Lower Bounds on the Capacity of Asymmetric Two-Dimensional (d, ∞)-Constraints

Jiangxin Chen
Qualcomm, Inc.
5775 Morehouse Drive
San Diego, CA 92122, U.S.A.
Email: chenj@qualcomm.com

Paul H. Siegel
Department of Electrical and Computer Engineering
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093, U.S.A.
Email: psiegel@ucsd.edu

***Abstract*— We derive lower bounds on the capacity of asymmetric two-dimensional (d, ∞)-constraints from bounds on the output entropy of bit-stuffing encoders for general asymmetric (d, ∞)-constraints on both the square lattice and the hexagonal lattice. For the $(d, \infty; 1, \infty)$-constraint on the square lattice and the $(d, \infty; 1, \infty; 1, \infty)$-constraint on the hexagonal lattice, we derive exact encoder output entropies which provide even tighter bounds on the capacity of these constraints.**

I. Introduction

One-dimensional runlength-limited (RLL) (d, k)-constraints have been widely used in magnetic recording systems to improve the reliability of data storage [1], where d and k represent, respectively, the minimum and the maximum number of 0's that separate two consecutive 1's in a binary sequence. Recently, two-dimensional RLL (d, k)-constraints have arisen in the context of page-oriented storage technologies, such as holographic memories, that can potentially offer data storage densities much higher than those thought to be achievable using conventional storage techniques [2]. These new technologies store the data in a two-dimensional array over, for example, a square lattice or hexagonal lattice. The readback channel suffers from two-dimensional intersymbol interference, and two-dimensional RLL (d, k)-constraint can be applied to reduce the number of errors in the data recovery process.

A two-dimensional array on a square lattice satisfies the $(d_1, k_1; d_2, k_2)$-constraint if it satisfies a one-dimensional (d_1, k_1)-constraint horizontally and a one-dimensional (d_2, k_2)-constraint vertically. A hexagonal lattice can be mapped to a square lattice [3] in such a way that a $(d_1, k_1; d_2, k_2; d_3, k_3)$-constrained array on the hexagonal lattice becomes equivalent to a constrained array on a square lattice which satisfies a (d_1, k_1)-constraint horizontally, a (d_2, k_2)-constraint vertically, and a (d_3, k_3)-constraint along the diagonal from the upper right to the lower left. If the one-dimensional constraints in different two-dimensional directions are not the same, we call the constraint an asymmetric two-dimensional constraint. If they are identical in all directions, we call it a symmetric two-dimensional constraint. For a two-dimensional constraint $\mathcal{S}$, we denote by $S(\Delta_{m,n})$ the set of arrays on the parallelogram

$$\Delta_{m,n} = \left\{(i, j) \in Z^2 : 0 \le i < m,\ 0 \le i + j < n\right\}$$

that satisfy the constraint $\mathcal{S}$. The capacity of the constraint is defined as:

$$\mathsf{cap}(S) = \lim_{m,n\to\infty} \frac{\log_2 |S(\Delta_{m,n})|}{mn}.$$

An important observation on the capacity of an asymmetric two-dimensional constraint on the square lattice was made in [9]. We repeat it here and extend it to constraints on the hexagonal lattice.

Lemma 1.1: On the square lattice,

$$\mathsf{cap}\left(S_{sq}(d_1, k_1; d_2, k_2)\right) = \mathsf{cap}\left(S_{sq}(d_2, k_2; d_1, k_1)\right).$$

On the hexagonal lattice,

$$\mathsf{cap}\left(S_{hex}(d_1, k_1; d_2, k_2; d_3, k_3)\right) = \mathsf{cap}\left(S_{hex}(d_a, k_a; d_b, k_b; d_c, k_c)\right),$$

where $\{a, b, c\}$ is a permutation of $\{1, 2, 3\}$.

