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

Zsigmond Nagy and Kenneth Zeger, *Fellow, IEEE*

***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.**

***Index Terms*—Bit allocation, lossy source coding, quantization, transform coding.**

I. Introduction

A binary sequence satisfies the (d,k) run-length constraint if the number of consecutive 0's is at most k, and between any two 1's in the sequence are at least d 0's. A subset of $\boldsymbol{Z}^n$ 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 [1]. 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 [2]–[7]. Three-dimensional constraints were studied in [8] and [9], 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 [6], [11], and [12].

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 [13] and Engel [14], and for three dimensions it was studied in [9]. 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 [15] for the one-dimensional $(0,k)$ constraint. Bit stuffing was then generalized in 1993 by Bender and Wolf [16] to the one-dimensional (d,k) constraint and in 1998 by Siegel and Wolf [17] to the two-dimensional (d,∞) constraint. In 2002, Halevy *et al.* [18] 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 [19]. 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 $\boldsymbol{Z}^2$. In a subsequent paper [20], 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 $\boldsymbol{Z}^2$ in order to facilitate analysis.

The coding rate calculations in [17], [19], and [20] 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 $\boldsymbol{Z}^2$. 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 an 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 $\boldsymbol{Z}^2$, 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 [17], [19], and [20]. 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). While the Roth–Siegel–Wolf algorithm is implementable but not easily analyzable, our modification is analyzable but apparently not easily implementable. However, the strong resemblence between the two algorithms can give confidence in the theoretical basis for the implementable

Manuscript received November 16, 2002; revised March 26, 2004. This work was supported in part by the National Science Foundation and the UCSD Center for Wireless Communications.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093-0407 USA (e-mail: nagy@code.ucsd.edu; zeger@ucsd.edu).

Communicated by K. A. S. Abdel-Ghaffar, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2004.838112

version. We then modify the two-dimensional algorithm to create a three-dimensional algorithm based on bit stuffing that maps an input binary sequence into $\boldsymbol{Z}^3$ 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 three-dimensional algorithm studied, while only marginally stronger (in terms of coding rate) than a trivial algorithm that places information bits in positions whose coordinates sum to an even number, nevertheless demonstrates that improved performance can be achieved with bit stuffing and can be rigorously analyzed as in two dimensions.

The paper is organized as follows. In Section II, basic definitions and terminology are introduced. Two-dimensional bit stuffing is described in Section III and our variable-to-variable length algorithm and analysis are given in Section IV. The two-dimensional coding rate derivation is given in Theorem IV.1. Three-dimensional bit stuffing is described in Section V and our variable-to-variable length algorithm and analysis are given in Section VI. The three-dimensional coding rate result is given in Theorem VI.2 and its maximum value is given in Theorem VI.5. Various tedious calculations are relegated to the appendices.

II. Preliminaries

For any binary string s let $l(s)$ denote its length, $|s|$ the number of 1's in the string, and s_i the ith bit in the string. Let $\boldsymbol{Z}$ denote the integers and $\boldsymbol{Z}^+$ the positive integers. For any $n \in \boldsymbol{Z}^+$, let $\boldsymbol{Z}^n$ 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

$$\begin{aligned} &\boldsymbol{P}(\hat{u}_n = a_n | \hat{u}_{n-1} = a_{n-1}, \ldots, \hat{u}_0 = a_0) \\ &\qquad = \boldsymbol{P}(\hat{u}_n = a_n | \hat{u}_{n-1} = a_{n-1}) \end{aligned}$$

for all $n \in \boldsymbol{Z}^+$ and $a_n \in A$. A Markov chain is *homogeneous* (or *time invariant*) if

$$\boldsymbol{P}(\hat{u}_n = a | \hat{u}_{n-1}) = \boldsymbol{P}(\hat{u}_1 = a | \hat{u}_0)$$

for all $n \in \boldsymbol{Z}^+$ and $a \in A$. For every $a, b \in A$ the conditional probabilities

$$\boldsymbol{P}(\hat{u}_1 = a | \hat{u}_0 = b)$$

of a homogeneous Markov chain are called the *transition probabilities*. A Markov chain is *stationary* if

$$\boldsymbol{P}(\hat{u}_n = a) = \boldsymbol{P}(\hat{u}_0 = a)$$

for all $n \in \boldsymbol{Z}^+$ 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 $S \subset \boldsymbol{Z}^n$, 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], [21] that if $k > d \geq 1$ and $n \geq 2$, then $C_{d,k}^{(n)} = 0 \iff k = d+1$. Numerical upper and lower bounds on $C_{1,\infty}^{(2)}$ were established in [13], and these bounds were later improved in [7] and then in [9]. The best known bounds on $C_{1,\infty}^{(2)}$ agree in the first nine 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 [9] as

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

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 $\boldsymbol{Z}^n$. The lower bound in (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 $\boldsymbol{Z}^n$ such that the constraint is not violated and such that the information source can be perfectly recovered from the labeling of $\boldsymbol{Z}^n$. 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 $\boldsymbol{Z}^n$ 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

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

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 $\boldsymbol{Z}^n$. 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 $\boldsymbol{Z}^n$, 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 $S_v \subset \boldsymbol{Z}^n$. An n-dimensional $(1,\infty)$-constrained *word encoder* is an injection

$$\mathcal{E}_V^{(n)} : V \longrightarrow \bigcup_{S \subset \boldsymbol{Z}^n} \Lambda_{1,\infty}^{(n)}(S) \tag{2}$$

that maps the elements of V into labelings of subsets of $\boldsymbol{Z}^n$. 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 $\boldsymbol{Z}^n$ are a distance 1 apart, then we call them *neighbors*. For any set $S \subset \boldsymbol{Z}^n$, 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

$$\{t_i \in \boldsymbol{Z}^n : i \in \boldsymbol{Z}^+\}$$

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

$$\mathcal{E}^{(n)}(z)(u) = \begin{cases} \mathcal{E}_V^{(n)}(z^{(i)})(u - t_i), & \text{if } u \in t_i + S_{z^{(i)}} \\ 0, & \text{if } u \in t_i + \bar{S}_{z^{(i)}} \\ & \text{and } u \notin S_{z^{(i)}}. \end{cases}$$

That is, $\mathcal{E}^{(n)}(z)$ is a labeling of translates of the sets $S_{z^{(1)}}, S_{z^{(2)}}, \ldots$ composed of the labelings

$$\mathcal{E}_V^{(n)}(z^{(1)}),\ \mathcal{E}_V^{(n)}(z^{(2)}), \ldots.$$

The labeling of points in $\boldsymbol{Z}^n$ 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 $t_1, t_2, \ldots$ such that the composite encoder is injective (i.e., is an encoder).

Define the following quantities for a word encoder:

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

These quantities upper- and lower-bound, respectively, the average ratio between the input length and the number of points in $\boldsymbol{Z}^n$ that are labeled for a particular prefix code V. The probability $\boldsymbol{P}(v)$ 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.

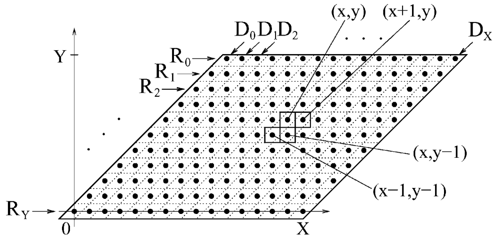


Fig. 1. The parallelogram $\Delta_{X,Y}$ and its diagonals D_i and rows R_j.

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 [22] that the capacity upper-bounds the coding rate.

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 [17] 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 $\boldsymbol{Z}^2$. The main idea is that a string is copied directly into a parallelogram bit-by-bit but skipping over 0's which were added whenever a 1 appeared previously in $z^{(i)}$. For the encoder defined in Section IV, 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.

III. 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, we let $\hat{w}$ be a $1/2$-sequence and $\hat{s}$ a p-sequence. For $X, Y \in \boldsymbol{Z}^+$, $i \in \{0,\ldots,X\}$, and $j \in \{0,\ldots,Y\}$ let

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

[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.

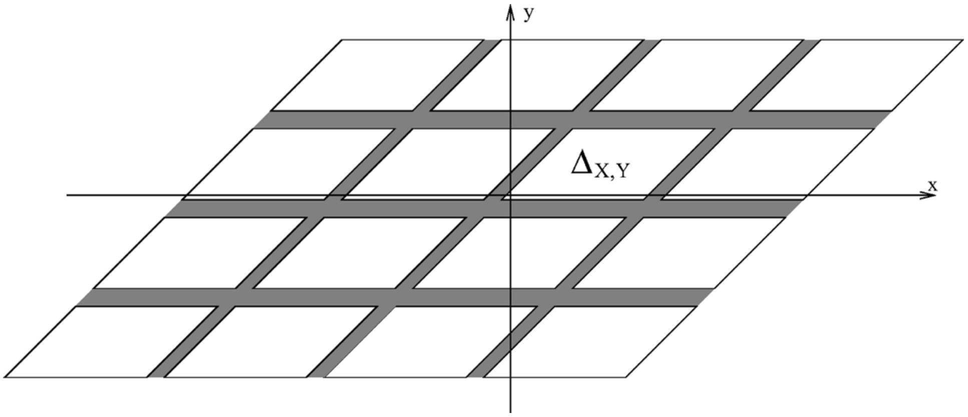


Fig. 2. A method of mapping an infinite binary source into $\boldsymbol{Z}^2$ satisfying the $(1,\infty)$ constraint. The source is mapped into labelings of translates of $\Delta_{X,Y}$ separated by rows and diagonals of 0's. The shaded areas indicate the padding 0's.

The set $\Delta_{X,Y}$ is a parallelogram whose diagonals and rows are D_i and R_j, respectively, as shown in Fig. 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 0's 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.

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 Fig. 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 [17] 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 0's and 1's 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 1's in $\hat{s}$ implies fewer stuffed 0's in $\Delta_{X,Y}$. To make the mathematical analysis of the algorithm simpler in [19] and [20], 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 [19] studied a certain two-dimensional bit-stuffing algorithm, and computed that the expected coding rate is within 1% of the capacity $C^{(2)}_{1,\infty}$. The algorithm was later improved in [20] with an encoding rate within 0.1% of the capacity $C^{(2)}_{1,\infty}$.

