

MCMC Methods for Drawing Random Samples from the Discrete-Grains Model of a Magnetic Medium

Abhinav Das N. V.

Navin Kashyap

Abstract—The discrete-grains model is a simple model for the distribution of “grains” on a magnetic medium. In this model, grains on the medium are taken to be one of four basic rectangular shapes (“tiles”) — 1×1 , 1×2 , 2×1 and 2×2 . The magnetic medium is then modeled as an $N \times N$ square tiled by these four basic tiles. In this paper, we present Markov chain Monte Carlo (MCMC) methods for generating random tilings of an $N \times N$ square consisting only of the four basic tiles. To be precise, given a target probability distribution, e.g., the uniform distribution, on the space, \mathcal{S}_N , of all possible such tilings, we make use of the Metropolis-Hastings algorithm to design an MCMC method for sampling from a probability distribution arbitrarily close, in total variation distance, to the target distribution. We further extend this approach to enable sampling from probability distributions on certain subsets of \mathcal{S}_N , namely, those consisting of tilings in which each kind of tile occurs a fixed number of times. We finally present some bounds and conjectures on the mixing times of the underlying Markov chains, which provide estimates of the amount of time taken by the MCMC methods to generate a random tiling sampled from the target probability distribution.

Index Terms—discrete-grains model, Markov Chain Monte Carlo (MCMC) methods, Metropolis-Hastings algorithm, two-dimensional magnetic recording (TDMR)

I. INTRODUCTION

The two-dimensional magnetic recording (TDMR) technology conceived by Wood et al. [1] proposes to use sophisticated two-dimensional coding and signal processing techniques to drastically increase the areal density of bits that can be reliably recorded on a magnetic medium. The key to developing such techniques is having good models for the distribution of the fundamental magnetizable units called “grains” on the medium. Models for grain distributions yield useful channel models for the magnetic medium that try to capture the deviations from ideal behavior in the write and readback processes. While *write channel models* attempt to characterize the discrepancy between what the write head intends to write on the magnetic medium and what actually gets recorded on the medium, *read channel models* try to pin down the non-idealities in the readback process [2]–[6]. At the kind of Terabits-per-square-inch areal densities targeted by TDMR, the

This work was supported in part by research grants from the Department of Science & Technology and the Department of Electronics & Information Technology, Government of India.

A. Das is with Bharat Electronics Limited (BEL), Bangalore, India. Email: abhinavdas.n.v@gmail.com

N. Kashyap is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore. Email: nkashyap@ece.iisc.ernet.in

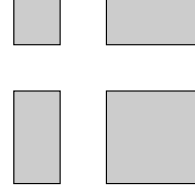


Fig. 1. The rectangular tileset.

distribution of grains in the medium is a significant source of non-ideality in both the write and readback processes. A nice discussion of the effect of grain distributions on the write and readback processes in TDMR can be found in the original paper of Wood et al. [1].

Several models for grain distributions on a magnetic medium have been considered in the literature [2], [3], [4], [7]. Perhaps the simplest of these is the discrete-grains model, which seems to have been first proposed by Krishnan et al. [2] and Chan et al. [3]. As pointed out by Chan et al. [3], this model is an oversimplified view of the medium, but it has the advantage of being easily simulated, and at the same time, being fairly amenable to analysis [5]. Subsequent works have used this model as a test-bed for novel coding and signal detection algorithms for TDMR [8], [9].

In the discrete-grains model, grains are assumed to have one of finitely many pre-defined shapes, and the magnetic medium is modeled as a tiling of a bounded region of the plane by these pre-defined shapes. We will call each such pre-defined shape a *basic tile*, or simply a *tile*, and the set of these shapes as the *tileset*. The most commonly used tileset consists of four rectangular tiles of dimensions 1×1 , 1×2 , 2×1 and 2×2 , the dimensions being specified with respect to some fundamental unit of length – see Figure 1. In this paper, we will restrict our attention to this rectangular tileset, although our techniques can be readily extended to other tilesets as well.

To meaningfully talk about the channel capacity of a granular magnetic medium [1], [5], [9], or indeed to evaluate the performance of two-dimensional coding and signal processing algorithms, one requires a probabilistic model for the overall channel formed by the cascade of the write and read channels. To this end, it is necessary to have a good probabilistic model for the underlying grain distribution. Within the context of the discrete-grains model, this means that one must have a probability distribution on the set of *valid* tilings of the plane, i.e., tilings that use only tiles from the given tileset. Typically,

this probability distribution is described by specifying the relative frequency of occurrence of each basic tile within a random tiling sampled from the distribution [2], [5], [8]. Strictly speaking, the relative frequencies of occurrence by themselves do not uniquely determine a probability distribution on valid tilings; nevertheless, from specifications of these, one can obtain a useful channel model in practice.

For the purpose of carrying out simulations — for example, to evaluate the performance of novel coding and signal processing schemes designed for TDMR — one needs algorithms to generate samples of the underlying granular magnetic medium that are consistent with the probabilistic model being assumed. Thus, given a probability distribution Π on valid tilings, it is necessary to have a means of generating a random tiling drawn according to the distribution Π . This is what we call *sampling* from the probability distribution Π . More concretely, our interest lies in generating random valid tilings of an $N \times N$ square (which represents the magnetic medium), for large values of N (say, $N \geq 20$), sampled from a given probability distribution $\Pi^{(N)}$ on such tilings. The sampling methods described in the prior literature [2], [5], [8] are essentially *ad hoc* in nature. They involve “randomly” filling in tiles into the $N \times N$ square in a manner consistent with the desired relative frequency of occurrence of each basic tile, starting with the hardest-to-fit tile (the 2×2 tile in the rectangular tileset) and ending with the easiest (the 1×1 tile). It is far from clear what overall probability distribution on valid tilings is induced by this procedure.

In this paper, we propose a mathematically rigorous procedure that, in a certain asymptotic sense, samples from a desired probability distribution on valid tilings of an $N \times N$ square. At the heart of our sampling procedure is the Metropolis-Hastings algorithm, a Markov chain Monte Carlo (MCMC) method that we describe in detail in the next section. It should be clarified that our choice of squares as the regions to be tiled is completely arbitrary — there is no inherent difficulty in using MCMC methods to generate random tilings of rectangular regions or indeed regions of other shapes.

The rest of this paper is organized as follows. The Metropolis-Hastings algorithm is described in Section II. In Section III, we explain how this algorithm can be tuned to yield an MCMC method for sampling from a probability distribution arbitrarily close (in a sense that we will make precise) to the uniform distribution on valid tilings of the $N \times N$ square. Then, in Section IV, we modify the algorithm so that, for a pre-specified set of relative frequencies of occurrence of the four basic tiles, it generates a tiling drawn nearly uniformly at random from the set of all valid tilings of the $N \times N$ square in which each basic tile occurs with the desired relative frequency. The issue of convergence of these algorithms is discussed in Section V. The paper ends in Section VI with some concluding remarks.

