Fixed-Rate Tiling Encoders for 2-D Constraints

Artyom Sharov Ron M. Roth
Computer Science Department, Technion, Haifa 32000, Israel.
{sharov, ronny}@cs.technion.ac.il

***Abstract*—A new fixed-rate tiling-based coding scheme is presented for two-dimensional (2-D) constraints. The new scheme is shown to improve on the best known rates of formerly published fixed-rate encoders for certain constraints, such as the "no isolated bits" constraint and several 2-D runlength limited (RLL) constraints. Methods of efficient implementation of the suggested scheme are discussed.**

I. INTRODUCTION

In [6], we presented a variable-rate coding scheme, based on a periodic tiling of the 2-D plane, for a family of 2-D constraints. In its basic setting, the tiling divides the 2-D plane into two classes of patterns (tiles), referred to, respectively, as "white" and "black" tiles. The encoder assigns values to the white tiles independently of each other, while the value assigned to each black tile depends only on a finite neighborhood of white tiles. The scheme realizes a probability measure μ_s on $s \times s$ arrays that satisfy the constraint, for any positive integer s, and the expected rate of the scheme is given by the per-symbol entropy of μ_s. The limit of that entropy for $s \to \infty$ was computed exactly in [6, Th. 3.1], and that limit yields a lower bound on the capacity of the constraint. In [6, Sec. 4-B], we introduced a generalization of the basic coding scheme, where the values assigned to the white tiles form a stationary Markov chain along each diagonal of white tiles.

In this work, we present a fixed-rate coding scheme which, in some sense, mimics the probability μ_s. Our encoder will encode into 2-D arrays (which, for the sake of convenience, will assume the shape of a parallelogram) by alternating between diagonals of white tiles and diagonals of black tiles. The encoded white diagonals will have a prescribed transition count which is computed from a Markov chain (the same Markov chain as in [6]), while guaranteeing a desired empirical distribution of the white neighborhoods of the black tiles. By taking the size of the 2-D arrays to infinity, the resulting fixed-rate encoder will achieve a rate which is at least that of the respective variable-rate encoder from [6].

II. NOTATION AND BACKGROUND

Let U be a nonempty finite subset of the integer plane $\mathbb{Z}^2$ and let Σ be a finite alphabet. A *U-configuration* is a mapping $\varphi : U \to \Sigma$. The restriction of a U-configuration φ to a domain $U' \subseteq U$ will be denoted by $\varphi(U')$. For $(h, v) \in \mathbb{Z}^2$, we let $\sigma_{h,v}(U)$ stand for the shifted set $\{(r, s) : (r-h, s-v) \in U\}$. For a positive integer m, we denote by $[m]$ the integer set $\{0, 1, \ldots, m-1\}$ and by Q_m the square subset $\{(r, s) \in \mathbb{Z}^2 : r, s \in [m]\}$ of $\mathbb{Z}^2$.

A *2-D constraint* is a set $\mathbb{S}$ of square arrays over a finite alphabet Σ that satisfy certain requirements (constraints) defined through labeled directed graphs; for a full formal definition of 2-D constraints see, for example, [2, Sec. 1]. We will regard the elements of $\mathbb{S}$ as configurations $x : \sigma_{h,v}(Q_m) \to \Sigma$, for varying m, h, and v. A U-configuration $\varphi : U \to \Sigma$ is $\mathbb{S}$-*compatible* if there exists an array x in $\mathbb{S}$ such that $x(U) = \varphi$. We denote by $\mathbb{S}(U)$ the set of all $\mathbb{S}$-compatible U-configurations.

As was the case in [6], we define our coding scheme through a partition of the plane $\mathbb{Z}^2$ with a regular pattern (periodic tiling) by means of two types of tiles, referred to as white and black tiles. We will define tiles using shifted copies of finite subsets $B, W \subset \mathbb{Z}^2$ and will characterize all the positioning of the black tiles (which are shifts of B) by an integer lattice $\mathcal{L}$. The positioning of the white tiles (shifts of W) will be characterized by the shifted lattice $\sigma_{\ell,\ell'}(\mathcal{L})$ for some $\ell, \ell' \in \mathbb{Z}$. A tiling of the plane $\mathbb{Z}^2$ will be thus completely defined by the triple $(B, W, \mathcal{L})$ (see [6, Sec. 2]).

Example 2.1: Figure 1 shows a tiling of $\mathbb{Z}^2$ with $B = W = Q_m$, and

$$\mathcal{L} = \left\{ \left(mz, m(z + 2w)\right) : z, w \in \mathbb{Z} \right\} ,$$

with the shift $(\ell, \ell') = (-m, 0)$ (the figure is drawn to scale for $m = 3$). For this tiling, the black tile $B = Q_m$ (with $(0, 0)$ as its lower leftmost point) has four neighboring white tiles, which are positioned at

$$\mathcal{N} = \{(-m, 0), (m, 0), (0, -m), (0, m)\} . \qquad (1)$$

□

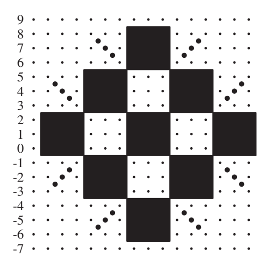

Fig. 1. Example of a tiling of $\mathbb{Z}^2$.

