Efficient Coding for a Two-Dimensional Runlength-Limited Constraint

Ron M. Roth[a], Paul H. Siegel[b], and Jack K. Wolf [c]

[a]Computer Science Department, Technion, Haifa 32000, Israel
[b]ECE, 0407, University of California, San Diego, 9500 Gilman Drive
La Jolla, CA 92093-0407
[c]CMRR, 0401, University of California, San Diego, 9500 Gilman Drive
La Jolla, CA 92093-0401

ABSTRACT

Runlength-limited (d, k) constraints and codes are widely used in digital data recording and transmission applications. Generalizations of runlength constraints to two dimensions are of potential interest in page-oriented information storage systems. However, in contrast to the one-dimensional case, little is known about the information-theoretic properties of two-dimensional constraints or the design of practical, efficient codes for them.

In this paper, we consider coding schemes that map unconstrained binary sequences into two-dimensional, runlength-limited (d, ∞) constrained binary arrays, in which 1's are followed by at least d 0's in both the horizontal and vertical dimensions. We review the derivation of a lower bound on the capacity of two-dimensional (d, ∞) constraints, for $d \geq 1$, obtained by bounding the average information rate of a variable-to-fixed rate encoding scheme, based upon a“bit-stuffing” technique.

For the special case of the two-dimensional $(1, \infty)$ constraint, upper and lower bounds on the capacity that are very close to being tight are known. For this constraint, we determine the exact average information rate of the bit-stuffing encoder, which turns out to be within 1% of the capacity of the constraint.

We then present a fixed-rate, row-by-row encoding scheme for the two-dimensional $(1, \infty)$ constraint, somewhat akin to permutation coding, in which the rows of the code arrays represent “typical” rows for the constraint. It is shown that, for sufficiently long rows, the rate of this encoding technique can almost achieve that of the variable-rate, bit-stuffing scheme.

Keywords: Constrained arrays, holographic data storage, bit-stuffing encoder

1. INTRODUCTION

Many data storage systems, such as those based upon magnetic and optical recording technology, require the use of constrained modulation codes. These codes efficiently, and invertibly, transform streams of arbitrary binary data into binary sequences that satisfy certain pre-specified constraints. The ensemble of sequences from which the code sequences may be drawn is referred to as a constrained system.

Historically, many digital recording applications have made use of codes over the binary alphabet $\{0, 1\}$ called runlength-limited (RLL) (d, k) codes. The parameters (d, k) represent, respectively, the minimum and maximum admissible number of 0's separating consecutive 1's in any allowable sequence $\mathbf{b} = b_0 b_1 \ldots$. With the advent of page-oriented storage technologies, such as holographic storage, interest in constrained arrays in two or more dimensions has arisen. Among the constraints of theoretical and possible practical interest are 2-D, RLL (d, k) constraints, in which the 1-D RLL (d, k) constraint is satisfied both horizontally and vertically. In both one and two dimensions, the relevant range of parameters is $0 \leq d < k \leq \infty$.

In this paper, we consider coding schemes that map unconstrained binary sequences into 2-D, $(d, k) = (1, \infty)$ constrained arrays. The exact capacity (maximum code rate) of this constraint is not known, but very tight upper and lower bounds have been computed. We present a simple, variable-rate encoding scheme, based upon a “bit-stuffing” technique, and precisely analyze the average code rate, which is shown to fall within 1% of the capacity.

Send correspondence to P. H. Siegel. E-mail: psiegel@ucsd.edu

We then present a fixed-rate, row-by-row encoding scheme, somewhat akin to permutation coding, in which the rows of the code arrays represent "typical" rows for the constraint. It is shown that, for sufficiently long rows, the rate of this encoding technique can almost achieve that of the variable-rate, bit-stuffing scheme.

1.1. Capacity and code construction

The base-2 Shannon capacity[1] $C(\mathcal{S})$ of a one-dimensional (1-D) constrained system $\mathcal{S}$ reflects the growth rate of the number $N(n;\mathcal{S})$ of words of length n in $\mathcal{S}$

$$C(\mathcal{S}) = \lim_{n\to\infty} \frac{1}{n} \log_2 N(n;\mathcal{S}).$$

The capacity represents a tight upper bound on the achievable rates of decodable, finite-state encoders from unconstrained binary data to the system $\mathcal{S}$. Specifically, the finite-state coding theorem[2] states that, for any rate $p/q \leq C(\mathcal{S})$, there exists a decodable, finite-state encoder that encodes a sequence of length-p input strings into a sequence of length-q codewords. Moreover, for a large class of constraints, the decoder can be made state-independent, thereby limiting error-propagation. (For more details regarding 1-D constrained systems and codes, the reader is referred to a recent exposition on this subject.[2])

The capacity of 1-D (d,k) constraints has been computed for a large range of parameters.[3] The capacity, denoted by $C_1(d,k)$, is given by

$$C_1(d,k) = \log_2 \lambda$$

where λ is the largest real root of the polynomial

$$f_{d,k}(x) = x^{k+1} - x^{k-d} - x^{k-d-1} - \cdots - x - 1, \tag{1}$$

for $k < \infty$, and

$$f_{d,\infty}(x) = x^{d+1} - x^{d} - 1 \tag{2}$$

for $k = \infty$.

