(H_3(\phi_2,\phi_3)-\epsilon)+\gamma_3(H_2(\psi_0)-\epsilon)+\gamma_4(H_2(\psi_1)-\epsilon))}\\
&\quad= |A\cap B|\cdot 2^{-\tau N^2(\gamma_1(H_3(\phi_0,\phi_1)-\epsilon)+\gamma_2(H_3(\phi_2,\phi_3)-\epsilon)+\gamma_3(H_2(\psi_0)-\epsilon)+\gamma_4(H_2(\psi_1)-\epsilon))}
\end{aligned}$$

which implies

$$|A\cap B| \geq \left((1-\epsilon)^4-\epsilon\right)\cdot 2^{\tau N^2(\gamma_1(H_3(\phi_0,\phi_1)-\epsilon)+\gamma_2(H_3(\phi_2,\phi_3)-\epsilon)+\gamma_3(H_2(\psi_0)-\epsilon)+\gamma_4(H_2(\psi_1)-\epsilon))}. \tag{6.14}$$

Similarly,

$$\begin{aligned}
1 &\geq \mathbf{P}\left(\left(\hat{s}^{(1)},\hat{s}^{(2)},\hat{s}^{(3)},\hat{s}^{(4)}\right)\in A\cap B\right)\\
&\geq \sum_{(s^{(1)},s^{(2)},s^{(3)},s^{(4)})\in A\cap B} 2^{-\tau N^2(\gamma_1(H_3(\phi_0,\phi_1)+\epsilon)+\gamma_2(H_3(\phi_2,\phi_3)+\epsilon)+\gamma_3(H_2(\psi_0)+\epsilon)+\gamma_4(H_2(\psi_1)+\epsilon))}\\
&= |A\cap B|\cdot 2^{-\tau N^2(\gamma_1(H_3(\phi_0,\phi_1)+\epsilon)+\gamma_2(H_3(\phi_2,\phi_3)+\epsilon)+\gamma_3(H_2(\psi_0)+\epsilon)+\gamma_4(H_2(\psi_1)+\epsilon))}
\end{aligned}$$

which implies

$$|A \cap B| \leq 2^{\tau N^2(\gamma_1(H_3(\phi_0,\phi_1)+\epsilon)+\gamma_2(H_3(\phi_2,\phi_3)+\epsilon)+\gamma_3(H_2(\psi_0)+\epsilon)+\gamma_4(H_2(\psi_1)+\epsilon))}. \tag{6.15}$$

Every string $z \in G$ has length $\lfloor \log_2 |A \cap B| \rfloor$ or one longer. Recall that $U_{X,N}$ is extended to $A_{X,N}$ by zero padding, and $A_{X,N}$ occupies $2(X+1)(N+1)$ points of T. Therefore (6.14), the definition of the set B, and $\sum_{z \in G} \mathbf{P}(z) = 1$ imply that the coding rate is lower bounded as

$$\begin{aligned} r(\mathcal{E}) &= \sum_{z \in G} \mathbf{P}(z) \frac{l(z)}{2(X+1)(N+1)} \\ &\geq \frac{\lfloor \log_2 |A \cap B| \rfloor}{2(\tau(N+1)+1)(N+1) + \frac{2N+2}{2N+1}(16\tau\epsilon + 2N + 1)} \sum_{z \in G} \mathbf{P}(z) \\ &\geq \frac{\log_2((1-\epsilon)^4 - \epsilon) - 1}{2(\tau(N+1)+1)(N+1) + \frac{2N+2}{2N+1}(16\tau\epsilon + 2N + 1)} \\ &\quad + \frac{\lfloor \tau\gamma_1 N^2 \rfloor \cdot (H_3(\phi_0,\phi_1) - \epsilon) + \lfloor \tau\gamma_2 N^2 \rfloor \cdot (H_3(\phi_2,\phi_3) - \epsilon)}{2(\tau(N+1)+1)(N+1) + \frac{2N+2}{2N+1}(16\tau\epsilon + 2N + 1)} \\ &\quad + \frac{\lfloor \tau\gamma_3 N^2 \rfloor \cdot (H_2(\psi_0) - \epsilon) + \lfloor \tau\gamma_4 N^2 \rfloor \cdot (H_2(\psi_1) - \epsilon))}{2(\tau(N+1)+1)(N+1) + \frac{2N+2}{2N+1}(16\tau\epsilon + 2N + 1)}. \end{aligned} \tag{6.16}$$

The term $16\tau\epsilon + 2N + 1$ in the denominator is an upper bound on the number of points occupied by the input symbols and zero padding in the overflow diagonals of $U_{X,N}$, and the factor $\frac{2N+2}{2N+1}$ accounts for the padding 0s in $A_{X,N} \setminus U_{X,N}$ in the overflow diagonals. Using (6.15) and the fact that $X \geq \tau(N+1)$ for any input string $z \in G$, the coding rate is upper bounded as

$$\begin{aligned} r(\mathcal{E}) &= E\left[\frac{l(z)}{2(X+1)(N+1)}\right] \\ &\leq \frac{\lceil \log_2 |A \cap B| \rceil}{2(\tau(N+1)+1)(N+1)} \\ &\leq \frac{\tau\gamma_1 N^2(H_3(\phi_0,\phi_1)+\epsilon)+\tau\gamma_2 N^2(H_3(\phi_2,\phi_3)+\epsilon)}{2(\tau(N+1)+1)(N+1)} \\ &\quad + \frac{\tau\gamma_3 N^2(H_2(\psi_0)+\epsilon)+\tau\gamma_4 N^2(H_2(\psi_1)+\epsilon)+1}{2(\tau(N+1)+1)(N+1)}. \end{aligned} \tag{6.17}$$

Taking limits first as $N \to \infty$ and then $\epsilon \to 0$, the theorem follows from (6.16) and (6.17). $\square$

For the parameters N, ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, ψ_1, ϵ the value of τ is induced by Lemma 6.2 and Theorem 6.3. For a fixed value of τ, the existence of the initial labeling λ and the auxiliary sequences $\sigma^{(1)}, \dots, \sigma^{(4)}$ used by the encoder $\mathcal{E}$ is given in Lemma 6.2. The parameters $\gamma_1, \dots, \gamma_4$ are calculated in (6.29)-(6.32).