Although the definition of a two-dimensional (d, k)-constraint is a simple extension of its one-dimensional counterpart, the capacity of such a constraint is much more difficult to analyze. Major efforts have been focused on the evaluation of symmetric (d, k)-constraints, especially the $(1, \infty)$ constraint (see *e.g.* [4]–[7]). Kato and Zeger [8] studied the zero-capacity region for the symmetric two-dimensional (d, k)-constraints and provided a series of bounds on the (d, k)-constraints with non-zero capacity. Siegel and Wolf [5] used a two-dimensional bit-stuffing encoder to derive a lower bound on the capacity of the symmetric (d, ∞)-constraint. These bounds were further improved in [10]. The capacities of the asymmetric (d, k)-constraints are even less known. Kato and Zeger [9] partially characterized the positive capacity region for the asymmetric (d, k)-constraints on the square lattice. No good capacity bounds are known for these constraints.

In this paper, we derive lower bounds on the capacity of asymmetric (d, ∞)-constraints by extending some of the bounds on the capacity of symmetric (d, ∞)-constraints developed in [6], [10]. Specifically, we modify the bit-stuffing encoders in [5], [6] to generate two-dimensional arrays that satisfy the asymmetric constraints, and the lower bounds on capacity are obtained by evaluating the output entropies induced by the encoders. Due to space limitations, details of some derivations and proofs are omitted.

The paper is organized as follows. Section II presents lower bounds on the capacity of the general $(d_1, \infty; d_2, \infty)$-constraint on the square lattice and the general $(d_1, \infty; d_2, \infty; d_3, \infty)$-constraint on the hexagonal lattice. These bounds are obtained by analysis of the lower bound on the output entropy of corresponding bit-stuffing encoders for these general constraints. In Section III, we show that the encoder outputs for the $(d, \infty; 1, \infty)$-constraint and the $(d, \infty; 1, \infty; 1, \infty)$-constraint are stationary dth order Markov processes ($d \geq 2$). This result allows us to obtain a tighter lower bound for these specific constraints through an exact evaluation of the encoder output entropy. Some numerical results are given for small values of d.

II. Bit-Stuffing Bounds on General Asymmetric (d, ∞)-Constraints

A. Lower Bound for $S_{sq}(d_1, \infty; d_2, \infty)$

The encoder for the $(d_1, \infty; d_2, \infty)$-constraint on the square lattice is a modification of the encoder for the symmetric (d, ∞)-constraint that was presented in [5]. It first converts a binary, equiprobable, i.i.d. data sequence to a sequence of statistically independent binary digits with the probability of 1 equal to p and the probability of 0 equal to $1 - p$. The conversion incurs a rate penalty of $H(p)$ where $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$. The resulting p-biased sequence is then written into the parallelogram $\Delta_{m,n}$ along successive rows according to the following "stuffing" rule. We begin from the origin. Whenever a 1 is written, additional 0's are stuffed into the d_1 positions immediately to the right of it and the d_2 positions immediately below it. During the writing process, the encoder skips those positions that are already occupied by stuffed 0's and writes into the next available position in the row. The decoder can uniquely decode the array by reading along the rows and discarding the stuffed 0's.

This encoder generates output arrays x on the parallelogram $\Delta_{m,n}$ with probability $\mu_{m,n}(x) = \text{Prob}\{X = x\}$. The entropy of the output array

$$H(\mu_{m,n}) = -\frac{1}{mn} \sum_{x \in S(\Delta_{m,n})} \mu_{m,n}(x) \log_2 \mu_{m,n}(x)$$

satisfies the following inequality

$$H(\mu_{m,n}) \leq \frac{\log_2 |S(\Delta_{m,n})|}{mn}.$$

Thus

$$H(\mu) = \lim_{m,n\to\infty} H(\mu_{m,n}) \leq \mathsf{cap}\left(S_{sq}(d_1, \infty; d_2, \infty)\right).$$

In [10], it was demonstrated that a good lower bound on the capacity of a symmetric two-dimensional (d, ∞)-constraint can be obtained by carefully studying the "double-stuffing" events (*i.e.*, when a location occupied by a stuffed 0 is actually stuffed by two 1's). In this paper, we extend this approach to asymmetric constraints and obtain the following lower bound.