Similarly, for a 2-D constrained system, the capacity $C_2(\mathcal{S})$ measures the growth rate of the number $N(m,n;\mathcal{S})$ of $m \times n$ arrays in $\mathcal{S}$, and is given by :

$$C(\mathcal{S}) = \lim_{m\to\infty, n\to\infty} \frac{1}{mn} \log_2 N(m,n:\mathcal{S}).$$

For a class of 2-D constrained systems that includes the 2-D (d,k) constraints, it has been shown that, this limit exists even as $m \to \infty$ and $n \to \infty$ independently.[4–7] We will denote the capacity of the 2-D (d,k) constraint by $C_2(d,k)$.

It is easy to see that $C_2(d,k) \leq C_1(d,k)$. However, the corresponding 1-D and 2-D capacities may be quite different. For example,[8] the capacity of the 1-D $(d,k) = (1,2)$ constraint satisfies $C_1(1,2) \approx 0.4057$, whereas the capacity of the corresponding 2-D constraint is $C_2(1,2) = 0$. In fact, any 2-dimensional $(d,k) = (1,2)$ array $\mathbf{x} = x_{i,j}$ containing the row $x_{0,j} = x_j$ must satisfy either $x_{i,j} = x_{i-j}$ for all i,j; or, $x_{i,j} = x_{i+j}$ for all i,j. Thus, the growth rate of such $n \times n$ arrays is only exponential in n, rather than exponential in n^2. Recently, this result has been generalized to a complete characterization of the (d,k) constraints in two dimensions and higher with zero capacity.[7,9] Specifically, $C_n(d,k) = 0$ if and only if $d > 0$ and $k = d+1$, for $n \geq 2$.

The general determination of the capacity of 2-dimensional (d,k) constraints appears to be quite difficult, however. The example of the $(d,k) = (1,\infty)$ constraint, or equivalently the $(d,k) = (0,1)$ constraint obtained by interchanging the roles of 0 and 1, illustrates this point. The problem of computing the capacity of this constraint has arisen in various forms in statistical mechanics and combinatorics, as well as in the information-theoretic context. Calkin and Wilf[10] used a transfer matrix method to derive close lower and upper bounds for the 2-D $(d,k) = (1,\infty)$ constraint, namely

$$0.5879 \leq C_2(1,\infty) \leq 0.5883 \tag{3}$$

These bounds were further improved by Weeks and Blahut,[11] and further improved by Nagy and Zeger,[12] who also extended the technique to three dimensions. These lower and upper bounds now agree out to 9 decimal places

$$0.587891161775 \leq C_2(1,\infty) \leq 0.587891161868. \tag{4}$$

Kato and Zeger[7] used the bounds on $C_2(1,\infty)$ to derive lower bounds on $C_2(d,\infty)$, for $d \geq 2$, and $C_2(0,k)$, for $k \geq 2$. (They noted that Talyansky[13] and Talyansky, *et al.*[14] described a construction that yields a lower bound on $C_2(0,k)$ that is stronger that the Kato-Zeger bound for all $k \geq 8$.)

The bounds on $C_2(d,\infty)$ were obtained by constructing (d,∞) arrays from $(1,\infty)$ arrays, and the bounds on $C_2(0,k)$ were obtained by constructing $(0,k)$ arrays from $(0,1)$ arrays. The lower bounds on $C_2(0,k)$ were then used to derive lower bounds on $C_2(d,k)$ for remaining cases where $k \neq d+1$. Upper bounds on $C_2(d,\infty)$ and $C_2(0,k)$ were also derived.[7] Together with the lower bounds, they imply that, as d grows, $C_2(d,\infty)$ converges to zero exactly at the rate $(\log_2 d)/d$, and they give asymptotic bounds on how fast, as k grows, $C_2(0,k)$ converges to one.

Siegel and Wolf[15] used a different approach to derive lower bounds on $C_2(d,\infty)$, for $d \geq 1$. They computed a simple lower bound on the average information rate of a variable-rate, bit-stuffing encoding algorithm that creates 2-D (d,∞)-constrained arrays from a 1-D sequence produced by a possibly biased binary source. These lower bounds were then optimized with respect to the 1-D binary source probability. The bit-stuffing approach is closely related to one introduced by Lee[16] and Bender and Wolf[17] for 1-D, RLL, charge-constrained $(d,k;c)$ sequences.[3]

For the case $d = 1$, the bit-stuffing bound is $C_2(1,\infty) \geq 0.5514$, which is not as good as the previously cited bounds. However, for a range of values of the parameter d, the lower bounds on $C_2(d,\infty)$ are the best known.[15]

1.2. Outline of paper

In this paper, we concentrate upon coding techniques for the 2-D $(d,k) = (1,\infty)$ constraint. We determine precisely the efficiency of the variable-rate bit-stuffing approach, as well a lower bound on the efficiency of a new fixed-rate row-by-row encoding algorithm. The achievable rate of the variable-rate, bit-stuffing scheme is shown to fall within 1% of the capacity of the constraint. An asymptotic lower bound on the achievable rate of the fixed-rate scheme lies within 1.2% of the capacity.