TABLE I
PARAMETERS USED IN SECTIONS III AND IV

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.

A list of variables defined in Sections III and IV and the parameters they depend on is given in Tables I and II as a reference.

A. A Variable-to-Fixed Length Bit-Stuffing Encoder

In Appendix I, we formally define a variable-to-fixed length encoder

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

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 $\boldsymbol{Z}^2$. Then we take $N \to \infty$.

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

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

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

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

TABLE II
VARIABLES INTRODUCED IN SECTIONS III AND IV AND THE PARAMETERS THEY DEPEND ON

Notation	Parameters	Description
$\tilde{\mathcal{E}}^{(2)}$	$N, \quad, \lambda$	Variable-to-fixed length encoder. Labels $\Delta_{N,N}$.
$\bar{\mathcal{E}}^{(2)}$	$N, \gamma, \lambda, \tau, \qquad, \sigma$	Fixed-to-variable length encoder. Labels $\Delta_{X,N}$.
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 $\Delta_{X,N}$.

TABLE III
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 $\lambda$. 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.
```

The encoder $\tilde{\mathcal{E}}^{(2)}$ is completely determined by the integer N and the initial labeling λ (defined in Appendix I). If N is a fixed constant then $\tilde{\mathcal{E}}^{(2)}$ is a variable-to-fixed length encoder, as defined in [17] and [19] (they actually used a more general parallelogram $\Delta_{X,Y}$ instead of $\Delta_{N,N}$). In Section IV-A, 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 IV-B, 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.

IV. A TWO-DIMENSIONAL VARIABLE-TO-VARIABLE LENGTH ENCODER

Using a finite complete prefix code defined in Section IV-B 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 $\boldsymbol{Z}^2$. The translates are separated by one diagonal and one row of zero padding (see Fig. 3). The tiling can be generalized to all of $\boldsymbol{Z}^2$ by alternately placing the parallelograms in the four quadrants. Henceforth, we abbreviate $X(w^{(i)})$ with X.

A. 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 $u \in \boldsymbol{Z}^2$:

$$\Delta_{N,N}^{(i)} = \Delta_{N,N} + i(N+2, 0)$$

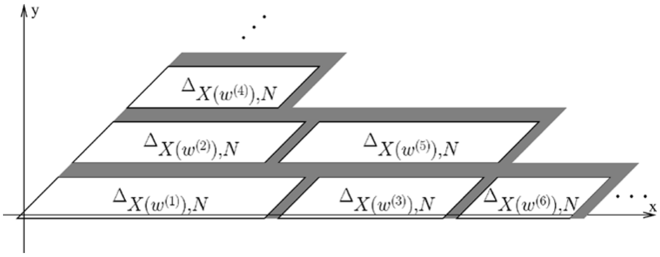


Fig. 3. Translates of the parallelograms $\Delta_{X(w^{(1)}),N}, \ldots, \Delta_{X(w^{(6)}),N}, \ldots$ are used to encode the words $w^{(1)}, \ldots, w^{(6)}, \ldots$, respectively. The shaded areas indicate the padding 0s.

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

Let γ be a positive real number. Let $\tau \in \boldsymbol{Z}^+$, 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*.

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 $\Delta_{N,N}^{(0)}, \ldots, \Delta_{N,N}^{(\tau-1)}$ are labeled

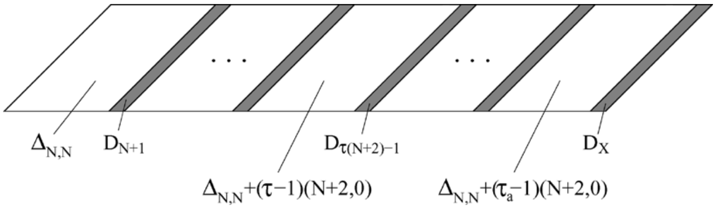

$\Delta_{N,N}$
D_{N+1}
. . .
$\Delta_{N,N}+(\tau-1)(N+2,0)$
$D_{\tau(N+2)-1}$
. . .
$\Delta_{N,N}+(\tau_a-1)(N+2,0)$
D_X

Fig. 4. The set $\Delta_{X,N}$ consists of τ_a translates of $\Delta_{N,N}$, and a diagonal of zero padding after each translate.

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 0's (see Fig. 4).

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 $\boldsymbol{Z}^2$ 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 $\Delta_{N,N}^{(\tau)}, \ldots, \Delta_{N,N}^{(\tau_a-1)}$, using the auxiliary sequence σ to finish filling the last translate $\Delta_{N,N}^{(\tau_a-1)}$, 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 0's.

The fixed-to-variable length encoder

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

described above is formally defined in Appendix II. 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 IV-B. Note that s can be recovered from the labeling of $\Delta_{X,N}$.

In Section IV-C, 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.

The set B is defined in Appendix II. It consists of 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 $\frac{\epsilon}{\gamma N^2}$. The set B is determined by N, γ, λ, τ, and ϵ.

B. Restriction to Typical Sequences

An ϵ-typical set A of block length $\lfloor \tau\gamma N^2 \rfloor$ is formally defined in Appendix II.

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

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

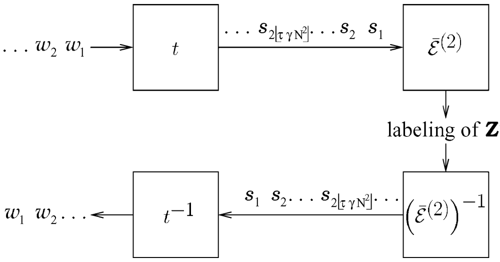

$\ldots w_2\ w_1$
t
$\ldots s_{2\lfloor\tau\gamma N^2\rfloor} \ldots s_2\ s_1$
$\bar{\mathcal{E}}^{(2)}$
labeling of $\boldsymbol{Z}^2$
$w_1\ w_2 \ldots$
t^{-1}
$s_1\ s_2 \ldots s_{2\lfloor\tau\gamma N^2\rfloor} \ldots$
$\left(\bar{\mathcal{E}}^{(2)}\right)^{-1}$

Fig. 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 $\boldsymbol{Z}^2$ by $\bar{\mathcal{E}}^{(2)}$.

be any bijection. Both t and T are determined by the parameters N, γ, λ, τ, ϵ, and 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, and 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 Fig. 5).

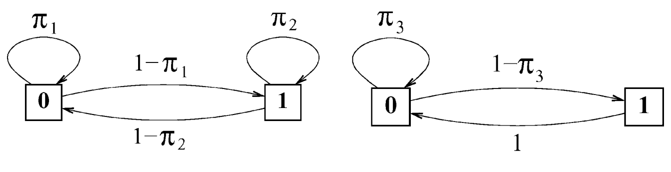
Note that it is guaranteed by the encoder $\mathcal{E}^{(2)}$ that the last diagonal of $\Delta_{X,N}$ is filled with 0's. An additional row R_{N+1} of padding 0's is added to $\Delta_{X,N}$ to ensure that a tiling by the parallelograms defines a valid labeling of $\boldsymbol{Z}^2$.

C. 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 Fig. 6(a) and (b)).

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

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



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

Fig. 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.

the origin, which is used to initiate the labeling of R_0. It follows from [19], [20] 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 IV.2). 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

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

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 $\tilde{\mathcal{E}}^{(2)}$.

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

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

The proof of the following theorem can be found in Appendix III.

Theorem IV.1: The two-dimensional bit-stuffing algorithm achieves a coding rate of

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

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 III.1 and Theorem IV.1. For a fixed value of τ, the existence of the initial labeling λ used by the encoder $\mathcal{E}^{(2)}$ is given in Lemma III.1. The parameter γ is a function of p (see (4)).

Note that if p is close to 0, then the p-sequence $\hat{s}$ contains fewer 1's, and fewer stuffed 0's 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.)

D. Coding Rate Maximization

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 IV-C. Let $\hat{F}(v)$ denote the random label assigned to the point $v \in \boldsymbol{Z}^2$. 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 will be given in Theorem IV.2 (proof in Appendix III). In the theorem, the random initial labeling assigned to the boundary elements $D_0 \cup R_0$ is that defined in Section IV.C, and the parameters x and y are given in Fig. 1.

Theorem IV.2: 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

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

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.

Remark IV.3: Equations (C12)–(C14) imply that

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

where $\alpha = \boldsymbol{P}\big(\hat{F}((0,0)) = 0\big)$. The conditions of Theorem IV.2 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 IV.1, IV.2, Remark IV.3, and the fact that

$$\gamma = \alpha \pi_1 \tag{4}$$

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

$$r^{(2)} = \gamma \cdot H(p) = \alpha \pi_1 \cdot H(p) \tag{5}$$

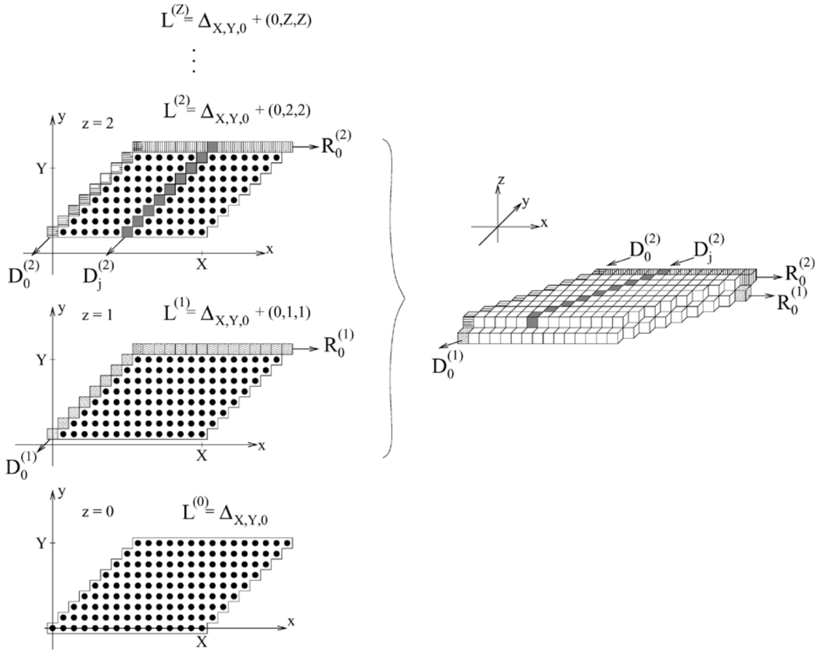


Fig. 7. Layers of $\Delta_{X,Y,Z}$ are shown on the left-hand side. The relative position of layers $L^{(1)}$ and $L^{(2)}$ is illustrated on the right-hand side. The diagonals $D_0^{(1)}$, $D_0^{(2)}$, $D_j^{(2)}$, and the rows $R_0^{(1)}$, $R_0^{(2)}$ are shaded.

since α and π_1 are implicit functions of p. The largest coding rate is found by maximizing (5) over the parameter p. In [19], 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 [20]. 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 $\mathcal{E}^{(2)'}$ as

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

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 VI.5) that the three-dimensional algorithm achieves the approximate coding rate of

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

E. Remarks on Computational Complexity

Although the bit-stuffing algorithm analyzed here closely resembles the Roth–Siegel–Wolf algorithm and is easier to analyze, it nevertheless appears computationally difficult to implement. The largest contribution to complexity occurs in constructing the sets A and B in the map $t : T \longrightarrow A \cap B$, defined in Section IV-B, and determining the initial labeling λ in Lemma III.1. An exhaustive construction of A consists of examining at most $2^{\tau\gamma N^2}$ binary sequences and determining (in $O(\tau\gamma N^2)$ time) whether each sequence is typical. For each of these binary sequences, it can be determined in polynomial time whether it lies in B by computing the $Q^{(i)}$'s by a stuffing simulation. Once $A \cap B$ is constructed, the prefix code T (and thus the bijection t) can be constructed in time linear in $|A \cap B|$. An initial labeling λ implied in Lemma III.1 can be found by an exhaustive search in at most $2^{2\tau N + \tau\gamma N^2}$ steps. The mapping $\bar{\mathcal{E}}^{(2)}$ is polynomial time contructible, and thus, the total complexity of determining $\mathcal{E}^{(2)}$ is at most $2^{O(N^2)}$. Similarly, the complexity occurring for the three-dimensional algorithm to be described in Section V is at most $2^{O(N^3)}$.

V. 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 $X, Y, Z \in \boldsymbol{Z}^+$ and $i \in \{0, \dots, Z\}$, define the sets

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

as shown in Fig. 7. The set $L^{(i)}$ is a translate of the parallelogram $\Delta_{X,Y,0} = \Delta_{X,Y}$ defined in Section III, and is called the ith *layer* of $\Delta_{X,Y,Z}$. For $j \in \{0, \dots, X\}$ and $m \in \{0, \dots, Y\}$ let

$$D_j^{(0)} = \{(y + j, y, 0) : 0 \le y \le Y\}$$
$$R_m^{(0)} = \{(x + m, m, 0) : 0 \le x \le X\}$$

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

$$D_j^{(i)} = D_j^{(0)} + (0, i, i) = \{(y + j, y + i, i) : 0 \le y \le Y\}$$
$$R_m^{(i)} = R_m^{(0)} + (0, i, i) = \{(x + m, m + i, i) : 0 \le x \le X\}$$

for $j \in \{0, \dots, X\}$ and $m \in \{0, \dots, Y\}$ (see Fig. 7). The points in $L^{(0)}$, $D_0^{(i)}$, $R_0^{(i)}$, for $i \in \{1, \dots, Z\}$, are called the *boundary* of $\Delta_{X,Y,Z}$. That is, the boundary points consist of the entire

TABLE IV
PARAMETERS USED IN SECTIONS V AND VI

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.

TABLE V
VARIABLES INTRODUCED IN SECTIONS V AND VI AND THE PARAMETERS THEY DEPEND ON

Notation	Parameters	Description
$\tilde{\mathcal{E}}^{(3)}$	$N, \quad\quad , \lambda$	Variable-to-fixed length encoder. Labels $\Delta_{N,N,N}$.
$\bar{\mathcal{E}}^{(3)}$	$N, \gamma_1, \gamma_2, \lambda, \tau, \quad\quad\quad , \sigma^{(1)}, \sigma^{(2)}$	Fixed-to-variable length encoder. Labels $\Delta_{N,N,Z}$.
B	$N, \gamma_1, \gamma_2, \lambda, \tau, \epsilon, \quad\quad\quad , \sigma^{(1)}, \sigma^{(2)}$	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 $\|A \cap B\|$.
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 $\Delta_{N,N,Z}$.

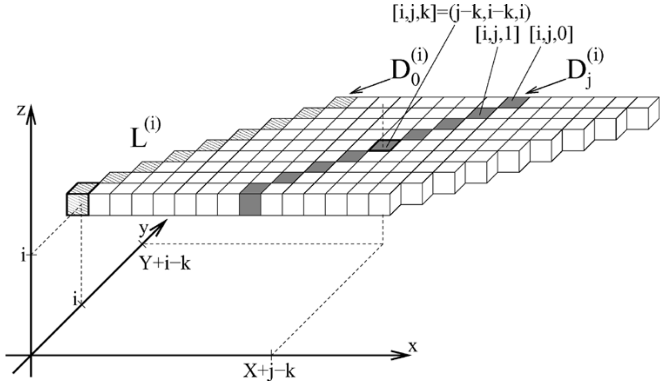


Fig. 8. The coordinates $[i, j, k]$ denote the point on the ith layer, jth diagonal, kth position. In the Cartesian coordinate system this point is $[i, j, k] = (X + j - k, Y + i - k, i)$.

first layer $L^{(0)}$ and the first diagonal and first row on every other layer. The nonboundary 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 $[i, j, k]$, 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 Fig. 8).

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 $\boldsymbol{Z}^3$. 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 IV.

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 $\left(s^{(i,1)}, s^{(i,2)}\right)$, 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 VI-C; and p_1 and p_2 are calculated in Section VI-D. 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 VI.

A list of variables defined in Sections V and VI and the parameters they depend on are given in Tables IV and V as a reference.

A. A Variable-to-Fixed Length Encoder

The ordering $\prec$ defined in Section III-A is extended to $\boldsymbol{Z}^3$ in the following way. For any

$$(x_1, y_1, z_1),\ (x_2, y_2, z_2) \in \boldsymbol{Z}^3$$

TABLE VI
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 $\lambda$.
    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 $s^{(1)}_{m_1}$. Let $m_1 = m_1 + 1$.
8.          Else
9.                  Label $(x_0, y_0, z_0)$ with $s^{(2)}_{m_2}$. Let $m_2 = m_2 + 1$.
10. If all of $\Delta_{N,N,N}$ is labeled then stop, else go to 2.
```

