Approximate Enumerative Coding for 2-D Constraints through Ratios of Matrix Products

Erik Ordentlich
Hewlett–Packard Laboratories
Palo Alto, CA 94304
erik.ordentlich@hp.com

Ron M. Roth*
Computer Science Department
Technion, Haifa, Israel
ronny@cs.technion.ac.il

***Abstract*—We show how to improve on the technique of approximate enumerative coding for two-dimensional constraints by encoding according to lower bounds based on the worst-case behavior of certain ratios of matrix products. For the case of the two-dimensional $(d{=}2,\infty)$ run-length limited (RLL) constraint, the improved approach yields a lower bound of 0.4453 on the capacity of the constraint.**

I. Introduction

Enumerative coding [1] is a well-known technique for mapping arbitrary binary sequences (messages) of fixed length into a sequence of symbols that are required to satisfy certain constraints, such as there being at least one 0 between any two 1's or the same number of 1's as 0's. Enumerative coding requires the computation or storage of the number of constraint-satisfying sequences starting with any prefix. This is not a problem for most one-dimensional (1-D) constraints and sequences of reasonable length. However, we are interested in two-dimensional (2-D) constraints for which conventional enumerative coding is not practical.

A 2-D constrained code maps arbitrary fixed-length binary messages into 2-D arrays satisfying a 2-D constraint. Examples of such constraints are requiring that every row and column obey a 1-D constraint (such as the above examples), or requiring that no 1 be surrounded by all 0's and no 0 surrounded by all 1's. Codes for certain 2-D constraints may find applications in denser conventional magnetic and optical data storage technologies, holographic data storage, nano-crossbar storage technologies, and 2-D bar codes.

The application of enumerative coding to 2-D constraints would require computing the number of constraint-satisfying arrays starting with any partially filled-in array, which, for all 2-D constraints of conceivable practical interest, is feasible only for arrays of very small extent along at least one dimension (narrow stripes). In [6] (for 2-D balanced arrays) and [7] (for 2-D finite-state row–column constraints), we introduced an alternative *approximate* enumerative coding technique for 2-D constraints (approximate enumerative coding was applied in a different way to 1-D coding in [4]). In this approach, only *lower bounds* on the number of constraint-satisfying arrays with any fixed initial portion are needed. The idea is to use lower bounds that are easily computed (unlike the exact cardinalities). For the encoding and decoding to work correctly, the lower bounds are required to satisfy a consistency condition, and the coding rate achieved (measured as the number of arbitrary bits encoded divided by the size of the array into which they are encoded) depends on the tightness of the lower bounds. It was noted in [7] that for certain finite-state row–column constraints (in which every row and column must satisfy a 1-D finite-state constraint) tighter lower bounds can be obtained by grouping array symbols into sub-blocks, treating these sub-blocks as super symbols which must satisfy a "new" constraint induced by the original constraint, and applying the approach of [6] to this "new" constraint. The resulting codes were found to achieve higher rates than previously proposed fixed-rate codings for certain constraints.

We shall confine our discussion to row–column constraints, with the understanding that the results in [7] and herein are more generally applicable. The formulation of such 2-D constraints is presented in Section II. In Section III, we review our result in [7] in some more detail. Then, in Sections IV–V, we present an improved lower bound, as compared to [7], on the number of partially filled constrained arrays, and use it to obtain new lower bounds on capacity. The improvement is obtained through bounds on the worst-case behavior of a ratio of matrix products, where the sequences of matrices comprising the products must satisfy certain constraints induced by the 2-D constraint at hand. Finally, in Section VI, we demonstrate how our improved method can be used to tune the parameters of a fixed-rate approximate enumerative encoder thereby achieving coding rates which are higher than those obtained via the technique in [7].

II. Constraints and notation

Consider an ordered, finite alphabet Σ and two 1-D constraints with memory 1 on sequences over this alphabet determined by the $|\Sigma|\times|\Sigma|$ transition matrices H and V. The rows and columns of H and V are indexed by Σ, and $H(x,y)$ (respectively, $V(x,y)$) denotes the entry in H (respectively, V) that is indexed by $x,y\in\Sigma$. In the sequel, the notation H and V is used to also mean the constraints themselves (not just the matrices). Let $\mathcal{H}_n$ and $\mathcal{V}_n$, respectively, denote the set of sequences of length n satisfying the constraints represented by

* Work done while visiting Hewlett–Packard Laboratories, Palo Alto, CA.