We remark that the $(1,\infty)$ arrays produced by these efficient encodings, and the $(0,1)$ arrays obtained by interchanging the symbols 0 and 1, can be used to construct arrays satisfying other 2-D (d,k) constraints.[7]

The remainder of the paper is organized as follows. In Section 2, we review the bit-stuffing algorithms for the 2-D (d,∞) constraints and the derivation of the lower bounds on their achievable rates.[15] In Section 3, we determine the exact value of the average information rate of the bit-stuffing encoder for the 2-D $(d,k) = (1,\infty)$ constraint. In Section 4, we describe a fixed-rate encoding algorithm for the 2-D $(d,k) = (1,\infty)$ constraint and present a lower bound on the efficiency of this method. Section 5 concludes the paper.

2. BIT STUFFING LOWER BOUNDS ON $C_2(d,\infty)$

Lee[16] described a variable rate algorithm for creating one-dimensional binary sequences that satisfied the $(0,k;c)$ constraint by stuffing binary digits into an unconstrained data binary sequence. Bender and Wolf[17] generalized this algorithm to create one-dimensional binary $(d,k;c)$ sequences. This generalization was shown to be optimal – in the sense that it created sequences with average rate equal to the capacity of the constrained sequence – for the 1-dimensional $(d,\infty;\infty)$, $(d,d+1,\infty)$, and $(2c-2,\infty;c)$ constraints, and nearly optimal for the other 1-dimensional $(d,k;c)$ constraints. Siegel and Wolf[15] showed that a similar technique can be applied to generate certain 2-D constrained arrays, leading to a lower bound for the capacity of the 2-D constraint, as we now review.

In order to apply bit-stuffing techniques to 2-D constraints, we first specify the order by which binary digits are inserted into the 2-dimensional array, which we represent as follows. A *parallelogram* $\Delta_{m,n}$ is a subset of the integer plane defined by

$$\Delta_{m,n} \;=\; \{(i,j) \in \mathcal{Z}^2 \;:\; 0 \leq i < m,\;\; 0 \leq i+j < n\}$$

(see Fig. 1). *Row* i in $\Delta_{m,n}$ consists of all locations (i,j) such that $-i \leq j < n-i$. *Diagonal* d consists of all locations $(i,d-i)$ such that $0 \leq i < m$. Row 0 will be denoted by $\Delta_n^{(\mathrm{h})}$ and will be referred to as the *horizontal boundary* of $\Delta_{m,n}$. Similarly, Diagonal 0, denoted $\Delta_m^{(\mathrm{d})}$, will be referred to as the *diagonal boundary* of $\Delta_{m,n}$. Those boundaries are depicted as thick lines in Fig. 1.

We write the binary digits along 45 degree diagonals from top right to bottom left. (We may want to initialize the boundary values.[18]) The binary data sequence is first converted by a distribution transformer $\mathcal{E}$ to a sequence of statistically independent binary digits with the probability of a 1 equal to p and the probability of a 0 equal to $(1-p)$. This conversion occurs at a rate penalty of $H_2(p)$, where $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary

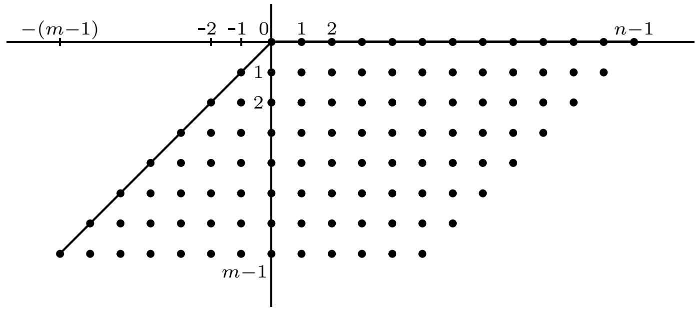

Figure 1. Parallelogram $\Delta_{m,n}$.

entropy function. The purpose of creating an unbalanced sequence will be to write more 0's than 1's. The optimal value of p will be chosen later. We now write the unbalanced sequence (without further coding) down successive diagonals, skipping all positions that contain "stuffed" 0's, which arise in a manner which will now be explained. Whenever a 1 in the unbalanced source sequence is written, d 0's are inserted – or "stuffed"– in the d positions to the right of it and in the d positions below it. It will sometimes occur that a 0 has already been stuffed in some of the positions to the right of the 1, in which case it is not necessary to stuff another 0. In writing the unbalanced sequence down diagonals, any position already filled by a previously stuffed 0 is skipped.

Decoding the array is accomplished by reading down diagonals in a similar manner. The unbalanced binary digits are read successively from the array, with certain 0 bits being ignored. Specifically, whenever a 1 is read from the array, the stuffed 0's to the right of it and below it are normally deleted. It may occur that the stuffed 0's to the right of the 1 have already been deleted, in which case only the stuffed 0's below it are deleted. This procedure reproduces the encoded unbalanced sequence. The original binary data is then obtained from the unbalanced stream by the inverse of the mapping used to create the unbalanced stream of bits.

Define

$$R_d(p) = \frac{H_2(p)}{1+2dp}.$$

The following lemma proves that, for $d > 0$ and any $0 < p < 1$, $R_d(p)$ is a lower bound for $C_2(d, \infty)$.

