Random Walks on Combinatorial Objects

Martin Dyer and Catherine Greenhill

Summary Approximate sampling from combinatorially-defined sets, using the Markov chain Monte Carlo method, is discussed from the perspective of combinatorial algorithms. We also examine the associated problem of discrete integration over such sets. Recent work is reviewed, and we re-examine the underlying formal foundational framework in the light of this. We give a detailed treatment of the coupling technique, a classical method for analysing the convergence rates of Markov chains. The related topic of perfect sampling is examined. In perfect sampling, the goal is to sample exactly from the target set. We conclude with a discussion of negative results in this area. These are results which imply that there are no polynomial time algorithms of a particular type for a particular problem.

1 Introduction

The focus of this paper is approximate sampling and approximate counting (or approximate integration), using the Markov chain Monte Carlo (MCMC) method, and viewed from the perspective of combinatorial algorithms. There has been much work in this area in recent years, some of which we survey below in Section 4. We illustrate this work with a closer examination of one particular technique which has proved successful recently, that of *coupling*. This is a classical method from applied probability, but its application in this area has involved some new insights.

Formal foundations for work in this area were provided in the seminal paper of Jerrum, Valiant and Vazirani [50]. However, the subject seems subsequently to have outgrown the framework it provided. The present paper makes a modest attempt to update the situation.

We begin, in Section 2, by fixing notation which we use throughout. In Section 3 we offer some formal definitions of the central concepts, following [50] and [73]. A review of recent developments in the areas of approximate sampling, approximate counting and perfect sampling is given in Section 4.

The coupling method is described in Section 5, and is illustrated on a simple Markov chain for independent sets in graphs. This example forms a running theme throughout the paper. A variant of coupling which has been employed since 1997, called *path coupling*, is presented in Section 6.

Much attention has recently been paid to the topic of "exact" or "perfect" sampling using Markov chains. In Section 7, we explore some aspects of perfect sampling, in particular the relationship between perfect and approximate sampling.

Another area of recent interest has been in showing that the MCMC (or any) technique fails on certain problems. As an illustration, we conclude by

discussing two typical negative results in Section 8. (Some others are included in the survey of Section 4.)

2 Notation and preliminaries

Throughout $\mathbb{N} = \{0, 1, 2, \ldots\}$, $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$, $\mathbb{Q}_+ = \{q \in \mathbb{Q} : q > 0\}$, and $[n] = \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}_+$.

Let $\mathcal{M}$ be a Markov chain on finite state space Ω, with transition matrix P, i.e. if X_t is the state at time t,

$$\Pr(X_t = \sigma \mid X_{t-1} = \omega) = P(\omega, \sigma) \qquad (t = 1, 2, \ldots),$$

which we also write as P^{ω}_{σ}. We assume that $\mathcal{M}$ is ergodic, and therefore has unique stationary distribution π. In most cases of interest, $\mathcal{M}$ is *reversible*, i.e.

$$\pi(\omega)P(\omega, \sigma) = \pi(\sigma)P(\sigma, \omega) \qquad (\forall \omega, \sigma \in \Omega). \tag{1}$$

The central role of reversible chains in applications rests on the fact that π can be deduced from (1). If $\mu : \Omega \to \mathbb{R}$ satisfies (1), then it determines π up to normalization. In fact, we usually *design* the chain to satisfy (1). Without reversibility, there is no apparent method of determining π, other than to explicitly construct the transition matrix, an exponential time (and space) computation in our setting.

If $p_0(\omega) = \Pr(X_0 = \omega)$, then $p_t(\sigma) = \sum_\omega p_0(\omega)P^t(\omega, \sigma)$ is the distribution at time t. As a measure of convergence, the natural choice in this context is *variation distance*,

$$\mathrm{d}_{\mathrm{TV}}(p_t, \pi) = \tfrac{1}{2} \sum_{\omega \in \Omega} |p_t(\omega) - \pi(\omega)| = \max_{A \subseteq \Omega} \left(p_t(A) - \pi(A)\right). \tag{2}$$

[Some authors, e.g. Lindvall [59], define this without the $\frac{1}{2}$.] The *mixing time* of the chain is then

$$\tau(\varepsilon) = \max_{p_0} \min_t \{\mathrm{d}_{\mathrm{TV}}(p_t, \pi) \le \varepsilon\},$$

and it is easy to show that the maximum occurs when $X_0 = \omega_0$, with probability one, for some state ω_0.