For a nonempty finite subset $\mathcal{N} \subset \mathbb{Z}^2$, we let $\mathcal{N} \oplus W$ denote the list $(\sigma_{h,v}(W))_{(h,v)\in\mathcal{N}}$ of $|\mathcal{N}|$ shifted copies of W. We define $\mathbb{S}(\mathcal{N} \oplus W)$ accordingly to denote the set $\left\{\boldsymbol{y} = (y_{h,v})_{(h,v)\in\mathcal{N}} : y_{h,v} \in \mathbb{S}(\sigma_{h,v}(W))\right\}$. That is, each element $\boldsymbol{y} \in \mathbb{S}(\mathcal{N} \oplus W)$ is a list of $\mathbb{S}$-compatible configurations which are defined on $|\mathcal{N}|$ shifted copies of W, where the shifts are determined by $\mathcal{N}$. Given now a list $\boldsymbol{y} = (y_{h,v})_{(h,v)\in\mathcal{N}}$ in $\mathbb{S}(\mathcal{N}\oplus W)$, we define $\mathbb{S}(B; \boldsymbol{y})$ to be the set of all configurations $\psi \in \mathbb{S}(B)$ such that ψ can be extended to an array x in $\mathbb{S}$ such that $x(B) = \psi$ and $x(\sigma_{h,v}(W)) = y_{h,v}$ for every $(h, v) \in \mathcal{N}$.

A tiling $(B, W, \mathcal{L})$ is *valid* with respect to a given 2-D constraint $\mathbb{S}$ if it satisfies the following two conditions:

[CW] *White tiles are freely configurable.* For every two nonempty finite subsets $\mathcal{M} \subset \sigma_{\ell,\ell'}(\mathcal{L})$ and $U \subset \mathbb{Z}^2$ and every list $\boldsymbol{y} \in \mathbb{S}(\mathcal{M} \oplus W)$ one has $\mathbb{S}(U; \boldsymbol{y}) \neq \emptyset$.

[CB] *Black configurations are constrained only by a finite neighborhood of white tiles.* There is a nonempty finite subset $\mathcal{N} \subset \sigma_{\ell,\ell'}(\mathcal{L})$ such that for every list $\boldsymbol{y} \in \mathbb{S}(\mathcal{N} \oplus W)$ and every nonempty finite subset $U \subset \mathbb{Z}^2 \setminus B$, $|\mathbb{S}(B \cup U; \boldsymbol{y})| = |\mathbb{S}(B; \boldsymbol{y})| \cdot |\mathbb{S}(U; \boldsymbol{y})|$.

It is easy to see that the tiling in Example 2.1 is valid for the 2-D (d, ∞)-RLL constraint for $m \geq d$, and for the 2-D $(0, k)$-RLL constraint for $m \geq k$, where we take $\mathcal{N}$ in condition [CB] to be the set (1). In this tiling $B = W$ (thus $|B| = |W|$); an example for a valid tiling for a constraint where $|B| \neq |W|$ is the tiling from [6, Ex. 2.1] for the "no isolated bits" constraint, where $B = Q_m$ and $W = Q_{m+1}$.

Next, we assume a partitioning of the white lattice $\sigma_{\ell,\ell'}(\mathcal{L})$ into infinite diagonals, defined by shifts of some infinite 1-D sub-lattice $\mathcal{D}$: these shifts are given by $\mathcal{D}^{(b)} = \sigma_{\tau b, \tau' b}(\mathcal{D})$, for some fixed $\tau, \tau' \in \mathbb{Z}$ and every $b \in \mathbb{Z}$. A (valid) tiling $(B, W, \mathcal{L})$ with such a partition of the white lattice will be denoted by $(B, W, \mathcal{L}, \mathcal{D})$ and will be called a *(valid) partitioned tiling*. A partitioned tiling $(B, W, \mathcal{L}, \mathcal{D})$ for the tiling in Example 2.1 is obtained, for instance, by taking

$$\mathcal{D} = \{(mz-m, mz) : z \in \mathbb{Z}\} \quad \text{and} \quad (\tau, \tau') = (m, -m) .$$

For the sake of simplicity of the presentation, we will henceforth specialize to this partitioned tiling and assume that it is valid for the 2-D constraint of interest $\mathbb{S}$, with condition [CB] holding for the neighborhood set $\mathcal{N}$ in (1).

Let $\mathcal{D}_N = \{(mz-m, mz) : z \in [N]\}$ be (a finite interval of) a diagonal of length N, let $\mathcal{D}_N^{(b)} = \sigma_{mb,-mb}(\mathcal{D}_N)$ be a shift of $\mathcal{D}_N$ for $b \in \mathbb{Z}$, and define the (rhombus-shaped) set $\Delta_N = \bigcup_{b \in [N]} \mathcal{D}_N^{(b)}$. Given an irreducible stationary Markov chain with a conditional probability distribution $\rho : \mathbb{S}(W) \times \mathbb{S}(W) \to [0, 1]$ and respective stationary distribution $\pi : \mathbb{S}(W) \to [0, 1]$, we define the following measure on the set $\mathbb{S}(\Delta_N \oplus W)$: for every $b \in [N]$, the restriction of a random $(\varphi_{h,v})_{(h,v)} \in \mathbb{S}(\Delta_N \oplus W)$ to the diagonal $\mathcal{D}_N^{(b)}$ (i.e., the configuration list $(\varphi_{h,v})_{(h,v)} \in \mathbb{S}(\mathcal{D}_N^{(b)} \oplus W)$) forms a Markov process with distribution ρ, and these processes are statistically independent for distinct b. This measure on $\mathbb{S}(\Delta_N \oplus W)$ has per-tile entropy

$$\mathsf{H}(\rho) = -\sum_{i \in \mathbb{S}(W)} \pi(i) \sum_{j \in \mathbb{S}(W)} \rho(j|i) \log \rho(j|i)$$

(hereafter all logarithms are taken to base 2). This measure on $\mathbb{S}(\Delta_N \oplus W)$, in turn, readily determines the probability $\mathsf{P}_\rho\{\boldsymbol{\varphi}\}$ of any assignment $\boldsymbol{\varphi} \in \mathbb{S}(\mathcal{N} \oplus W)$ to the white neighborhood of the black tiles that are fully "surrounded" by $\Delta_N \oplus W$. Reformulating [6, Th. 3.1] for this new setting yields the following result.