H and V. Thus a sequence $x_1 x_2 \ldots x_n$ is in $\mathcal{H}_n$ if and only if $H(x_i, x_{i+1}) = 1$ for all $i = 1, 2, \ldots, n-1$.

We are interested in the generic 2-D constraint on arrays over Σ in which rows and columns are constrained by H and V, respectively. For the sake of simplicity of the exposition, we will further assume that there is a "wild card" symbol $0 \in \Sigma$ that can precede and follow any symbol in Σ, both horizontally and vertically, i.e., $H(0,x) = H(x,0) = V(0,x) = V(x,0) = 1$ for every $x \in \Sigma$. Let $\mathcal{A}_{m,n}$ be the set of $m \times n$ arrays whose rows and columns, respectively, belong to $\mathcal{H}_n$ and $\mathcal{V}_m$. The capacity $C_{H,V}$ of this constraint is given by

$$C_{H,V} = \lim_{m,n \to \infty} \frac{1}{mn} \log |\mathcal{A}_{m,n}|,$$

where hereafter all logarithms are to the base 2.

The following notation is used in the sequel. For any array x, let $x(i,j)$ denote the entry in the ith row and jth column and let $x(i_1{:}i_2, j_1{:}j_2)$ denote the sub-array $(x(i,j))_{i=i_1}^{i_2}\,{}_{j=j_1}^{j_2}$. The notation $x(i,:)$ and $x(:,j)$ will stand for the ith row and jth column of x, respectively. A sequence of ℓ symbols $x_1 x_2 \ldots x_\ell$ may also be denoted as x^ℓ. For a positive integer ℓ and a symbol $y \in \Sigma$, we define the subset $\mathcal{H}_\ell(y) \subseteq \mathcal{H}_\ell$ by

$$\mathcal{H}_\ell(y) = \{z^\ell \in \mathcal{H}_\ell : y z^\ell \in \mathcal{H}_{\ell+1}\}.$$

The subset $\mathcal{V}_\ell(y) \subseteq \mathcal{V}_\ell$ is defined in a similar manner.

Finally, for any $y^m \in \mathcal{V}_m$, define

$$\mathcal{A}_{m,\ell}(y^m) = \{x \in \mathcal{A}_{m,\ell} : y^m x \in \mathcal{A}_{m,\ell+1}\}.$$

III. Previous results

In this section, we recall our approach from [7] (which, in turn, follows closely our even earlier approach from [6]) to approximate enumerative coding for constraints of the type presented in Section II.

For each m and ℓ, let $q_{m,\ell}$ be non-negative real numbers that satisfy

$$q_{m,\ell} \le \min_{y^m \in \mathcal{V}_m} \frac{\sum_{x^m \in \mathcal{A}_{m,1}(y^m)} \prod_{j=1}^m |\mathcal{H}_{\ell-1}(x_j)|}{\prod_{j=1}^m |\mathcal{H}_\ell(y_j)|}. \qquad (1)$$

An argument similar to that used in the proof of Theorem 1 in [7] shows that for any $y^m \in \mathcal{V}_m$,

$$|\mathcal{A}_{m,\ell}(y^m)| \ge L_{m,\ell}(y^m), \qquad (2)$$

where

$$L_{m,\ell}(y^m) \triangleq \Big(\prod_{j=1}^m |\mathcal{H}_\ell(y_j)|\Big) \cdot \Big(\prod_{i=1}^\ell q_{m,i}\Big); \qquad (3)$$

namely, $L_{m,\ell}(y^m)$ bounds from below the number of $m \times \ell$ constraint-satisfying extensions of any partial array. The lower bound (2), when computed for values $q_{m,\ell}$ that satisfy (1) with equality, was used in [7] to compute a lower bound on $C_{H,V}$.

Now, it follows from (1) that for any y^m,

$$\Big(\prod_{j=1}^m |\mathcal{H}_\ell(y_j)|\Big) \cdot \Big(\prod_{i=1}^\ell q_{m,i}\Big) \le \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} \Big(\prod_{j=1}^m |\mathcal{H}_{\ell-1}(x_j)|\Big) \cdot \Big(\prod_{i=1}^{\ell-1} q_{m,i}\Big),$$

which is equivalent to

$$L_{m,\ell}(y^m) \le \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} L_{m,\ell-1}(x^m). \qquad (4)$$

This consistency condition makes it possible to use the lower bound (2) not just for computing lower bounds on capacity, but also for approximate enumerative coding.