6.5.4 Coding rate maximization

In this section we consider the labeling of the set $U_{N,N}$ by $\tilde{\mathcal{E}}$, where the input sequences $\hat{s}^{(1)}, \dots, \hat{s}^{(4)}$ are the random sequences introduced in Section 6.5.3, and where we construct a specific random initial labeling $\hat{\lambda} : D_0 \cup R_0 \longrightarrow \{0,1\}$ to be defined in what follows. The random labels of the points $[i,j], [i,j]' \in U_{N,N}$ are denoted by $\hat{F}(i,j)$ and $\hat{F}(i,j)'$, respectively. We will also use $\hat{F}(a)$ and $\hat{F}(a)'$ to denote the random labels of the points $a = [i,j]$ and $a' = [i,j]'$.

The elements of D_0 are assigned labels from the stationary homogeneous Markov chain $\hat{\mu}^{(1)}$ and the elements of R_0 are assigned labels from the stationary homogeneous Markov chain $\hat{\mu}^{(2)}$ (see Figures 6.10a and 6.10b).

The transition probabilities π_1, π_2, π_3, π_4 are constrained such that the stationary

(a): $\hat{\mu}^{(1)}$ (b): $\hat{\mu}^{(2)}$

Figure 6.10: The homogeneous Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$.

distributions of the Markov chains $\hat{\mu}^{(1)}$ and $\hat{\mu}^{(2)}$ are the same; thus the labeling of D_0 determines the label $\hat{F}(0, N)$, which is used to initiate the labeling of R_0. Furthermore, in this section we show that the parameters ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, ψ_1, π_1, π_2, π_3, π_4 can be chosen such that the resulting labeling of $U_{N,N}$ is a standard labeling, i.e. each γ_k takes on a constant value for every internal point-pair $([i,j],[i,j]')$. Specifically, we prove that Conditions (i), (ii), and (iii) below imply that γ_k is constant for every internal point of $U_{N,N}$, and in Appendix 6.7 we show that the parameters can be chosen to satisfy Conditions (i)-(iii).

Condition (i): The probability distribution of the random variables $\Big(\hat{F}(0, N-1), \hat{F}(1, N-1)\Big)$ is identical to the probability distribution of the random variables $\Big(\hat{F}(0, N), \hat{F}(1, N)\Big)$.

Condition (ii): The probability distribution of the random variables $\Big(\hat{F}(1, N), \hat{F}(1, N-1)\Big)$ is identical to the probability distribution of the random variables $\Big(\hat{F}(0, N), \hat{F}(0, N-1)\Big)$.

Condition (iii): The random variables $\hat{F}(0, N-1)$ and $\hat{F}(1, N)$ are conditionally independent given $\hat{F}(1, N-1)$.

The points $[1, N-1]$, $[0, N-1]$, $[0, N]$, $[1, N]$ appearing in Conditions (i)-(iii) are illustrated in Figure 6.11.

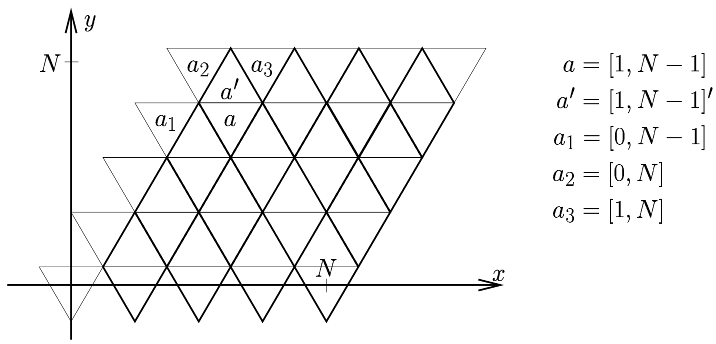


Figure 6.11: The points appearing in Conditions (i)-(iii).

Lemma 6.4. *Let $b \in D_1 \setminus R_0$. If Condition (i) holds, then the probability distribution of the random variables $\left(\hat{F}(b),\ \hat{F}(b'), \hat{F}(b_1), \hat{F}(b_2),\ \hat{F}(b_3)\right)$ is identical to the probability distribution of $\left(\hat{F}(1, N-1),\ \hat{F}(1, N-1)', \hat{F}(0, N-1), \hat{F}(0, N),\ \hat{F}(1, N)\right)$.*

Proof. The lemma is trivially true for $b = [1, N-1]$. Suppose the lemma is true for $b = [1, k+1]$ (where $0 \leq k < N-1$). We show that it holds for $b = [1, k]$. Let $v_1, v_2, \ldots, v_5 \in \{0, 1\}$. Then

$$\begin{aligned}
&\mathbf{P}\left(\hat{F}(b) = v_1, \hat{F}(b') = v_2, \hat{F}(b_1) = v_3, \hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\
&= \mathbf{P}\left(\hat{F}(b) = v_1, \hat{F}(b') = v_2 \middle| \hat{F}(b_1) = v_3, \hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\
&\quad \cdot \mathbf{P}\left(\hat{F}(b_1) = v_3 \middle| \hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\
&\quad \cdot \mathbf{P}\left(\hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right). \qquad (6.18)
\end{aligned}$$

In what follows we rewrite each of the factors of (6.18). First, by the definition of the encoder $\tilde{\mathcal{E}}$,

$$\begin{aligned}
&\mathbf{P}\left(\hat{F}(b) = v_1, \hat{F}(b') = v_2 \middle| \hat{F}(b_1) = v_3, \hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\
&= \mathbf{P}\Big(\hat{F}(1, N-1) = v_1, \hat{F}(1, N-1)' = v_2 \Big| \\
&\qquad \hat{F}(0, N-1) = v_3, \hat{F}(0, N) = v_4, \hat{F}(1, N) = v_5\Big). \qquad (6.19)
\end{aligned}$$