For further information on Markov chains, see [2].

Throughout the paper, we refer to the well-known complexity classes P, NP and $\#\mathsf{P}$. For their definitions and further information, see [64].

3 A computational framework

Some foundations for approximate counting and uniform sampling were set out in the influential paper of Jerrum, Valiant and Vazirani [50]. However,

the ideas of that paper have rather more generality than the results contained within it. Also the paper concentrates its attention on the class of *self-reducible* problems, as introduced by Schnorr [72]. The restrictive definition of self-reducibility makes this concept difficult to apply, as we discuss below. For these reasons, we re-examine these foundations, to reflect the substantial body of subsequent work which has used the ideas, rather than the results, of [50]. The reader who is more interested in techniques and applications might prefer to skim this material, at least on first reading.

The sample spaces we have in mind are sets of combinatorial objects. However, in order to discuss the computational complexity of generation, it is necessary to consider a sequence of instances of increasing size. We therefore work within the following formal framework, which extends the ideas of Jerrum, Valiant and Vazirani [50]. The models of computation are the Turing Machine (TM) for deterministic computations and the Probabilistic Turing Machine (PTM) for randomized computations. (A PTM is a TM with a source of uniform and independent random bits.) The following definition generalises that of [50] to allow non-uniform distributions, and is closely related to that of Sinclair [73, pp. 86–87] for the same purpose. We must confine ourselves to some class of distributions which are "easily described", from a computational viewpoint, in large instances. We identify this below with a class of unnormalized measures which we call "weight functions".

Let Σ be a fixed alphabet of at least two symbols, and $W : (\Sigma^*)^2 \to \mathbb{N}$ be such that, for some polynomial b, $W(\sigma, \omega) = 0$ unless $|\omega| \leq b(|\sigma|)$. Moreover $W(\sigma, \omega)$ must be computable in time polynomial in $|\sigma|$ whenever $W(\sigma, \omega) > 0$. (If the TM for W may ignore part of its input, this implies that W is *always* computable in polynomial time.) Let us call W a *weight function*. Here σ may be thought of as an encoding of an instance of some combinatorial problem, and the ω of interest are encodings of the structures we wish to generate.

Let $\Omega_\sigma = \{\omega : W(\sigma, \omega) > 0\}$. Then the sequence of discrete probability spaces determined by W is $(\Omega_\sigma, \pi_\sigma)$, where π_σ is the *density*

$$\pi_\sigma(\omega) = W(\sigma, \omega)/Z(\sigma), \quad \text{with} \quad Z(\sigma) = \sum_{\omega' \in \Omega_\sigma} W(\sigma, \omega')$$

being the corresponding *normalising function*. It is easy to see that the class of normalising functions so defined is essentially Valiant's [76] class #P. The definition implies that, for some fixed $c \in \mathbb{N}$, $|\Omega_\sigma| \leq Z(\sigma) \leq 2^{|\sigma|^c}$. If $Z(\sigma) = 0$, then $\Omega_\sigma = \emptyset$ and π_σ is the unique (improper) measure on Ω_σ.

In our definition, two distinct weight functions may define the same sequence of spaces. Therefore let us say weight functions W_1, W_2 are *equivalent* if there exists $\kappa : \Sigma^* \to \mathbb{Q}_+$ so that $W_2(\sigma, \omega) = \kappa(\sigma) W_1(\sigma, \omega)$ $(\forall \sigma, \omega \in \Sigma^*)$. Then there is a bijection between sequences of probability spaces $(\Omega_\sigma, \pi_\sigma)$ and equivalence classes of weight functions. Thus, if we write $\widetilde{W}$ for the equivalence class containing W, we may identify it with the sequence $(\Omega_\sigma, \pi_\sigma)$.