let

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

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 $\Delta_{X,Y,Z}$.

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

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

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 (A1)). 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 pseudocode description of $\tilde{\mathcal{E}}^{(3)}$ is given in Table VI. 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 VI.4). The inverse mapping scans the elements of $\Delta_{N,N,N}$ in increasing order with respect to the ordering $\prec$, skipping over stuffed 0's 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.

VI. 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 VI-B, 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 VI-C when the input is the $1/2$-sequence $\hat{w}$.

A. An Intermediate Fixed-to-Variable Length Encoder

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

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

The translates

$$\Delta_{N,N,N} + i[N+2,0,0]$$

are labeled with information bits and stuffed 0's, the layers $L^{(j(N+2)-1)}$ are padded with 0's, and some additional "overflow" layers

$$L^{(\tau(N+2))}, \ldots, L^{(Z)}$$

are filled randomly. 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 VI.1). The translates of $\Delta_{N,N,N}$ are labeled with input strings

$$s^{(1)} \in \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor}$$
$$s^{(2)} \in \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor}$$

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 0's 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 0's. Finally, an additional layer of padding 0's is added, whose index is defined to be Z.

The fixed-to-variable length encoder

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

described above is formally defined in Appendix IV. 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

$$\sigma^{(1)} \in \{0,1\}^{\tau N^3}$$
$$\sigma^{(2)} \in \{0,1\}^{\tau N^3}$$

which are described in Lemma VI.1.

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.

The set B is defined in Appendix IV. It contains the pairs of strings that "fit well" into $\Delta_{N,N,\tau(N+2)-1}$.

B. Restriction to Typical Sequences

A set A of ϵ-typical pairs of strings of block length $\tau\gamma_1 N^3$ and $\tau\gamma_2 N^3$ is formally defined in Appendix IV. The sets A and B can be shown to have nonempty intersection, by an argument similar to that used to obtain the lower bound in (C6).

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* $\mathcal{E}^{(3)}$ 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 0's in three dimensions similarly as in two dimensions.

C. 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

$$\gamma_1(i,j,k) = \text{probability } [i,j,k] \text{ is labeled by a bit from } \hat{s}^{(1)}$$
$$\gamma_2(i,j,k) = \text{probability } [i,j,k] \text{ is labeled by a bit from } \hat{s}^{(2)}$$

as in Steps 7 and 9 of Table VI.

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

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

It is shown in Section VI-D that Ω is nonempty (in the paragraph preceding Remark VI.4).

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 (proof in Appendix V). The set B in the lemma is defined in (D1).

Lemma VI.1: 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 $N \in \boldsymbol{Z}^+$ and any $\epsilon > 0$, there exists $\tau_0 \in \boldsymbol{Z}^+$ such that for any $\tau \ge \tau_0$ there is an initial labeling $\lambda : U_\tau \longrightarrow \{0,1\}$ and auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$ such that

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

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 VI.2: 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

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

for $\tau \ge \tau_0$ large enough (see [23, pp. 51–52]). The rest of the proof is analogous to the proof of Theorem IV.1 using Lemma VI.1. □

As in two dimensions, for given parameters N, p_1, p_2, ϵ, the value of τ is induced by Lemma VI.1 and Theorem VI.2. For a fixed τ, the initial labeling λ and the auxiliary sequences $\sigma^{(1)}$ and $\sigma^{(2)}$ are implied in Lemma VI.1. The parameters γ_1 and γ_2 are given in (F3) and (F4) of Appendix VI.