LEMMA 2.1. *For $d > 0$ and any $0 \le p \le 1$, $C_2(d,\infty) \ge R_d(p)$.*

Sketch of proof: The information rate of the array will be $H_2(p)$ times the ratio of the number of unbalanced binary digits to the total number of binary digits written (unbalanced digits and stuffed digits). We call this ratio Q and our goal is to estimate Q as a function of p. Assume that we have written a long diagonal of length n. It should contain about Qn unbalanced binary digits and about $(1-Q)n$ stuffed 0's. But the $(1-Q)n$ stuffed 0's were caused by at least $[(1-Q)n]/2$ unbalanced 1's written on the previous $d-1$ diagonals. Assuming that the proportion of unbalanced binary digits written on the previous $d-1$ diagonals is also Q (which can be ensured by proper initialization of the boundary of the array[18]), it follows that the number of unbalanced 1's written on the previous $d-1$ diagonals should be, for large n, about $pQdn$. Thus we have that

$$pQdn > \frac{(1-Q)n}{2},$$

or

$$Q > \frac{1}{1+2dp}.$$

The resulting lower bound on the entropy of the 2-D (d,∞) constraint is then given by

$$C_2(d,\infty) = H_2(p)Q > \frac{H_2(p)}{1+2dp} = R_d(p).$$

This completes the derivation. □

The following proposition determines the value of p that maximizes the lower bound of Lemma 2.1.

PROPOSITION 2.2. *For $d > 0$ and any $0 \leq p \leq 1$, $C_2(d, \infty) \geq C_1(2d, \infty)$.*

Proof: To determine the value of p that maximizes this lower bound, we differentiate $R_d(p)$ with respect to p and set the result equal to 0. The result shows that the optimal value p_* satisfies the equation

$$p_* = (1 - p_*)^{2d+1}. \tag{5}$$

Setting $\lambda_* = (1 - p_*)^{-1}$, (5) can be rewritten as

$$\lambda_*^{2d+1} - \lambda_*^{2d} - 1 = 0.$$

The corresponding maximum value of the lower bound is given by

$$R_d(p_*) = -\log_2 (1 - p_*). \tag{6}$$

Referring to (2), we conclude from (6) that the corresponding lower bound $R_d(p_*)$ is precisely the capacity $C_1(2d, \infty)$ of the 1-dimensional $(2d, \infty)$ constraint. This completes the proof. □

Table 1 shows the lower bound $R_d(p_*) = C_1(2d, \infty)$ for small values of d.

Table 1. Bit-stuffing lower bounds for $C_2(d, \infty)$

d	$C_1(2d, \infty)$
1	0.5514
2	0.4057
3	0.3282
4	0.2788
5	0.2440

Referring to the bound on $C_2(1, \infty)$ we conclude that the simple bit-stuffing encoder for the 2-D $(1, \infty)$ constraint achieves an efficiency at least 0.937. In Section 3, we calculate precisely the average information rate of this bit-stuffing encoder, and conclude that its efficiency is, in fact, in excess of 99%.

3. EXACT ANALYSIS OF THE 2-D $(1, \infty)$ ENCODER

In this section, we determine the exact value of the average information rate of the bit-stuffing encoder for the 2-D $(1, \infty)$ constraint, as described in Section 2. We first establish some notation and terminology. Fig. 2 depicts the four possible configurations for a consecutive pair of diagonal entries in the array. Let D_1 denote the event that two consecutive entries on a diagonal, $x_{i-1,j}$ and $x_{i,j-1}$ both equal 1. Let D_2 denote the event that they satisfy $x_{i-1,j} = 1$, $x_{i,j-1} = 0$. Let D_3 correspond to the event $x_{i-1,j} = 0$, $x_{i,j-1} = 1$, and, finally, let D_4 denote the event that $x_{i-1,j} = x_{i,j-1} = 0$.

Suppose that the biased source generates a 1 with probability p, and a 0 with probability $q = 1 - p$. Over the ensemble of the 2-D arrays in the image of the bit-stuff encoder, let $x = Pr(D_1)$, $y = Pr(D_2) = Pr(D_3)$, and $z = Pr(D_4)$. Note that $Pr(D_2) = Pr(D_3)$ by symmetry properties of the family of 2-D $(1, \infty)$ constrained arrays.

The following proposition gives the exact information rate $\mathcal{R}(p)$ of the bit-stuffing encoder for the 2-D $(1, \infty)$ constraint with specified biasing probability $0 < p < 1$.

PROPOSITION 3.1.

$$\mathcal{R}(p) = H_2(p) \frac{(4 - 3q) + \sqrt{(4 - 3q)^2 - 4(1 - q)(4 - 3q)}}{2(1 - q)(4 - 3q)}. \tag{7}$$

Proof: It is easy to see that the average information rate of the bit-stuff encoder, $\mathcal{R}(p)$, is given by

$$\mathcal{R}(p) = H_2(p)Pr(D_4) = H_2(p)z. \tag{8}$$

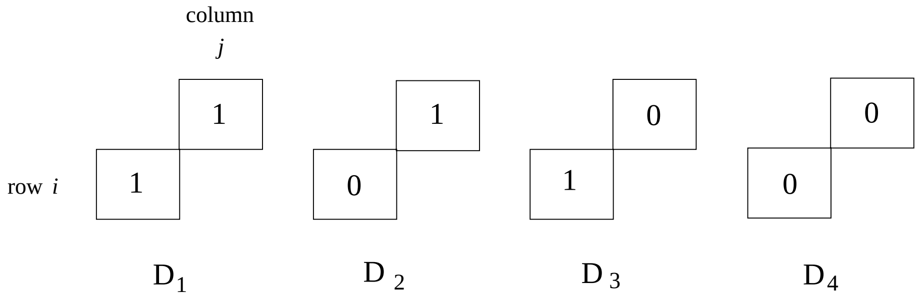

Figure 2. Configurations of consecutive diagonal entries.