Example 3.1 Let $G = (V, E)$ be a (simple) graph, with maximum degree Δ and $|V| = n$, and let $\mathcal{I}(G)$ denote the collection of independent sets in G. It is known [75, 38] to be #P-complete to compute $|\mathcal{I}(G)|$ exactly for $\Delta \geq 3$, and is easily shown to be in P for $\Delta < 3$. For given $\lambda > 0$, the *hard-core gas model* (see for example [4]) samples from a density where independent sets of size s have probabilities proportional to λ^s. We use this problem as a running example. In our setting, let $\lambda = r/q$ for integers r, q. Then $W(\sigma, \omega) = 0$ unless σ encodes a graph, and ω an independent set of some size s in G. If $W(\sigma, \omega) \neq 0$, then $W(\sigma, \omega) = q^n \lambda^s = r^s q^{(n-s)} \in \mathbb{N}_+$. ∎

We insist that sample spaces are discrete, and weight functions are integer valued. Computationally, discrete spaces are essential. If we wish to work with continuous spaces, then approximations must be made to some predetermined number of bits. The same is true if we are interested in real-valued densities (as in some statistical applications). However, the effect of such approximations can be absorbed into the variation distance of the sampling procedure. The reader may still wonder why we require W to have codomain $\mathbb{N}$ rather than $\mathbb{Q}$, which would seem more natural. This is because we use unnormalised measures, and we wish to avoid the following technical difficulty. In a large sample space it is possible to specify polynomial size rationals for the unnormalised measure which result in exponential size rationals for the probabilities. An example is the set $[2^n]$, with the measure assigning probability proportional to $1/i$ to $i \in [2^n]$. (See [73, pp. 27–28] for details, where this example is used for a slightly different purpose.) In such spaces there is no possibility of *exact* sampling in sub-exponential expected time, and we must accept approximations. We prefer not to deal with these anomalous spaces, but to insist that these approximations be made explicit. Thus, in this example we could use weights $\lfloor K/i \rfloor$ for some suitably large integer K.

A *fully polynomial approximate sampler* (which we shorten to *good sampler*) for $(\Omega_\sigma, \pi_\sigma)$ is a PTM which, on inputs σ and $\varepsilon \in \mathbb{Q}_+$ $(0 < \varepsilon \leq 1)$, outputs $\omega \in \Sigma^*$, according to a measure μ_σ satisfying $\mathrm{d_{TV}}(\mu_\sigma, \pi_\sigma) \leq \varepsilon$, in time bounded by a bivariate polynomial in $|\sigma|, \log \varepsilon^{-1}$. We allow $\omega \notin \Omega_\sigma$. If $\Omega_\sigma = \emptyset$, the algorithm does not terminate within its time bound. However, this can be detected, and we may construct a polynomial time algorithm which terminates either with a random ω or a proof that Ω_σ is empty.

A good sampler essentially coincides with the *almost uniform generator* of Jerrum, Valiant and Vazirani [50]. Their definition is given in terms of a relation $\mathcal{R}$ rather than a function, so is clearly contained within ours by restricting $W : \Sigma^* \to \{0, 1\}$. However, the following converse is easily proved. Let $(\Omega_\sigma, \pi_\sigma) = \widetilde{W}$ and $\Omega'_\sigma = \{(\omega, i) : \omega \in \Omega_\sigma, 1 \leq i \leq W(\sigma, \omega)\}$. Then there is a good sampler for $(\Omega_\sigma, \pi_\sigma)$ if and only if there is an almost uniform generator for $\mathcal{R} = \{(\sigma, (\omega, i)) : (\omega, i) \in \Omega'_\sigma\}$. So our definition is reducible to that of [50], but it seems more natural to work directly with non-uniform distributions.

Our real interest here is in combinatorial Markov chains, which we define

as follows. Let $M : (\Sigma^*)^3 \to \mathbb{N}$ and define

$$\mathcal{R}_\sigma = \{(\omega, \omega') : M(\sigma, \omega, \omega') > 0\}, \quad \Omega_\sigma = \{\omega : \exists w' \text{ with } (\omega, \omega') \in \mathcal{R}_\sigma\}.$$

Let M have the following properties.