II. THE METROPOLIS-HASTINGS ALGORITHM

The main issue with sampling from some probability distribution, for example, the uniform distribution, on the set of all valid tilings of an $N \times N$ square is that it is impractical to

list out all the valid tilings. In fact, it is very difficult even to determine the total number of such tilings. It is in situations such as this that MCMC methods are most useful. The overview of MCMC in this section assumes some familiarity with the basic definitions and terminology of Markov chain theory — see e.g., [10, Chapters 1 and 2]. For more details on MCMC, the reader is referred to the handbook [11].

Let \mathcal{S} be the sample space, which we assume to be finite, on which the desired probability distribution Π is defined. The idea behind the MCMC method is to define a suitable discrete-time Markov chain with \mathcal{S} as its state space, such that Π is the unique stationary distribution of the chain. The Markov chain $(X_n)_{n \geq 0}$ to be defined must be time-homogeneous, irreducible and aperiodic, so that starting from any choice of the initial distribution on X_0 , the Markov chain converges to the stationary distribution Π . The MCMC method then consists of simulating the Markov chain for T time steps starting from an arbitrary choice of the initial distribution, where T is chosen large enough that the distribution of X_T is assured to be close (say, in terms of total variation distance¹) to the stationary distribution Π . Thus, the realization of X_T obtained from one such simulation run may be taken to be a sample drawn from the distribution Π . For this method to be useful, one must design the Markov chain so that it can be efficiently simulated and so that it rapidly converges to the stationary distribution.

The Metropolis-Hastings (MH) algorithm [12], [13] is a recipe for designing a Markov chain with a desired stationary distribution Π . A time-homogeneous Markov chain on the state space \mathcal{S} is specified by its transition probabilities $p(s'|s) := \Pr[X_{n+1} = s' | X_n = s]$, defined for all $s, s' \in \mathcal{S}$. The MH choice of $p(s'|s)$ is given by

$$p(s'|s) = \begin{cases} \alpha(s, s')q(s'|s) & \text{if } s' \neq s \\ 1 - \sum_{s'' \neq s} \alpha(s, s'')q(s''|s) & \text{if } s' = s, \end{cases} \quad (1)$$

where $q(s'|s)$ is the *proposal* or *candidate* transition probability, and $\alpha(s, s')$ is the *acceptance probability* defined by

$$\alpha(s, s') = \min \left\{ 1, \frac{\Pi(s')q(s|s')}{\Pi(s)q(s'|s)} \right\}. \quad (2)$$

The probabilities $q(\cdot|s)$ and $\alpha(\cdot, s)$ are to be interpreted as follows: given that the Markov chain is in state s at time n (i.e., $X_n = s$), the next state, X_{n+1} , is *proposed* to be s' with probability $q(s'|s)$.² With probability $\alpha(s, s')$, the proposal is accepted, and we have $X_{n+1} = s'$; on the other hand, with probability $1 - \alpha(s, s')$, the proposal is rejected and the chain remains in state s at the next time instant (i.e., $X_{n+1} = s$). With this choice of $p(s'|s)$, the resulting Markov chain is easily verified to be reversible, with Π as a stationary distribution. The MH algorithm is summarized in Algorithm 1.

The transition probability matrix $Q = (q(s'|s))_{s, s' \in \mathcal{S}}$ should be chosen so that the Markov chain governed by the transition probabilities $p(s'|s)$, as defined by (1), is irreducible and aperiodic. Beyond that, Q is a design parameter to be

¹The total variation distance between two probability distributions Π and Π' on \mathcal{S} is defined as $\|\Pi - \Pi'\|_{\text{TV}} = \max_{A \subseteq \mathcal{S}} |\Pi(A) - \Pi'(A)|$. It is a standard fact that $\|\Pi - \Pi'\|_{\text{TV}} = \frac{1}{2} \sum_{s \in \mathcal{S}} |\Pi(s) - \Pi'(s)|$.

²To be clear, $q(s|s)$, i.e., $q(s'|s)$ for $s' = s$, is also allowed to be strictly positive.

Algorithm 1 The Metropolis-Hastings algorithm

Initialization: $X_0 \leftarrow s_0$
for $n = 0$ to $T - 1$ **do**
 If $X_n = s$, generate an $s' \in \mathcal{S}$ according to the proposal distribution $q(\cdot|s)$
 Set $X_{n+1} = \begin{cases} s' & \text{with probability } \alpha(s, s') \\ s & \text{with probability } 1 - \alpha(s, s') \end{cases}$
end for
return X_T

fine-tuned to obtain good performance — Q governs both the rate of convergence of the Markov chain as well as the computational complexity of picking a proposed next state.

The MH algorithm has one very useful property: it works even in situations when the desired probability distribution $\Pi = (\Pi(s))_{s \in \mathcal{S}}$ is specified only as being proportional to a weight distribution $(w(s))_{s \in \mathcal{S}}$, where $w(s)$ is a non-negative weight function computable directly from s . The constant of proportionality — usually denoted by Z^{-1} , where Z is the normalization constant $\sum_{s \in \mathcal{S}} w(s)$ — need not be exactly computed. This is very useful in situations where it is difficult or impractical to exhaustively enumerate the elements of the state space \mathcal{S} . Indeed, the acceptance probability in (2) does not require knowledge of Z , since it is equivalently computable as $\alpha(s, s') = \min \left\{ 1, \frac{w(s')q(s|s')}{w(s)q(s'|s)} \right\}$. In particular, if Π is a uniform distribution on \mathcal{S} , then we have

$$\alpha(s, s') = \min \left\{ 1, \frac{q(s|s')}{q(s'|s)} \right\}. \quad (3)$$

This makes the MH algorithm well-suited for our purposes.

Our work is inspired largely by the well-known application of the MCMC method to produce random domino tilings of a square (see e.g., [14]). A *domino tiling* of an $N \times N$ square, N being an even integer, is a tiling using only two basic tiles — a 1×2 horizontal domino and a 2×1 vertical domino. Starting from an arbitrary domino tiling of the $N \times N$ square, pick a 2×2 window uniformly at random. If this 2×2 window is a *domino square*, i.e., a square covered by a pair of horizontal dominos or by a pair of vertical dominos, then perform a *domino flipping* move: rotate the pair of dominos by 90° , as shown in Figure 2. It can be shown (again, see [14]) that this defines an irreducible Markov chain on the state space \mathcal{S} of all domino tilings of the square, with the stationary distribution being the uniform distribution on \mathcal{S} . It is easy to check that this Markov chain can be realized by the MH algorithm by taking the proposal distribution to be

$$q(s'|s) = \begin{cases} \frac{1}{(n-1)^2} & \text{if } s' \in \mathcal{D}_1(s) \\ 1 - \frac{1}{(n-1)^2} |\mathcal{D}_1(s)| & \text{if } s' = s \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{D}_1(s)$ is the set of all domino tilings s' reachable from s via exactly one domino flipping move, and taking the acceptance probability to be $\alpha(s, s') = 1$ for all $s, s' \in \mathcal{S}$.