Furthermore, since the labels of D_0 form a homogeneous Markov chain,

$$\begin{aligned} & \mathbf{P}\left(\hat{F}(b_1) = v_3 \middle| \hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\ = & \ \mathbf{P}\left(\hat{F}(0, N-1) = v_3 \middle| \hat{F}(0, N) = v_4, \hat{F}(1, N) = v_5\right). \end{aligned} \tag{6.20}$$

Finally,

$$\begin{aligned} & \mathbf{P}\left(\hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\ = & \ \mathbf{P}\left(\hat{F}(0, N-1) = v_4, \hat{F}(1, N-1) = v_5\right) \qquad (6.21) \\ = & \ \mathbf{P}\left(\hat{F}(0, N) = v_4, \hat{F}(1, N) = v_5\right) \qquad (6.22) \end{aligned}$$

where (6.21) follows from the induction hypothesis, and Condition (i) implies (6.22). Combining (6.18) with (6.19), (6.20), and (6.22) yields

$$\begin{aligned} & \mathbf{P}\left(\hat{F}(b) = v_1, \hat{F}(b') = v_2, \hat{F}(b_1) = v_3, \hat{F}(b_2) = v_4, \hat{F}(b_3) = v_5\right) \\ = & \ \mathbf{P}\left(\hat{F}(1, N-1) = v_1, \hat{F}(1, N-1)' = v_2 \middle| \right. \\ & \qquad \left. \hat{F}(0, N-1) = v_3, \hat{F}(0, N) = v_4, \hat{F}(1, N) = v_5\right) \\ & \cdot \mathbf{P}\left(\hat{F}(0, N-1) = v_3 \middle| \hat{F}(0, N) = v_4, \hat{F}(1, N) = v_5\right) \\ & \cdot \mathbf{P}\left(\hat{F}(0, N) = v_4, \hat{F}(1, N) = v_5\right) \\ = & \ \mathbf{P}\left(\hat{F}(1, N-1) = v_1, \hat{F}(1, N-1)' = v_2, \hat{F}(0, N-1) = v_3, \hat{F}(0, N) = v_4, \right. \\ & \qquad \left. \hat{F}(1, N) = v_5\right). \end{aligned}$$

□

Corollary 6.5. *If Condition (i) holds, then the sequence of labels* $\left\{\left(\hat{F}(0, k), \hat{F}(1, k)\right)\right\}_{k=N}^{0}$ *forms a stationary homogeneous Markov chain.*

Proof. The fact that the sequence $\left\{\left(\hat{F}(0, k), \hat{F}(1, k)\right)\right\}_{k=N}^{0}$ is a Markov chain follows from the definition of $\tilde{\mathcal{E}}$. Lemma 6.4 implies that this Markov chain is stationary and homogeneous. □

Lemma 6.6. *If Conditions (i)-(iii) hold, then the labels of D_1 form a stationary homogeneous Markov chain identical to the labels of D_0.*

Proof. For $k \in \{0, \ldots, N-1\}$, let $b = [1, k]$ and $b_3 = [1, k+1]$ be two neighboring points on the solid diagonal D_1. By Lemma 6.4, the joint probability distribution of $\left(\hat{F}(b), \hat{F}(b_3)\right)$ is identical to the joint probability distribution of $\left(\hat{F}(1, k-1), \hat{F}(1, k)\right)$. This fact combined with Condition (ii) implies that the probability distribution of $\left(\hat{F}(b), \hat{F}(b_3)\right)$ is identical to the probability distribution of $\left(\hat{F}(0, k-1), \hat{F}(0, k)\right)$. Therefore, to prove the lemma it suffices to show that the labels of D_1 form a Markov chain.

The sequence $\left\{\hat{F}(1, k)\right\}_{k=N}^{0}$ is a Markov chain if and only if the reverse sequence $\left\{\hat{F}(1, k)\right\}_{k=0}^{N}$ is a Markov chain. We show that $\left\{\hat{F}(1, k)\right\}_{k=0}^{N}$ is a Markov chain. Let $v \in \{0,1\}$, and $2 \leq k \leq N$. Then,

$$\begin{aligned}
&\mathbf{P}\left(\hat{F}(1,k) = v \middle| \hat{F}(1,k-1), \ldots, \hat{F}(1,0)\right) \\
&= \sum_{u_0,\ldots,u_{k-1} \in \{0,1\}} \mathbf{P}\left(\hat{F}(1,k) = v \middle| \hat{F}(1,k-1), \ldots, \hat{F}(1,0), \right. \\
&\qquad\qquad \left. \hat{F}(0,k-1) = u_{k-1}, \ldots, \hat{F}(0,0) = u_0\right) \\
&\qquad\qquad \cdot \mathbf{P}\left(\hat{F}(0,k-1) = u_{k-1}, \ldots, \hat{F}(0,0) = u_0 \middle| \hat{F}(1,k-1), \ldots, \hat{F}(1,0)\right) \\
&= \sum_{u_0,\ldots,u_{k-1} \in \{0,1\}} \mathbf{P}\left(\hat{F}(1,k) = v \middle| \hat{F}(1,k-1), \hat{F}(0,k-1) = u_{k-1}\right) \\
&\qquad\qquad \cdot \mathbf{P}\left(\hat{F}(0,k-1) = u_{k-1}, \ldots, \hat{F}(0,0) = u_0 \middle| \hat{F}(1,k-1), \ldots, \hat{F}(1,0)\right) \qquad (6.23) \\
&= \sum_{u_0,\ldots,u_{k-1} \in \{0,1\}} \mathbf{P}\left(\hat{F}(1,k) = v \middle| \hat{F}(1,k-1)\right) \\
&\qquad\qquad \cdot \mathbf{P}\left(\hat{F}(0,k-1) = u_{k-1}, \ldots, \hat{F}(0,0) = u_0 \middle| \hat{F}(1,k-1), \ldots, \hat{F}(1,0)\right) \qquad (6.24) \\
&= \mathbf{P}\left(\hat{F}(1,k) = v \middle| \hat{F}(1,k-1)\right)
\end{aligned}$$

where (6.23) is obtained using Corollary 6.5; and (6.24) follows from Condition (iii) and Lemma 6.4. $\square$