Theorem 2.1: Let $\mathbb{S}$ be a 2-D constraint for which $(B, W, \mathcal{L}, \mathcal{D})$ is a valid partitioned tiling. The capacity of $\mathbb{S}$ is bounded from below by

$$\frac{1}{|W|+|B|} \sup_\rho \Big(\mathsf{H}(\rho) + \sum_{\boldsymbol{\varphi}} \mathsf{P}_\rho\{\boldsymbol{\varphi}\} \log |\mathbb{S}(B; \boldsymbol{\varphi})| \Big) , \quad (2)$$

where $\boldsymbol{\varphi} = (\varphi_{h,v})_{(h,v) \in \mathcal{N}}$ ranges over $\mathbb{S}(\mathcal{N} \oplus W)$ and the supremum is taken over all irreducible stationary Markov chains $\rho : \mathbb{S}(W) \times \mathbb{S}(W) \to [0, 1]$.

Without real loss of generality, we assume that the supremum in (2) is actually attained by an irreducible stationary Markov chain ρ; this maximizing ρ will be used in the sequel in our fixed-rate encoder construction.

III. Fixed-rate encoder

We start by laying out the framework for encoding the white tiles positioned at Δ_N. Let $w \leq N$ be a positive integer and, for some $b \in [w]$, let $\boldsymbol{d} = (d_{r,s})_{(r,s) \in \mathcal{D}_w^{(b)}}$ be a configuration list in $\mathbb{S}(\mathcal{D}_w^{(b)} \oplus W)$ (assigned to the white tiles along $\mathcal{D}_w^{(b)}$). For the sake of simplicity, we will re-index the entries in $\boldsymbol{d}$ as $(d_z)_{z \in [w]}$, where $d_z = d_{m(z+b-1), m(z-b)}$. The size of $\mathbb{S}(W)$ will be denoted hereafter by n. For every $i, j \in \mathbb{S}(W)$, define $f_{i,j} = f_{i,j}(\boldsymbol{d})$ to be the number of indexes $z \in [w]$ such that $(d_z, d_{z+1}) = (i, j)$, where d_w is defined to be some fixed configuration in $\mathbb{S}(W)$, denoted hereafter by o. We will represent the (transition) counts $f_{i,j}$ of the configuration list $\boldsymbol{d}$ by the $n \times n$ *count matrix* $\mathcal{F}(\boldsymbol{d}) = (f_{i,j})_{i,j \in \mathbb{S}(W)}$.

Let $\Gamma = (\gamma_{i,j})_{i,j \in \mathbb{S}(W)}$ be an $n \times n$ nonnegative integer matrix, and denote $\gamma_{i,*} = \sum_{j \in \mathbb{S}(W)} \gamma_{i,j}$ and $\gamma_{*,j} = \sum_{i \in \mathbb{S}(W)} \gamma_{i,j}$. Suppose that for some $u \in \mathbb{S}(W)$ and every $i \in \mathbb{S}(W)$,

$$\gamma_{i,*} - \gamma_{*,i} = \delta(i, u) - \delta(i, o) , \quad (3)$$

where $\delta(\cdot, \cdot)$ is the Kronecker delta function. (Notice that (3) holds for $\Gamma = \mathcal{F}(\boldsymbol{d})$ where $u = d_0$.) By Whittle's formula [1, Th. 2.1], the number of configuration lists $\boldsymbol{d} \in \mathbb{S}(\mathcal{D}_w^{(b)} \oplus W)$ with $d_0 = u$ and $\mathcal{F}(\boldsymbol{d}) = \Gamma$ is given by

$$\mathsf{T}_u^{(w)}(\Gamma) = \mathsf{M}(\Gamma) \cdot \mathsf{C}_u(\Gamma) , \quad (4)$$

where

$$\mathsf{M}(\Gamma) = \Big(\prod_{i \in \mathbb{S}(W)} \gamma_{i,*}! \Big) \Big/ \Big(\prod_{i,j \in \mathbb{S}(W)} \gamma_{i,j}! \Big) \quad (5)$$

and $\mathsf{C}_u(\Gamma)$ denotes the (o, u)th cofactor of the following matrix $\Gamma^* = (\gamma^*_{i,j})_{i,j \in \mathbb{S}(W)}$:

$$\gamma^*_{i,j} = \begin{cases} \delta(i,j) - (\gamma_{i,j}/\gamma_{i,*}) & \text{if } \gamma_{i,*} > 0 \\ \delta(i,j) & \text{if } \gamma_{i,*} = 0 \end{cases} . \quad (6)$$

We remark that each $\boldsymbol{d} \in \mathbb{S}(\mathcal{D}_w^{(b)} \oplus W)$ with $d_0 = u$ corresponds—in a one-to-one and onto manner—to an Eulerian path of length w (edges) from state u to state o in the directed graph $\mathcal{G} = \mathcal{G}(A) = (\mathbb{S}(W), E)$ (possibly with parallel edges) whose adjacency matrix is Γ.

Our encoder will force a certain count matrix F on the configuration list along each diagonal $\mathcal{D}_N^{(b)}$ in Δ_N. We compute such an F from the maximizing Markov chain ρ in (2) by resorting to the method suggested in [7, Sec. 4]. Specifically, given the matrix $P = (p_{i,j})_{i,j \in \mathbb{S}(W)}$ with entries $p_{i,j} = \pi(i)\rho(j|i)$ and a positive integer N', the method in [7] yields a nonnegative integer matrix F with the following properties:

(P1) $N' - \lfloor n/2 \rfloor \leq \sum_{i,j \in \mathbb{S}(W)} f_{i,j} \leq N'$,
(P2) $f_{i,*} = f_{*,i}$ for any $i \in \mathbb{S}(W)$,
(P3) $(N' - \lfloor n/2 \rfloor) p_{i,j} - f_{i,j} \leq 1$ for any $i, j \in \mathbb{S}(W)$, and—
(P4) $\mathsf{H}(\rho) = (1/N') \lfloor \log \mathsf{M}(F) \rfloor - O\big((n^2 \log N')/N'\big)$.