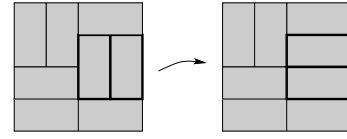


Fig. 2. Domino flipping.

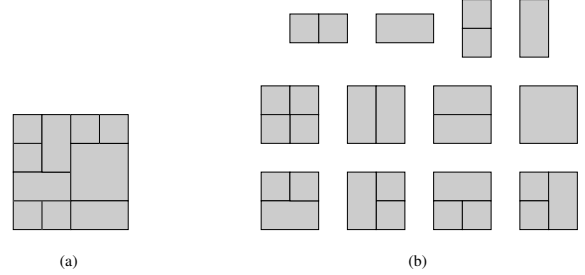


Fig. 3. (a) In the 4×4 tiled square shown, the top-left and bottom-left 2×2 windows are tile-closed rectangles, while the top-right and bottom-right ones are not. (b) All the possible tile-closed rectangles of sizes 1×2 , 2×1 and 2×2 .

III. GENERATING A UNIFORMLY RANDOM TILING USING THE RECTANGULAR TILESET

We denote by \mathcal{S}_N the state space consisting of all the possible tilings of an $N \times N$ square (for some fixed integer $N > 1$) that use only tiles from the rectangular tileset depicted in Figure 1. Determining the exact size of \mathcal{S}_N is a difficult problem. Indeed, even if we dropped the 2×2 tile from the rectangular tileset of Figure 1, counting the number of tilings of an $N \times N$ square using the reduced tileset is a famous open problem of statistical mechanics — see e.g., [15]. For even N , a crude lower bound on the size of \mathcal{S}_N is given by the number of domino tilings of the $N \times N$ square, for which an exact formula is known [16]. From this, we obtain $|\mathcal{S}_N| \gtrsim (1.338515 \dots)^{N^2}$ for even N . For odd N , we could bound $|\mathcal{S}_N|$ from below by the number of domino tilings of the $(N-1) \times N$ rectangle, which, again by Kasteleyn's formula [16], is approximately $(1.338515 \dots)^{N(N-1)}$. These bounds show that $|\mathcal{S}_N|$ grows exponentially in N^2 , which makes listing out all the tilings in \mathcal{S}_N highly impractical for all but a few small values of N . This is precisely the kind of situation in which the MH algorithm is most useful.

A. The MH Algorithm for Sampling from the Uniform Distribution on \mathcal{S}_N

Given a tiling $s \in \mathcal{S}_N$, define an $m \times n$ *tile-closed rectangle* to be an $m \times n$ window within s with the property that no tile of s crosses the boundary of the window. Figure 3(a) shows examples of 2×2 windows within a 4×4 tiled square that are tile-closed rectangles and others that are not. Figure 3(b) lists all the possible tile-closed rectangles of sizes 1×2 , 2×1 and 2×2 .

Recall that in the domino flipping move described in the previous section (see Figure 2), one domino square is replaced by another. We extend that idea to our present context by defining a collection of *valid local moves*. Each valid local

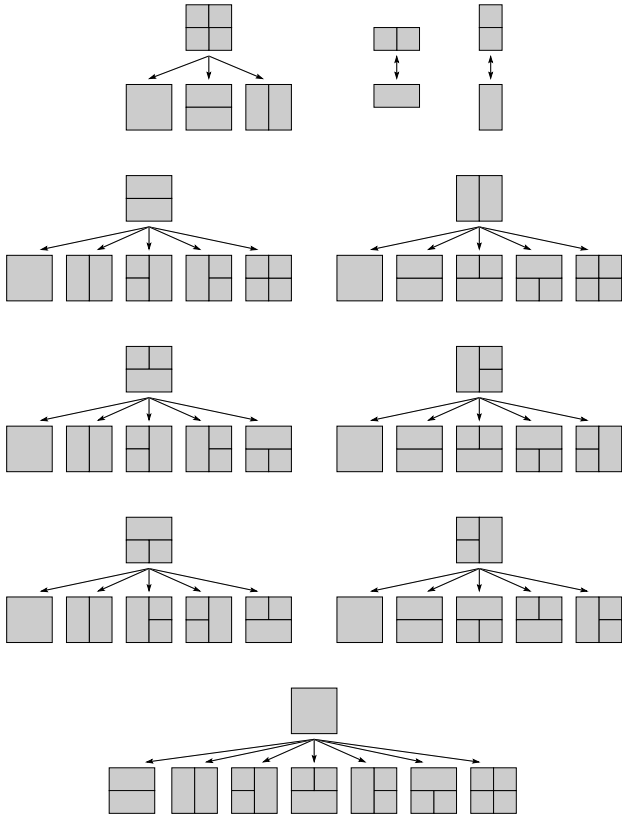


Fig. 4. Valid local moves for tilings in \mathcal{S}_N .

move involves replacing one tile-closed rectangle in a tiling $s \in \mathcal{S}_N$ by another tile-closed rectangle of the same size. To facilitate our description, the tile-closed rectangle to be replaced will be called the *object*, while the rectangle that replaces the object will be called the *target*. The specific collection of valid local moves that we use is depicted in Figure 4. Note that these moves only involve the tile-closed rectangles shown in Figure 3(b).

For a given tile-closed rectangle R , let $\text{Tgt}(R)$ denote the set of all targets of valid moves starting from R . Thus, if R is one of the 1×2 or 2×1 rectangles, then $\text{Tgt}(R)$ consists solely of the other rectangle of the same size. Also, from Figure 4, we see that if R consists of four 1×1 tiles, then $|\text{Tgt}(R)| = 3$; if R is a 2×2 rectangle consisting of at least one domino tile (1×2 or 2×1), then $|\text{Tgt}(R)| = 5$; and finally, if R consists of the 2×2 tile, then $|\text{Tgt}(R)| = 7$.

We are now in a position to define our proposal transition probabilities $q(\cdot)$. Given an $s \in \mathcal{S}_N$, let τ_s denote the total number of 1×2 , 2×1 and 2×2 tile-closed rectangles in s . Moreover, define $\mathcal{M}_1(s)$ to be the set of all $s' \in \mathcal{S}_N$ that can be obtained from s using exactly one valid local move. Note that for each $s' \in \mathcal{M}_1(s)$, there is a unique valid local move that takes s to s' . This move is identified by the smallest window within which s and s' contain different tile-closed rectangles. Thus, for $s' \in \mathcal{M}_1(s)$, let $R_{s \rightarrow s'}$ denote the (unique) object rectangle in s that gets converted to a target rectangle in s' by the (unique) valid local move that takes s

to s' . We then set

$$q(s'|s) = \begin{cases} \frac{1}{\tau_s |\text{Tgt}(R_{s \rightarrow s'})|} & \text{if } s' \in \mathcal{M}_1(s) \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

To complete our description of the MH algorithm, we need to define the acceptance probability $\alpha(s, s')$ for all $s' \in \mathcal{M}_1(s)$. Since we want the stationary distribution to be uniform, $\alpha(s, s')$ is given by (3). Note that, by construction, $p(s'|s) = \alpha(s, s')q(s'|s) > 0$ for all $s' \in \mathcal{M}_1(s)$. In other words, for any $s \in \mathcal{S}_N$, each valid local move acting on s has a positive probability of occurring.