The full encoding and decoding algorithms are detailed in Figures 1 and 2, where it is assumed for simplicity that $x(0,:)$ and $x(:,0)$ are set to the wild-card symbol 0 (the encoding–decoding can be easily modified when this is not the case). For a particular i and j, the function $L(w)$ in the decoder coincides with (5) in the encoder. The correctness of the encoding hinges on the consistency condition (4) and is established in a manner similar to that in [6].

Input: Integer-valued message $M \in \ 0, \lceil L_{m,n}(0^m) \rceil$.
Output: Array $x \in \mathcal{A}_{m,n}$.
$M' \leftarrow M$
for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** m **do**
    for each $w \in \Sigma$, **let**

$$L(w) \leftarrow \sum_{\substack{y^m \in \mathcal{A}_{m,1}(x(:,j-1)): \\ y^{i-1} = x(1:i-1,j:j),\, y_i = w}} \lceil L_{m,n-j}(y^m) \rceil \qquad (5)$$

    end
    $x(i,j) \leftarrow \max\{y' : \sum_{w<y'} L(w) \le M' < \sum_{w \le y'} L(w)\}$
    $M' \leftarrow M' - \sum_{w < x(i,j)} L(w)$
  end
end

Fig. 1. Encoder.

Input: Array $x \in \mathcal{A}_{m,n}$.
Output: Integer-valued message M.
$M \leftarrow 0$
for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** m **do**
    $M \leftarrow M + \sum_{w < x_{i,j}} L(w)$
  end
end

Fig. 2. Decoder.

The summation over an exponential number of terms appearing in the expression (5) for $L(w)$ can be computed efficiently (in linear time) using matrix multiplication; see [7, §IV] for details.

As mentioned in Section I, higher coding rates can be achieved by grouping symbols into sub-blocks and considering the 2-D constraint induced on arrays comprised of these sub-blocks. The improvement in coding rate obtained this way comes at the cost of increased computational complexity due to the larger super-alphabet size.

IV. New lower bounds

Our new approach for bounding $C_{H,V}$ will be based on the inequality (2), but with a different, tighter choice for the values (3). Using the tighter values, we will also improve on the coding rate achieved by the encoding and decoding steps of Figures 1 and 2. The method presented herein has a memory parameter k that controls the trade-off between the values of the computed lower bounds and the complexity of computing those bounds.

For any $y^m \in \mathcal{V}_m$ and $m \times k$ array $x^{m\times k}$, define

$$p^{(\ell)}_{X^{m\times k}|Y^m}(x^{m\times k}|y^m) = \prod_{j=1}^{m} \frac{|\{\tilde{x}^\ell \in \mathcal{H}_\ell(y_j) : \tilde{x}^k = x(j,:)\}|}{|\mathcal{H}_\ell(y_j)|}. \tag{6}$$

We will regard $p^{(\ell)}_{X^{m\times k}|Y^m}(\cdot|\cdot)$ as a probability measure in the following manner. Assume a uniform distribution over all $m \times \ell$ extensions of y^m (into an $m \times (\ell{+}1)$ array) with rows satisfying the constraint H; then $p^{(\ell)}_{X^{m\times k}|Y^m}(x^{m\times k}|y^m)$ equals the probability that the first (leftmost) k columns in such an extension equal $x^{m\times k}$.

In the limit when $\ell \to \infty$, we get that

$$\begin{aligned} p_{X^{m\times k}|Y^m}(x^{m\times k}|y^m) &\triangleq p^{(\infty)}_{X^{m\times k}|Y^m}(x^{m\times k}|y^m) \\ &= \prod_{j=1}^{m} p_{X^{1\times k}|Y}(x(j,:)|y_j), \end{aligned}$$

where $p_{X^{1\times k}|Y}(w|y)$ is the probability, according to a max-entropic Markov chain on H, that we see $w \in \mathcal{H}_k$ following $y \in \Sigma$, conditioned on y [5, §3.2.3].

Fix a memory parameter $k > 0$, and for each m, ℓ, and $y^m \in \mathcal{V}_m$, define $Q^{[k]}_{m,\ell}$ to be the probability that the first k columns of an extension of a column y^m by an $m \times \ell$ array forms with y^m an $m \times (\ell{+}1)$ constrained array; that is,

$$Q^{[k]}_{m,\ell}(y^m) = \sum_{x^{m\times k} \in \mathcal{A}_{m,k}(y^m)} p^{(\ell)}_{X^{m\times k}|Y^m}(x^{m\times k}|y^m).$$