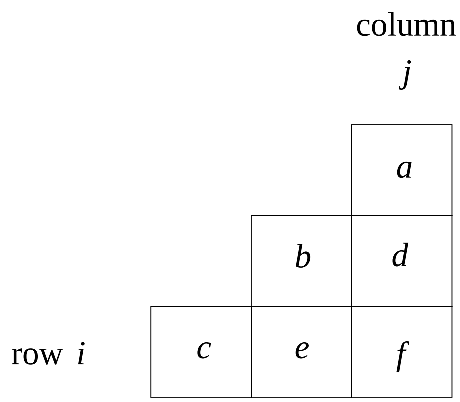

Figure 3. Array configuration.

We now develop a set of relations among the quantities x, y, and z, involving the source parameter p. From these, we will derive an expression for z in terms of p, leading to (7). We first note that

$$Pr(x_{i,j} = 1) = x + y, \tag{9}$$

and, therefore,

$$Pr(x_{i,j} = 0) = 1 - (x + y). \tag{10}$$

Since any consecutive pair of diagonal entries belongs to exactly one of the disjoint events D_1, D_2, D_3, D_4, we have the relation

$$x + 2y + z = 1. \tag{11}$$

For convenience, denote the array entries $x_{i-2,j}, x_{i-1,j-1}, x_{i,j-2}, x_{i-1,j}, x_{i,j-1}$, and $x_{i,j}$ by a, b, c, d, e, and f, respectively, as depicted in Fig. 3. Under suitable boundary conditions on the array, we can assume that the entries along diagonals form a first-order Markov process.[18] Using the implied independence of entries separated by 2 positions along a diagonal, we can deduce another relation involving x, y, z and p. Specifically, noting that $z = Pr(d = e = 0)$, we can write

$$\begin{aligned} z \;\; = \;\; & Pr(b=1) + Pr(b=0) \cdot [Pr(a=1|b=0)Pr(c=1|b=0) + Pr(a=0|b=0)Pr(c=1|b=0)q \\ & + Pr(a=1|b=0)Pr(c=0|b=0)q + Pr(a=0|b=0)Pr(c=0|b=0)q^2\big] \,. \end{aligned} \tag{12}$$

This relation can be rewritten using the following expression for the constituent conditional probabilities

$$Pr(a = 1|b = 0) = \frac{Pr(a = 1, b = 0)}{Pr(b = 0)} = \frac{y}{1 - (x + y)}, \tag{13}$$

$$Pr(c=1|b=0) = \frac{Pr(a=1,b=0)}{Pr(b=0)} = \frac{y}{1-(x+y)}, \tag{14}$$

and

$$Pr(a=0|b=0) = Pr(c=0|b=0) = \frac{z}{1-(x+y)}. \tag{15}$$

Substituting (9), (10), (13), (14), (15) into (12) yields:

$$z = x + y + \frac{(y+zq)^2}{1-(x+y)}. \tag{16}$$

Now, referring again to Fig. 3, we can see that

$$Pr(f=0) = Pr(d=e=1) + Pr(d=1,e=0) + Pr(d=0,e=1) + Pr(d=e=0)q, \tag{17}$$

or, equivalently, from (10)

$$1 - x - y = x + 2y + zq. \tag{18}$$

Using (11), we conclude that

$$x + y = z - zq. \tag{19}$$

Substitution of (19) into (16), followed by some straightforward algebraic manipulations, yields

$$y^2 + (2zq)y + z(z-1)q = 0. \tag{20}$$

From (11), we know

$$x + y = 1 - y - z \tag{21}$$

and substitution of this into (19) yields the relation

$$y = 1 + zq - 2z. \tag{22}$$

Combining (22) with (20) yields an equation which is quadratic in z, namely

$$z^2(1-q)(4-3q) - z(4-3q) + 1 = 0. \tag{23}$$

Solving (23) for z via the quadratic formula, we get

$$z = \frac{(4-3q) + \sqrt{(4-3q)^2 - 4(1-q)(4-3q)}}{2(1-q)(4-3q)}. \tag{24}$$

Combining (24) and (8) yields (7), completing the proof. □

We numerically determined the value of $q = 1 - p$ that maximizes the average information rate $\mathcal{R}(p) = H_2(p)z$ of the bit-stuff encoder. The optimal value $q_{opt} \approx 0.644400$ yields the information rate $\mathcal{R}(q_{opt}) \approx 0.583056$, which, referring to the bounds in (4), is within 1% of the capacity $C_2(1,\infty)$.