Lemma 6.7. *Let $b \in D_i \setminus R_0$ for $i \in \{1, \ldots, N\}$. If Conditions (i)-(iii) hold, then the probability distribution of $\left(\hat{F}(b),\ \hat{F}(b'), \hat{F}(b_1), \hat{F}(b_2),\ \hat{F}(b_3)\right)$ is identical to the probability distribution of $\left(\hat{F}(1,N-1),\ \hat{F}(1,N-1)', \hat{F}(0,N-1), \hat{F}(0,N),\ \hat{F}(1,N)\right)$.*

Proof. The lemma has been proved for $b \in D_1 \setminus R_0$ in Lemma 6.4. Suppose that $b \in D_2 \setminus R_0$. By Lemma 6.6, the probability distribution of the labels of D_1 is identical to the probability distribution of the labels of D_0. Since the labels of R_0 form a stationary homogeneous Markov chain, it follows that Lemma 6.4, Corollary 6.5, and Lemma 6.6 can be generalized to the solid diagonals D_1 and D_2. Repeatedly applying this idea to

Table 6.5: The numerical values of the parameters that maximize r_0.

π_1	π_2	π_3	π_4
0.780481	0.674449	0.781640	0.670889

ϕ_0	ϕ_1	ϕ_2	ϕ_3	ψ_0	ψ_1
0.414654	0.170691	0.376711	0.246578	0.707065	0.578868

γ_1	γ_2	γ_3	γ_4
0.460253	0.129451	0.294192	0.116104

consecutive solid diagonals proves the lemma. □

Theorem 6.8. *Let $b \in U_{N,N} \cap T_1$ be an internal point of $U_{N,N}$. If Conditions (i)-(iii) hold, then the probability distribution of $\left(\hat{F}(b_1), \hat{F}(b_2), \hat{F}(b_3)\right)$ is independent of b and N.*

Proof. Follows from Lemma 6.7. □

Corollary 6.9. *If Conditions (i)-(iii) hold, then γ_k is a constant for every internal point-pair and for all $k = 1, 2, 3, 4$.*

Conditions (i)-(iii) translate into a set of equations for the parameters π_1, π_2, π_3, π_4, ϕ_0, ϕ_1, ϕ_2, ϕ_3, ψ_0, and ψ_1 listed in Appendix 6.7. Optimization of (6.13) subject to equations (6.25)-(6.32) gives the numerical results in Table 6.5.

Theorem 6.10. *The hard-triangle constrained encoder $\mathcal{E}$ achieves a coding rate of $r_0 = 0.628831217$, which is within 1% of the capacity.*

Proof. Substituting the numerical values of Table 6.5 into (6.13) gives $r_0 = 0.628831217$. Using the upper bound on C_T from Section 6.3 gives

$$\frac{C_T - r_0}{C_T} = 1 - \frac{r_0}{C_T} \leq 1 - \frac{0.628831217}{0.634775895} < 0.009365003 < 1\%.$$

□

Theorem 6.10 and (6.2) give our main result, as summarized in the following corollary.

Corollary 6.11. *The capacity C_T of the hard-triangle constraint is bounded as*

$$0.628831217 \leq C_T \leq 0.634775895.$$

6.6 Acknowledgments

The authors thank Alexander Vardy for suggesting this problem.

This chapter, in full, has been submitted for publication as: Zs. Nagy and K. Zeger, Capacity Bounds for the Hard-Triangle Model, *IEEE Trans. Inform. Theory*, November 2002. The dissertation author was the primary investigator of this paper.

Appendix

6.7 Equations used to obtain the numerical values of Table 6.5

Recall that $a = [1, N-1]$, $a_1 = [0, N-1]$, $a_2 = [0, N]$, $a_3 = [1, N]$ as shown in Figure 6.11. Let $\mathbf{P}\left(\hat{F}(0,N) = 0\right) = \alpha$. Corresponding to the labelings of a_2, a_3 and the labelings of a_1, a there are 4 equations implied by Condition (i). These equations, generated by $\mathbf{P}\left(\left(\hat{F}(a_2), \hat{F}(a_3)\right) = \left(\hat{F}(a_1), \hat{F}(a)\right) = (v_1, v_2)\right)$ are given below for $(v_1, v_2) =$

$$
\begin{aligned}
0\,0: \quad & \alpha\pi_3 = \pi_2\psi_0 + \alpha(\pi_1\pi_3(1-\phi_1) - \pi_2\psi_0 + \pi_1(1-\pi_3)\psi_0) \\
0\,1: \quad & \alpha - \alpha\pi_3 = \pi_2 - \pi_2\psi_0 + \alpha(\pi_1(1-\psi_0 - \pi_3(1-\phi_1-\psi_0)) - \pi_2(1-\psi_0)) \\
1\,0: \quad & \pi_4 - \alpha\pi_4 = \psi_1 - \pi_2\psi_1 + \alpha((\pi_2-\pi_1)\psi_1 + (1-\pi_1)\pi_3(1-\phi_3-\psi_1)) \\
1\,1: \quad & (1-\alpha)(1-\pi_4) = \ (1-\pi_2)(1-\psi_1) + \alpha(\pi_2 - \pi_2\psi_1 - \pi_3(1-\phi_3-\psi_1) \\
& \qquad\qquad + \pi_1(\psi_1 + \pi_3(1-\phi_3-\psi_1) - 1)).
\end{aligned}
$$