D. Coding Rate Maximization

In this subsection, 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 IV.2. 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 $i \in \{1,\dots,N\}$) 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.

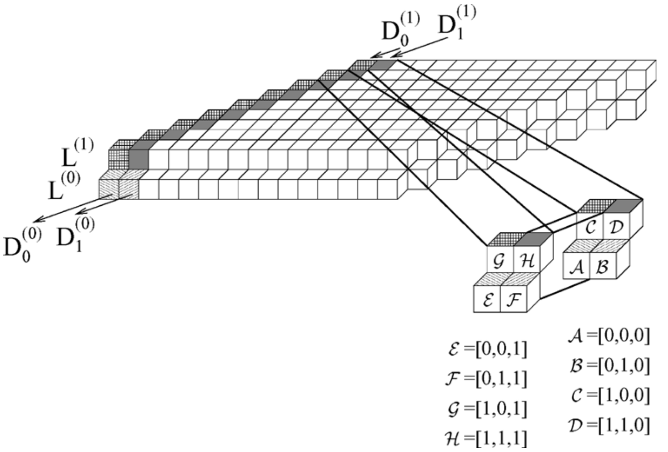


Fig. 9. Elements of $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$ in Conditions **a)**-**d)**. Labels represent the centers of cubes.

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 VI, 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 Fig. 9.

- **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 $\hat{F}(\mathcal{F},\mathcal{H})$.

Theorem VI.3: 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 Fig. 13 for $i,j,k \in \{1,\dots,N\}$. Lemma V.8 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. □

Conditions **a)**-**d)** translate into a set of equations for the parameters $\pi_1, \pi_2, \pi_3, \alpha, p_1, p_2, p$ (see Appendix VI). Using Condition 3 in Theorem IV.2, (3), and (F1)–(F4), we expressed the

TABLE VII
THE NUMERICAL VALUES OF THE PARAMETERS THAT MAXIMIZE $r^{(3)}$

p	π_1	π_2	π_3	α	p_1	p_2	γ_1	γ_2
0.255838	0.832393	0.212958	0.787042	0.824431	0.343793	0.255838	0.350456	0.21531

parameters $\pi_1, \pi_2, \pi_3, \alpha, p_1, p_2, \gamma_1, \gamma_2$ in terms of p. Therefore, for

$$\pi_1, \pi_2, \pi_3, \alpha, p_1, p_2, p, \gamma_1, \gamma_2 \in [0, 1]$$

that also satisfy Condition 3 in Theorem IV.2, (3), and (F1)–(F4), 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

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

over $p \in \left[\frac{1}{2}, 1\right]$ gives the numerical results in Table VII.

Remark VI.4: 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 VI.5: 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 VII into the formula $\gamma_1 H(p_1) + \gamma_2 H(p_2)$ determined in Theorem VI.2 gives $r^{(3)} = 0.50200500727$. Using the bounds in (1) we get

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

VII. CONCLUSION

In order to improve the theoretical foundation of the Roth–Siegel–Wolf bit-stuffing algorithm for two-dimensional $(1, \infty)$ run-length constrained coding, we rigorously analyzed a close variant of the algorithm. The studied algorithm closely resembles the original bit-stuffing algorithm, is precisely defined, and achieves the same coding rate. It is, however, not as readily implementable due to higher computational complexity. We generalized the algorithm to three dimensions, and then rigorously analyzed it using techniques similar to those we used in two dimensions. The rate of the three-dimensional coder is less than 4% from the capacity and is higher (although only slightly) than a trivial coder that places information bits in locations whose coordinates sum to an even number, and places zeros elsewhere.

APPENDIX I
DEFINITION OF A TWO-DIMENSIONAL VARIABLE-TO-FIXED LENGTH ENCODER

This appendix formally defines an encoder used in Section III-A.

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

$$\tilde{\mathcal{E}}^{(2)}(s)(u) = \begin{cases} \lambda(u), & \text{if } u \in R_0 \cup D_0 \\ 0, & \text{if } \tilde{\mathcal{E}}^{(2)}(s)(u_l) = 1 \\ & \quad \text{or } \tilde{\mathcal{E}}^{(2)}(s)(u_t) = 1 \\ s_{\beta(u)}, & \text{otherwise} \end{cases}$$

$$\beta(u) = 1 + \Big|\{v \in \Delta_{N,N} \setminus (D_0 \cup R_0) : v \prec u, \\ \tilde{\mathcal{E}}^{(2)}(s)(v_l) \neq 1, \tilde{\mathcal{E}}^{(2)}(s)(v_t) \neq 1\}\Big|.$$

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

$$V_{N,\lambda} = \{s \in \{0, 1\}^* : l(s) = \beta(u_b) - 1\} \qquad \text{(A1)}$$

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

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

is a word encoder, as defined in (2).

APPENDIX II
DEFINITION OF A TWO-DIMENSIONAL INTERMEDIATE FIXED-TO-VARIABLE LENGTH ENCODER

This appendix formally defines an encoder used in Sections IV-A and IV-B.

For each i, the labeling λ_i induces a prefix code V_{N,λ_i} based on labeling $\Delta_{N,N}^{(i)}$, as in (A1). 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

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

be the number of translates of $\Delta_{N,N}$ used to perform the labeling. That is, s will be encoded into a labeling of

$$\Delta_{N,N}^{(0)}, \ldots, \Delta_{N,N}^{(\tau_a - 1)}.$$

Later, we will force τ_a to be close to the target τ, with high probability. Let

$$X = \tau_a(N+2) - 1$$

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

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

TABLE VIII
ADDITIONAL VARIABLES INTRODUCED IN APPENDIX II AND THE PARAMETERS THEY DEPEND ON

Notation	Parameters	Description
$Q^{(i)}$	$N,\gamma,\lambda,\tau,\sigma,s$	Number of bits $\bar{\mathcal{E}}^{(2)}$ maps into $\Delta^{(i)}_{N,N}$.
q	$N,\gamma,\lambda,\tau,\sigma$	Number of bits $\bar{\mathcal{E}}^{(2)}$ does not map into 1st τ translates of $\Delta_{N,N}$.

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

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

by specifying the labeling

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

as

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

where $\tilde{\mathcal{E}}^{(2)}$ is the encoder defined in Section III-A.

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 (see Table VIII)

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

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

$$B = \left\{ s \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor} : q(s) < \tau\epsilon \right\}. \tag{B1}$$

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 $q(s) < 0$.

For any $\epsilon > 0$, the *typical set* A of block length $\lfloor \tau\gamma N^2 \rfloor$ with respect to p is defined as [23, p. 51]

$$A = \Big\{ s \in \{0,1\}^{\lfloor \tau\gamma N^2 \rfloor} : 2^{-\lfloor \tau\gamma N^2 \rfloor (H(p)+\epsilon)} \leq p^{|s|}(1-p)^{\lfloor \tau\gamma N^2 \rfloor - |s|} \leq 2^{-\lfloor \tau\gamma N^2 \rfloor (H(p)-\epsilon)} \Big\}.$$

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 $N, \gamma, \lambda, \tau, \epsilon, p$, and an element of A is called an *ϵ-typical sequence*.

APPENDIX III
TWO LEMMAS AND THE PROOFS OF THEOREMS IV.1 AND IV.2

This appendix gives a lemma and a proof used in Section IV-C. An initial labeling λ of U_∞ used by $\mathcal{E}^{(2)}$ is implied in the lemma. The set B in the lemma is defined in Appendix II.

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

$$\lambda : U_\infty \longrightarrow \{0,1\}$$

such that

$$\boldsymbol{P}(\hat{s} \in B) > 1 - \epsilon. \tag{C1}$$

Proof: Suppose that for some $\tau \in \boldsymbol{Z}^+$ 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 $D^{(i)}_0$ $(0 \leq i < \tau)$, the Markov chain $\hat{\mu}^{(2)}$ for the translates $R^{(i)}_0$ $(0 \leq i < \tau)$, and initializes $D^{(i)}_0$ and $R^{(i)}_0$ with 0's for $i \geq \tau$. By choosing the auxiliary sequence $\hat{\sigma}$ to be a p-sequence, we will demonstrate that there is at least one initial labeling $\lambda : U_\infty \longrightarrow \{0,1\}$ such that for any auxiliary sequence σ, (C1) holds.

Since each $\Delta^{(i)}_{N,N}$ is initialized by $\hat{\lambda}_i$, the labeling of each $\Delta^{(i)}_{N,N}$ 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 $i,j \in \{0,1,\ldots,\tau-1\}$, the random variables $\hat{Q}^{(i)}$ and $\hat{Q}^{(j)}$ are independent and have finite variances (independent of τ). Therefore, by the weak law of large numbers, for every $\epsilon > 0$, we have

$$\lim_{\tau\to\infty} \boldsymbol{P}\left(\left| \frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}^{(i)} - \gamma N^2 \right| < \epsilon \right) = 1. \tag{C2}$$

The random variables $\hat{Q}^{(i)}$ in (C2) 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 (C2) and the inequalities

$$\begin{aligned} &\boldsymbol{P}\left(\left| \frac{1}{\tau}\sum_{i=0}^{\tau-1} \hat{Q}^{(i)} - \gamma N^2 \right| < \epsilon \right) \\ &\qquad \leq \boldsymbol{P}\left(\tau\gamma N^2 - \sum_{i=0}^{\tau-1} \hat{Q}^{(i)} < \tau\epsilon \right) \\ &\qquad \leq \boldsymbol{P}\left(\lfloor \tau\gamma N^2 \rfloor - \sum_{i=0}^{\tau-1} \hat{Q}^{(i)} < \tau\epsilon \right) \\ &\qquad = \boldsymbol{P}\left(q(\hat{s}) < \tau\epsilon\right) \end{aligned} \tag{C3}$$

imply that

$$\lim_{\tau\to\infty} \boldsymbol{P}\left(q(\hat{s}) < \tau\epsilon\right) = 1. \tag{C4}$$

It follows from (C4) that there exists a τ_0 such that

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

for all $\tau \geq \tau_0$. Thus, there must exist at least one initial labeling λ (depending on τ) such that

$$\boldsymbol{P}\left(q(\hat{s}) < \tau\epsilon \middle| \lambda\right) > 1-\epsilon \tag{C5}$$

where the conditioning in (C5) is on the event that the random labeling $\hat{\lambda}$ equals the fixed labeling λ.