Theorem 2.1: A lower bound on the capacity of $(d_1, \infty; d_2, \infty)$-constraint ($d_1 \neq d_2$) on the square lattice is

$$\mathsf{cap}\left(S_{sq}(d_1, \infty; d_2, \infty)\right) \geq \max_{0<p<1} \frac{H(p)}{1 + (d_1 + d_2)p - p^2\left(1 - p^{2d_{min}}\right)},$$

where $d_{min} = \min\{d_1, d_2\}$.

B. Lower Bound for $S_{hex}(d_1, \infty; d_2, \infty; d_3, \infty)$

For constraints on the hexagonal lattice, we apply the transformation mentioned above, and describe the encoder in terms of the equivalent constraints on the square lattice. The encoder operation is very similar to that of the encoder for the $(d_1, \infty; d_2, \infty)$-constraint on the square lattice, except for the following modified "stuffing" rule: after writing a 1 into the array on the parallelogram $\Delta_{m,n}$, we insert additional 0's in the d_1 positions immediately to the right of it, the d_2 positions immediately below it, and the d_3 positions along the upper-right-to-lower-left diagonal immediately below it.

The entropy of the encoder output arrays also satisfies

$$H(\mu) = \lim_{m,n\to\infty} H(\mu_{m,n}) \leq \mathsf{cap}\left(S_{hex}(d_1, \infty; d_2, \infty; d_3, \infty)\right).$$

Using an approach similar to that used in [10], we can obtain the following lower bound on the capacity of an asymmetric (d, ∞)-constraint on the hexagonal lattice by carefully studying the local properties of the encoded array.

Theorem 2.2: Let d_{med} be the median of d_1, d_2 and d_3. When $d_{med} > 1$, a lower bound on the capacity of the $(d_1, \infty; d_2, \infty; d_3, \infty)$-constraint on the hexagonal lattice is

$$\mathsf{cap}\left(S_{hex}(d_1, \infty; d_2, \infty; d_3, \infty)\right) \geq \max_{0<p<1} \frac{H(p)}{1 + (d_1 + d_2 + d_3)p - p^2}.$$

When $d_{med} = 1$, we can derive a lower bound based on the exact output entropy of the bit-stuffing encoder, as will be shown in the next section.

III. Improved Lower Bounds for $S_{sq}(d, \infty; 1, \infty)$ and $S_{hex}(d, \infty; 1, \infty; 1, \infty)$

The bounds in the previous section are obtained by evaluating a lower bound on the output entropy of the corresponding bit-stuffing encoder. For the $(d, \infty; 1, \infty)$-constraint and the $(d, \infty; 1, \infty; 1, \infty)$-constraint ($d > 1$), we can obtain a tighter bound by evaluating the exact output entropy of another encoder, as we now describe.

The encoder $\mathcal{E}$ that we investigate is a generalization of that used for the symmetric $(1, \infty)$-constraint [6]. First, the encoder $\mathcal{E}$ converts the binary, equiprobable, i.i.d. data sequence into two streams, each with statistically independent binary digits, denoted stream 0 and stream 1. The probability of 0 is q_0 in data stream 0 while the probability of 0 is q_1 in data stream 1. The encoder then writes the two biased data streams along successive rows in the parallelogram $\Delta_{m,n}$ according to the following rules. When the position to be written is (i, j), the encoder inserts a 0 into the array element $X_{i,j}$ if any of the elements $X_{i-1,j}$, $X_{i,j-1}, X_{i,j-2}, \ldots, X_{i,j-d}$ has the value 1. The configuration of these array elements is shown below.

$$\begin{array}{ccccc} & & & X_{i-1,j} & X_{i-1,j+1} \\ X_{i,j-d} & \ldots & X_{i,j-1} & X_{i,j} & \end{array}$$

Otherwise, it inserts an information bit from data stream 0 if $X_{i-1,j+1} = 0$, and from data stream 1 if $X_{i-1,j+1} = 1$.