Corresponding to the labelings of a_2, a_1 and the labelings of a_4, a there are 4 equations implied by Condition (ii). These equations, generated by $\mathbf{P}\left(\left(\hat{F}(a_2), \hat{F}(a_1)\right) = \left(\hat{F}(a_4), \hat{F}(a)\right) = (v_1, v_2)\right)$ are given below for $(v_1, v_2) =$

$$
\begin{aligned}
0\,0: \quad & \alpha\pi_1 = \ \pi_2\pi_4\psi_0 + \alpha(\pi_3 - \pi_1\pi_3\phi_1 - \pi_3\phi_3 + \pi_1\pi_3\phi_3 - \pi_2\pi_4\psi_0) \\
& \qquad + (1-\alpha)(1-\pi_2)\pi_4\psi_1 \\
0\,1: \quad & \alpha - \alpha\pi_1 = \alpha\pi_3(\pi_1(\phi_1-\phi_3) + \phi_3) + (1-\alpha)\pi_4(1-\pi_2(\psi_0-\psi_1) - \psi_1) \\
1\,0: \quad & \pi_2 - \alpha\pi_2 = \ \alpha(\pi_1 - \pi_1\pi_3 - \pi_2(1-\pi_4))\psi_0 \\
& \quad + \alpha(\pi_2 - \pi_1(1-\pi_3) - \pi_3 + \pi_4 - \pi_2\pi_4)\psi_1 + (1-\pi_4)(\pi_2(\psi_0-\psi_1) + \psi_1)
\end{aligned}
$$

$$
\begin{aligned}
1\,1: \quad & (1-\alpha)(1-\pi_2) = (1-\pi_4)(1-\pi_2(\psi_0-\psi_1)-\psi_1) + \alpha(\pi_4 - \pi_1\psi_0 + \pi_2\psi_0 \\
& \quad -\pi_2\pi_4\psi_0 + (\pi_1 - \pi_2 - (1-\pi_2)\pi_4)\psi_1 - \pi_3(1-\pi_1(\psi_0-\psi_1)-\psi_1)).
\end{aligned}
$$

The equations corresponding to Condition (iii) are of the form

$$
\frac{\mathbf{P}\left(\left(\hat{F}(a_3), \hat{F}(a), \hat{F}(a_1)\right) = (v_1, v_2, 0)\right)}{\mathbf{P}\left(\left(\hat{F}(a), \hat{F}(a_1)\right) = (v_2, 0)\right)} = \frac{\mathbf{P}\left(\left(\hat{F}(a_3), \hat{F}(a), \hat{F}(a_1)\right) = (v_1, v_2, 1)\right)}{\mathbf{P}\left(\left(\hat{F}(a), \hat{F}(a_1)\right) = (v_2, 1)\right)}
$$

where $v_1, v_2 \in \{0, 1\}$. The list of equations is given below for $(v_1, v_2) =$

$$
\begin{aligned}
0\,0: \quad & \frac{\pi_2\pi_4\psi_0+\alpha(\pi_1\pi_3(1-\phi_1)-\pi_2\pi_4\psi_0)}{\pi_2\psi_0+\alpha(\pi_1\pi_3(1-\phi_1)-\pi_2\psi_0+\pi_1(1-\pi_3)\psi_0)} = \frac{\alpha(1-\pi_1)\pi_3(1-\phi_3)+(1-\alpha)(1-\pi_2)\pi_4\psi_1}{\psi_1-\pi_2\psi_1+\alpha((\pi_2-\pi_1)\psi_1+(1-\pi_1)\pi_3(1-\phi_3-\psi_1))} \\
0\,1: \quad & \frac{\alpha\pi_1\pi_3\phi_1+(1-\alpha)\pi_2\pi_4(1-\psi_0)}{\pi_2-\pi_2\psi_0-\alpha(\pi_2(1-\psi_0)-\pi_1(1-\psi_0-\pi_3(1-\phi_1-\psi_0)))} \\
& \qquad = \frac{\alpha(1-\pi_1)\pi_3\phi_3+(1-\alpha)(1-\pi_2)\pi_4(1-\psi_1)}{(1-\pi_2)(1-\psi_1)+\alpha(\pi_2-\pi_2\psi_1-\pi_3(1-\phi_3-\psi_1)-\pi_1(1-\psi_1-\pi_3(1-\phi_3-\psi_1)))} \\
1\,0: \quad & \frac{\alpha\pi_1(1-\pi_3)\psi_0+(1-\alpha)\pi_2(1-\pi_4)\psi_0}{\pi_2\psi_0+\alpha(\pi_1\pi_3(1-\phi_1)-\pi_2\psi_0+\pi_1(1-\pi_3)\psi_0)} = \frac{\alpha(1-\pi_1)(1-\pi_3)\psi_1+(1-\alpha)(1-\pi_2)(1-\pi_4)\psi_1}{\psi_1-\pi_2\psi_1+\alpha((\pi_2-\pi_1)\psi_1+(1-\pi_1)\pi_3(1-\phi_3-\psi_1))} \\
1\,1: \quad & \frac{(\pi_2(1-\pi_4)-\alpha(\pi_2-\pi_1(1-\pi_3)-\pi_2\pi_4))(1-\psi_0)}{\pi_2-\pi_2\psi_0-\alpha(\pi_2(1-\psi_0)-\pi_1(1-\psi_0-\pi_3(1-\phi_1-\psi_0)))} \\
& \qquad = \frac{(\alpha(1-\pi_1)(1-\pi_3)+(1-\alpha)(1-\pi_2)(1-\pi_4))(1-\psi_1)}{(1-\pi_2)(1-\psi_1)+\alpha(\pi_2-\pi_2\psi_1-\pi_3(1-\phi_3-\psi_1)-\pi_1(1-\psi_1-\pi_3(1-\phi_3-\psi_1)))}.
\end{aligned}
$$

The above set of equations corresponding to Conditions (i)-(iii) can be reduced the following set of independent equations:

$$
\alpha\pi_3 = \pi_2\psi_0 + \alpha(\pi_1\pi_3(1-\phi_1) - \pi_2\psi_0 + \pi_1(1-\pi_3)\psi_0) \tag{6.25}
$$

$$
\pi_4 - \alpha\pi_4 = \psi_1 - \pi_2\psi_1 + \alpha((\pi_2-\pi_1)\psi_1 + (1-\pi_1)\pi_3(1-\phi_3-\psi_1)) \tag{6.26}
$$

$$
\begin{aligned}
\alpha\pi_1 = {} & \pi_2\pi_4\psi_0 + \alpha(\pi_3 - \pi_1\pi_3\phi_1 - \pi_3\phi_3 + \pi_1\pi_3\phi_3 - \pi_2\pi_4\psi_0) \\
& +(1-\alpha)(1-\pi_2)\pi_4\psi_1
\end{aligned} \tag{6.27}
$$

$$
\begin{aligned}
& \frac{\pi_2\pi_4\psi_0 + \alpha(\pi_1\pi_3(1-\phi_1) - \pi_2\pi_4\psi_0)}{\pi_2\psi_0 + \alpha(\pi_1\pi_3(1-\phi_1) - \pi_2\psi_0 + \pi_1(1-\pi_3)\psi_0)} \\
& \quad = \frac{\alpha(1-\pi_1)\pi_3(1-\phi_3) + (1-\alpha)(1-\pi_2)\pi_4\psi_1}{\psi_1 - \pi_2\psi_1 + \alpha((\pi_2-\pi_1)\psi_1 + (1-\pi_1)\pi_3(1-\phi_3-\psi_1))}.
\end{aligned} \tag{6.28}
$$