Lemma III.1. *Let $p(s'|s)$, $\alpha(s, s')$ and $q(s'|s)$ be defined by (1), (3) and (4), respectively. Then, the Markov chain having transition probability matrix $P = (p(s'|s))_{s, s' \in \mathcal{S}_N}$ is irreducible and aperiodic.*

Proof: Let $s^* \in \mathcal{S}_N$ denote the tiling consisting of 1×1 tiles only. Note that for any $s \in \mathcal{S}_N$, $s \neq s^*$, there is a sequence of valid local moves that converts s to s^* . This is because valid local moves allow us to replace any 1×2 or 2×1 tile by two 1×1 tiles, and any 2×2 tile by four 1×1 tiles. Moreover, reversing such a sequence of valid local moves would take us from s^* back to s . Thus, for any $s, s^\dagger \in \mathcal{S}_N$, there is a sequence of valid local moves that takes us from s to s^\dagger . Since each valid local move has a positive probability of occurring, the Markov chain defined by P is irreducible.

To show aperiodicity, it is enough to check that $p(s^*|s^*) > 0$. Let k be the total number of 1×2 and 2×1 tile-closed rectangles in s^* , and ℓ be the number of 2×2 tile-closed rectangles in s^* . Thus, $\tau_{s^*} = k + \ell$. It is easy to verify that $k = 2N(N-1)$ and $\ell = (N-1)^2$, but for our purposes, we only need that $\ell > 0$. We then have

$$\begin{aligned} \sum_{s' \in \mathcal{M}_1(s^*)} \alpha(s, s')q(s'|s) &\leq \sum_{s' \in \mathcal{M}_1(s^*)} q(s'|s) \\ &= \frac{1}{k + \ell} \sum_{s' \in \mathcal{M}_1(s^*)} \frac{1}{|\text{Tgt}(R_{s^* \rightarrow s'})|}. \end{aligned} \quad (5)$$

Since $R_{s^* \rightarrow s'}$ is a tile-closed rectangle in s^* , it is composed of 1×1 tiles only. Thus, from Figure 4, we note that $|\text{Tgt}(R_{s^* \rightarrow s'})| = 1$ if $R_{s^* \rightarrow s'}$ is of size 1×2 or 2×1 , and $|\text{Tgt}(R_{s^* \rightarrow s'})| = 3$ if $R_{s^* \rightarrow s'}$ is of size 2×2 . Thus, each of the k tile-closed rectangles of size 1×2 or 2×1 in s^* contributes 1 to the sum in (5), and each of the ℓ tile-closed rectangles of size 2×2 in s^* contributes $\frac{1}{3}$ to the sum. Hence,

$$\sum_{s' \in \mathcal{M}_1(s^*)} \frac{1}{|\text{Tgt}(R_{s^* \rightarrow s'})|} = k + \frac{1}{3}\ell.$$

Plugging this into (5), we obtain

$$\sum_{s' \in \mathcal{M}_1(s^*)} \alpha(s, s')q(s'|s) \leq \frac{1}{k + \ell} \left(k + \frac{1}{3}\ell\right),$$

which is strictly less than 1, since $\ell > 0$. Therefore, $p(s^*|s^*) = 1 - \sum_{s' \in \mathcal{M}_1(s^*)} \alpha(s, s')q(s'|s) > 0$. ■

As a consequence of the above lemma, we have that the Markov chain $(X_n)_{n \geq 0}$ with P as its transition probability

matrix has a unique stationary distribution, and moreover, for any choice of the initial distribution on X_0 , the Markov chain converges to the stationary distribution. By our choice of P , the stationary distribution is the uniform distribution, which we will denote by Υ_N , on \mathcal{S}_N . Thus, the MH algorithm, when run for a long enough time T , will produce a sample drawn from a distribution close (in terms of the total variation distance) to Υ_N . We will comment upon how large T must be to guarantee this in Section V.

The MH algorithm for sampling a (nearly) uniformly random tiling from \mathcal{S}_N is summarized in Algorithm 1' below. The initial state s_0 can be chosen arbitrarily. For example, it can be taken to be the state s^* composed of 1×1 tiles only.

Algorithm 1' The MH algorithm with $q(\cdot|\cdot)$ given by (4) and $\alpha(\cdot, \cdot)$ defined by (3).

Initialization: $X_0 \leftarrow s_0$

for $n = 0$ to $T - 1$ **do**

If $X_n = s$, generate an $s' \in \mathcal{M}_1(s)$ according to the proposal distribution $q(\cdot|s)$

Set $X_{n+1} = \begin{cases} s' & \text{with probability } \alpha(s, s') \\ s & \text{with probability } 1 - \alpha(s, s') \end{cases}$

end for

return X_T

B. Analysis of Computational Complexity

For a fixed T (and fixed $N > 1$), the computational complexity of Algorithm 1' can be easily analyzed. The first step within each iteration of the **for** loop involves sampling, for a given $s \in \mathcal{S}_N$, an $s' \in \mathcal{M}_1(s)$ according to the proposal distribution $q(\cdot|s)$ defined by (4). This can be done as follows:

- 1) Identify the set \mathcal{T}_s of all 1×2 , 2×1 and 2×2 tile-closed rectangles in s .
- 2) Select an R uniformly at random from \mathcal{T}_s .
- 3) Obtain s' by changing R in s to a target R' chosen uniformly at random from $\text{Tgt}(R)$.

Steps 2) and 3) above can be executed in $O(1)$ time using a random number generator. Step 1) can be carried out by scanning through the $N \times N$ tiled square s , a process that requires $O(N^2)$ time. Note, however, that a scan of the entire $N \times N$ square only needs to be performed in the first iteration ($n = 0$) of the **for** loop. For each subsequent iteration ($n = 1, \dots, T - 1$), the computation of the set \mathcal{T}_s can actually be carried out in $O(1)$ time in the *previous* iteration itself, by means of the following extra step:

- 4) Modify \mathcal{T}_s to obtain $\mathcal{T}_{s'}$ via a scan of s' in the immediate neighbourhood of R' .

This extra step is useful because the s in Iteration n is either the same as the s in Iteration $n - 1$, or is equal to the $s' \in \mathcal{M}_1(s)$ determined in Iteration $n - 1$. In summary, the first step within the **for** loop in Algorithm 1' requires $O(N^2)$ time for the first iteration of the loop, and $O(1)$ time for each subsequent iteration.