When $0 < q_0, q_1 < 1$ and $q_0 \neq q_1$, encoder $\mathcal{E}$ generates arrays that satisfy the $(d, \infty; 1, \infty)$-constraint on the square lattice and is distinct from that used in Section II-A. Only when $0 < q_0 = q_1 < 1$, does it become the bit-stuffing encoder for the $(d, \infty; 1, \infty)$-constraint described in Section II-A. When $q_1 = 1$ and $0 < q_0 < 1$, encoder $\mathcal{E}$ generates arrays which satisfy the $(d, \infty; 1, \infty; 1, \infty)$-constraint on the hexagonal lattice and it is identical to the bit-stuffing encoder described in Section II-B. The encoder output entropy, parameterized by q_0 and q_1, will provide a lower bound on capacity for both the $(d, \infty; 1, \infty)$-constraint and the $(d, \infty; 1, \infty; 1, \infty)$-constraint.

For each array $x \in S(\Delta_{m,n})$ generated by the bit-stuffing encoder $\mathcal{E}$, the probability measure $\mu_{m,n}(x) = \text{Prob}\{X = x\}$ takes the following form:

$$\begin{aligned} \mu_{m,n}(x) &= \mu_0(\underline{x}_{0,d-1}) \cdot \mu_n^{(h)}(x_{0,d} \ldots x_{0,n-1} | \underline{x}_{0,d-1}) \\ &\cdot \ \mu_m^{(d)}(\underline{x}_{1,d-2}, \underline{x}_{2,d-3}, \cdots \underline{x}_{m-1,d-m} | \underline{x}_{0,d-1}) \\ &\cdot \ \prod_{i=1}^{m-1} \prod_{j=-i+d}^{n-1-i} \vartheta(x_{i,j} | \underline{x}_{i,j-1}, x_{i-1,j}, x_{i-1,j+1}), \end{aligned}$$

where $\underline{x}_{i,j} = [x_{i,j}, x_{i,j-1}, \ldots, x_{i,j-d+1}]$, and the function $\vartheta(x_{i,j} | \underline{x}_{i,j-1}, x_{i-1,j}, x_{i-1,j+1})$ is defined as:

$$\vartheta(0 | \underline{u}, y, v) = \begin{cases} q_0 & \text{if } \underline{u} = \underline{0}_d, y = v = 0 \\ q_1 & \text{if } \underline{u} = \underline{0}_d, y = 0, v = 1 \\ 1 & \text{otherwise} \end{cases},$$

where $\vartheta(1 | \underline{u}, y, v) = 1 - \vartheta(0 | \underline{u}, y, v)$ and $\underline{0}_d$ is a row vector of length d with all zero entries. Due to the horizontal (d, ∞)-constraint, there is at most one 1 in a length-d sequence within the encoded array. Therefore, there are only $d + 1$ possible binary sequences for $\underline{x}_{i,j}$:

$$\begin{aligned} \underline{s}_0 &= \underline{0}_d, \\ \underline{s}_1 &= \underline{0}_{d-1}1, \\ &\vdots \\ \underline{s}_i &= \underline{0}_{d-i}1\underline{0}_{i-1}, \\ &\vdots \\ \underline{s}_d &= 1\underline{0}_{d-1}. \end{aligned} \tag{1}$$