Thus, F can be seen as an integer approximation of the matrix $N' \cdot P$ and, in addition, (P2) implies (3) for $u = o$.

We now take $N = \sum_{i,j \in \mathbb{S}(W)} f_{i,j}$ and, for every given $b \in [N]$, we assign a configuration list $\boldsymbol{d}$ with $\mathcal{F}(\boldsymbol{d}) = F$ (and $d_0 = o$) to $\mathcal{D}_N^{(b)} \oplus W$ using enumerative coding [3, Ch. 6]. A procedure for implementing such an assignment is shown in Fig. 2. The procedure gets the count matrix F and an index $\zeta \in \left[\mathsf{T}_o^{(N)}(F)\right]$ as its arguments and generates a configuration list $\boldsymbol{d} = (d_z)_{z \in [N]}$. In that figure, $\mathcal{J}_t(A)$ stands for the set of configurations u such that $a_{t,u} > 0$ (equivalently, $u \in \mathcal{J}_t(A)$ if and only if u is adjacent to t in the graph $\mathcal{G}(A)$), and $\partial_{t,u}(A)$ is the matrix obtained when the (t,u)th entry in A is decremented by 1.

$A = (a_{i,j}) \leftarrow F$; $d_0 \leftarrow o$;
for $z \in [N{-}1]$ do {
  $t \leftarrow d_z$;
  for $u \in \mathcal{J}_t(A)$ do {
    $\Gamma \leftarrow \partial_{t,u}(A)$;
    if $\zeta < \mathsf{T}_u^{(N-z-1)}(\Gamma)$ { $A \leftarrow \Gamma$; $d_{z+1} \leftarrow u$; break; }
    else { $\zeta \leftarrow \zeta - \mathsf{T}_u^{(N-z-1)}(\Gamma)$; }
  }
}

Fig. 2. Enumerative encoding into a configuration list $\boldsymbol{d} = (d_z)_{z \in [N]}$.

The procedure in Fig. 2 defines a one-to-one encoding due to the following equality, which holds at the beginning of each iteration z of the outer loop:

$$\textstyle\sum_{u \in \mathcal{J}_t(A)} \mathsf{T}_u^{(N-z-1)}(\partial_{t,u}(A)) = \mathsf{T}_t^{(N-z)}(A) \, . \tag{7}$$

The decoding algorithm for the procedure in Fig. 2 can be readily obtained, and is omitted due to space limitations.

Recall that the matrix F is an integer approximation of an irreducible matrix $N' \cdot P$; furthermore, by (P3) it follows that F is irreducible for sufficiently large N'. Therefore, the matrix F^* (defined as in (6)) is rational and its cofactors are all rational and (strictly) positive. In particular, $\mathsf{C}_o(F) \geq \left(\prod_{i \in \mathbb{S}(W)} f_{i,*}\right)^{-1} \geq (n/N)^n$. Combining this with (P1), (P4), and (4), we obtain that the per-tile coding rate along each diagonal $\mathcal{D}_N^{(b)} \oplus W$ equals

$$R(F) = (1/N)\left\lfloor \log \mathsf{T}_o^{(N)}(F) \right\rfloor = \mathsf{H}(\rho) - O\big((n^2 \log N)/N\big) \, . \tag{8}$$

We now turn to the encoding of the black tiles that are fully surrounded by white tiles positioned at Δ_N. Based on (2), we would like the per-tile contribution of the black tiles to the rate to be at least $\sum_{\boldsymbol{\varphi}} \mathsf{P}_\rho\{\boldsymbol{\varphi}\} \log |\mathbb{S}(B; \boldsymbol{\varphi})|$. To this end, we will shift the (already encoded) contents of the white diagonals cyclically in order to obtain a good empirical distribution of the white neighborhoods of black tiles.

Consider two adjacent white diagonals, $\mathcal{D}_N^{(b)}$ and $\mathcal{D}_N^{(b+1)}$, and respective configuration assignments, $\boldsymbol{d} = (d_z)_{z \in [N]}$ in $\mathbb{S}(\mathcal{D}_N^{(b)} \oplus W)$ and $\boldsymbol{d}' = (d'_z)_{z \in [N]}$ in $\mathbb{S}(\mathcal{D}_N^{(b+1)} \oplus W)$, where $\mathcal{F}(\boldsymbol{d}) = \mathcal{F}(\boldsymbol{d}') = F = (f_{i,j})$. For every $(i,j,k,\ell) \in (\mathbb{S}(W))^4$, let $f_{i,j,k,\ell}(\boldsymbol{d}, \boldsymbol{d}')$ denote the number of indexes $z \in [N]$ for which $(d_z, d_{z+1}, d'_z, d'_{z+1}) = (i,j,k,\ell)$. The per-tile coding rate of the black tiles that are positioned between $\mathcal{D}_N^{(b)}$ and $\mathcal{D}_N^{(b+1)}$ is given by

$$\mathsf{R}(\boldsymbol{d}, \boldsymbol{d}') = (1/N) \sum_{i,j,k,\ell \in \mathbb{S}(W)} f_{i,j,k,\ell}(\boldsymbol{d}, \boldsymbol{d}') \log |\mathbb{S}(B; \boldsymbol{\varphi})| \, ,$$

where $\boldsymbol{\varphi} = (\varphi_{-m,0}, \varphi_{m,0}, \varphi_{0,-m}, \varphi_{0,m}) = (i,j,k,\ell)$. Let $\theta_r(\boldsymbol{d}')$ denote the cyclic right shift of $\boldsymbol{d}'$ by r positions, namely, $\theta_r(\boldsymbol{d}') = (d'_{-r}, d'_{-r+1}, \ldots, d'_{-r+N-1})$ (where indexes are taken modulo N). We have the following lower bound (with proof omitted) on $\mathsf{R}(\boldsymbol{d}, \theta_g(\boldsymbol{d}'))$, for *some* shift $g \in [N]$.