The second step within the **for** loop requires computation of $\alpha(s, s')$. But this is trivial once $\tau_s = |\mathcal{T}_s|$, $\tau_{s'} = |\mathcal{T}_{s'}|$, R

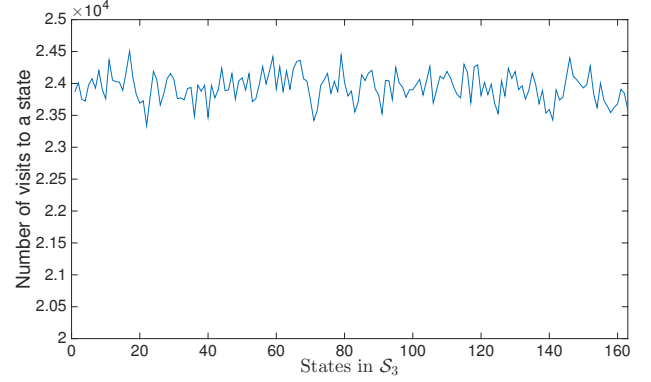


Fig. 5. Plot of the number of times each of the 163 states in \mathcal{S}_3 is visited in one run of the MH Markov chain for 3.9 million time steps.

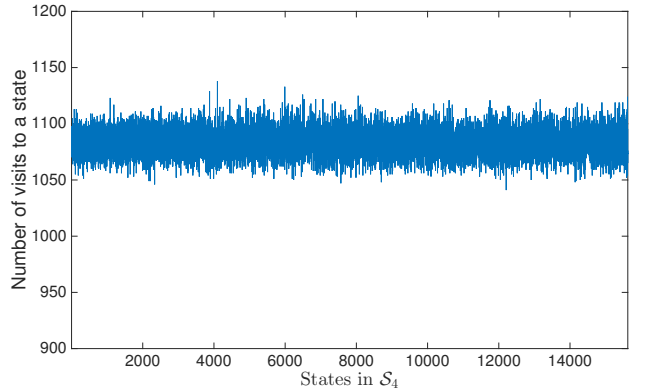


Fig. 6. Plot of the number of times each of the 15623 states in \mathcal{S}_4 is visited in one run of the MH Markov chain for 17 million time steps.

and R' are known from Steps 1)–4) above:

$$\alpha(s, s') = \min \left\{ 1, \frac{\tau_s |\text{Tgt}(R)|}{\tau_{s'} |\text{Tgt}(R')|} \right\}.$$

Thus, overall, Algorithm 1' has a run-time of $O(N^2) + (T - 1)O(1) = O(N^2) + O(T)$.

C. Simulation results

For $N = 3$ and $N = 4$, we were able to exhaustively list out all the tilings in \mathcal{S}_N by means of a computer search. It turns out that there are 163 tilings in \mathcal{S}_3 and 15623 in \mathcal{S}_4 . For the state space \mathcal{S}_3 , we ran the Markov chain described in Section III-A for 3.9 million time steps ($T = 3.9 \times 10^6$ in Algorithm 1'). Figure 5 plots the number of times each of the 163 states was visited during one run of the Markov chain. For \mathcal{S}_4 , we ran the Markov chain for 17 million time steps, and Figure 6 shows the corresponding plot. The plots show that each state in the state space is visited approximately the same number of times. This is consistent with what one would expect the long-term behaviour of an ergodic Markov chain, whose stationary distribution is uniform on the state space, to be.

It should be said that these plots took a long time to generate. They were generated by a Matlab code running on a standard desktop computer. The code for generating the plot in Figure 5 ran for about a week, while that for the plot in Figure 6 ran for considerably longer. No attempt was made to optimize either the Matlab code or the choices being made in the MH algorithm described in this section. We expect that better implementation as well as cleverer choices of the local transformations and the proposal transition probabilities should reduce the runtime needed considerably.

IV. GENERATING A RANDOM TILING WITH FIXED TILE FREQUENCIES

As discussed in the Introduction, in many practical situations, one would like to specify at the outset a desired relative frequency of occurrence of each tile from the tileset, and then generate a random tiling of an $N \times N$ square in which each basic tile has the desired relative frequency [2], [5], [8]. In this section, we briefly sketch how this can be accomplished using the MH algorithm.

We define the *type* of a tiling $s \in \mathcal{S}_N$ to be the vector $[a_1 \ a_2 \ a_3 \ a_4]$, where a_1 , a_2 , a_3 and a_4 are the numbers of 1×1 , 1×2 , 2×1 and 2×2 tiles, respectively, in s . Considering the area covered by each basic tile, we have

$$a_1 + 2a_2 + 2a_3 + 4a_4 = N^2. \quad (6)$$

Now, suppose that in a tiling, we would like the relative frequency of occurrence of tiles of size 1×1 , 1×2 , 2×1 and 2×2 to be p_1 , p_2 , p_3 and p_4 , respectively. For this to happen, the type of the tiling must satisfy $p_i = \frac{a_i}{\sum_i a_i}$ for $i = 1, 2, 3, 4$. These conditions along with (6) yield the unique solution

$$a_i = \frac{N^2 p_i}{p_1 + 2p_2 + 2p_3 + 4p_4}, \quad i = 1, 2, 3, 4.$$

Since the right-hand side above need not be an integer, some rounding to integer values may be required, keeping in mind that (6) must be satisfied. This yields a type vector $\mathbf{a} = [a_1 \ a_2 \ a_3 \ a_4]$ such that in any tiling of type \mathbf{a} , the individual tiles from the tileset occur with (approximately) the desired relative frequencies. Thus, what we require is a means of sampling uniformly from the subset of \mathcal{S}_N consisting of tilings of fixed type \mathbf{a} . This can be done via the MH algorithm.

Given a type vector \mathbf{a} , let us denote by $\mathcal{S}_N(\mathbf{a})$ the subset of \mathcal{S}_N consisting of tilings of type \mathbf{a} . We describe here a set of local moves and corresponding proposal transition probabilities $q(\cdot)$ that, when used in the MH algorithm, will allow us to sample uniformly at random from a large subset of (but perhaps not all of) $\mathcal{S}_N(\mathbf{a})$. Note that there may be other choices of local moves that yield similar or even better results. We make some comments on this at the end of this section.

Figure 7 shows the local moves we use on tilings in $\mathcal{S}_N(\mathbf{a})$. As was the case with the local moves defined in Figure 4, the objects and targets of the local moves in Figure 7 are tile-closed rectangles. Note that these local moves simply rearrange the tiles in each tile-closed rectangle, thus preserving the type of the overall tiling.

We set the proposal transition probabilities $q(\cdot)$ in a manner analogous to (4). For $s \in \mathcal{S}_N(\mathbf{a})$, let $\tau_s^{\mathbf{a}}$ denote the

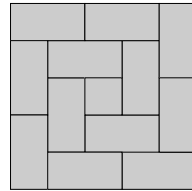


Fig. 8. An isolated state in $\mathcal{S}_5([1 \ 6 \ 6 \ 0])$.