The initialization of the horizontal boundary $\{x_{0,j} : 0 \leq j < n\}$ and the first d diagonals

$$\{\underline{x}_{i,-i+d-1} : 0 \leq i < m\}$$

is similar to that in [6], albeit more complicated. While the horizontal boundary is initialized as a first-order Markov process in [6], it is now initialized as a Markov process of order d, namely

$$\mu_n^{(h)}(w_d \ldots w_{n-1} | w_{d-1} \ldots w_0) = \prod_{j=d}^{n-1} \mu^{(h)}(w_j | w_{j-1} \ldots w_{j-d}),$$

where

$$\mu^{(h)}(0 | u_{d-1} \ldots u_0) = \begin{cases} \alpha & \text{if } u_{d-1} = \ldots = u_0 = 0 \\ 1 & \text{otherwise} \end{cases},$$

and $\mu^{(h)}(1 | u_{d-1} \ldots u_0) = 1 - \mu^{(h)}(0 | u_{d-1} \ldots u_0)$. The initial distribution $\mu_0(\underline{x}_{0,d-1})$ is set to the stationary distribution of the dth-order Markov process $\mu^{(h)}$:

$$\mu_0(\underline{x}_{0,d-1} = \underline{s}_i) = \sigma_i.$$

It is easy to see that $\sigma_i = P\{\underline{0}_d 1 \underline{0}_d\}$ for $1 \leq i \leq d$, where $P\{\underline{0}_d 1 \underline{0}_d\}$ is the stationary probability of the length-$(2d+1)$ sequence $\underline{0}_d 1 \underline{0}_d$ in the encoded array. Therefore,

$$\sigma_0 + \sum_{i=1}^{d} \sigma_i = \sigma_0 + d\sigma_1 = 1.$$

The initialization along the first d diagonals forms a first-order Markov process, if we treat the d elements in each row as a Markov state. There are $(d+1)$ states, as listed in (1). Given $\underline{X}_{i,-i+d-1}$, the encoder initializes the array elements $\underline{X}_{i+1,-i+d-2}$ with the transition probabilities

$$\mu^{(d)}(\underline{x}_{i+1,-i+d-2} | \underline{x}_{i,-i+d-1}) = \beta_{k,j},$$

if $\underline{x}_{i,-i+d-1} = \underline{s}_k$, $\underline{x}_{i+1,-i+d-2} = \underline{s}_j$, and $0 \leq k, j \leq d$. It is obvious that

$$\beta_{k,k+1} = 0, \quad \text{for } 1 \leq k \leq d-1,$$

and

$$\sum_{j=0}^{d} \beta_{k,j} = 1, \quad \text{for } 0 \leq k \leq d.$$

If we initialize the horizontal boundary and the first d diagonals properly, we can show that each row of the encoder output array forms a dth-order Markov process identical to the horizontal boundary.

Proposition 3.1: The elements in each row of the output array X generated by encoder $\mathcal{E}$ on the parallelogram $\Delta_{m,n}$ ($m \geq 2, n \geq d+1$) form a stationary dth-order Markov process whose distribution is identical to that of the horizontal boundary if the initial distribution σ_i and the transition probabilities α and $\beta_{k,j}$ are chosen properly.

More specifically, we can show that to ensure the stationarity of the output array, the transition probability α should be the real root of the following equation:

$$f(\alpha) = (1 - q_0)\alpha^{d+1} + q_1 q_0^{d-1} \alpha - q_1 q_0^{d-1} = 0. \tag{2}$$

By Descartes' rule of signs [11, p. 96] – which states that the number of positive roots of a polynomial with real coefficients is no more than the number of "changes of sign in the list of coefficients – there is at most one positive root of this equation when $0 < q_0 < 1$ and $0 < q_1 \leq 1$. On the other hand, since $f(0) = -q_0^d q_1 < 0$ and $f(1) = 1 - q_0 > 0$, there is at least one root in the region $(0, 1)$. Thus, there is, in fact, a unique root $0 < \alpha < 1$ that satisfies equation (2) above. To ensure the stationarity of the output array, the initial distribution σ_i should also satisfy

$$\sigma_0 = \frac{1}{1 + d(1 - \alpha)}, \quad \sigma_1 = \ldots = \sigma_d = \frac{1 - \alpha}{1 + d(1 - \alpha)},$$