Proposition 3.1: For at least one $g \in [N]$,

$$\mathsf{R}(\boldsymbol{d}, \theta_g(\boldsymbol{d}')) \geq (1/N^2) \sum_{i,j,k,\ell \in \mathbb{S}(W)} f_{i,j} f_{k,\ell} \log |\mathbb{S}(B; (i,j,k,\ell))| \, .$$

By (P3) it follows that the difference between this lower bound and the term $\sum_{\boldsymbol{\varphi}} \mathsf{P}_\rho\{\boldsymbol{\varphi}\} \log |\mathbb{S}(B; \boldsymbol{\varphi})|$ from (2), is $O(n^2/N)$. We refer to a shift $g \in [N]$ for which Proposition 3.1 holds as a *good shift of $\boldsymbol{d}'$ with respect to $\boldsymbol{d}$*. Clearly, once the encoder performs such a shift, it needs to convey the value of g to the decoder. Yet the description of g requires only $\lceil \log N \rceil$ bits and will result in a negligible additive penalty of $O\big((\log N)/N\big)$ to the per-tile rate.

We now proceed to formally define our fixed-rate coding scheme. Let $\nu = \lceil (\log N)/(\log n) \rceil + 1$. The encoder maps unconstrained input bit sequences into arrays in $\mathbb{S}(\mathcal{P}_N)$, where $\mathcal{P}_N$ is a (quasi) parallelogram formed by the union of the following two subsets of $\mathbb{Z}^2$:

- A union of N consecutive (white) diagonals of length $N{+}\nu$, i.e.: $\bigcup_{b \in [N], (h,v) \in \mathcal{D}_{N+\nu}^{(b)}} \sigma_{h,v}(W)$.
- A union of $N{-}1$ consecutive (black) diagonals of length $N{+}\nu{-}1$, i.e.: $\bigcup_{b \in [N-1], (h,v) \in \sigma_{m,0}(\mathcal{D}_{N+\nu-1}^{(b)})} \sigma_{h,v}(B)$.

The size of $\mathcal{P}_N$ is given by

$$|\mathcal{P}_N| = N(N{+}\nu)|W| + (N{-}1)(N{+}\nu{-}1)|B| \, . \tag{9}$$

The number of input bits encoded to each white diagonal will be given by

$$\omega = \left\lfloor \log \mathsf{T}_o^{(N)}(F) \right\rfloor \, , \tag{10}$$

and the respective number for each black diagonal will be

$$\beta = \left\lfloor (1/N) \sum_{i,j,k,\ell \in \mathbb{S}(W)} f_{i,j} f_{k,\ell} \log |\mathbb{S}(B; (i,j,k,\ell))| \right\rfloor \, . \tag{11}$$

The fixed-rate procedure shown in Fig. 3 takes $N\omega + (N{-}1)\beta$ input bits and produces an array $(x_{r,s})_{(r,s) \in \mathcal{P}_N} \in \mathbb{S}(\mathcal{P}_N)$. In the procedure, the first N tiles along each white diagonal $\mathcal{D}_{N+\nu}^{(b)}$ are used to encode the input, while the $(N{+}1)$st white tile is assigned the same configuration as the first tile, in order to ensure a correct count of white neighborhoods for the black tiles. The last ν tiles along the white diagonal are used to record the (good) shift of the next ($(b{+}1)$st) white diagonal. The encoding is carried out diagonal-by-diagonal in a three-steps-forward one-step-backward manner: the encoding of each new white diagonal is followed by the encoding of the black diagonal which *precedes* it, thereby ensuring that the black tiles are encoded after the configurations in their white neighborhoods have already been fully assigned. The decoding of an array encoded by Fig. 3 is carried out in a similar diagonal-by-diagonal fashion.

From (8)–(11) we get the following result.

1) *(Encode the first white diagonal)*
   a) Apply the procedure of Fig. 2 to the integer represented by the first ω input bits, to produce a configuration list $\boldsymbol{\varphi}^{(0)} = (\varphi_{h,v})$ in $\mathbb{S}(\mathcal{D}_N^{(0)} \oplus W)$ (where (h,v) ranges over $\mathcal{D}_N^{(0)}$). Extend $\boldsymbol{\varphi}^{(0)}$ by letting the $(N{+}1)$st configuration equal o.
   b) For $(h,v) \in \mathcal{D}_{N+1}^{(0)}$ set $x(\sigma_{h,v}(W)) \leftarrow \varphi_{h,v}$.
2) *(Encode the remaining diagonals)* For $b \leftarrow 1, 2, \ldots, N{-}1$ do:
   a) Apply the procedure of Fig. 2 to the integer represented by the next ω input bits, to produce a configuration list $\boldsymbol{\varphi}^{(b)} = (\varphi_{h,v})$ in $\mathbb{S}(\mathcal{D}_N^{(b)} \oplus W)$ ((h,v) ranges over $\mathcal{D}_N^{(b)}$).
   b) Find $g \in [N]$ such that $\boldsymbol{\xi}^{(b)} = (\xi_{h,v}^{(b)}) = \theta_g(\boldsymbol{\varphi}^{(b)})$ is a good shift of $\boldsymbol{\varphi}^{(b)}$ with respect to $\boldsymbol{\varphi}^{(b-1)}$. Extend $\boldsymbol{\xi}^{(b)}$ by letting the $(N{+}1)$st configuration equal the first.
   c) Select the gth configuration list $\boldsymbol{\psi} = (\psi_{h,v})$ in $\mathbb{S}(\mathcal{H}^{(b)} \oplus W)$, where $\mathcal{H}^{(b)} = (\mathcal{D}_{N+\nu}^{(b-1)} \setminus \mathcal{D}_{N+1}^{(b-1)}) \oplus W$.
   d) *(Encode a good shift)* For $(h,v) \in \mathcal{H}^{(b)}$ set $x(\sigma_{h,v}(W)) \leftarrow \psi_{h,v}$.
   e) *(Encode white diagonal)* For $(h,v) \in \mathcal{D}_{N+1}^{(b)}$ set $x(\sigma_{h,v}(W)) \leftarrow \xi_{h,v}^{(b)}$.
   f) *(Encode black diagonal)* Let ζ be the integer represented by the next β input bits. Set $T \leftarrow 2^\beta$.
      For $(h,v) \in \sigma_{m,0}(\mathcal{D}_{N+\nu-1}^{(b-1)})$ do:
      i) Set $T \leftarrow \lceil T / |\mathbb{S}(\sigma_{h,v}(B); \boldsymbol{\varphi}_{h,v})| \rceil$, where $\boldsymbol{\varphi}_{h,v}$ is the configuration list in $\mathbb{S}(\sigma_{h,v}(\mathcal{N}) \oplus W)$ assigned to the white neighborhood of $\sigma_{h,v}(B)$.
      ii) Set $q \leftarrow \lfloor \zeta / T \rfloor$.
      iii) Set $x(\sigma_{h,v}(B))$ to the qth configuration in $\mathbb{S}(\sigma_{h,v}(B); \boldsymbol{\varphi}_{h,v})$.
      iv) Set $\zeta \leftarrow \zeta - q \cdot T$.