(a) There is a polynomial b such that $|\omega|, |\omega'| \leq b(|\sigma|)$ if $M(\sigma, \omega, \omega') > 0$, and M is computable in time polynomial in $|\sigma|$ whenever $M(\sigma, \omega, \omega') > 0$.

(b) There exist constants $K(\sigma) \in \mathbb{N}_+$, of polynomial size, such that

$$\sum_{\omega' \in \Sigma^*} M(\sigma, \omega, \omega') = K(\sigma) \qquad (\forall \omega \in \Omega_\sigma).$$

(c) The transitive closure of $\mathcal{R}_\sigma$ is $\Omega_\sigma \times \Omega_\sigma$, and for some ω, $(\omega, \omega) \in \mathcal{R}_\sigma$.

(d) Writing $M_\omega(\sigma, \omega') = M(\sigma, \omega, \omega')$ $(\omega \in \Sigma^*)$, it follows from (a) that M_ω is a weight function. We require that there is a good sampler for $\widetilde{M_\omega}$ $(\forall \omega)$.

We call M a *density matrix*, and associate with it a sequence of Markov chains $\mathcal{M}_\sigma = (\Omega_\sigma, P_\sigma)$, with transition matrices

$$P_\sigma(\omega_1, \omega_2) = M(\sigma, \omega_1, \omega_2)/K(\sigma) \qquad (\omega_1, \omega_2 \in \Omega_\sigma).$$

Properties (a) and (c) ensure that $\mathcal{M}_\sigma$ is finite and ergodic. Property (d) ensures that we can efficiently simulate $\mathcal{M}_\sigma$ to a close approximation for any given number of steps. Property (b) ensures that polynomial powers of the transition matrix cannot generate rationals of superpolynomial size, and hence the state probabilities at any polynomial time cannot be rationals of superpolynomial size. We include this property since we do not wish to preclude exact generation using Markov chains. In any case, this condition can always be satisfied to any desired approximation, and is usually satisfied naturally. There is little loss in restricting $K(\sigma)$ to be a power of 2. If any such $K(\sigma)$ exist, it is easy to show that there is a chain with the same stationary distribution and K a power of 2, simply by increasing the "self-loop" probability on all states. Since we are interested in the stationary distribution, we can use this slightly slower chain. Thus we may insist on K being a power of 2 where convenient.

Density matrices M_1, M_2 are *equivalent* if there exists $\kappa : \Sigma^* \to \mathbb{Q}_+$ such that $M_2(\sigma, \omega, \omega') = \kappa(\sigma) M_1(\sigma, \omega, \omega')$ for all $\sigma, \omega, \omega' \in \Sigma^*$. We can identify the equivalence class $\widetilde{M}$ with the sequence $\mathcal{M}_\sigma$. We say that $\mathcal{M}_\sigma$ is a *rapidly mixing Markov chain* if its mixing time $\tau_\sigma(\varepsilon)$ is bounded by a polynomial in $|\sigma|, \log \varepsilon^{-1}$.

If $\mathcal{M}_\sigma$ is a Markov chain sequence, let π_σ denote the stationary distribution of $\mathcal{M}_\sigma$. Then, if W is a weight function, $\mathcal{M}_\sigma$ is a *Monte Carlo Markov chain* (MCMC) for $\widetilde{W}$ if both $\widetilde{W}$, $\mathcal{M}_\sigma$ determine the same sequence of probability

spaces $(\Omega_\sigma, \pi_\sigma)$. (This slight overloading of the MCMC abbreviation should not cause confusion.) The usual way to establish this is by reversibility, i.e. if $W(\sigma,\omega)M(\sigma,\omega,\omega') = W(\sigma,\omega')M(\sigma,\omega',\omega)$ for all $\sigma \in \Sigma^*$ and $\omega, \omega' \in \Omega_\sigma$. Clearly we have a good sampler for $\widetilde{W}$ if $\mathcal{M}_\sigma$ is a rapidly mixing Markov chain.