Let $V^{[k]}$ denote the $|\mathcal{H}_k|\times|\mathcal{H}_k|$ adjacency matrix of the 1-D constraint whose elements are all the k-width vertical stripes (seen over the alphabet $\mathcal{H}_k$) that satisfy the 2-D constraint. For each y in Σ, define $D^{[k]}_\ell(y)$ to be the $|\mathcal{H}_k| \times |\mathcal{H}_k|$ diagonal matrix whose rows and columns are indexed by the elements of $\mathcal{H}_k$, and the xth diagonal entry equals $p^{(\ell)}_{X^{1\times k}|Y}(x^{1\times k}|y)$ (as obtained by substituting $m = 1$ in (6)). Also, let $\mathbf{e}$ denote the real row vector which is 1 in the component corresponding to the all-0 (i.e., all-wild-card) array element of $\mathcal{H}_k$ and zero in all other components. We can express $Q^{[k]}_{m,\ell}(y^m)$ as

$$Q^{[k]}_{m,\ell}(y^m) = \mathbf{e}\Big(\prod_{j=1}^{m}(V^{[k]}D^{[k]}_\ell(y_j))\Big)\mathbf{1}, \tag{7}$$

where $\mathbf{1}$ denotes the real column vector of all 1's. In the limit when $\ell \to \infty$, Equation (7) becomes

$$\lim_{\ell\to\infty} Q^{[k]}_{m,\ell}(y^m) = \mathbf{e}\Big(\prod_{j=1}^{m}(V^{[k]}D^{[k]}(y_j))\Big)\mathbf{1}, \tag{8}$$

where $D^{[k]}$ is a diagonal matrix whose rows and columns are indexed by $\mathcal{H}_k$, and the xth diagonal entry equals $p_{X^{1\times k}|Y}(x|y)$.

When $Q^{[k-1]}_{m,\ell}(y^m) \neq 0$, the ratio $Q^{[k]}_{m,\ell}(y^m)/Q^{[k-1]}_{m,\ell}(y^m)$ equals the probability that the first k columns of an extension of a column y^m by an $m\times\ell$ array forms with y^m a constrained array, conditioned on the first $k{-}1$ columns of the extension already forming with y^m an $m \times k$ constrained array.

In what follows, we fix for every m and ℓ a non-negative constant $q^{[k]}_{m,\ell}$ that satisfies

$$q^{[k]}_{m,\ell} \le \min_{y^m\in\mathcal{V}_m} \frac{Q^{[k]}_{m,\ell}(y^m)}{Q^{[k-1]}_{m,\ell}(y^m)} \tag{9}$$

(for $k = 1$, we take the denominator $Q^{[0]}_{m,\ell}(y^m)$ to be 1, and throughout we adopt the convention that $0/0$ is ∞).

We have the following lemma.

Lemma 4.1: Given k, m, and ℓ, for every $y^m \in \mathcal{V}_m$,

$$|\mathcal{A}_{m,\ell}(y^m)| \ge L^{[k]}_{m,\ell}(y^m), \tag{10}$$

where

$$L^{[k]}_{m,\ell}(y^m) \triangleq \Big(\prod_{j=1}^{m}|\mathcal{H}_\ell(y_j)|\Big)Q^{[k-1]}_{m,\ell}(y^m)\Big(\prod_{i=k}^{\ell} q^{[k]}_{m,i}\Big) \tag{11}$$

(the rightmost product is defined to be 1 if $\ell < k$).

Proof: The proof is similar to an argument used in the proof of Theorem 1 of [7]. □

Note that for $k = 1$, Equations (9)–(11) coincide with (1)–(3). For $k > 1$, we will get an improvement over [7] in that the right-hand side of (11) is partially comprised of an exact conditional probability, namely, the factor $Q^{[k-1]}_{m,\ell}(y^m)$, in addition to the lower bounds $q^{[k]}_{m,\ell}$. These latter lower bounds also differ from their counterparts $q_{m,\ell}$ in [7], in terms of what they bound from below. In particular, the right-hand side of (9) can be shown to be never smaller than the right-hand side of (1), its counterpart in [7], thus allowing for the potential to improve in the tightness of the values $L^{[k]}_{m,\ell}(y^m)$ as lower bounds. In fact, for $\ell < k$, these lower bounds are tight (equality holds in (10) for $\ell < k$).

The next theorem generalizes Theorem 2.1 in [7] and applies to any 2-D row–column constraint with a wild-card symbol 0.