Fig. 3. Encoding into an array $(x_{r,s}) \in \mathbb{S}(\mathcal{P}_N)$.

Theorem 3.2: The coding rate of the procedure in Fig. 3 is, up to an additive term $O\big((n^2 \log N)/(N \log n)\big)$,

$$\frac{1}{|W|+|B|}\Big(R(F) + \frac{1}{N^2}\sum_{i,j,k,\ell \in \mathbb{S}(W)} f_{i,j} f_{k,\ell} \log |\mathbb{S}(B; (i,j,k,\ell))|\Big)$$

which, for $N \to \infty$, is at least the lower bound (2).

Table I presents lower bounds on the rates of fixed-rate encoders constructed using the procedure of Fig. 3. The rates are taken from [6, Tables I, III] and are, in fact, the rates of encoders constructed by the variable-rate coding scheme therein. For the 2-D RLL constraints in Table I we used the valid tiling of Example 2.1, whereas for the n.i.b. constraint we used the valid tiling from [6, Ex. 2.1] with $\mathcal{D} = \{(mz, (m{+}1)(z{-}1)) : z \in \mathbb{Z}\}$. To the best of our knowledge, the rates in Table I are the best known for fixed-rate encoders for the respective constraints, except for the 2-D $(2,\infty)$-RLL constraint, where our result falls short of the rate 0.4453 of the fixed-rate encoder from [5].

TABLE I
LOWER BOUNDS ON THE RATES OF SEVERAL FIXED-RATE ENCODERS.

Constraint	$(1,\infty)$-RLL	$(2,\infty)$-RLL	$(3,\infty)$-RLL	$(0,2)$-RLL	n.i.b.
Rate	**0.586485**	0.44417	**0.36562**	**0.81600**	**0.92086**

The practicality of our encoding procedure clearly depends on the efficiency of implementation of the enumerative coding of the white diagonals in Steps 1a and 2a, the method for finding good shifts in Step 2b, and the encoding of the black diagonals in Step 2f. For the former two, we will present such efficient implementations in Sections IV and V, respectively. Step 2f is essentially a binary to mixed-radix conversion, which, in turn, is a special case of enumerative coding. As such, it can be implemented, similarly to the white tiles, using the method of approximate enumerative coding to be discussed in Section IV.

IV. ENUMERATIVE CODING OF WHITE DIAGONALS

In Fig. 2, we may need to compute the values $\mathsf{T}_u^{(\cdot)}(\cdot)$ for up to $n = |\mathbb{S}(W)|$ distinct configurations u in each iteration of the outer loop. If we use Whittle's formula (4), then a naive computation of the cofactors $\mathsf{C}_u(\cdot)$ may require $O(n^4)$ operations over numbers of $O(\log N)$ bits (if determinants are computed through Gaussian elimination). This complexity can be significantly improved (at the price of a negligible rate penalty) if we use lower bounds rather than the exact values of the cofactors. We elaborate next on this approach.

Let $A = A(z)$ and $t = d_z$ be the respective values at the beginning of iteration z of the outer loop in Fig. 2, and consider the matrix $\Gamma = \partial_{t,u}(A)$ used in iteration u of the inner loop. Recalling the definition of Γ^* from (6), the cofactor $\mathsf{C}_u(\Gamma)$ is bounded from below by

$$\widehat{\mathsf{C}}_u(\Gamma) = \begin{cases} \left(\prod_{i \in \mathbb{S}(W)} \gamma_{i,*}\right)^{-1} & \text{if } \mathsf{C}_u(\Gamma) > 0 \\ 0 & \text{if } \mathsf{C}_u(\Gamma) = 0 \end{cases} \qquad (12)$$

and, consequently, $\mathsf{T}_u^{(N-z-1)}(\Gamma)$ is bounded from below by

$$\widehat{\mathsf{T}}_u^{(N-z-1)}(\Gamma) = \mathsf{M}(\Gamma) \cdot \widehat{\mathsf{C}}_u(\Gamma) \, .$$

The following lemma implies that replacing $\mathsf{T}_u^{(\cdot)}(\cdot)$ with $\widehat{\mathsf{T}}_u^{(\cdot)}(\cdot)$ in Fig. 2 does not affect the validity of the procedure therein (compare with (7)). We omit the proof.