Example 3.2 Continuing Example 3.1, a possible MCMC for this problem is:

INSERT/DELETE CHAIN

Let T be a time bound and $X_0 \in \mathcal{I}(G)$ arbitrary.
Let X_t be the current independent set.
(a) Choose $v \in V$ uniformly at random.
(b) (Delete) If $v \in X_t$, set $X_{t+1} \leftarrow X_t \setminus \{v\}$ with probability $1/(1+\lambda)$.
(Insert) If $v \notin X_t$, set $X_{t+1} \leftarrow X_t \cup \{v\}$ with probability $\lambda/(1+\lambda)$, if this is an independent set.
(c) Set $t \leftarrow t+1$. If $t < T$, return to (a).

To express this in our formalism, $M(\sigma,\omega,\omega') = 0$ unless σ encodes a graph and ω, ω' encode independent sets which differ in at most one vertex of G. Otherwise, as $\lambda = r/q$, $M(\sigma,\omega,\omega') = q$ corresponds to a possible deletion from ω, and $M(\sigma,\omega,\omega') = r$ to a possible insertion. Then $M(\sigma,\omega,\omega)$ is chosen to ensure $\sum_{\omega'} M(\sigma,\omega,\omega') = K(\sigma) = n(r+q)$. It is easy to see that M satisfies our conditions and, using (1), that the weight function W from Example 3.1 determines its stationary distribution. ∎

One of the main applications of sampling is to *approximate integration*. In our setting this means estimating $Z(\sigma)$ to some specified relative error. In the important case where W is a characteristic function, we call the approximate integration problem *approximate counting*. Specifically, a *fully polynomial randomized approximation scheme* (fpras) for $Z(\sigma)$ is a PTM which on input σ, ϵ outputs $\hat{Z}$ so that

$$\Pr(1/(1+\epsilon) \le \hat{Z}/Z \le 1+\epsilon) \ge \tfrac{3}{4},$$

and which runs in time polynomial in $|\sigma|$ and $1/\epsilon$.

The theoretical foundations for approximate uniform sampling and counting were laid in [50]. There, an equivalence between sampling and approximate counting was shown for the class of so-called self-reducible problems [72]. Subsequent work has exposed several disadvantages of this class as a general framework. First, whether a problem is self-reducible depends strongly on the encoding. The problem may have to be re-encoded in an unnatural manner in order to render it self-reducible. An example is adding an edge to a graph, where we may have to encode a graph by its complement to have self-reducibility. Furthermore, many natural reductions employed do not satisfy the self-reducibility criteria. An example is adding an arbitrary ordered pair to a finite partial order (and forming the appropriate closure), which occurs in one

approach to the problem of approximately counting linear extensions. (See [78, pp. 301–302].) More seriously, some problems to which the techniques of [50] have been applied, do not seem to be self-reducible under any re-encoding. Examples are the volume approximation problem of [26], and approximately counting contingency tables [31, 25].

We therefore propose a modified framework. We emphasise that the techniques are those of [50], and we wish only to emphasise the extent of their applicability. Let size : $\Sigma^* \to \mathbb{N}$ be such that size(σ) is polynomially bounded in $|\sigma|$, and if size(σ') < size(σ) then $|\sigma'|$ is polynomially bounded in $|\sigma|$. If size(σ) = 0, we call the problem a *base problem*. For the class of base problems, we assume the existence of a good sampler and a fpras for $Z(\sigma)$.

For all σ, let $\Xi(\sigma)$ be a polynomial time computable set such that

(a) size(ξ) < size(σ) ($\forall \xi \in \Xi$).

(b) There exist polynomial time computable constants $k_\xi(\sigma) \in \mathbb{Q}_+$ and injections $\phi_\xi(\sigma) : \Omega_\xi \to \Omega_\sigma$ ($\forall \xi \in \Xi$), such that

$$k_\xi W(\xi, \omega) \le W(\sigma, \phi_\xi(\omega)) \qquad (\forall \omega \in \Omega_\xi).$$

Both $\phi_\xi(\omega)$ and ${\phi_\xi}^{-1}(\omega)$ must be computable in polynomial time, given $\omega \in \Omega_\xi$ and $\omega \in \Omega_\sigma$, respectively.