Theorem 4.2: Let C_H denote the capacity of the 1-D constraint defined by H. The capacity $C_{H,V}$ of the 2-D constraint with rows and columns constrained, respectively, by H and V satisfies

$$C_{H,V} \ge C_H + \nu^{[k]}_{H,V},$$

where $\nu^{[k]}_{H,V}$ stands for the expression

$$\limsup_{m\to\infty} \frac{1}{m} \min_{y^m\in\mathcal{V}_m} \left\{ \log \frac{\mathbf{e}\big(\prod_{j=1}^{m}(V^{[k]}D^{[k]}(y_j))\big)\mathbf{1}}{\mathbf{e}\big(\prod_{j=1}^{m}(V^{[k-1]}D^{[k-1]}(y_j))\big)\mathbf{1}} \right\}. \tag{12}$$

Sketch of proof: We assume equality in (9), then apply Lemma 4.1, and finally we use (8) when taking ℓ to infinity. □

The next corollary (which follows from Perron–Frobenius Theorem [5, §3.1]) applies to the special case where (the) minimizing sequences y^m in (12) are all periodic extensions of the same sequence. For a square non-negative real matrix U, let $\lambda(U)$ be the largest real eigenvalue of U.

Corollary 4.3: Under the conditions of Theorem 4.2, suppose that there exists a sequence $y^* = y_1^* y_2^* \dots y_J^*$ such that for all sufficiently large m, the minimum in (12) is attained by a periodic sequence of period J obtained by repeating the sequence y^*. Then,

$$\nu_{H,V}^{[k]} = \frac{1}{J} \log \frac{\lambda(\prod_{j=1}^{J} V^{[k]} D^{[k]}(y_j^*))}{\lambda(\prod_{j=1}^{J} V^{[k-1]} D^{[k-1]}(y_j^*))}. \tag{13}$$

V. Verifying the periodicity condition

Based on Corollary 4.3, we can now proceed by minimizing the expression in the right-hand side of (13) over a range of (small) J and (short) sequences $y_1 y_2 \dots y_J$ satisfying $y_1 \in \mathcal{V}_1(y_J)$, and, for each minimizing sequence, checking whether the periodicity assumption in Corollary 4.3 holds (up to some controlled discrepancy). We describe this approach in more detail below.

Let $y_1^* y_2^* \dots y_J^*$ be a sequence that minimizes the right-hand side of (13), for some J. Next, based on $y_1^* y_2^* \dots y_J^*$, for any penalty (discrepancy) factor $\alpha \in (0,1]$ and for each $i \in \{1,2,\dots,J\}$, we shall define a partial order "$\preceq_{\alpha,i}$" on sequences of equal length in V. To this end, we introduce the following additional notation. For $s \in \{k-1,k\}$ define $U^{[s]}(\cdot) \stackrel{\triangle}{=} V^{[s]} D^{[s]}(\cdot)$, and, for each $i \in \{1,2,\dots,J\}$, let $\mathbf{p}_i^{[s]}$ be a non-negative left eigenvector associated with the largest real eigenvalue of the matrix $\prod_{j=i}^{J} U^{[s]}(y_j^*) \prod_{j=1}^{i-1} U^{[s]}(y_j^*)$. For s and i as above and a sequence $y^m \in \mathcal{V}_m$, define $\mathbf{v}_{s,i}(y^m) = \mathbf{p}_i^{[s]} \Big(\prod_{j=1}^{m} U^{[s]}(y_j) V^{[s]} \Big)$. And for real row vectors $\mathbf{a} = (a_j)_j$ and $\mathbf{b} = (b_j)_j$, define the tensor division:

$$\mathbf{a} \oslash \mathbf{b} = ((a_1/b_1)\ (a_1/b_2)\ \dots\ (a_2/b_1)\ (a_2/b_2)\ \dots).$$

Next, for $\alpha \in (0,1]$, $i \in \{1,2,\dots,J\}$, and $y^m \in \mathcal{V}_m$, define

$$\mathbf{r}_{\alpha,i}(y^m) = \begin{cases} \alpha \mathbf{v}_{k,i}(y^m) \oslash \mathbf{v}_{k-1,i}(y^m) & \text{if } y_1 = y_i^* \\ \mathbf{v}_{k,i}(y^m) \oslash \mathbf{v}_{k-1,i}(y^m) & \text{if } y_1 \neq y_i^* \end{cases}.$$

Finally, for any two sequences $x = x_1 x_2 \dots x_m$ and $y = y_1 y_2 \dots y_m$ in $\mathcal{V}_m$, define the partial order

$$x \preceq_{\alpha,i} y \iff \big(\mathcal{V}_1(y_m) \subseteq \mathcal{V}_1(x_m) \text{ and } \mathbf{r}_{\alpha,i}(x) \leq \mathbf{r}_{\alpha,i}(y)\big), \tag{14}$$

where the vector inequality is component-wise.