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

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

Proof of Theorem IV.1: For $N \in \boldsymbol{Z}^+$ and $\epsilon > 0$, let τ_0 be defined as in Lemma III.1. It is known [23, 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

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

Therefore, Lemma III.1 implies

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

which then implies

$$|A \cap B| \geq (1-2\epsilon) \cdot 2^{\lfloor \tau\gamma N^2 \rfloor (H(p)-\epsilon)}. \tag{C6}$$

Similarly, we have

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

which implies

$$|A \cap B| \leq 2^{\lfloor \tau\gamma N^2 \rfloor (H(p)+\epsilon)}. \tag{C7}$$

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 $\boldsymbol{Z}^2$. Therefore, (C6), the definition of the set B, and the fact that

$$\sum_{z \in T} \boldsymbol{P}(z) = 1$$

imply that the coding rate is lower bounded as

$$\begin{aligned} r\left(\mathcal{E}^{(2)}\right) &= \sum_{z\in T} \boldsymbol{P}(z) \frac{l(z)}{(X+1)(N+2)} \\ &\geq \frac{\lfloor \log_2 |A \cap B| \rfloor}{\tau(N+2)^2 + \left\lceil \frac{3\tau\epsilon}{N^2} \right\rceil (N+2)^2} \sum_{z \in T} \boldsymbol{P}(z) \\ &\geq \frac{\log_2(1-2\epsilon) + \lfloor \tau\gamma N^2 \rfloor (H(p)-\epsilon) - 1}{\tau(N+2)^2 + \left\lceil \frac{3\tau\epsilon}{N^2} \right\rceil (N+2)^2}. \end{aligned} \tag{C8}$$

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 (C7) and the fact that

$$X \geq \tau(N+2) - 1$$

for any input word $z \in T$, the coding rate is upper-bounded as

$$\begin{aligned} r\left(\mathcal{E}^{(2)}\right) &= E\left[\frac{l(z)}{(X+1)(N+2)}\right] \\ &\leq \frac{\lceil \log_2 |A \cap B| \rceil}{\tau(N+2)^2} \\ &\leq \frac{\tau\gamma N^2 (H(p)+\epsilon) + 1}{\tau(N+2)^2}. \end{aligned} \tag{C9}$$

Taking limits as $N \to \infty$ and $\epsilon \to 0$, the theorem follows from (C8) and (C9). □

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

Lemma III.2: 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 $n \in \boldsymbol{Z}^+$, 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 $a^* \in A^*$. Then

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

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, we have

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

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

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

Hence,

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

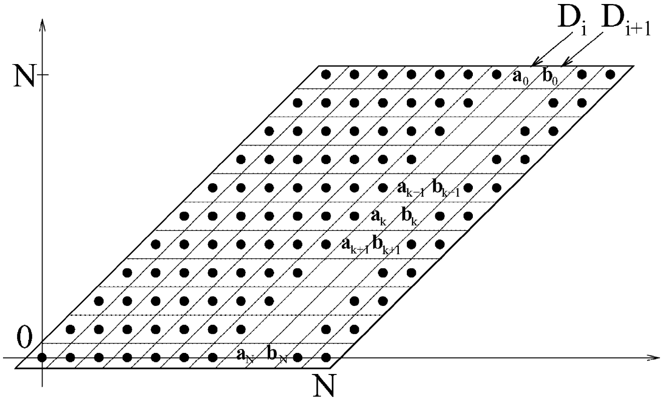


Fig. 10. The diagonals D_i and D_{i+1} of $\Delta_{N,N}$.

Proof of Theorem IV.2: The equivalence of Conditions 1), 2), and 3) follows from the results in [19][3] and [20]. 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

$$a_j = (i+j, j)$$
$$b_j = (i+j+1, j)$$

for $0 \le j \le N$, of the diagonals D_i and D_{i+1} (see Fig. 10). 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 $0 < k \le N$. Then

$$\begin{aligned}
&\boldsymbol{P}\big(\hat{F}(a_{k-1}, b_{k-1}, a_k, b_k)=(v_1, v_2, v_3, v_4)\big)\\
&\quad=\boldsymbol{P}\big(\hat{F}(a_{k-1}) = v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(b_{k-1}, a_k)=(v_2, v_3)\big|\hat{F}(a_{k-1})=v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(b_k)=v_4\big|\hat{F}(a_{k-1}, b_{k-1}, a_k)=(v_1, v_2, v_3)\big)\\
&\quad=\boldsymbol{P}\big(\hat{F}(a_{k-1}) = v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(b_{k-1}, a_k)=(v_2, v_3)\big|\hat{F}(a_{k-1})=v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(b_k)=v_4\big|\hat{F}(a_{k-1}, a_k)=(v_1, v_3)\big) \qquad \text{(C10)}\\
&\quad=\boldsymbol{P}\big(\hat{F}(a_{k-1}) = v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(a_k)=v_3\big|\hat{F}(a_{k-1})=v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(b_{k-1})=v_2\big|\hat{F}(a_{k-1})=v_1\big)\\
&\qquad\cdot \boldsymbol{P}\big(\hat{F}(b_k)=v_4\big|\hat{F}(a_{k-1}, a_k)=(v_1, v_3)\big) \qquad \text{(C11)}
\end{aligned}$$

where (C10) follows from the definition of the bit-stuffing encoder; and (C11) follows from Lemma III.2 with

$$\hat{u}_j = \hat{F}(a_j)$$
$$\hat{u}^* = \hat{F}(b_{k-1})$$

since the labels $\big\{\hat{F}(a_j)\big\}_{j=N}^{0}$ form a Markov chain, and $\hat{F}(b_{k-1})$ is independent of $\hat{F}(a_k)$ if

$$\hat{F}(a_{k-1}), \hat{F}(a_{k-2}), \ldots, \hat{F}(a_0)$$

are given. Conditions 1) and 2) imply that the first three terms of (C11) are independent of the diagonal i and the position k, and the last term of (C11) 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

$$\alpha = \alpha\pi_1 + (1-\alpha)(1-\pi_2) \qquad \text{(C12)}$$
$$\alpha = \alpha\pi_3 + (1-\alpha). \qquad \text{(C13)}$$

Furthermore, by Condition 4) we have

$$\begin{aligned}
&\boldsymbol{P}\big(\hat{F}((0,0)) = \hat{F}((-1,-1))=0\big)\\
&\qquad=\boldsymbol{P}\big(\hat{F}((1,0)) = \hat{F}((0,-1)) = 0\big)
\end{aligned}$$

which implies

$$\begin{aligned}
\alpha\pi_1 =&\, \boldsymbol{P}\Big(\hat{F}((0,0)) = 0, \hat{F}((-1,-1)) = 0,\\
&\qquad \hat{F}((1,0))=0, \hat{F}((0,-1))=0\Big)\\
&+\boldsymbol{P}\Big(\hat{F}((0,0))=0, \hat{F}((-1,-1))=1,\\
&\qquad \hat{F}((1,0))=0, \hat{F}((0,-1))=0\Big)\\
&+\boldsymbol{P}\Big(\hat{F}((0,0))=1, \hat{F}((-1,-1))=0,\\
&\qquad \hat{F}((1,0))=0, \hat{F}((0,-1))=0\Big)\\
&+\boldsymbol{P}\Big(\hat{F}((0,0))=1, \hat{F}((-1,-1))=1,\\
&\qquad \hat{F}((1,0))=0, \hat{F}((0,-1))=0\Big)\\
=&\,\alpha\pi_1\pi_3(1-p) + \alpha\pi_3(1-\pi_1)\\
&+(1-\alpha)(1-\pi_2) + \pi_2(1-\alpha)\\
=&\,\alpha\pi_3(\pi_1(1-p) + 1 - \pi_1) + 1 - \alpha
\end{aligned}$$

and

$$\begin{aligned}
&\boldsymbol{P}\big(\hat{F}((0,0)) = \hat{F}((1,0)) = 0\big)\\
&\qquad=\boldsymbol{P}\big(\hat{F}((-1,-1)) = \hat{F}((0,-1)) = 0\big)
\end{aligned}$$

which implies

$$\begin{aligned}
\alpha\pi_3 =&\, \boldsymbol{P}\Big(\hat{F}((0,0)) = 0, \hat{F}((1,0)) = 0,\\
&\qquad \hat{F}((-1,-1))=0, \hat{F}((0,-1))=0\Big)\\
&+\boldsymbol{P}\Big(\hat{F}((0,0))=0, \hat{F}((1,0))=1,\\
&\qquad \hat{F}((-1,-1))=0, \hat{F}((0,-1))=0\Big)\\
&+\boldsymbol{P}\Big(\hat{F}((0,0))=1, \hat{F}((1,0))=0,\\
&\qquad \hat{F}((-1,-1))=0, \hat{F}((0,-1))=0\Big)\\
=&\,\alpha\pi_1\pi_3(1-p) + \alpha\pi_1(1-p)(1-\pi_3)\\
&+(1-\alpha)(1-\pi_2)\\
=&\,\alpha\pi_1(1-p) + (1-\alpha)(1-\pi_2). \qquad \text{(C14)}
\end{aligned}$$

The solution of (C12)–(C14) for π_1, π_2, π_3 in terms of p gives the formulas in Condition 3). □

[3] [19, eq. (24)] is incorrect. It should read

$$z = \frac{(4-3q) - \sqrt{(4-3q)^2 - 4(1-q)(4-3q)}}{2(1-q)(4-3q)}.$$

TABLE IX
ADDITIONAL VARIABLES INTRODUCED IN APPENDIX IV AND THE PARAMETERS THEY DEPEND ON

Notation	Parameters	Description
$Q_i^{(j)}$	$N, \gamma_1, \gamma_2, \lambda, \tau, \sigma^{(1)}, \sigma^{(2)}, s^{(1)}, s^{(2)}$	Number of bits $\bar{\mathcal{E}}^{(3)}$ maps into ith translate of $\Delta_{N,N,N}$ from jth input string.
q_i	$N, \gamma_1, \gamma_2, \lambda, \tau, \sigma^{(1)}, \sigma^{(2)}$	Number of bits $\bar{\mathcal{E}}^{(3)}$ does not map into 1st τ translates of $\Delta_{N,N,N}$ from ith input string.