(c) For some $\zeta \in \Xi$, $Z(\sigma)/\left(k_\zeta(\sigma) Z(\zeta)\right)$ is polynomially bounded in $|\sigma|$.

If $\widetilde{W}$ satisfies these conditions, we call the problem *self-contractible*. Summing over $\omega \in \Omega_\xi$ and using the injectivity of ϕ_ξ shows that (b) implies $k_\xi Z(\xi) \le Z(\sigma)$ $\forall \xi \in \Xi$. Now, suppose we have a good sampler for $\widetilde{W}$. Then, following [50], we may estimate $k_\xi Z(\xi)/Z(\sigma)$ by rejection sampling. We sketch the method, ignoring details concerning the allowable closeness of intermediate approximations, for which we refer the reader to [50]. We sample ω from $W(\sigma, \cdot)$, and accept with probability $k_\xi W(\xi, \phi^{-1}(\omega))/W(\sigma, \omega)$ if $\phi^{-1}(\omega) \ne \emptyset$. The overall acceptance probability is

$$\sum_{w \in \phi(\Omega_\xi)} \frac{k_\xi W(\xi, \phi^{-1}(\omega))}{W(\sigma, \omega)} \frac{W(\sigma, \omega)}{Z(\sigma)} = \frac{k_\xi Z(\xi)}{Z(\sigma)}.$$

Moreover, from (c) there is some $\zeta \in \Xi$ such that we can estimate this ratio to sufficient relative accuracy in polynomial time. Since size(ζ) < size(σ), we may repeat this process with ζ replacing σ. Then, letting $\sigma_0 = \sigma$, $\sigma_1 = \zeta$, ..., we may iterate until size(σ_r) = 0. Now $|\sigma_i|$ is polynomially bounded in $|\sigma|$ for all $i = 0, 1, \ldots, r$. For σ_r we can approximate $Z(\sigma_r)$ in polynomial time. Then we may multiply estimates together to approximate

$$Z(\sigma_r) \prod_{i=1}^{r} \frac{Z(\sigma_{i-1})}{k_{\sigma_i}(\sigma_{i-1}) Z(\sigma_i)} = \frac{Z(\sigma)}{\prod_{i=1}^{r} k_{\sigma_i}(\sigma_{i-1})}$$

to the required relative error, and hence $Z(\sigma)$.

Example 3.3 Consider the independent set problem of Example 3.1. We take size(G) to be its number of edges, $|E|$. The set Ξ contains the n subgraphs G_v given by deleting a particular nonisolated vertex v. The injections ϕ_{G_v} map independent set X in G_v to X in G. The k_{G_v} are all q. It is easy to check that this satisfies (a),(b) above. Moreover, since the ranges of the ϕ_{G_v} cover $\mathcal{I}(G)$ provided G has an edge, it follows that there exists a v with $Z(G)/qZ(G_v) \leq n$ if $|E| > 0$. Hence (c) is also satisfied. If size(G) $= |E| = 0$, then we may sample from the independent sets directly (or see Example 5.2) and efficiently compute the normalising function. Thus the problem is self-contractible, and it follows that a fpras exists for the normalising function if we have a good sampler. It is shown in Example 6.1 that this is true for $\lambda \leq 2/(\Delta - 2)$. This is not a self-reducibility reduction, though (a different) one does exist for this problem. (See Example 3.4.) ∎

A converse result may be obtained under rather stronger conditions. First suppose that the base problems are such that $Z(\sigma)$ may be determined *exactly*, and suppose that (b) and (c) are strengthened to

(b)′ There exist polynomial time computable constants $k_\xi(\sigma) \in \mathbb{Q}_+$ and injections $\phi_\xi(\sigma) : \Omega_\xi \to \Omega_\sigma$ $(\forall \xi \in \Xi)$, such that

$$k_\xi W(\xi, \omega) = W(\sigma, \phi_\xi(\omega)) \qquad (\forall \omega \in \Omega_\xi).$$

Both $\phi_\xi(\omega)$ $(\omega \in \Omega_\xi)$ and ${\phi_\xi}^{-1}(\omega)$ $(\omega \in \Omega_\sigma)$ must be computable in polynomial time.