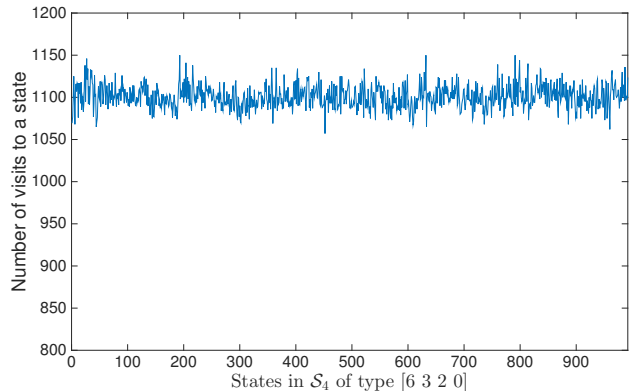


Fig. 9. Plot of the number of times each of the 992 states in $\mathcal{S}_4([6 \ 3 \ 2 \ 0])$ is visited in one run of the MH Markov chain for 1 million time steps.

total number of occurrences in s of the tile-closed rectangles depicted in Figure 7. Also, let $\mathcal{M}_1^{\mathbf{a}}(s)$ denote the set of all $s' \in \mathcal{S}_N(\mathbf{a})$ that can be obtained from s using exactly one of the local moves in Figure 7. With this, we define

$$q(s'|s) = \begin{cases} \frac{1}{\tau_s^{\mathbf{a}}} & \text{if } s' \in \mathcal{M}_1^{\mathbf{a}}(s) \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

As before, the acceptance probability $\alpha(s, s')$, for $s' \in \mathcal{M}_1^{\mathbf{a}}(s)$, is defined as in (3).

The initialization of the MH algorithm can be done in an arbitrary fashion. One could, for example, take the initial state s_0 to be a tiling in $\mathcal{S}_N(\mathbf{a})$ generated by the simple algorithm described in [5, Section II-A].

One issue with the MH algorithm in this case is that the concomitant Markov chain on the state space $\mathcal{S}_N(\mathbf{a})$ need not be irreducible. Indeed, for certain type vectors \mathbf{a} , the state space $\mathcal{S}_N(\mathbf{a})$ contains some isolated states that are not reachable from any other state, as they contain none of the tile-closed rectangles shown in Figure 7 — one such example is shown in Figure 8. However, such states are typically only a small subset of the overall state space $\mathcal{S}_N(\mathbf{a})$, and the MH Markov chain traverses most of $\mathcal{S}_N(\mathbf{a})$. As an illustrative example, the plot in Figure 9 shows that the MH Markov chain, when run long enough, visits all the 992 states in $\mathcal{S}_4([6 \ 3 \ 2 \ 0])$ roughly the same number of times.

In principle, it is possible to mitigate the problem of isolated states identified above by expanding the set of local moves to include, for example, 5×5 tile-closed rectangles as objects and targets. Indeed, in general, the richer the collection of local moves, the more effective will be the MH Markov chain's exploration of the state space, potentially lowering the mixing

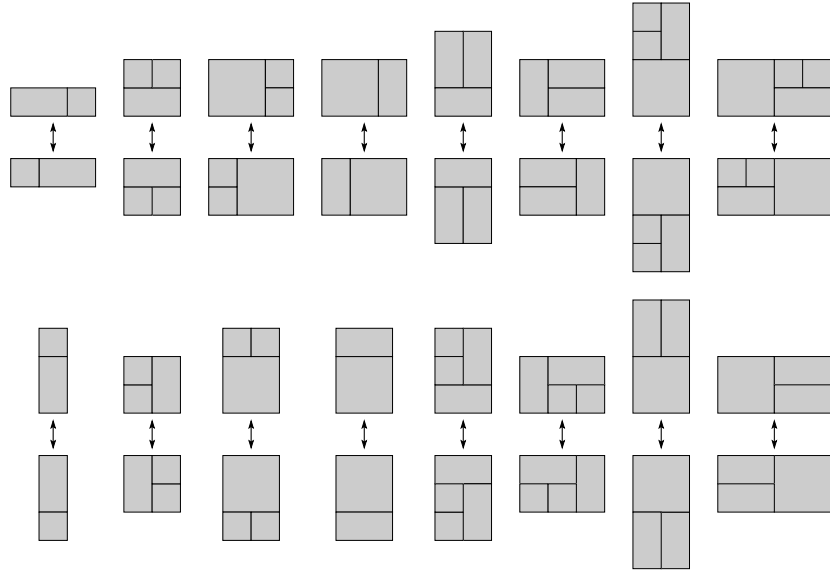


Fig. 7. Local moves for tilings of a fixed type.

time of the chain. However, this generally comes at the cost of making the proposal probability distribution $q(\cdot|s)$ more complex, which increases the computational complexity of sampling from this distribution. Thus, the collection of local moves should be chosen so as to strike a balance between a faster and more thorough exploration of the state space, on the one hand, and an increase in the computational complexity of sampling from the proposal distribution, on the other. Our choice of local moves, shown in Figure 7, has been made keeping this balance in mind.

V. CONVERGENCE AND MIXING TIME

We now turn to the question of how long (in terms of the number of iterations of the main **for** loop) an MH algorithm needs to be run in order to guarantee that it will produce a sample from a distribution close, in a sense that we will make precise below, to the desired stationary distribution. To keep our discussion concrete, we will address this question specifically for Algorithm 1' described in Section III-A. The methods we use can also be applied to the MH algorithm in Section IV, but we do not do so here.

We begin by introducing some notation to be used in this section. Let P be the transition probability matrix of a time-homogeneous Markov chain (X_n) with state space \mathcal{S} . For $s, s' \in \mathcal{S}$, we use $P(s, s')$ to denote the (s, s') th entry of P , namely, the one-step transition probability $p(s'|s) = \Pr[X_1 = s' | X_0 = s]$. Then, for any positive integer n , the (s, s') th entry of the matrix P^n , which we denote by $P^n(s, s')$, is the n -step transition probability $\Pr[X_n = s' | X_0 = s]$. Thus, the s th row of P^n , denoted by $P^n(s, \cdot)$, represents the probability distribution of X_n given $X_0 = s$.