One can also consider more sophisticated bit-stuffing encoders, in which two distribution transformers $\mathcal{E}_0$ and $\mathcal{E}_1$ are used. These transformers take sequences of independent, equiprobable bits and convert them into sequences of independent bits with the probability of a 0 equal to q_0 and q_1, respectively. These conversions occurs at a rate penalty of $H_2(q_0)$ and $H_2(q_1)$, respectively. When the bit-stuffing encoder is preparing to write a bit into the open diagonal position i,j (which implies $x_{i-1,j} = x_{i,j-1} = 0$), it conditions the choice of the transformer from which the bit to be written will be selected upon the value $x_{i-1,j+1}$, the preceding bit in the diagonal being written. This enhanced encoding algorithm generalizes the bit-stuffing encoder that we analyzed above, which corresponds to the case $q_0 = q_1$. The determination of the average information rate of this enhanced bit-stuffing encoder is rather involved,[18] but the result implies that the maximum encoding rate is obtained by setting $q_0 \approx 0.671833$ and $q_1 \approx 0.566932$. These biasing probabilities yield an average rate $\mathcal{R}(q_0, q_1) \approx 0.587277$, which is within 0.1% of the capacity $C_2(1,\infty)$.

Remark: Forchhammer and Justesen[19] and Justesen and Shtarkov[20] have also studied the entropy of 2-D constrained random fields with simple structure on rows/columns/diagonals. Some of their results pertain to the $C_2(1,\infty)$ constrained system investigated in this paper.

4. FIXED-RATE ENCODER FOR THE 2-D $(1, \infty)$ CONSTRAINT

In this section, we describe a fixed-rate coding scheme into rectangular arrays satisfying 2-D $(1, \infty)$ constraints and we state an asymptotic lower bound on the achievable rate of the encoder, which is within 1.2% of the capacity.[18]

Denote by $B_{m,n}$ the rectangle defined as the following subset of the integer plane

$$B_{m,n} = \{(i,j) \in \mathcal{Z}^2 \ : \ 0 \le i < m, \ 0 \le j < n\}. \tag{25}$$

A $B_{m,n}$–*array* is an assignment of binary digits to entries in the rectangle $B_{m,n}$. A $B_{m,n}$–array $\boldsymbol{x} = (x_{i,j})$ is called *circular with respect to the* $(1, \infty)$ *constraint* if $\boldsymbol{x}$ satisfies the constraint and for every $0 \le i < m$, the entries $x_{i,0}$ and $x_{i,n-1}$ are not both 1. In other words, every cyclic shift of the columns of $\boldsymbol{x}$ results in a $B_{m,n}$–array that satisfies the constraint. We denote the set of all $B_{m,n}$–arrays that are circular with respect to the $(1, \infty)$ constraint by $\mathcal{S}^\circ_{m,n}$.

We now present a fixed-rate coding scheme into $\mathcal{S}^\circ_{m,n}$. The circular property is not necessary for the coding, but it simplifies the analysis of the scheme. The $B_{m,n}$–arrays generated by the encoder will have the property that all rows in all of the arrays have the same Hamming weight δn for a value of $\delta \in [0, 1]$ specified below.

Let $\boldsymbol{x}$ be in $\mathcal{S}^\circ_{m,n}$ and assume that for some i in the range $1 \le i < m$, row $i-1$ in $\boldsymbol{x}$ has weight t. Let $j_1, j_2, \ldots, j_t$ be the indexes j for which $x_{i-1,j} = 1$. Then, it follows that $x_{i,j_k} = 0$ for every $1 \le k \le t$. We define the words

$$x_i^{(k)} = x_{i,j_k+1} x_{i,j_k+2} \ldots x_{i,j_{k+1}-1}, \ 1 \le k < t,$$

and

$$x_i^{(t)} = x_{i,j_t+1} \ldots x_{i,n-1} x_{i,0} \ldots x_{i,j_1-1}.$$

We refer to $x_i^{(k)}$ as the *kth phrase* of row i in $\boldsymbol{x}$. Row i can be obtained by shifting the word

$$0 \, x_i^{(1)} \, 0 \, x_i^{(2)} \, 0 \ldots 0 x_i^{(t)}$$

cyclically j_1 entries to the right. We refer to the length of $x_i^{(k)}$ as the *kth phrase length* in row i of $\boldsymbol{x}$. Denoting that length by l_k, the list of phrase lengths $(l_1, l_2, \ldots, l_t)$ is called the phrase profile of row i in $\boldsymbol{x}$. It is immediate that $l_k = j_{k+1} - j_k - 1$, where $j_t + 1$ is defined to be j_1. Thus, the phrase profile of row i is completely determined by row $i-1$ and $\sum_{k=1}^{t} l_k = n - t$.

For a positive integer ℓ, let $\mathcal{S}_\ell = \mathcal{S}_{1,\ell}$ denote the set of all words of length ℓ that satisfy the one-dimensional $(1, \infty)$-RLL constraint. Similarly, we define $\mathcal{S}^\circ_\ell = \mathcal{S}^\circ_{1,\ell}$. Also, denote by $\mathcal{S}_\ell(r)$ (respectively, $\mathcal{S}^\circ_\ell(r)$) the set of words in $\mathcal{S}_\ell$ (respectively, $\mathcal{S}^\circ_\ell$) of weight r. It is easy to see that a $B_{m,n}$–array $\boldsymbol{x} \in \mathcal{S}_{m,n}$ is in $\mathcal{S}^\circ_{m,n}$ if and only if every row in $\boldsymbol{x}$ is in $\mathcal{S}^\circ_n$.