Note that $D_0 \cap R_0 = [0, N]$, and therefore the stationary distribution of $\hat{\mu}^{(1)}$ and

$\hat{\mu}^{(2)}$ generating the labels of D_0 and R_0 must be identical. Thus α is the probability that any point of $D_0 \cup R_0$ is labeled with 0. The requirement that the labels of D_0 and R_0 form stationary homogeneous Markov chains implies the following equations.

$$\begin{aligned}
\alpha &= \alpha\pi_1 + (1-\alpha)\pi_2 \\
\alpha &= \alpha\pi_3 + (1-\alpha)\pi_4.
\end{aligned}$$

The parameters γ_1, γ_2, γ_3, γ_4 in terms of the other parameters are given as:

$$\gamma_1 = \alpha\pi_1\pi_3 \tag{6.29}$$

$$\gamma_2 = \alpha(1-\pi_1)\pi_3 \tag{6.30}$$

$$\gamma_3 = \pi_2 - \alpha(\pi_2 - \pi_1(1-\pi_3)) \tag{6.31}$$

$$\gamma_4 = 1 - \pi_2 + \alpha(\pi_2 - \pi_3 - \pi_1(1-\pi_3)). \tag{6.32}$$

References

[1] J. J. Ashley and B. H. Marcus. Two-Dimensional Low-Pass Filtering Codes. *IEEE Trans. on Communications*, 46:724–727, June 1998.

[2] R. B. Bapat and T. E. S. Raghavan. *Nonnegative Matrices and Applications.* Cambridge University Press, Cambridge, U.K., 1997.

[3] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982.

[4] R. J. Baxter and S. K. Tsang. Entropy of Hard Hexagons. *J. Physics A*, 13(3):1023–1030, March 1980.

[5] P. Bender and J. K. Wolf. A Universal Algorithm for Generating Optimal and Nearly Optimal Run-Length-Limited, Charge Constrained Binary Sequences. *Proc. 1993 IEEE Intl. Symp. on Inform. Theory*, page 6, San Antonio, TX, January 1993.

[6] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., New York, 1991.

[8] K. Engel. On the Fibonacci Number of an $M \times N$ Lattice. *Fibonacci Quarterly*, 28:72–78, 1990.

[9] W. H. Erxleben and M. W. Marcellin. Error-Correcting Two-Dimensional Modulation Codes. *IEEE Trans. Inform. Theory*, 41:1116–1126, July 1995.

[10] T. Etzion. Cascading methods for runlength-limited arrays. *IEEE Trans. Inform. Theory*, 43:319–324, January 1997.

[11] T. Etzion and K. G. Paterson. Zero/positive Capacities of Two-Dimensional Runlength Constrained Arrays. *Proc. 2001 IEEE Intl. Symp. on Inform. Theory*, page 269, Washington, DC, USA, June 2001.

[12] T. Etzion and A. Vardy. Two-Dimensional Interleaving Schemes With Repetitions: Constructions and Bounds. *IEEE Trans. Inform. Theory*, 48(2):428–457, February 2002.

[13] S. Forchhammer and J. Justesen. Entropy Bounds for Constrained Two-Dimensional Random Fields. *IEEE Trans. Inform. Theory*, 45:118–127, January 1999.

[14] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf. Improved Bit-Stuffing Bounds on 2-Dimensional Constraints. *Proc. 2002 IEEE Intl. Symp. on Inform. Theory*, page 385, Lausanne, Switzerland, June-July 2002.

[15] S. Halevy and R. M. Roth. Parallel Constrained Coding with Application to Two-Dimensional Constraints. *IEEE Trans. Inform. Theory*, 48(5):1009–1020, May 2002.

[16] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, U.K., 1985.

[17] G. S. Joyce. On the Hard-hexagon Model and the Theory of Modular Functions. *Philos. Trans. Roy. Soc. London Ser. A*, 325(1588):643–702, 1988.

[18] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[19] Z. Kukorelly and K. Zeger. The Capacity of Some Hexagonal (d, k) Constraints. *Proc. 2001 IEEE Intl. Symp. on Inform. Theory*, page 64, Washington, DC, June 2001.

[20] M. Larsen. The Problem of the Kings. *Elec. J. of Combinatorics*, 2, 1995.

[21] P. Lee. Combined Error-Correcting/Modulation Recording Codes. *Ph.D. dissertation*, University of California, San Diego, 1988.

[22] Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel. *IEEE Trans. Inform. Theory*, 46(3):1030–1033, May 2000.

[23] Zs. Nagy and K. Zeger. Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints. *IEEE Trans. Inform. Theory* (submitted July 2002).

[24] Zs. Nagy and K. Zeger. Bit Stuffing Algorithms and Analysis for Run Length Constrained Channels in Two and Three Dimensions *IEEE Trans. Inform. Theory* (submitted November 2002).

[25] E. Ordentlich and R. M. Roth. Two-Dimensional Weight-Constrained Codes through Enumeration Bounds. *IEEE Trans. Inform. Theory*, 46(4):1292–1301, July 2000.

[26] M. P. Richey and C. A. Tracy. Equation of State and Isothermal Compressibility for the Hard Hexagon Model in the Disordered Regime. *J. Phys. A*, 20(16):L1121–L1126, November 1987.