and the transition probabilities $\beta_{k,0}$ should be

$$\begin{aligned}
\beta_{0,0} &= \frac{\alpha}{(1 - \alpha)q_1} c, \\
\beta_{k,0} &= \frac{q_0^{d-k-1}}{\alpha^{d-k}(1 - \alpha)} c, \qquad \text{for } 1 \leq k \leq d - 1, \\
\beta_{d,0} &= \frac{1}{1 - \alpha} c,
\end{aligned}$$

where c is a scaling constant,

$$c = \frac{1}{\frac{\alpha}{(1-\alpha)q_1} + \sum_{k=1}^{d-1} \frac{q_0^{d-k-1}}{\alpha^{d-k}} + 1}.$$

Now we can calculate the entropy $H(\mu_{m,n})$, making use of Proposition 3.1. Specifically, we have

$$\begin{aligned}
H(\mu_{m,n}) = \frac{1}{mn} h(\underline{\sigma}) + \frac{n - d}{mn} \sigma_0 h(\alpha) + \frac{m - 1}{mn} \sum_{i=0}^{d} \sigma_i h(\underline{\beta}_i) \\
+ \frac{(m - 1)(n - d)}{mn} \Big[\beta_{0,0} \sigma_0 (\alpha h(q_0) + \\
(1 - \alpha) h(q_1)) + \sum_{i=1}^{d-1} \beta_{i,0} \sigma_i h(q_0) \Big],
\end{aligned}$$

where

$$\begin{aligned}
h(\underline{\sigma}) &= -\sum_{i=0}^{d} \sigma_i \log_2 \sigma_i, \\
h(\underline{\beta}_i) &= -\sum_{j=0}^{d} \beta_{i,j} \log_2 \beta_{i,j}, \\
h(\alpha) &= -\alpha \log_2(\alpha) - (1 - \alpha) \log_2(1 - \alpha), \\
h(q_i) &= -q_i \log_2(q_i) - (1 - q_i) \log_2(1 - q_i), \quad i = 0, 1.
\end{aligned}$$

Therefore,

$$\begin{aligned}
H(\mu) &= \lim_{m,n \to \infty} H(\mu_{m,n}) \\
&= \beta_{0,0} \sigma_0 (\alpha h(q_0) + (1 - \alpha) h(q_1)) \\
&\quad + \sum_{i=1}^{d-1} \beta_{i,0} \sigma_i h(q_0).
\end{aligned}$$

The maximum output entropy is determined by numerically evaluating the expression for $H(\mu)$ over all possible q_1 and q_0 values. Tables I and II list various lower bounds on the capacity of $(d, \infty; 1, \infty)$-constraints and $(d, \infty; 1, \infty; 1, \infty)$-constraints for small values of d. In Table I, the maximum output entropy of encoder $\mathcal{E}$ for the $(d, \infty; 1, \infty)$-constraint is shown, along with the maximum output entropy of the bit-stuffing encoder, obtained by setting $q_1 = q_0$ and optimizing q_0 over the range $(0, 1)$. For the sake of comparison, the general lower bound derived in Section II-A for the square lattice is also shown. In Table II, we show the maximum output entropy of the bit-stuffing encoder for the $(d, \infty; 1, \infty; 1, \infty)$-constraint, obtained by setting $q_1 = 1$ and optimizing q_0 over the range $(0, 1)$.