Let x and y be two words in $\mathcal{S}_n$. We say that x is *consistent* with y if x and y form the rows of an array in $\mathcal{S}_{2,n}$. In other words, x and y do not have 1's in the same position.

Define

$$K(n,t) \ = \ \sum_{s=0}^{t-1} 2^s \cdot \binom{t-1}{s} \cdot |\mathcal{S}_{n-3t+2}(t-s)| \ . \tag{26}$$

The following lemma gives a lower bound on the number of words in $\mathcal{S}^\circ_n(t)$ that are consistent with a specified word x.

LEMMA 4.1. *For every word $x \in \mathcal{S}^\circ_n(t)$ there are at least $K(n,t)$ words $y \in \mathcal{S}^\circ_n(t)$ that are consistent with x.* □

Let $t_{\max} = t_{\max}(n)$ be the value of a nonnegative integer t for which $K(n,t)$ is maximized. The following result is a direct corollary of Lemma 4.1.

PROPOSITION 4.2.

$$\frac{\log_2 |\mathcal{S}^\circ_{m,n}|}{mn} \ \ge \ \frac{\log_2 K(n, t_{\max})}{n} \ .$$

□

Given n, m, and t (e.g., $t = t_{\max}(n)$), Lemma 4.1 suggests a coding scheme at a fixed rate $(\log_2 K(n,t))/n$ into the set $\mathcal{S}^\circ_{m,n}$ as follows. For $i = 0, 1, \ldots, m-1$, we select row i from $\mathcal{S}^\circ_n(t)$ so that it is consistent with row $i-1$ (for the case $i = 0$, we can assume a particular word from $\mathcal{S}_n^{\circ}(t)$ to serve as a 'phantom' row -1). Lemma 4.1 guarantees that we have at least $K(n,t)$ words in $\mathcal{S}_n^{\circ}(t)$ that can be selected for row i.

The effective computation of row i can be done by enumerative coding, as we now briefly describe.[18] Let $(\ell_1, \ell_2, \ldots, \ell_t)$ be the phrase profile of row i as induced by row $i{-}1$. For this particular phrase profile, denote by $M_{k,s}$ the number of possible assignments for the first k phrases of row i so that their overall weight is s, $0 \le s \le t$. Also define $T_{\ell,r} = |\mathcal{S}_\ell(r)|$.

The enumerative coding algorithm of row i proceeds as follows. The unconstrained input stream to be coded into row i is regarded as an integer p in the range $0 \le p < K(n,t)$, and the phrase profile of row i is also assumed to be available. The main loop of the algorithm computes the phrases of row i, in reverse order, starting with the tth phrase. In each iteration of the main loop, the variable η determines the weight of the kth phrase, and s equals the overall weight of the first $k{-}1$ phrases. It can be easily verified by descending induction on k that each loop iteration starts with a value of p in the range $0 \le p < M_{k,s}$, the induction base following from $0 \le p < K(n,t) \le M_{t,t}$. Similarly, the value of θ at the end of each loop iteration lies in the range $0 \le \theta < T_{\ell_k,\eta} = |\mathcal{S}_{\ell_k}(\eta)|$. The mapping from θ into a word in $\mathcal{S}_{\ell_k}(\eta)$ assumes an ordering on the elements of each set $\mathcal{S}_\ell(r)$. If the standard lexicographic ordering is used, then such a mapping can be efficiently implemented by a second level of enumerative coding, using recurrence relations for $T_{\ell,r}$.

We obtain an asymptotic estimate for $K(n,t)$ which enables us to compute an asymptotic lower bound on $(\log_2 K(n, t_{\max}))/n$. Let $\delta = t/n$. Then, the resulting bound[18] is given by

$$\limsup_{n\to\infty} \, (1/n) \cdot \log_2 K(n, \delta n) \;\ge\; \sup_{\rho} F(\delta,\rho) \, , \tag{27}$$

where

$$F(\delta,\rho) \;=\; \delta \cdot [1 + h((1/\delta - 3)\rho)] + (1{-}3\delta) \cdot [(1{-}\rho) \cdot h(\rho/(1{-}\rho)) - \rho] \, ,$$

and the supremum in the right-hand side of (27) is taken over all rational ρ in the range $0 \le \rho \le \min\{\delta/(1{-}3\delta), 1/2\}$. In fact, the bound (27) is tight, and, since the function $F(\delta,\rho)$ is continuous, the supremum in (27) can be replaced by a maximum over all real $\rho \in [0, \min\{\delta/(1{-}3\delta), 1/2\}]$.

By taking partial derivatives of $F(\delta,\rho)$ with respect to δ and ρ, we get the equations

$$(23\delta - 4)(29\delta - 4)(8357\delta^5 - 8357\delta^4 + 3098\delta^3 - 518\delta^2 + 38\delta - 1) \;=\; 0$$

and

$$\rho \;=\; \frac{\delta \cdot (369\delta^2 - 101\delta + 4)}{1469\delta^3 - 682\delta^2 + 95\delta - 4} \, .$$

The maximum is attained for $(\delta_{\max}, \rho_{\max}) = (0.216594, 0.248986)$, implying that

$$\begin{aligned} &\limsup_{n\to\infty} \, (1/n) \cdot \log_2 K(n, t_{\max}(n)) \\ &\quad \ge \; \max_{(\delta,\rho)} F(\delta,\rho) \;=\; F(\delta_{\max}, \rho_{\max}) \;=\; 0.581074, \end{aligned}$$

and it can be shown that the inequality can be replaced by equality. Thus, the asymptotic code rate lies within 1.2% of $C_2(1,\infty)$.