[27] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding for a Two-Dimensional Runlength-Limited Constraint. *Proc. of the SPIE - The Intl. Soc. for Optical Engineering*, Denver, CO, 3802:8–17, July 1999.

[28] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding Schemes for the Hard-Square Model. *IEEE Trans. Inform. Theory*, 47:1166–1176, March 2001.

[29] C. E. Shannon. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27(3 and 4):379–423 and 623–656, July and October 1948.

[30] P. H. Siegel and J. K. Wolf. Bit Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays. *Proc. 1998 IEEE Intl. Symp. on Inform. Theory*, page 323, MIT, Cambridge, MA, August 1998.

[31] E. Soljanin. Writing Sequences on the Plane. *IEEE Trans. Inform. Theory*, 48:1344–1354, June 2002.

[32] R.E. Swanson and J. K. Wolf. A New Class of Two-Dimensional Runlength-Limited Recording Codes. *IEEE Trans. on Mag. Recording*, 28:3407–3416, November 1992.

[33] R. Talyansky, T. Etzion, and R. M. Roth. Efficient Code Construction for Certain Two-Dimensional Constraints. *IEEE Trans. Inform. Theory*, 45(2):794–799, March 1999.

[34] C. A. Tracy, L. Grove, and M. F. Newman. Modular Properties of the Hard-hexagon Model. *J. Stat. Phys.*, 48(3 and 4):477–502, August 1987.

[35] B. V. Vasic. Capacity of Channels with Redundant Multitrack (d, k) Constraints: the $k < d$ Case. *IEEE Trans. Inform. Theory*, 42:1546–1548, September 1996.

[36] B. V. Vasic, S. W. McLaughlin, and O. Milenkovic. Shannon Capacity of M-ary Redundant Multitrack Runlength Limited Codes. *IEEE Trans. Inform. Theory*, 44:766–774, March 1998.

[37] W. Weeks and R. E. Blahut. The Capacity and Coding Gain of Certain Checkerboard Codes. *IEEE Trans. Inform. Theory*, 44(3):1193–1203, May 1998.

[38] H. S. Wilf. The Problem of the Kings. *Elec. J. of Combinatorics*, 2, 1995.

Chapter 7

Conclusions and Comments

The mathematical characterization of two and higher dimensional constrained channels is more difficult than the one-dimensional case. Unlike in one dimension, there are no known general formulas for the number of n-dimensional ($n \geq 2$) arrays satisfying a given constraint, or for the value of the capacity associated with the constraint. For (d, k) run length limited constraints in $n \geq 2$ dimensions the exact value of the capacity is only known when it equals zero [11], [13]. In $n = 2, 3$ dimensions, and for some values of d and k (such that $C_{d,k}^{(n)} > 0$), upper and lower bounds on $C_{d,k}^{(n)}$ have been given in [8], [10], [15], [17], [18], [20], and [23]. In particular, the two-dimensional $(0, 1)$ constraint (or, equivalently, $(1, \infty)$ constraint) has been studied by several authors (see [8], [17], [18], [20], and [23]) and $C_{0,1}^{(2)} = C_{1,\infty}^{(2)}$ is now known up to its first nine decimal digits. The ideas in [8] were generalized in [15] to obtain upper and lower bounds on $C_{0,1}^{(3)}$. However, the matrix computations used to establish both the upper and the lower bounds in [8] and [15] rely on the fact that the transfer matrices corresponding to the $(0, 1)$ constraint are symmetric. Since this property does not hold for any other values of d and k (in any dimension), the method of [8] can not be used to obtain numerical bounds on $C_{d,k}^{(n)}$ in general.

The $(1, \infty)$ constraint on a two-dimensional rectangular grid and the $(1, \infty)$ constraint on a two-dimensional hexagonal grid are known in statistical physics as the hard-square and hard-hexagon models, respectively. In a series of intriguing publications Baxter [2], [3], Baxter, Enting, and Tsang [4], and Baxter and Tsang [5] asserted exact solutions to the hard-square and hard-hexagon models in terms of infinite matrix equations and infinite products of polynomials. Based on these results and using the theory of modular functions, Joyce [12] presented a closed formula for the grand partition function per site for the hard-hexagon model. The special case when the activity is $z = 1$ gives the exact solution for the hexagonal $(1, \infty)$ constraint. However, both Baxter's and Joyce's papers are rather difficult to understand in terms of basic principles. An interesting open problem is to present an easy-to-understand derivation for the special case $z = 1$ based on the work by Baxter and Joyce.

Upper and lower bounds for certain two-dimensional checkerboard constraints can be found in [23], and the asymptotic value of the capacity associated with open convex symmetric checkerboard constraints was determined in [16]. The checkerboard constraints can be viewed as generalizations of the one-dimensional (d, ∞) constraints to two dimensions. A question that naturally arises is how one-dimensional (d, k) constraints can be generalized to two-dimensional convex constraints for $k < \infty$. A possible generalization and future research topic is the following. Let $S \subset \mathbf{R}^2$ be a checkerboard constraint, and let $\alpha_d, \alpha_k \in \mathbf{R}$, such that $0 < \alpha_d \leq \alpha_k$. For a labeling $f : \mathbf{Z}^2 \to \{0, 1\}$ of the integer lattice, let X be the collection of lattice points with label 1. We say that f is (S_d, S_k)-constrained, if the translates of $S_d = \alpha_d S$ by all points in X define a packing of $\mathbf{Z}^2$, and the translates of $S_k = \alpha_k S$ by all points in X define a covering of $\mathbf{Z}^2$. The analysis of two-dimensional (S_d, S_k)-constrained channels is a possible future research problem.

Encoding and decoding algorithms for one-dimensional run length limited sequences have been extensively studied. Optimal and nearly optimal variable length codes have been constructed for one-dimensional run length limited, charged constrained sequences [7]. Block codes for one-dimensional (d,k)-constrained channels are presented in [6], [21], and [22]. However, efficient two-dimensional encoders only exist for a few constraints. Encoders for the two-dimensional $(1,\infty)$ constraint have been studied in [17], [18], and [20]. Another possible future research problem is to devise efficient encoding algorithms for two-dimensional (d,k) constraints, and (S_d, S_k) constraints described in the previous paragraph.