APPENDIX IV
DEFINITION OF A THREE-DIMENSIONAL INTERMEDIATE FIXED-TO-VARIABLE LENGTH ENCODER

This appendix formally defines an encoder used in Sections VI-A and VI-B.

A fixed-to-variable length encoder

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

is defined by specifying the labeling

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

as

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

where

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

and $\tilde{\mathcal{E}}^{(3)}$ is the encoder defined in Section III-A; $(z^{(i,1)}, z^{(i,2)})$ is a parsing of $(s^{(1)}\sigma^{(1)}, s^{(2)}\sigma^{(2)})$ 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 intertranslate layers $L^{(j(N+2)-1)}$ for $j = 1, 2, \dots, \tau$, as well as those

$$u = [u_1, u_2, u_3] \in \Delta_{N,N,N}$$

for which

$$u_1 \in \{\tau(N+2), \dots, Z\}$$

and either u_2 is odd or else

$$\delta(u) > l\left(\bar{s}^{(1)}\bar{s}^{(2)}\right).$$

Note that in the middle case above, the subscript $\delta(u)$ refers to a bit position in the string $\bar{s}^{(1)}\bar{s}^{(2)}$.

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 (see Table IX)

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

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

$$B = \left\{(s^{(1)}, s^{(2)}) \in \{0,1\}^{\lfloor \tau\gamma_1 N^3 \rfloor} \times \{0,1\}^{\lfloor \tau\gamma_2 N^3 \rfloor} : q_1(s^{(1)}, s^{(2)}) < \tau\epsilon,\ q_2(s^{(1)}, s^{(2)}) < \tau\epsilon\right\}. \quad \text{(D1)}$$

Let

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

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.

APPENDIX V
DETAILS OF THREE-DIMENSIONAL CODING RATE MAXIMIZATION

This appendix provides details used in Section VI-D.

Proof of Lemma VI.1: 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.

The rest of the proof is similar to the proof of Lemma III.1 and is therefore omitted. □

The following lemmas about the two-dimensional standard labeling of $\Delta_{N,N}$ follow from Theorem IV.2 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 Fig. 10.

Lemma V.1: 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 $i \in \{0,\dots,N-1\}$, 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 $\{\hat{F}(a_j,b_j)\}_{j=0}^{N}$ forms a stationary homogeneous Markov chain.

Lemma V.2: 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 $i \in \{0,\dots,N\}$, let the random vector $\hat{F}(D_i)$ denote the labels of the diagonal D_i. The sequence $\{\hat{F}(D_i)\}_{i=0}^{N}$ forms a stationary homogeneous Markov chain.

In the following we show that Conditions **a)**-**d)** imply that for every $x \in \Delta_{N,N,N}$ if

$$\mathcal{A}+x, \mathcal{B}+x, \dots, \mathcal{H}+x \in \Delta_{N,N,N}$$

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 Fig. 11.

Lemma V.3: 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 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 $1 \le k < N$. 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

$$\begin{aligned}
&\boldsymbol{P}\Big(\hat{F}(a_k,b_k,c_k,d_k,a_{k+1},b_{k+1},c_{k+1},d_{k+1})\\
&\quad = (v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8)\Big)\\
&\quad = \boldsymbol{P}\Big(\hat{F}(a_k,b_k,c_k,d_k) = (v_1,v_2,v_3,v_4)\Big)\\
&\quad \cdot \boldsymbol{P}\Big(\hat{F}(a_{k+1},b_{k+1})=(v_5,v_6)\big|\\
&\quad \hat{F}(a_k,b_k,c_k,d_k)=(v_1,v_2,v_3,v_4)\Big)\\
&\quad \cdot \boldsymbol{P}\Big(\hat{F}(c_{k+1},d_{k+1})=(v_7,v_8)\big|\\
&\quad \hat{F}(a_k,b_k,c_k,d_k,a_{k+1},b_{k+1})=(v_1,v_2,v_3,v_4,v_5,v_6)\Big).
\end{aligned} \tag{E1}$$

In what follows, we rewrite each of the three terms in (E1). First we have

$$\begin{aligned}
&\boldsymbol{P}\Big(\hat{F}(a_k,b_k,c_k,d_k)=(v_1,v_2,v_3,v_4)\Big)\\
&\quad =\boldsymbol{P}\Big(\hat{F}(a_1,b_1,c_1,d_1) = (v_1,v_2,v_3,v_4)\Big) &\text{(E2)}\\
&\quad =\boldsymbol{P}\Big(\hat{F}(a_0,b_0,c_0,d_0) = (v_1,v_2,v_3,v_4)\Big) &\text{(E3)}
\end{aligned}$$

where (E2) follows from the induction hypothesis (summing out four terms), and (E3) follows from Condition **b)**. Furthermore

$$\begin{aligned}
&\boldsymbol{P}\Big(\hat{F}(a_{k+1},b_{k+1})=(v_5,v_6)\big|\\
&\qquad \hat{F}(a_k,b_k,c_k,d_k)=(v_1,v_2,v_3,v_4)\Big)\\
&\quad = \boldsymbol{P}\Big(\hat{F}(a_{k+1},b_{k+1}) = (v_5,v_6)\big|\hat{F}(a_k,b_k) = (v_1,v_2)\Big) &\text{(E4)}\\
&\quad = \boldsymbol{P}\Big(\hat{F}(a_1,b_1) = (v_5,v_6)\big|\hat{F}(a_0,b_0) = (v_1,v_2)\Big) &\text{(E5)}\\
&\quad = \boldsymbol{P}\Big(\hat{F}(a_1,b_1) = (v_5,v_6)\big|\\
&\qquad \hat{F}(a_0,b_0,c_0,d_0)=(v_1,v_2,v_3,v_4)\Big) &\text{(E6)}
\end{aligned}$$

where (E4) follows from Lemma III.2 with

$$\begin{aligned}
\hat{u}_i &= \hat{F}(a_i,b_i)\\
\hat{u}^* &= \hat{F}(c_k,d_k)
\end{aligned}$$

since the labels $\{\hat{F}(a_i,b_i)\}_{i=0}^{N}$ form a stationary homogeneous Markov chain by Lemma V.1; (E5) follows from Lemma V.1; and (E6) 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

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

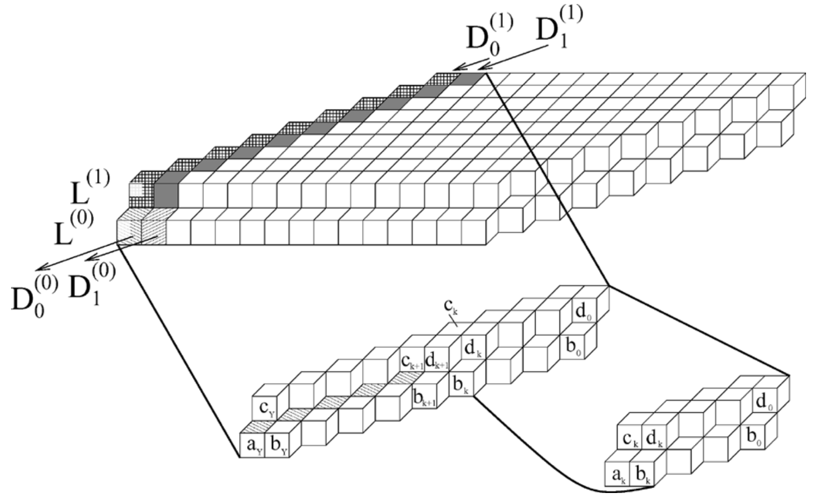


Fig. 11. The diagonals $D_0^{(0)}$, $D_1^{(0)}$, $D_0^{(1)}$, $D_1^{(1)}$.

$$= \boldsymbol{P}\left(\hat{F}(c_1, d_1) = (v_7, v_8)\middle| \hat{F}(a_0, b_0, c_0, d_0, a_1, b_1) = (v_1, v_2, v_3, v_4, v_5, v_6)\right). \tag{E7}$$

Combining (E1) with (E3), (E6), and (E7) gives

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

This completes the induction argument. □

The elements a_l, b_l, c_l, d_l in the following corollary are illustrated in Fig. 11.

Corollary V.4: 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 $\hat{\mathcal{E}}^{(3)}$ 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 V.3 implies that this Markov chain is stationary and homogeneous. □

Lemma V.5: 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 $(\hat{u}_i, \hat{u}'_i)$ 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

$$\begin{aligned}
&\boldsymbol{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right)\\
&= \sum_{u_k, \ldots, u_m \in A} \left[\boldsymbol{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k, \ldots, \hat{u}'_m, \hat{u}_k = u_k, \ldots, \hat{u}_m = u_m\right)\right.\\
&\quad \left. \cdot \boldsymbol{P}\left(\hat{u}_k = u_k, \ldots, \hat{u}_m = u_m \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right)\right] &\text{(E8)}\\
&= \sum_{u_k, \ldots, u_m \in A} \left[\boldsymbol{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k, \hat{u}_k = u_k\right)\right.\\
&\quad \left. \cdot \boldsymbol{P}\left(\hat{u}_k = u_k, \ldots, \hat{u}_m = u_m \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right)\right] &\text{(E9)}\\
&= \sum_{u_k, \ldots, u_m \in A} \left[\boldsymbol{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k\right)\right.\\
&\quad \left. \cdot \boldsymbol{P}\left(\hat{u}_k = u_k, \ldots, \hat{u}_m = u_m \middle| \hat{u}'_k, \ldots, \hat{u}'_m\right)\right]\\
&= \boldsymbol{P}\left(\hat{u}'_{k-1} = v \middle| \hat{u}'_k\right) &\text{(E10)}
\end{aligned}$$

where the summation in (E8) is taken over all values $u_k, \ldots, u_m$ such that the conditioning event has positive probability; (E9) follows since $(\hat{u}_i, \hat{u}'_i)$ is a Markov chain; and (E10) 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 Fig. 11.