5. CONCLUSIONS

In this paper, we have addressed problems pertaining to efficient coding for 2-dimensional runlength-limited (d,k) constrained arrays. Following a discussion of bounds on the Shannon capacity of these constrained systems, we presented and analyzed two efficient coding algorithms for the $(d,k) = (1,\infty)$ constraint: a variable-rate scheme based upon bit-stuffing, and a fixed-rate enumerative coding scheme that generates rows of equal Hamming weight. Both schemes are shown to be extremely efficient, asymptotically achieving code rates that are within 1% and 1.2% of the capacity, respectively.

ACKNOWLEDGMENTS

The authors thank Søren Forchhammer, Jørn Justesen, Erik Ordentlich, Alon Orlitsky, and Ken Zeger for very helpful comments. This work was supported in part by Grant No. 95-00522 from the United-States–Israel Binational Science Foundation (BSF), Jerusalem, Israel, by Grant No. NCR-9612802 of the National Science Foundation (NSF), and by the Center for Magnetic Recording Research at the University of California, San Diego.

REFERENCES

1. C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.,* vol. 27, no. 10, pp. 379–423, 623–656, Oct. 1948.
2. B. H. Marcus, R. M. Roth, and P. H. Siegel, "Constrained Systems and Coding for Recording Channels," *Handbook of Coding Theory,* Elsevier Scientific Publishers, 1998.
3. K. Norris and D.S. Bloomberg, "Channel capacity of charge-constrained run-length limited systems," *IEEE Trans. Magn.*, vol. MAG-17, no. 6, pp. 3452–3455, Nov. 1981.
4. N. Markley and M. Paul, "Maximal measures and entropy for Z^ν subshifts of finite type," *Classical Mechanics and Dynamical Systems,* Dekker Notes, eds. R. Devaney and Z. Nitecki, 70 (1981), pp. 135–157.
5. J. Justesen and Y. M. Shtarkov, "The combinatorial entropy of images," *Prob. Pered. Inform.* (in Russian), vol. 33, pp. 3–11, 1997.
6. Y. M. Shtarkov, S. Forchhammer, and J. Justesen, "On definitions and existence of combinatorial entropy of 2d fields," Tech. Univ. Denmark, Tech. Rep. TR-TELE-10, 1998.
7. A. Kato and K. Zeger, "On the capacity of two-dimensional run-length constrained channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1527–1540, July 1999.
8. J. J. Ashley and B. H. Marcus, "Two-dimensional low-pass filtering codes," IBM Research Division, Almaden Research Center, IBM Research Report RJ 10045 (90541), Oct. 1996.
9. H. Ito, A. Kato, A. Nagy, and K. Zeger, "Zero capacity region of multidimensional run length constraints," submitted to *Electron. J. Combinatorics,* Apr. 24, 1999.
10. N. J. Calkin and H.S. Wilf, "The number of independent sets in a grid graph," *SIAM J. Discrete Mathematics,* vol. 11, pp. 54–60, Feb. 1998.
11. W. Weeks IV an R. E. Blahut, "The capacity and coding gain of certain checkerboard codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1193–1203, May 1998.
12. Z. Nagy and K. Zeger, "Capacity bounds for the 3-dimensional (0,1) runlength limited channel," submitted to *IEEE Trans. Inform. Theory,* Dec. 17, 1998.
13. R. Talyansky, *Coding for Two-Dimensional Constraints,* M.Sc. Thesis, Computer Science Department, Technion-Israel Inst. Technol., Haifa, Israel, 1997 (in Hebrew).
14. R. Talyansky, T. Etzion, and R. M. Roth, "Efficient constructions for certain two-dimensional constraints," *IEEE Trans. Inform. Theory,* vol. 45, no. 2, pp. 794-799, Mar. 1999.
15. P. H. Siegel and J. K. Wolf, "Bit-stuffing bounds on the capacity of 2-dimensional constrained arrays," *Proc. 1998 IEEE Int. Symp. Inform. Theory,* Cambridge, Massachusetts, p. 323, Aug. 1998.
16. P. Lee, Combined error-correcting/modulation recording codes, Ph.D. Dissertation, University of California, San Diego, 1988.
17. P. Bender and J. K. Wolf, "A universal algorithm for generating optimal and nearly optimal run-length-limited, charge constrained binary sequences," *Proc. 1993 IEEE Int. Symp. Inform. Theory,* San Antonio, Texas, Jan. 17–22, 1997, p. 6.
18. R. M. Roth, P. H. Siegel, and J. K. Wolf, "Efficient coding schemes for the two-dimensional runlength-limited $(1, \infty)$ constraint," in preparation.
19. S. Forchhammer and J. Justesen,"Entropy bounds for constrained two-dimensional random fields," *IEEE Trans. Inform. Theory,* vol. 45, no. 1, pp. 118–127, Jan. 1999.
20. J. Justesen and Y. M. Shtarkov, "Simple models of two-dimensional information sources and codes," preprint, 1998.