As is well known from Markov chain theory, an irreducible and aperiodic Markov chain (X_n) on a finite state space \mathcal{S} converges to its (unique) stationary distribution $\Pi = (\Pi(s))_{s \in \mathcal{S}}$. What this convergence precisely means is that if P is the transition probability matrix of the Markov chain (X_n) , then

every row of the matrix P^n converges in total variation distance to Π : $\lim_{n \rightarrow \infty} \|P^n(s, \cdot) - \Pi\|_{\text{TV}} = 0$ for all $s \in \mathcal{S}$. The convergence here is in fact exponentially fast. Indeed, let $r > 0$ be such that all entries of P^r are strictly positive; such an r always exists under the irreducible and aperiodic assumption. Let $\delta > 0$ be such that $P^r(s, s') \geq \delta \Pi(s')$ for all $s, s' \in \mathcal{S}$; since \mathcal{S} is finite, such a δ exists. Then (see proof of [17, Theorem 4.9]), for any integers $k \geq 1$ and $j \geq 0$,

$$\|P^{rk+j}(s, \cdot) - \Pi\|_{\text{TV}} \leq (1 - \delta)^k. \quad (8)$$

Precise knowledge of r and the largest possible choice of δ for that r would allow us to obtain a good estimate of the rate of convergence to the stationary distribution. In particular, it would allow us to get an estimate, via (8), of the *mixing time*, $t_{\text{mix}}(\epsilon)$, of the Markov chain, which for an arbitrarily small $\epsilon > 0$ is defined as [17, Section 4.5]

$$t_{\text{mix}}(\epsilon) = \min\{n : \max_{s \in \mathcal{S}} \|P^n(s, \cdot) - \Pi\|_{\text{TV}} \leq \epsilon\}. \quad (9)$$

This definition makes precise the notion of how long the Markov chain (X_n) must be run so that, irrespective of the starting state X_0 , the distribution of X_n is ϵ -close in total variation distance to the stationary distribution. Thus, in Algorithms 1 and 1', we should set $T = t_{\text{mix}}(\epsilon)$, for some suitably small $\epsilon > 0$, to be assured that the output X_T is sampled from a distribution close to the stationary distribution of the MH Markov chain. For a Markov chain satisfying (8), if we set $k = \left\lceil \frac{\log \epsilon}{\log(1-\delta)} \right\rceil$, then we have $\|P^{rk}(s, \cdot) - \Pi\|_{\text{TV}} \leq \epsilon$, and hence, $t_{\text{mix}}(\epsilon) \leq r \left\lceil \frac{\log \epsilon}{\log(1-\delta)} \right\rceil$.

For the Markov chain associated with Algorithm 1' and defined on the state space \mathcal{S}_N , we can take $r = N^2$, as shown by the lemma below.

Lemma V.1. *Let $p(s'|s)$, $\alpha(s, s')$ and $q(s'|s)$ be defined by (1), (3) and (4), respectively. Then, for the transition probability matrix $P = (p(s'|s))_{s, s' \in \mathcal{S}_N}$, all the entries of P^{N^2} are strictly positive.*

Proof: The proof is a refinement of the proof of Lemma III.1. As in that proof, let $s^* \in \mathcal{S}_N$ denote the tiling consisting of 1×1 tiles only. We have already shown there that $p(s^*|s^*) > 0$, from which we infer that $P^n(s^*, s^*) > 0$ for all $n \geq 1$.

Now, given $s, s' \in \mathcal{S}$, let d and d' be total number of tiles in s and s' , respectively, that are not of size 1×1 . Note that $d, d' \leq N^2/2$, so that $d + d' \leq N^2$. The argument in the proof of Lemma III.1 shows that s can be converted to s^* by a sequence of d valid local moves, and s^* can be converted to s' by a sequence of d' valid local moves. Thus, $P^d(s, s^*)P^{d'}(s^*, s') > 0$, and hence, $P^{N^2}(s, s') \geq P^d(s, s^*)P^{N^2-(d+d')}(s^*, s^*)P^{d'}(s^*, s') > 0$. ■

So, for the Markov chain on \mathcal{S}_N associated with Algorithm 1', we may take $r = N^2$ and

$$\delta = \min_{s, s'} \frac{P^{N^2}(s, s')}{\Pi(s')} = |\mathcal{S}_N| \left(\min_{s, s'} P^{N^2}(s, s') \right),$$

since the stationary distribution in this case is the uniform distribution on \mathcal{S}_N . With this, as argued above, we obtain the bound

$$t_{\text{mix}}(\epsilon) \leq N^2 \left\lceil \frac{\log \epsilon}{\log(1 - \delta)} \right\rceil.$$

Unfortunately, it seems to be difficult to derive from this a good upper bound on the mixing time expressible as a function of ϵ and N alone. We therefore look for some alternative approaches.

Many techniques have been established in the literature for bounding the mixing time of a Markov chain (see, e.g., [17]). We attempted to use two techniques that have been previously applied to MCMC algorithms for tiling problems. One was the ‘‘canonical paths’’ method developed by Sinclair [18], [19]. Applying this technique to our Markov chain, we could only obtain an $O(N^2|\mathcal{S}_N| \log(|\mathcal{S}_N|/\epsilon))$ upper bound on $t_{\text{mix}}(\epsilon)$. We refer the reader to the Master’s thesis of the first author [20, Section 7.2] for the details of this approach. Given that $|\mathcal{S}_N| \geq c^{N^2}$ for some constant $c > 1$, we believe that this bound is overly pessimistic.

We in fact conjecture that the mixing time of the Markov chain under consideration is $\text{poly}(N)$. Our conjecture is based in part on empirical evidence gathered from our observations of the running time of Algorithm 1', and in part on analysis using a coupling time approach described in [14]. Using this approach, it was shown in [14, Theorem 4.3] that the mixing time for the MCMC method for domino tilings described in Section II is at most $8eN^8 \lceil \ln \epsilon^{-1} \rceil$. While we were unable to harness this method to bound the mixing time of the Markov chain associated with Algorithm 1', we were able to use it to analyze a class of Markov chains on \mathcal{S}_N with a certain non-uniform stationary distribution. To be a little more precise, we designed an MH Markov chain on the state space, $\hat{\mathcal{S}}_N$, of $N \times N$ tilings composed of three basic tiles only — 1×1 , 1×2 and 2×1 . The designed stationary distribution was $\Pi_\gamma(s) = \frac{1}{Z_\gamma} \gamma^{b(s)}$ for $s \in \hat{\mathcal{S}}_N$, where $\gamma \in [0, 1]$ is a constant, $b(s)$ is the total number of 1×2 and 2×1 tiles in s , and $Z_\gamma = \sum_{s \in \hat{\mathcal{S}}_N} \gamma^{b(s)}$ is the normalization constant. Observe that when $\gamma = 1$, Π_γ is the uniform distribution on $\hat{\mathcal{S}}_N$. For

$\gamma \leq 1/3$, we could use the coupling time method from [14] to show that the mixing time $t_{\text{mix}}(\epsilon)$ of the Markov chain is $O(N^8 \ln(\epsilon^{-1}))$. The details of the analysis can be found in [20, Section 7.3].

VI. CONCLUDING REMARKS

In this paper, we presented MCMC methods, based on the MH algorithm, for generating tilings nearly uniformly at random from the set, \mathcal{S}_N , of all tilings of an $N \times N$ square that use only tiles of sizes 1×1 , 1×2 , 2×1 and 2×2 . We also showed how these methods could be adapted to generate almost uniformly random samples from a subset of \mathcal{S}_N consisting of only those tilings in which the four basic tiles occur with given relative frequencies. The underlying Markov chains in these algorithms are shown to converge rapidly (exponentially quickly in N), although we are unable to pin down the precise rate of convergence or mixing time of these chains. We leave it as a conjecture that the mixing time is polynomial in N .