Lemma V.6: 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 $\hat{\mathcal{E}}^{(3)}$ 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 $\left\{\hat{F}(a_l, c_l)\right\}_{l=0}^{N}$.

Proof: Let $\hat{u}_l = \hat{F}(a_l, c_l)$ and $\hat{u}'_l = \hat{F}(b_l, d_l)$. Corollary V.4 implies that $(\hat{u}_l, \hat{u}'_l)$ is a Markov chain. Furthermore, Condition **d**) and Lemma V.3 imply that $\hat{u}_l$ is conditionally independent of $\hat{u}'_{l-1}$ given $\hat{u}'_l$. Hence, the conditions of Lemma V.5 hold for $\hat{u}_l$ and $\hat{u}'_l$, and, therefore, $\hat{u}'_l$ is a Markov chain. Condition **a**) and Lemma V.3 imply that the Markov chain $\left\{\hat{F}(b_l, d_l)\right\}_{l=0}^{N}$ is identical to the Markov chain $\left\{\hat{F}(a_l, c_l)\right\}_{l=0}^{N}$. □

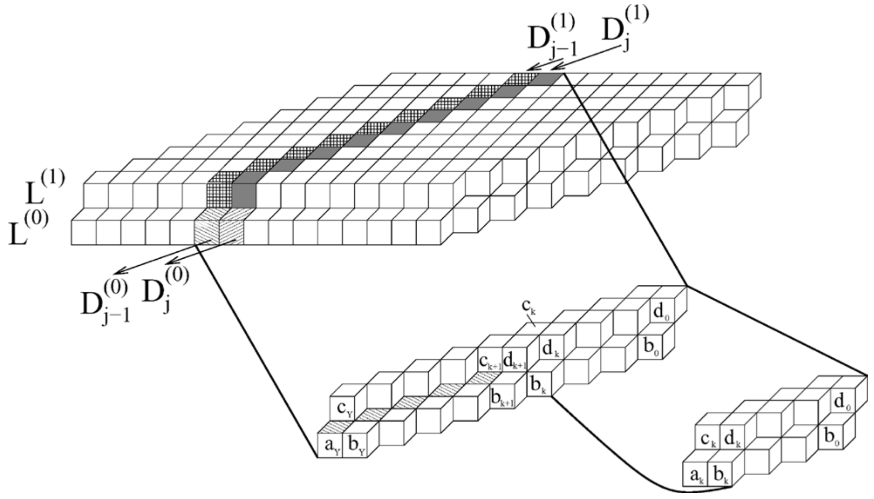


Fig. 12. The diagonals $D_{j-1}^{(0)}$, $D_j^{(0)}$, $D_{j-1}^{(1)}$, $D_j^{(1)}$.

TABLE X
EQUATIONS IMPLIED BY CONDITION **a**) AND GENERATED BY $\boldsymbol{P}(\hat{F}(\mathcal{A},\mathcal{C},\mathcal{E},\mathcal{G}) = \hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H}) = (v_1, v_2, v_3, v_4))$

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

Lemma V.7 below generalizes Lemma V.3 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 Fig. 12.

Lemma V.7: Let $j \in \{1,\dots,N\}$. 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 $D_{j-1}^{(0)}$, $D_j^{(0)}$, $D_{j-1}^{(1)}$, $D_j^{(1)}$, respectively. If the labeling of $\Delta_{N,N,N}$ by $\tilde{\mathcal{E}}^{(3)}$ satisfies Conditions **a**), **b**), and **d**), then for all $l \in \{1,\dots,N\}$, 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: Recall that the random vector $\hat{F}(D_j^{(i)})$ denotes the labels of the diagonal $D_j^{(i)}$ for $i, j \in \{0,\dots,N\}$. Since the labels of $L^{(0)}$ form a standard two-dimensional labeling, $\{\hat{F}(D_j^{(0)})\}_{j=0}^{N}$ is a stationary homogeneous Markov chain by Lemma V.2. Using Lemma III.2 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 V.6. 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 V.2. 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 V.3, Corollary V.4, and Lemma V.6 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 $i = 2, 3, \dots, N-1$) generalizes Lemma V.3 to the elements of $L^{(0)} \cup L^{(1)}$. □

TABLE XI

EQUATIONS IMPLIED BY CONDITION **a**) AND GENERATED BY $\boldsymbol{P}(\hat{F}(\mathcal{B}, \mathcal{D}, \mathcal{A}, \mathcal{C}) = \hat{F}(\mathcal{F}, \mathcal{H}, \mathcal{E}, \mathcal{G}) = (v_1, v_2, v_3, v_4))$

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

TABLE XII

EQUATIONS IMPLIED BY CONDITION **c**) AND GENERATED BY $\boldsymbol{P}(\hat{F}(\mathcal{A}, \mathcal{B}, \mathcal{E}, \mathcal{F}) = \hat{F}(\mathcal{C}, \mathcal{D}, \mathcal{G}, \mathcal{H}) = (v_1, v_2, v_3, v_4))$

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

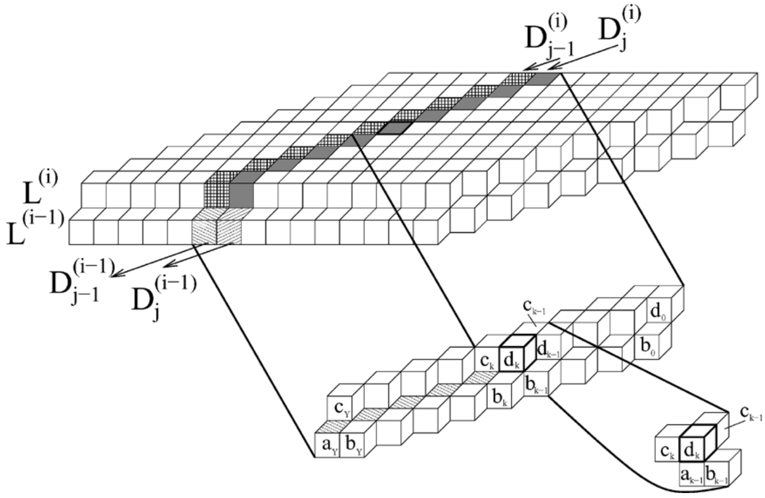


Fig. 13. The diagonals $D_{j-1}^{(i-1)}$, $D_j^{(i-1)}$, $D_{j-1}^{(i)}$, $D_j^{(i)}$, the internal point d_k, and its neighbors.

Lemma V.8 generalizes Lemma V.7 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 Fig. 13.

Lemma V.8: 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 Fig. 12). It follows from Lemma V.7 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 Fig. 9). Thus, the labels of $L^{(1)}$ satisfy Condition 4) of Theorem IV.2. This implies that the probability distribution of the labelings 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 V.3, Corollary V.4, Lemma V.6, and Lemma V.7 for the elements of $L^{(0)} \cup L^{(1)}$ can be used to prove the same results for the layers