Lemma 4.1: For $z \in [N{-}1]$,

$$\textstyle\sum_{u \in \mathcal{J}_t(A)} \widehat{\mathsf{T}}_u^{(N-z-1)}(\partial_{t,u}(A)) \geq \widehat{\mathsf{T}}_t^{(N-z)}(A)$$

and $\widehat{\mathsf{T}}_t^{(N-z)}(A) \leq \mathsf{T}_t^{(N-z)}(A)$, where $A = A(z)$ and $t = d_z$.

In order to make the lower bound (12) useful, we will also need to check efficiently whether $\mathsf{C}_u(\partial_{t,u}(A)) = 0$; a configuration $u \in \mathcal{J}_t(A)$ for which this equality holds will be referred to as a *deadlock* (for A). We have the following lemma (which is stated without proof).

Lemma 4.2: There is at most one deadlock in $\mathcal{J}_t(A)$, and $u \in \mathcal{J}_t(A)$ is the deadlock if and only if $a_{t,*} > a_{t,u} = 1$ and there is no path from u to t in $\mathcal{G}(A)$.

Next, we turn to the computation of the multinomial $\mathsf{M}(\Gamma)$ in (5). A naive computation of these multinomials incurs at least the order of $N \log N$ bit operations per tile. In order to reduce that complexity, we resort to the technique of *approximate enumerative coding* [4],[7, Sec. 7].

Let α be a positive real. We will approximate α with a *floating-point* number of the form $\overline{\alpha} = c \cdot 2^h$, where c and h are integers satisfying $2^\mu \leq c < 2^{\mu+1}$ and $-2^{\epsilon-1} \leq h < 2^{\epsilon-1}$, for some prescribed positive integers μ and ϵ. The value of ϵ should be taken large enough to allow us store $\mathsf{M}(F) = \mathsf{M}((f_{i,j}))$, and, to this end, it suffices to take $\epsilon = (1 + o(1)) \log N$. We set $\eta(n) = 2n^2{+}2n{+}1$ (for $n = |\mathbb{S}(W)|$) and take $\mu = \lceil \log (N^2 + N\eta(n)) \rceil$. Next, we prepare two lookup tables: one containing approximations of $1!$, $2!$, $\ldots$, $(\max_i f_{i,*})!$, and the other containing approximations of the values $\kappa(1), \kappa(2), \ldots, \kappa(N)$ of a function $\kappa(\cdot)$, which is defined recursively by $\kappa(1) = \big(1 - (1/(2^\mu+1))\big)^{\eta(n)+2(N-1)}$ and $\kappa(w+1) = \kappa(w)\big(1-(2/(2^\mu+1))\big)^{\eta(n)+2(N-w)}(1+2^{-\mu})^2$ for $1 \le w < N$. Altogether the space complexity for storing both tables is $O(N(\mu+\epsilon))$.

Using the floating-point representation and the lower bound $\widehat{\mathsf{C}}_u(\Gamma)$ (from (12)) on $\mathsf{C}_u(\Gamma)$, the encoding procedure remains like in Fig. 2, except that we replace any *nonzero* $\mathsf{T}_u^{(N-z-1)}(\Gamma)$ by

$$\overline{\mathsf{T}}_u^{(N-z-1)}(\Gamma) = \lceil \overline{\kappa}(N-z-1) \times \widehat{\mathsf{M}}(\Gamma) \rceil \tag{13}$$

where $\widehat{\mathsf{M}}(\Gamma) = \big(\prod_{i\in\mathbb{S}(W)} \overline{(\gamma_{i,*}-1)!}\big) \div \big(\prod_{i,j\in\mathbb{S}(W)} \overline{\gamma_{i,j}!}\big)$, with multiplications $\times$ and divisions $\div$ being all floating-point. The computation of $\widehat{\mathsf{M}}(F)$ takes $O(n^2)$ floating-point operations due to the lookup tables and, once we have the value of $\widehat{\mathsf{M}}(F)$, the computation of any $\widehat{\mathsf{M}}(\Gamma)$ in Fig. 2 takes only $O(1)$ floating-point operations.

It turns out that Lemma 4.1 holds even when we replace all the *nonzero* terms $\widehat{\mathsf{T}}_\cdot^{(\cdot)}(\cdot)$ therein by the respective approximations $\overline{\mathsf{T}}_\cdot^{(\cdot)}(\cdot)$. The additive penalty to the coding rate due to the suggested approximation method is $O\big((n \log N)/(N \log n)\big)$, whereas the number of floating-point operations per tile required for the computations of the multinomials reduces to just $O(n)$.

Figure 4 shows a procedure for encoding an assignment $\boldsymbol{d} = (d_z)_{z\in[N]}$ to a white diagonal $\mathcal{D}_N^{(b)}$ using the approximation (13) given the count matrix F and an index $\zeta \in \big[\overline{\mathsf{T}}_o^{(N)}(F)\big]$ as arguments. The main difference between this procedure and Fig. 2 is the mechanism for handling deadlocks. The ordering that is assumed on $\mathcal{J}_t(A)$ guarantees that this mechanism indeed works (we skip the details). The check whether a given u is a deadlock will be carried out using Lemma 4.2, namely, by finding whether there is no path in $\mathcal{G}(\partial_{t,u}(A))$ from u to t. The latter check, in turn, can be carried out by, say, *Depth First Search* (DFS), totaling $O(n^4/N)$ operations per tile (on average).

$A = (a_{i,j}) \leftarrow F$; $d_0 \leftarrow 0$;
for $z \in [N-1]$ do {
  $t \leftarrow d_z$;
  assume an ordering on $\mathcal{J}_t(A)$ such that $\forall u, u' \in \mathcal{J}_t(A)$:
    $a_{t,u} > a_{t,u'} \Longrightarrow u < u'$;
  for u in $\mathcal{J}_t(A)$ in ascending order do {
    $\Gamma \leftarrow \partial_{t,u}(A)$;
    if $\zeta < \overline{\mathsf{T}}_u^{(N-z-1)}(\Gamma)${
      if u is a deadlock { $u \leftarrow \min(u' \in \mathcal{J}_t(A) \,:\, u' > u)$; }
      $A \leftarrow \partial_{t,u}(A)$; $d_{z+1} \leftarrow u$; break;
    }
    else { $\zeta \leftarrow \zeta - \overline{\mathsf{T}}_u^{(N-z-1)}(\Gamma)$; }
  }

Fig. 4. Approximate enumerative encoding.

