Bit-Stuffing Bounds on the Capacity of 2-Dimensional Constrained Arrays[1]

Paul H. Siegel
ECE, 0407
University of California, San Diego
La Jolla, CA 92093-0407 USA
Email: psiegel@ucsd.edu

Jack Keil Wolf
CMRR, 0401
University of California, San Diego
La Jolla, CA 92093-0401 USA
Email: jwolf@ucsd.edu

Abstract — **Bit-stuffing constructions of binary 2-dimensional constrained arrays satisfying (d,∞) or $(0,k)$ runlength constraints in both horizontal and vertical dimensions are described. Lower bounds on the capacity of these constrained arrays are derived.**

I. Introduction

With the advent of page-oriented storage technologies, such as holographic storage, interest in constrained arrays in two or more dimensions has arisen. We consider binary arrays satisfying 2-dimensional (d,k) constraints, where the parameters d and k represent, respectively, the minimum and maximum admissible number of 0's separating consecutive 1's in any row and any column. The capacity $C_2(d,k)$ of a 2-dimensional (d,k) constraint measures the growth rate of the number $N(m,n)$ of $m \times n$ (d,k) arrays. The capacity $C_1(d,k)$ of 1-dimensional (d,k) constrained sequences is exactly known. The only nontrivial result in two dimensions is that, for $d \geq 1$, $C_2(d,k) = 0$ if and only if $k = d+1$ [3]. General upper and lower bounds on $C_2(d,k)$ are also derived in [3]. Here we present different lower bounds on $C_2(d,\infty)$ and $C_2(0,k)$ by analyzing a bit-stuffing encoder.

II. Bit Stuffing Bounds on Capacity

We first describe a mapping of 1-dimensional $(2d,\infty)$ constrained sequences to 2-dimensional (d,∞) constrained arrays. We represent the 2-dimensional constrained array as the lower right quadrant of a rectangular grid. Starting at the origin, we write the sequence digits into open positions along successive 45 degree diagonals, from upper right to lower left. However, the $2d$ 0's following each 1 are written into the d positions immediately to the right and immediately below the position of the 1. Any one of these $2d$ 0's that would overwrite a position already containing a 0 is discarded. This mapping is clearly invertible. It follows that $C_2(d,\infty) \geq C_1(2d,\infty)$. Table 1 shows the lower bounds, for $1 \leq d \leq 5$. (It is easy to extend this construction to include runlength constraints along diagonals, as well.)

Another approach to deriving a lower bound on $C_2(d,\infty)$ is by analyzing a bit-stuffing encoder that operates as follows. A binary data sequence is first converted 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 $H_2(p)$ is the binary entropy function.

We successively write these digits into open positions along diagonals as before, subject to the rule that whenever a 1 is written, additional 0's are inserted (or "stuffed") into any of the d positions immediately to the right of it or immediately below it that are empty. To decode the array, one reads down diagonals, sensing and discarding the 0's that have been stuffed.

Table 1: Bit-stuffing lower bounds $C_2(d,\infty)$ and $C_2(0,k)$

d	$C_1(2d,\infty)$	k	$r_k(p')$
1	0.5515	1	0.5515
2	0.4057	2	0.7769
3	0.3282	3	0.8788
4	0.2788	4	0.9320
5	0.2440	5	0.9616

If we assume that the fraction of stuffed digits on diagonals approaches a constant value as the length of the diagonal grows, a simple analysis leads to the lower bound $C_2(d,\infty) \geq R_d(p) = H_2(p)/(1+2dp)$. The value p_* that maximizes this lower bound is the largest real solution of the equation $p = (1-p)^{2d+1}$. The corresponding lower bound $R_d(p_*)$ turns out to be precisely $C_1(2d,\infty)$.

For the $(1,\infty)$ constraint, the lower bound can be improved by a more careful accounting. Specifically, the exact rate R_{max} of the bit-stuffing encoder can be computed, and satisfies $R_{max} \approx 0.5831$. The best known bounds [2] are $0.5879 \leq C_2(1,\infty) \leq 0.5883$. Thus, bit-stuffing achieves a rate that is only 1% below capacity. We remark that the optimality of bit-stuffing for certain 1-dimensional (d,k) constraints was shown in [1].

One can generate $(0,k)$-constrained arrays by stuffing 1's to enforce the k constraint. A similar analysis leads to the lower bound $C_2(0,k) \geq r_k(p) = H_2(p)\gamma(p)$, where $\gamma(p)$ is the largest real solution of $2\gamma^k(1-p)^k + \gamma - 1 = 0$. The value p' that maximizes the lower bound was found numerically. The results are shown in Table 1. Simulation results are consistent with these lower bounds.

Acknowledgements

The authors thank Ron Roth and Alon Orlitsky for helpful discussions, and William Hung for conducting the simulations.

References

[1] 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, TX, Jan. 17–22, 1993, p. 6.

[2] N.J. Calkin and H.S. Wilf, "The number of independent sets in a grid graph," *SIAM J. Discr. Math.*, to appear.

[3] A. Kato and K. Zeger, "On the capacity of two-dimensional run-length-limited codes," *Proc. 1998 IEEE Int. Symp. Inform. Theory*, Cambridge, MA, Aug. 16–21, 1998.

[1] This work was supported in part by NSF Grants NCR-9612802 and NCR-9405008.