$$L^{(1)} \cup L^{(2)}, \ldots, L^{(N-1)} \cup L^{(N)}. \qquad \square$$

Remark V.9: As noted in the proof of Theorem IV.2, since

$$D_0^{(i)} \cap R_0^{(i)} = \{[i, 0, 0]\}$$

TABLE XIII
EQUATIONS IMPLIED BY CONDITION **d**) FOR DIFFERENT VALUES OF $(v_1, v_2, v_3, v_4, v_5, v_6)$

0 0 0 0 0 0 :	$\frac{(1-\alpha)(1-\pi_2)(1-\pi_2-\alpha(1-\pi_2-\pi_1\pi_3(2-p)))+\alpha^2\pi_1^2\pi_3^2(1-p)(1-p_1)}{(1-\pi_2)^2-\alpha(1-\pi_2)(2-\pi_1-2\pi_2-\pi_1(1-p))+\alpha^2((1-\pi_2)^2-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2)))} = \frac{1+\alpha(\pi_3(2-\pi_1 p)-2+\alpha(1+\pi_3(\pi_1 p-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1)))}{1+\alpha\pi_1(-p+\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2))}$
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 :	$\frac{(\alpha-1)\pi_2(\alpha(1+\pi_3-2\pi_1\pi_3)-1)-((\alpha-1)^2\pi_2^2)+\alpha(1-\pi_1)\pi_3(1+\alpha(\pi_1\pi_3(1-p_1)-1))}{\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+\pi_1(1-p_2)(1-\pi_1)(1-\pi_3)-1)} = \frac{1+\alpha(\pi_3(2-\pi_1 p)+\alpha(1+\pi_3(\pi_1 p-2)+\pi_3^2(1-\pi_1 p)(1-\pi_1 p_1))-2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
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 :	$\frac{\alpha\pi_1(1-\pi_3)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p)(1-p_1))}{(1-\pi_2)^2+\alpha(1-\pi_2)(2\pi_2+\pi_1(2-p)-2)+\alpha^2((1-\pi_2)^2-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2)))} = \frac{-\alpha(1-\pi_3)(1+\alpha(\pi_3(1-\pi_1 p)(1-\pi_1 p_1)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
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 :	$\frac{-\alpha\pi_1(1-\pi_3)((\alpha-1)\pi_2-\alpha(1-\pi_1)\pi_3(1-p_1))}{\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+(1-\pi_1)\pi_1(1-\pi_3)(1-p_2)-1)} = \frac{-\alpha(1-\pi_3)(1+\alpha(\pi_3(1-\pi_1 p)(1-\pi_1 p_1)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
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 :	$\frac{\alpha\pi_1(1-\pi_3)(1-p)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p_2))}{(1-\pi_2)^2+\alpha(1-\pi_2)(2\pi_2+\pi_1(2-p)-2)+\alpha^2((1-\pi_2)^2-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2)))} = \frac{-\alpha(1-\pi_3)(1-\pi_1 p)(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
1 0 0 0 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 :	$\frac{-\alpha(1-\pi_1)(1-\pi_3)((\alpha-1)(1-\pi_2)-\alpha\pi_1\pi_3(1-p_2))}{\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+(1-\pi_1)\pi_1(1-\pi_3)(1-p_2)-1)} = \frac{-\alpha(1-\pi_3)(1-\pi_1 p)(1+\alpha(\pi_3(1-\pi_1 p_2)-1))}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
1 0 0 0 1 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 :	$\frac{\alpha^2\pi_1^2(1-p)(1-p_2)(1-\pi_3)^2}{(1-\pi_2)^2+\alpha(1-\pi_2)(2\pi_2+\pi_1(2-p)-2)+\alpha^2((1-\pi_2)^2-\pi_1(1-\pi_2)(2-p)+\pi_1^2(1-p)(\pi_3(p_2-p_1)+(1-p_2)))} = \frac{\alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
1 1 0 0 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 :	$\frac{\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2(1-p_2)}{\pi_2-\pi_2^2+\alpha(\pi_1+\pi_2-1)(2\pi_2-1)+\alpha^2(\pi_1+2\pi_2-2\pi_1\pi_2-\pi_2^2+\pi_1\pi_3(1-p_1)-\pi_1^2\pi_3(1-p_1)+(1-\pi_1)\pi_1(1-\pi_3)(1-p_2)-1)} = \frac{\alpha^2(1-\pi_3)^2(1-\pi_1 p)(1-\pi_1 p_2)}{1+\alpha\pi_1(\alpha(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)-p)}$
1 1 0 0 1 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 :	$\frac{-\alpha^2\pi_1^2(1-\pi_3)^2(1-p)p_2}{\alpha^2\pi_1^2(1-p)(\pi_3(p_2-p_1)-p_2)} = \frac{-\alpha^2\pi_1(1-\pi_3)^2(1-\pi_1 p)p_2}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$
1 1 0 1 1 0 :	$\frac{-\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2 p_2}{\alpha^2\pi_1(1-\pi_1)(\pi_3(p_2-p_1)-p_2)} = \frac{-\alpha^2\pi_1(1-\pi_3)^2(1-\pi_1 p)p_2}{\alpha^2\pi_1(1-\pi_1 p)(\pi_3(p_2-p_1)-p_2)}$
1 1 1 0 0 0 :	$\frac{-\alpha^2\pi_1^2(1-\pi_3)^2 p(1-p_2)}{\alpha\pi_1 p((\alpha-1)(1-\pi_2)-\alpha\pi_1(1-p_2))} = \frac{\alpha^2\pi_1(1-\pi_3)^2 p(1-\pi_1 p_2)}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$
1 1 1 0 0 1 :	$\frac{-\alpha^2\pi_1(1-\pi_1)(1-\pi_3)^2 p}{\alpha\pi_1(\alpha(\pi_1+\pi_2-1)-\pi_2)p} = \frac{\alpha^2\pi_1(1-\pi_3)^2 p(1-\pi_1 p_2)}{\alpha\pi_1 p(1-\alpha\pi_1 p_2)}$
1 1 1 1 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 V.8 and Condition **c**) that the probability distribution of the labelings of $L^{(i)}$ (for $i \in \{0, \ldots, N\}$) is identical to the probability distribution of the labelings of $L^{(0)}$. Hence,

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

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

Appendix VI
Equations for Three-Dimensional Bit Stuffing Used to Obtain Table VII

In this appendix, we list the equations used to obtain Table VII. 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 Fig. 9.

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

$$\boldsymbol{P}\big(\hat{F}(\mathcal{A},\mathcal{C},\mathcal{E},\mathcal{G}) = \hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H}) = (v_1, v_2, v_3, v_4)\big)$$

are given in Table X for different values of (v_1, v_2, v_3, v_4).

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 nine equations implied by Condition **b**). These equations, generated by

$$\boldsymbol{P}\big(\hat{F}(\mathcal{B},\mathcal{D},\mathcal{A},\mathcal{C}) = \hat{F}(\mathcal{F},\mathcal{H},\mathcal{E},\mathcal{G}) = (v_1, v_2, v_3, v_4)\big)$$

are given in Table XI for different values of (v_1, v_2, v_3, v_4).

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 eight equations implied by Condition **c**). Some equations are tautologies—these are omitted. These equations, generated by

$$\boldsymbol{P}\big(\hat{F}(\mathcal{A},\mathcal{B},\mathcal{E},\mathcal{F}) = \hat{F}(\mathcal{C},\mathcal{D},\mathcal{G},\mathcal{H}) = (v_1, v_2, v_3, v_4)\big)$$

are given in Table XII for different values of (v_1, v_2, v_3, v_4).

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

$$\begin{aligned} &\frac{\boldsymbol{P}\big(\hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H},\mathcal{E},\mathcal{G}) = (v_1, v_2, v_3, v_4, v_5, v_6)\big)}{\boldsymbol{P}\big(\hat{F}(\mathcal{F},\mathcal{H},\mathcal{E},\mathcal{G}) = (v_3, v_4, v_5, v_6)\big)} \\ &\qquad = \frac{\boldsymbol{P}\big(\hat{F}(\mathcal{B},\mathcal{D},\mathcal{F},\mathcal{H}) = (v_1, v_2, v_3, v_4)\big)}{\boldsymbol{P}\big(\hat{F}(\mathcal{F},\mathcal{H}) = (v_3, v_4)\big)} \end{aligned}$$

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 in Table XIII for different values of $(v_1, v_2, v_3, v_4, v_5, v_6)$.

Substituting from Theorem IV.2 and from (3) the quantities

$$\begin{aligned} \pi_1 &= \frac{2}{1 + p + \sqrt{(1+3p)(1-p)}} \\ \pi_2 &= \frac{2p}{1 + p + \sqrt{(1+3p)(1-p)}} \\ \pi_3 &= \frac{2(1-p)}{1 - p + \sqrt{(1+3p)(1-p)}} \\ \alpha &= \frac{1}{2}\left(1 + \frac{1-p}{\sqrt{(1+3p)(1-p)}}\right), \end{aligned}$$

the preceding 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

$$\frac{2-(1-p)^2(1+p_1)+(2+(1-p)(1+p_1))p'}{1+p+p'} = 1 \qquad \text{(F1)}$$

$$\frac{1-p-2(1-p)^2+2(1-p)(1-p_2)+p'}{1-p+p'} = 1. \qquad \text{(F2)}$$

where

$$p' = \sqrt{(1+3p)(1-p)}.$$

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

$$\gamma_1 = \alpha^2 \pi_1 \pi_3 (1 - \pi_1 p) \qquad \text{(F3)}$$

$$\gamma_2 = \alpha^2 \pi_1 (1 - \pi_3 (1 - \pi_1 p)). \qquad \text{(F4)}$$

Acknowledgment

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

References

[1] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2260–2299, Oct. 1998.

[2] J. J. Ashley and B. H. Marcus, "Two-dimensional low-pass filtering codes," *IEEE Trans. Commun.*, vol. 46, pp. 724–727, June 1998.

[3] T. Etzion, "Cascading methods for runlength-limited arrays," *IEEE Trans. Inform. Theory*, vol. 43, pp. 319–324, Jan. 1997.

[4] Z. Nagy and K. Zeger, "Asymptotic capacity of two-dimensional channels with checkerboard constraints," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2115–2125, Sept. 2003.

[5] R. E. Swanson and J. K. Wolf, "A new class of two-dimensional runlength-limited recording codes," *IEEE Trans. Magn. Recording*, vol. 28, pp. 3407–3416, Nov. 1992.

[6] R. Talyansky, T. Etzion, and R. M. Roth, "Efficient code construction for certain two-dimensional constraints," *IEEE Trans. Inform. Theory*, vol. 45, pp. 794–799, Mar. 1999.

[7] W. Weeks and R. E. Blahut, "The capacity and coding gain of certain checkerboard codes," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1193–1203, May 1998.

[8] S. Forchhammer, "Upper bounds on the capacity of constrained three-dimensional codes," in *Proc. IEEE 2000 Int. Symp. Information Theory*, Sorrento, Italy, June 2000.

[9] Z. Nagy and K. Zeger, "Capacity bounds for the three-dimensional $(0, 1)$ run length limited channel," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1030–1033, May 2000.

[10] H. Ito, A. Kato, Z. Nagy, and K. Zeger, "Zero capacity region of multidimensional run length constraints," *Electron. J. Comb.*, vol. 6, no. 1, p. R33, 1999.

[11] E. Ordentlich and R. M. Roth, "Two-dimensional weight-constrained codes through enumeration bounds," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1292–1301, July 2000.

[12] A. Vardy, M. Blaum, P. H. Siegel, and G. T. Sincerbox, "Conservative arrays: Multidimensional modulation codes for holographic recording," *IEEE Trans. Inform. Theory*, vol. 42, pp. 227–230, Jan. 1996.

[13] N. J. Calkin and H. S. Wilf, "The number of independent sets in a grid graph," *SIAM J. Discr. Math.*, vol. 11, pp. 54–60, Feb. 1998.

[14] K. Engel, "On the Fibonacci number of an $m \times n$ lattice," *Fibonacci Quart.*, vol. 28, pp. 72–78, 1990.

[15] P. Lee, "Combined error-correcting/modulation recording codes," Ph.D. dissertation, Univ. Calif., San Diego, La Jolla, 1988.

[16] P. Bender and J. K. Wolf, "A universal algorithm for generating optimal and nearly optimal run-length-limited, charge constrained binary sequences," in *Proc. 1993 IEEE Int. Symp. Information Theory*, San Antonio, TX, Jan. 1993, p. 6.

[17] P. H. Siegel and J. K. Wolf, "Bit stuffing bounds on the capacity of 2-dimensional constrained arrays," in *Proc. 1998 IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 323.

[18] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf, "Improved bit-stuffing bounds on 2-dimensional constraints," in *Proc. 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, June/July 2002, p. 385.

[19] R. M. Roth, P. H. Siegel, and J. K. Wolf, "Efficient coding for a two-dimensional runlength-limited constraint," *Proc. SPIE— The International Society for Optical Engineering*, vol. 3802, pp. 8–17, July 1999.

[20] ——, "Efficient coding schemes for the hard-square model," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1166–1176, Mar. 2001.

[21] A. Kato and K. Zeger, "On the capacity of two-dimensional run length constrained channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1527–1540, July 1999.

[22] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3 and 4, pp. 379–423, July and Oct. 1948.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.