The respective decoding procedure is omitted.

V. Finding a good shift

A naive implementation of the procedure for finding a good shift requires, in the worst case, $O(N)$ operations per tile. In this section, we will demonstrate an acceleration method of the naive implementation based on the *discrete Fourier transform* (DFT) over the complex field $\mathbb{C}$.

Given two configuration assignments to adjacent white diagonals, $\boldsymbol{d} = (d_z)_{z\in[N]} \in \mathbb{S}(\mathcal{D}_N^{(b)} \oplus W)$ and $\boldsymbol{d}' = (d'_z)_{z\in[N]} \in \mathbb{S}(\mathcal{D}_N^{(b+1)} \oplus W)$, and fixing $i, j, k, \ell \in \mathbb{S}(W)$, we define two real vectors, $\boldsymbol{e} = (e_z)_{z\in[N]}$ and $\boldsymbol{e}' = (e'_z)_{z\in[N]}$, as follows: for every $z \in [N]$, we let $e_z = \delta(d_z, i) \cdot \delta(d_{z+1}, j)$ and $e'_z = \delta(d'_{-z}, k) \cdot \delta(d'_{1-z}, \ell)$ (with indexes taken modulo N). The cyclic convolution $\boldsymbol{e} * \boldsymbol{e}'$ equals a vector $\boldsymbol{\chi}_{i,j,k,\ell} = \boldsymbol{\chi} = (\chi_r)_{r\in[N]}$ whose entries are given by $\chi_r = f_{i,j,k,\ell}(\boldsymbol{d}, \theta_r(\boldsymbol{d}'))$. Computing $\boldsymbol{\chi}_{i,j,k,\ell}$ for all $i, j, k, \ell \in \mathbb{S}(W)$, in turn, can be carried out using fast implementations of DFT, thereby incurring $O(n^4 N \log N)$ operations over $\mathbb{C}$ (yet see Remark 5.1 below).

Figure 5 presents a DFT-based procedure for finding a good shift: the procedure takes two configuration lists, $\boldsymbol{d} = (d_z)_{z\in[N]}$ and $\boldsymbol{d}' = (d'_z)_{z\in[N]}$, and returns the value of a best shift of $\boldsymbol{d}$ with respect to $\boldsymbol{d}'$. The vector $\boldsymbol{X}$ in the figure equals $\mathsf{DFT}(\boldsymbol{\chi})$, and $\odot$ stands for component-wise multiplication.

$\boldsymbol{Y} = (Y_z)_{z\in[N]} \leftarrow \boldsymbol{0}$;
for $(i,j,k,\ell) \in (\mathbb{S}(W))^4$ do {
  $\boldsymbol{e} \leftarrow (\delta(d_z, i))_{z\in[N]} \odot (\delta(d_{z+1}, j))_{z\in[N]}$;
  $\boldsymbol{e}' \leftarrow (\delta(d'_{-z}, k))_{z\in[N]} \odot (\delta(d'_{1-z}, \ell))_{z\in[N]}$;
  $\boldsymbol{X} \leftarrow \mathsf{DFT}(\boldsymbol{e}) \odot \mathsf{DFT}(\boldsymbol{e}')$;
  $\boldsymbol{Y} \leftarrow \boldsymbol{Y} + \log|\mathbb{S}(B; (i,j,k,\ell))| \cdot \boldsymbol{X}$;
}
$(y_z)_{z\in[N]} \leftarrow \mathsf{DFT}^{-1}(\boldsymbol{Y})$;
return $\arg\max_{z\in[N]} y_z$;

Fig. 5. DFT-based procedure for finding a good shift.

If we use the approximate enumerative encoding of Fig. 4 and the procedure in Fig. 5 for shifting diagonals, then the accumulated additive rate penalty from all the stages of the presented coding scheme is dominated by the term $O\big((n^2 \log N)/(N \log n)\big)$ (from (P4)). On the other hand, the time complexity per tile is $O(n^4 \log N)$ floating-point operations over $O(\log N)$-bit number representations, where the dominating $O(n^4 \log N)$ term stems from Fig. 5.

Remark 5.1: The dependence of the time complexity on n can be further mitigated using specific properties of the constraint, such as symmetry or small memory. Specifically, the complexity of the procedure in Fig. 5 can be reduced to $\exp(\sqrt{\log n}) \cdot N \log N < O(nN \log N)$, where the base of the exponent depends on the memory of $\mathbb{S}$. We omit the details.$\square$

References

[1] P. Billingsley, "Statistical methods in Markov chains," *Ann. Math. Stat.,* vol. 32, pp. 12–40, 1961.

[2] S. Halevy R.M. Roth, "Parallel constrained coding with application to two-dimensional constraints," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1009–1020, 2002.

[3] K.A.S. Immink, *Codes for Mass Data Storage Systems,* Shannon Foundation Publishers, The Netherlands, 2004.

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

[5] E. Ordentlich, R.M. Roth, "Approximate enumerative coding for 2-D constraints through ratios of matrix products," *Proc. IEEE Symp. Inform. Theory (ISIT 2009),* Seoul, Korea, 2009, pp. 1050–1054.

[6] A. Sharov, R.M. Roth, "Two-dimensional constrained coding based on tiling," *Proc. IEEE Symp. Inform. Theory (ISIT 2008),* Toronto, ON, 2008, pp. 1468–1472.

[7] I. Tal, R.M. Roth, "On row-by-row coding for 2-D constraints," *IEEE Trans. Inform. Theory,* vol. 55, pp. 3565–3576, 2009.