While our focus has largely been on sampling from a uniform distribution on tilings, it ought to be clear from the description in Section II that the MH algorithm can be used to sample from an arbitrary distribution Π on \mathcal{S}_N . The only modification to be made in Algorithm 1' is to the acceptance probability, which should now follow (2) rather than (3). The method also carries over easily to tilings of regions other than squares.

One major drawback of our MCMC algorithms is that they may be prohibitively expensive from a computational perspective. To be assured that the MH algorithm produces a sample from a distribution close to the target distribution, it should be run for a length of time T that is at least as large as the mixing time of the associated Markov chain. Also, note that the samples produced at times T and $T + 1$ are not independent — their joint probabilities are governed by the transition probability matrix of the Markov chain. If we want almost independent samples, then they must again be spread apart by a length of time of the order of the mixing time. While we believe that the mixing time is a polynomial in N , it could be a polynomial of relatively large degree, say $O(N^8)$ or higher, which will get unreasonably large even for moderate values of N , say, $N > 10$.

Nevertheless, one can still use the MCMC method to add an extra level of randomization to a tiling that is generated using some ad-hoc method of low complexity. One could initialize the MH algorithm with a tiling generated by a simpler algorithm, such as the one described in [5, Section II-A], and then run the Markov chain for some reasonable number of time steps till one gets a tiling with an extra sprinkling of randomness to suit one’s taste.

REFERENCES

- [1] R. Wood, M. Williams, A. Kavcic and J. Miles, ‘‘The feasibility of magnetic recording at 10 Terabits per square inch on conventional media,’’ *IEEE Trans. Magn.*, vol. 45, no. 2, pp. 917–923, Feb. 2009.
- [2] A. R. Krishnan, R. Radhakrishnan and B. Vasic, ‘‘Read channel modeling for detection in two-dimensional magnetic recording systems,’’ *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3679–3682, Oct. 2009.

- [3] K. S. Chan, J. J. Miles, E. Hwang, B. V. K. Vijayakumar, J. G. Zhu, W. C. Lin and R. Negi, "TDMR platform simulations and experiments," *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3837–3843, Oct. 2009.
- [4] K. S. Chan, R. Radhakrishnan, K. Eason, M. R. Elidrissi, J. J. Miles, B. Vasic and A. R. Krishnan, "Channel models and detectors for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 46, no. 3, pp. 804–811, March 2010.
- [5] A. Kavcic, X. Huang, B. Vasic, W. Ryan and M. F. Erden, "Channel modeling and capacity bounds for two-dimensional magnetic recording," *IEEE Trans. Magn.*, vol. 46, no. 3, pp. 812–818, March 2010.
- [6] A. R. Iyengar, P. H. Siegel, and J. K. Wolf, "Write channel model for bit-patterned media recording," *IEEE Trans. Magn.*, vol. 47, no. 1, pp. 35–45, Jan. 2011.
- [7] S. M. Khatami and B. Vasic, "Generalized belief propagation detector for TDMR microcell model," *IEEE Trans. Magn.*, vol. 49, no. 7, pp. 3699–3702, July 2013.
- [8] L. Pan, W.E. Ryan, R. Wood and B. Vasic, "Coding and detection for rectangular-grain TDMR models," *IEEE Trans. Magn.*, vol. 47, no. 6, pp. 1705–1711, June 2011.
- [9] M. Khatami, V. Ravanmehr and B. Vasic, "GBP-based detection and symmetric information rate for rectangular-grain TDMR model," in *Proc. 2014 IEEE Int. Symp. Inf. Theory (ISIT 2014)*, Honolulu, Hawaii, USA, June 29–July 4, 2014, pp. 1618–1622.
- [10] P. G. Hoel, S. C. Port and C. J. Stone, *Introduction to Stochastic Processes*, Houghton-Mifflin Company, Boston, 1972.
- [11] *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. L. Jones and X.-L. Meng (Eds.), Chapman and Hall/CRC Press, 2011.
- [12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equations of state calculations by fast computing machines," *J. Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [13] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [14] M. Luby, D. Randall and A. Sinclair, "Markov chain algorithms for planar lattice structures," *SIAM J. Computing*, vol. 31, pp. 167–192, 2001.
- [15] C. Kenyon, D. Randall and A. Sinclair, "Approximating the number of monomer-dimer coverings of a lattice," *J. Stat. Phys.*, vol. 83, pp. 637–659, 1996.
- [16] P.W. Kasteleyn, "The statistics of dimers on a lattice. I. The number of dimer arrangements on a quadratic lattice," *Physica*, vol. 27, pp. 1209–1225, 1961.
- [17] D.A. Levin, Y. Peres and E.L. Wilmer, *Markov Chains and Mixing Times*, American Mathematical Society, 2009.
- [18] A. J. Sinclair, *Randomised Algorithms for Counting and Generating Combinatorial Structures*, PhD Thesis CST-58-88, Dept. Computer Science, University of Edinburgh, Nov. 1988.
- [19] A. J. Sinclair, "Improved bounds for mixing rates of Markov chains and multicommodity flow," *Combinatorics, Probability and Computing*, vol. 1, pp. 351–370, 1992.
- [20] Abhinav Das N. V., *Sampling from a Target Probability Distribution on a Two Dimensional Magnetic Recording Medium*, M. E. Project Report, Dept. Electrical Communication Engineering, Indian Institute of Science, June 2015.



Abhinav Das N. V. received the B. Tech. degree in Electronics and Communication Engineering from the Govt. College of Engineering, Kannur, in 2011, and the M. Tech. degree in Telecommunication Engineering from the Indian Institute of Science, Bangalore in 2015. In December 2015, he joined the Central Research Laboratory, Bharat Electronics Ltd., Bangalore, as a Member-Research Staff. His research interests lie in the areas of communication networks and coding theory.

Navin Kashyap (S'97-M'02-SM'07) received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Bombay, in 1995, the M.S. degree in Electrical Engineering from the University of Missouri-Rolla in 1997, and the M.S. degree in Mathematics and the Ph.D. degree in Electrical Engineering from the University of Michigan, Ann Arbor, in 2001. From November 2001 to November 2003, he was a postdoctoral research associate at the University of California, San Diego. From 2004 to 2010, he was at the Department of Mathematics and Statistics at Queen's University, Kingston, Ontario, first as an Assistant Professor, then as an Associate Professor. In January 2011, he joined the Department of Electrical Communication Engineering at the Indian Institute of Science as an Associate Professor. His research interests lie primarily in the application of combinatorial and probabilistic methods in information and coding theory. Prof. Kashyap served on the editorial board of the IEEE Transactions on Information Theory during the period 2009–2014.