TABLE I

MAXIMUM OUTPUT ENTROPY OF BIT-STUFFING BASED ENCODERS FOR $(d, \infty; 1, \infty)$-CONSTRAINTS.

d	Encoder $\mathcal{E}$	Bit-Stuffing Encoder	Lower Bound from Theorem 2.1
2	0.4965	0.4952	0.4847
3	0.4315	0.4309	0.4182
4	0.3831	0.3828	0.3706
5	0.3456	0.3455	0.3344

TABLE II

MAXIMUM OUTPUT ENTROPY OF BIT-STUFFING ENCODERS FOR $(d, \infty; 1, \infty; 1, \infty)$-CONSTRAINTS.

d	Bit-Stuffing Encoder
2	0.4357
3	0.3941
4	0.3580
5	0.3277

For the $(d, \infty; 1, \infty)$-constraint, the difference between the maximum output entropy of encoder $\mathcal{E}$ and that of the bit-stuffing encoder decreases as d increases. Similarly, the gap between the encoder output entropy for the $(d, \infty; 1, \infty)$-constraint and that for the $(d, \infty; 1, \infty; 1, \infty)$-constraint gets smaller as d increases. Comparing these bounds with the capacity of the one-dimensional (d, ∞)-constraint – an obvious upper bound on the capacities of the corresponding two-dimensional constraints – we see that these differences also shrink as d increases. These results are consistent with the intuitive reasoning that, as d increases, the horizontal (d, ∞)-constraint has the dominant effect on the capacity of the two-dimensional constraint, with the impact of the $(1, \infty)$-constraint in the other coordinate directions gradually becoming negligible. Thus, for large d, an efficient encoder should focus on generating horizontal (d, ∞)-constrained sequences maxentropically, while ensuring that the output arrays satisfy the $(1, \infty)$-constraint in the other directions. Noting that the specific encoders considered here are based upon bit-stuffing techniques, and recalling that bit-stuffing encoders achieve the capacity of one-dimensional (d, ∞)-constraints [12], it might be expected that, as d increases, the two-dimensional encoders considered here have coding ratios that converge to the capacity of the one-dimensional (d, ∞)-constraint.

ACKNOWLEDGMENT

This work was supported in part by the Center for Magnetic Recording Research at UC San Diego and by the National Science Foundation under Grant CCR-0219582.

REFERENCES

[1] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2260–2299, October 1998.

[2] D. Psaltis and F. Mok, "Holographic memories," *Scientific American*, pp. 70–76, November 1995.

[3] Z. Kukorelly, K. Zeger, "The capacity of some hexagonal (d, k)-constraints," in *Proc. of IEEE Int'l Symp. Inform. Theory*, Washington, DC, pp. 64, June 2001.

[4] N. Calkin and H. S. Wilf, "The number of independent sets in a grid graph," *SIAM J. Discr. Math.*, vol. 11, pp. 54–60, 1997.

[5] P. H. Siegel and J. K. Wolf, "Bit-stuffing bounds on the capacity of 2-dimensional constrained arrays," in *Proc. of IEEE Int'l Symp. Inform. Theory*, Cambridge, MA, pp. 323, August 1998.

[6] R. M. Roth, P. H. Siegel, J. K. Wolf, "Efficient coding schemes for the hard-square model," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp.1166–1176, March 2001.

[7] R. J. Baxter, "Hard hexagons: exact solution," *J. Phy. A*, vol. 13, pp. 61–70, 1980.

[8] 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.

[9] A. Kato and K. Zeger, "Partial characterization of the positive capacity region of two-dimensional asymmetric run length constrained channels," *IEEE Trans. Inform. Theory*, vol. 46, no. 7, pp. 2666–2670, November 2000.

[10] S. Halevy, J. Chen, R. M. Roth, P. H. Siegel, and J. K. Wolf, "Improved bit-stuffing bounds on two-dimensional constraints," *IEEE Trans. Inform. Theory*, vol. 50, no. 5, pp. 824–838, May 2004.

[11] A. S. Householder, *Principles of Numerical Analysis.* New York: McGraw-Hill, 1953.

[12] P. E. Bender and J. K. Wolf, "A universal algorithm for generating optimal and nearly optimal run-length-limited charge-constrained binary sequences," in *Proc. of IEEE Int'l Symp. Inform. Theory*, San Antonio, TX, pp. 6, January 1993.