The next step is to find a (largest) penalty factor $\alpha \leq 1$ for which the pruning algorithm of Figure 3 converges in reasonable time for each $i = 1,2,\dots,J$ and the partial order (14). As can be seen from Figure 3, reducing α has the effect of favoring sequences in $\mathcal{W}_i$ that start with y_i^*, thereby speeding the running time of the algorithm. On the other hand, as the following lemma shows, reducing α also carries a penalty in the lower bound that we get on $\nu_{H,V}^{[k]}$.

$\mathcal{W}_i \leftarrow \mathcal{V}_1(y_{i-1}^*)$
while $\exists w \in \mathcal{W}_i : \ w_1 \neq y_i^*$, **do**
  $\mathcal{W}_i \leftarrow \{$all minimal elements in $\mathcal{W}_i$ with respect to "$\preceq_{\alpha,i}$"$\}$
  $\mathcal{W}_i \leftarrow \{wx : w \in \mathcal{W}_i, x \in \mathcal{V}_1(w)\}$
end

Fig. 3. Pruning algorithm. The partial ordering "$\preceq_{\alpha,i}$" is defined in (14).

Lemma 5.1: If the algorithm in Figure 3 terminates for α, $y_1^* y_2^* \dots, y_J^*$, and all $1 \leq i \leq J$, then

$$\nu_{H,V}^{[k]} \geq \log \alpha + \frac{1}{J} \log \frac{\lambda(\prod_{j=1}^{J} V^{[k]} D^{[k]}(y_j^*))}{\lambda(\prod_{j=1}^{J} V^{[k-1]} D^{[k-1]}(y_j^*))}. \tag{15}$$

The proof of Lemma 5.1 relies on the following two propositions, the proofs of which we omit.

Proposition 5.2: There is a constant $c > 0$ (independent of m) such that for all m,

$$\min_{y^m \in \mathcal{V}_m} \frac{\mathbf{e} \prod_{j=1}^{m} V^{[k]} D^{[k]}(y_j) \mathbf{1}}{\mathbf{e} \prod_{j=1}^{m} V^{[k-1]} D^{[k-1]}(y_j) \mathbf{1}} \geq c \cdot \min_{\substack{y^m \in \mathcal{V}_m: \\ y_1 = y_1^*}} \frac{\mathbf{p}_1^{[k]} \prod_{j=1}^{m} V^{[k]} D^{[k]}(y_j) \mathbf{1}}{\mathbf{p}_1^{[k-1]} \prod_{j=1}^{m} V^{[k-1]} D^{[k-1]}(y_j) \mathbf{1}}.$$

Proposition 5.3: Given non-negative row vectors $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^n$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^q$, the following two conditions are equivalent:

(i) $\mathbf{a}_1 \oslash \mathbf{b}_1 \leq \mathbf{a}_2 \oslash \mathbf{b}_2$ (where the vector inequality holds component-wise).

(ii) For *any* pair of non-negative column vectors $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^q$,
$$\frac{\mathbf{a}_1 \cdot \mathbf{c}}{\mathbf{b}_1 \cdot \mathbf{d}} \leq \frac{\mathbf{a}_2 \cdot \mathbf{c}}{\mathbf{b}_2 \cdot \mathbf{d}}.$$

Sketch of proof of Lemma 5.1: Proposition 5.2 implies that $\nu_{H,V}^{[k]}$ is bounded from below by

$$\limsup_{m\to\infty} \frac{1}{m} \min_{\substack{y_2 y_3 \dots y_m \in \\ \mathcal{V}_{m-1}(y_1^*)}} \left\{ \log \frac{\mathbf{p}_1^{[k]} U^{[k]}(y_1^*) \big(\prod_{j=2}^{m} U^{[k]}(y_j)\big) \mathbf{1}}{\mathbf{p}_1^{[k-1]} U^{[k-1]}(y_1^*) \big(\prod_{j=2}^{m} U^{[k-1]}(y_j)\big) \mathbf{1}} \right\}. \tag{16}$$