Finally, an interesting theoretical problem for future research in the field of (d,k)-constrained codes is to determine the limit $\lim_{n\to\infty} C_{d,k}^{(n)}$ as a function of d and k. There are connections between multidimensional capacities and other works in coding theory and graph theory. The asymptotic number of binary codes with distance 2 was determined by Korshunov and Sapozhenko [14] and Sapozhenko [19], and the asymptotic number of independent sets in a regular graph was found by Alon [1]. In addition, Galvin and Kahn [9] studied the statistical mechanics problem of phase transitions in the hard-core model on $\mathbf{Z}^n$, which is closely related to the n-dimensional $(0,1)$ constraint. Each of the results in [1], [9], [14], and [19] can be used to (independently) conclude that $\lim_{n\to\infty} C_{0,1}^{(n)} = 1/2$ (and therefore also $\lim_{n\to\infty} C_{1,\infty}^{(n)} = 1/2$). We conjecture that in the special cases when $k=\infty$ and when $d=0$, the following hold.

Conjecture 7.1. *For all $d \geq 0$, the capacities of the n-dimensional (d,∞) run length constraint satisfy*

$$\lim_{n\to\infty} C_{d,\infty}^{(n)} = \frac{1}{d+1}.$$

Conjecture 7.2. *For all $k \geq 0$, the capacities of the n-dimensional $(0,k)$ run length constraint satisfy*

$$\lim_{n\to\infty} C_{0,k}^{(n)} = \frac{k}{k+1}.$$

References

[1] N. Alon. Independent Sets in Regular Graphs and Sum-free Subsets of Finite Groups. *Israel Journal of Mathematics*, 73(2):247–256, 1991.

[2] R. J. Baxter. Variational Approximations for Square Lattice Models in Statistical Mechanics. *J. Statist. Phys.*, 19(5):461–478, 1978.

[3] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics.* Academic Press, 1982.

[4] R. J. Baxter, I. G. Enting, and S. K. Tsang. Hard-Square Lattice Gas. *J. Statist. Phys.*, 22(4):465–489, 1980.

[5] R. J. Baxter and S. K. Tsang. Entropy of Hard Hexagons. *J. Physics A*, 13(3):1023–1030, March 1980.

[6] G. F. M. Beenker and K. A. S. Immink. Generalized Method for Encoding and Decoding Run-Length-Limited Binary Sequences. *IEEE Trans. Inform. Theory*, 29:751–754, September 1983.

[7] P. Bender and J. K. Wolf. A Universal Algorithm for Generating Optimal and Nearly Optimal Run-Length-Limited, Charge Constrained Binary Sequences. *Proc. 1993 IEEE Intl. Symp. on Inform. Theory*, San Antonio, TX, page 6, January 1993.

[8] N. J. Calkin and H. S. Wilf. The Number of Independent Sets in a Grid Graph. *SIAM Journal on Discrete Mathematics*, 11:54–60, February 1998.

[9] D. Galvin and J. Kahn. Two Problems Involving the Notion Of Phase Transition. *http://research.microsoft.com/˜galvin/pdf/thesis.pdf*, 2002.

[10] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf. Improved Bit-Stuffing Bounds on 2-Dimensional Constraints. *Proc. 2002 IEEE Intl. Symp. on Inform. Theory*, Lausanne, Switzerland, page 385, June-July 2002.

[11] H. Ito, A. Kato, Zs. Nagy, and K. Zeger. Zero Capacity Region of Multidimensional Run Length Constraints. *Elec. J. Combinatorics*, 6(1)(R33), 1999.

[12] G. S. Joyce. On the Hard-hexagon Model and the Theory of Modular Functions. *Philos. Trans. Roy. Soc. London Ser. A*, 325(1588):643–702, 1988.

[13] A. Kato and K. Zeger. On the Capacity of Two-Dimensional Run Length Constrained Channels. *IEEE Trans. Inform. Theory*, 45(4):1527–1540, July 1999.

[14] A. D. Korshunov and A. A. Sapozhenko. The Number of Binary Codes with Distance 2 (in Russian). *Problemy Kibernetiki*, 40:111–130, 1983.

[15] Zs. Nagy and K. Zeger. Capacity Bounds for the Three-Dimensional (0,1) Run Length Limited Channel. *IEEE Trans. Inform. Theory*, 46(3):1030–1033, May 2000.

[16] Zs. Nagy and K. Zeger. Asymptotic Capacity of Two-Dimensional Channels with Checkerboard Constraints. *IEEE Trans. Inform. Theory* (submitted July 2002).

[17] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding for a Two-Dimensional Runlength-Limited Constraint. *Proc. of the SPIE - The Intl. Soc. for Optical Engineering*, Denver, CO, 3802:8–17, July 1999.

[18] R. M. Roth, P. H. Siegel, and J. K. Wolf. Efficient Coding Schemes for the Hard-Square Model. *IEEE Trans. Inform. Theory*, 47:1166–1176, March 2001.

[19] A. A. Sapozhenko. The Number of Antichains in Ranked Partially Ordered Sets. *Diskret. Mat.*, 1:74–93, 1989. (Russian; translation in *Discrete Math. Appl.*, 1(1):35-58, 1991.)

[20] P. H. Siegel and J. K. Wolf. Bit Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays. *Proc. 1998 IEEE Intl. Symp. on Inform. Theory*, MIT, Cambridge, MA, page 323, August 1998.

[21] D. T. Tang and L. R. Bahl. Block Codes for a Class of Constrained Noiseless Channels. *Information and Control*, 17:436–461, 1970.

[22] J. H. Weber and K. A. S. Abdel-Ghaffar. Cascading Runlength-Limited Sequences. *IEEE Trans. Inform. Theory*, 39:1976–1984, November 1993.

[23] W. Weeks and R. E. Blahut. The Capacity and Coding Gain of Certain Checkerboard Codes. *IEEE Trans. Inform. Theory*, 44(3):1193–1203, May 1998.