Note that $\mathbf{p}_1^{[s]} U^{[s]}(y_1^*) = c_s \cdot \mathbf{p}_2^{[s]}$ for $s \in \{k-1,k\}$ and constants c_s. Thus, by invoking Proposition 5.3, it is straightforward to show (though we omit the details) that if the algorithm depicted in Figure 3 terminates for $i = 2$ then, for sufficiently large m, the minimum in (16) is achieved to within a factor of $1/\alpha$ by $y_2 = y_2^*$; more formally:

$$\min_{\substack{y_2 y_3 \dots y_m \in \\ \mathcal{V}_{m-1}(y_1^*)}} \frac{\mathbf{p}_1^{[k]} U^{[k]}(y_1^*) \big(\prod_{j=2}^{m} U^{[k]}(y_j)\big) \mathbf{1}}{\mathbf{p}_1^{[k-1]} U^{[k-1]}(y_1^*) \big(\prod_{j=2}^{m} U^{[k-1]}(y_j)\big) \mathbf{1}} \tag{17}$$

$$\geq \alpha \cdot \min_{\substack{y_3 y_4 \dots y_m \in \\ \mathcal{V}_{m-2}(y_2^*)}} \frac{\big(\mathbf{p}_1^{[k]} \prod_{j=1}^{2} U^{[k]}(y_j^*)\big) \big(\prod_{j=3}^{m} U^{[k]}(y_j)\big) \mathbf{1}}{\big(\mathbf{p}_1^{[k-1]} \prod_{j=1}^{2} U^{[k-1]}(y_j^*)\big) \big(\prod_{j=3}^{m} U^{[k-1]}(y_j)\big) \mathbf{1}}.$$

Again, noting that $\mathbf{p}_1^{[s]} \prod_{j=1}^{2} U^{[s]}(y_j^*) = c'_s \mathbf{p}_3^{[s]}$, invoking the termination of Figure 3 for $i = 3$, along with Proposition 5.3 for m sufficiently large, this process can be repeated to fix $y_3 = y_3^*$ and incur another factor of α. In fact, this process can be continued until all but a bounded suffix of the minimizing sequence y^m in (17) has been replaced by $y_1^* y_2^* \ldots y_J^*$, repeated, with each additional replaced entry incurring an additional factor of α. The boundedness of this suffix implies the lemma. $\square$

We have applied the above framework to the 2-D $(2,\infty)$-RLL constraint (where every row and column must have at least two 0's between any two 1's), after grouping bits into non-overlapping 2×1 super-symbols. For this grouping and a memory parameter value of $k = 6$, a choice of α and $y_1^* y_2^* \ldots y_J^*$ satisfying the assumptions of Lemma 5.1 has been found. (While the memory parameter k is measured with respect to 2×1 super-symbols, the minimizing sequence y^m in (12)—and hence also the sequence $y_1^* y_2^* \ldots y_J^*$ which was found—are actually over an alphabet of 2×2 super-symbols, so as to match the memory of the horizontal constraint.) The corresponding lower bound on $\nu_{H,V}^{[k]}$, when combined with Theorem 4.2, yields a new lower bound of $.4453$ on the capacity of the 2-D $(2,\infty)$-RLL constraint. For comparison, the best known *upper* bound on the capacity of this constraint is $.4459$ (this bound is obtained using the technique of [2], as reported in [3]).

VI. Encoder and decoder

An improved encoding–decoding algorithm can be obtained by using the values $L_{m,\ell}^{[k]}(y^m)$ in (11) in place of $L_{m,\ell}(y^m)$ to carry out the approximate enumerative encoding and decoding of Figures 1–2; specifically, we use the new values in the range of messages $M \in \big[0, \lceil L_{m,n}^{[k]}(0^m) \rceil\big)$ and in the computation of $L(w)$ in (5). In this case as well, $L(w)$ can be computed efficiently using matrix multiplication, as described in [7, §IV].

Like in [7], the correctness of the resulting algorithms hinges on the new values $L_{m,\ell}^{[k]}(y^m)$ satisfying the following consistency condition.

Lemma 6.1: For every $y \in \mathcal{V}_m$,

$$L_{m,\ell}^{[k]}(y^m) \le \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} L_{m,\ell-1}^{[k]}(x^m).$$

Proof: We have

$$\begin{aligned} L_{m,\ell}^{[k]}(y^m) &\overset{(11)}{=} \Big(\prod_{j=1}^{m} |\mathcal{H}_\ell(y_j)|\Big) Q_{m,\ell}^{[k-1]}(y^m) \Big(\prod_{i=k}^{\ell} q_{m,i}^{[k]}\Big) \\ &\overset{(9)}{\le} \Big(\prod_{j=1}^{m} |\mathcal{H}_\ell(y_j)|\Big) Q_{m,\ell}^{[k-1]}(y^m) \\ &\qquad \times \frac{Q_{m,\ell}^{[k]}(y^m)}{Q_{m,\ell}^{[k-1]}(y^m)} \Big(\prod_{i=k}^{\ell-1} q_{m,i}^{[k]}\Big) \\ &= \Big(\prod_{j=1}^{m} |\mathcal{H}_\ell(y_j)|\Big) Q_{m,\ell}^{[k]}(y^m) \Big(\prod_{i=k}^{\ell-1} q_{m,i}^{[k]}\Big). \end{aligned} \tag{18}$$

Now,

$$Q_{m,\ell}^{[k]}(y^m) = \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} p_{X^m|Y^m}^{(\ell)}(x^m|y^m) Q_{m,\ell-1}^{[k-1]}(x^m). \tag{19}$$

Also, from (6) we get that for any $x^m \in \mathcal{A}_{m,1}(y^m)$,

$$p_{X^m|Y^m}^{(\ell)}(x^m|y^m) = \frac{\prod_{j=1}^{m} |\mathcal{H}_{\ell-1}(x_j)|}{\prod_{j=1}^{m} |\mathcal{H}_\ell(y_j)|}. \tag{20}$$

Hence, by plugging (19) into (18) we obtain

$$\begin{aligned} L_{m,\ell}^{[k]}(y^m) &\le \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} p_{X^m|Y^m}^{(\ell)}(x^m|y^m) \Big(\prod_{j=1}^{m} |\mathcal{H}_\ell(y_j)|\Big) \\ &\qquad \times Q_{m,\ell-1}^{[k-1]}(x^m) \Big(\prod_{i=k}^{\ell-1} q_{m,i}^{[k]}\Big) \\ &\overset{(20)}{=} \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} \Big(\prod_{j=1}^{m} |\mathcal{H}_{\ell-1}(x_j)|\Big) \\ &\qquad \times Q_{m,\ell-1}^{[k-1]}(x^m) \Big(\prod_{i=k}^{\ell-1} q_{m,i}^{[k]}\Big) \\ &\overset{(11)}{=} \sum_{x^m \in \mathcal{A}_{m,1}(y^m)} L_{m,\ell-1}^{[k]}(x^m), \end{aligned}$$

thereby completing the proof. $\square$

The value of α and the sequence $y_1^* y_2^* \ldots y_J^*$ satisfying the assumptions of Lemma 5.1, along with the terminating sets $\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_J$ of the algorithm of Figure 3 (and some additional penalty factors to account for the non-asymptotic regime), can be leveraged to also obtain suitable values $q_{m,\ell}^{[k]}$ satisfying

$$\frac{1}{m} \log q_{m,\ell}^{[k]} \longrightarrow \log \alpha + \frac{1}{J} \log \frac{\lambda\big(\prod_{j=1}^{J} V^{[k]} D^{[k]}(y_j)\big)}{\lambda\big(\prod_{j=1}^{J} V^{[k-1]} D^{[k-1]}(y_j)\big)}$$

for increasing m and ℓ (we omit the details). Such values $q_{m,\ell}^{[k]}$, together with Figures 1 and 2, specify an encoder and decoder achieving a rate which approaches the sum of C_H and the right-hand side of (15).

References

[1] T.M. Cover, *Enumerative source encoding, IEEE Trans. Inform. Theory,* 19 (1973), 73–77.

[2] S. Forchhammer, J. Justesen, *Bounds on the capacity of constrained two-dimensional codes, IEEE Trans. Inform. Theory*, 46 (2000), 2659–2666.

[3] S. Forchhammer, T.V. Laursen, *Entropy of bit-stuffing-induced measures for two-dimensional checkerboard constraints, IEEE Trans. Inform. Theory*, 53 (2007), 1537–1546.

[4] K.A.S. Immink, *A practical method for approaching the channel capacity of constrained channels, IEEE Trans. Inform. Theory,* 43 (1997), 1389–1399.

[5] B.H. Marcus, R.M. Roth, P.H. Siegel, *Constrained systems and coding for recording channels,* in *Handbook of Coding Theory, Vol. II,* V.S. Pless, W.C. Huffman (Eds.), North-Holland, Amsterdam, 1998, pp. 1635–1764.

[6] E. Ordentlich, R.M. Roth, *Two-dimensional weight-constrained codes through enumeration bounds, IEEE Trans. Inform. Theory,* 46 (2000), 1292–1301.

[7] E. Ordentlich, R.M. Roth, *Capacity lower bounds and approximate enumerative coding for 2-D constraints,* in *Proc. IEEE Int'l Symp. on Inform. Theory,* Nice, France (2007), 1681–1685.