(c)′ The sets $\phi_\xi(\Omega_\xi)$ form a partition of Ω_σ.

Let us call such a problem *self-partitionable*. Clearly (b)′ implies (b). Also, from (b)′ and (c)′, since

$$\begin{aligned} \sum_{\xi\in\Xi} k_\xi Z(\xi) &= \sum_{\xi\in\Xi}\sum_{\omega\in\Omega_\xi} k_\xi W(\xi,\omega) = \sum_{\xi\in\Xi}\sum_{\omega\in\Omega_\xi} W(\sigma,\phi_\xi(\omega)) \\ &= \sum_{\omega\in\Omega_\sigma} W(\sigma,\omega) = Z(\sigma), \end{aligned} \tag{3}$$

and the polynomial size of Ξ now implies (c). We sketch the generation procedure, skipping details. Suppose we can estimate $Z(\sigma)$ by $\hat{Z}(\sigma)$ within relative error ϵ to high enough probability. We branch to $\xi \in \Xi$ with probability $k_\xi \hat{Z}(\xi)/(1+\epsilon)\hat{Z}(\sigma)$. If the total of these probabilities over $\Xi(\sigma)$ is more than 1 we "fail", i.e. we abandon this whole sampling "trial". If the total is less than 1, as we would expect, then we fail with the (small) unassigned probability. Otherwise we repeat, getting $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_r$ until size(σ_r) $= 0$, in which case $\hat{Z}(\sigma_r) = Z(\sigma_r)$. Then we generate ω' from $W(\sigma_r, .)$ with small enough variation distance that all probabilities have relative error at most ϵ' for some very small ϵ'. Then ω is determined from $\omega_{i-1} = \phi_{\sigma_i}(\omega_i)$ $(i = 1, \ldots, r)$, with

$\omega_0 = \omega$, $\omega_r = \omega'$. Then (with high probability) the probability that ω is generated is within relative error ϵ' of

$$\frac{k_{\sigma_1}\hat{Z}(\sigma_1)}{(1+\epsilon)\hat{Z}(\sigma_0)}\frac{k_{\sigma_2}\hat{Z}(\sigma_2)}{(1+\epsilon)\hat{Z}(\sigma_1)}\cdots\frac{k_{\sigma_r}Z(\sigma_r)}{(1+\epsilon)\hat{Z}(\sigma_{r-1})}\frac{W(\sigma_r,\omega_r)}{Z(\sigma_r)} = \frac{W(\sigma,\omega)}{(1+\epsilon)^r\hat{Z}(\sigma)},$$

after an easy induction. This is equivalent to the desired weight function. Provided that ϵ, ϵ' are sufficiently small, the failure probability and the variation distance can be kept small on a single trial. Then we may output an arbitrary ω if we fail after some large enough number of trials. Hence the overall variation distance is small. It follows that for self-partitionable problems, approximate integration and good sampling are equivalent. It is easy to see that self-reducible [50] problems are self-partitionable, but the converse is not necessarily true. An example is the volume approximation problem of [26]. (See [27] for some details.)

Example 3.4 Consider further the independent set problem of Example 3.3. We take size(G) to be its number of vertices, n. Choose a fixed vertex $v \in V$. Then the set Ξ contains two subgraphs, G_0 given by deleting v, and G_1 given by deleting v and all its neighbours. The injection ϕ_{G_0} maps independent set X in G_0 to X in G, and ϕ_{G_1} maps X in G_1 to $X \cup \{v\}$ in G. We take $k_{G_0} = q$, $k_{G_1} = r$. This satisfies (a), (b)$'$, (c)$'$ above. If size(G) = 0, then there is only one independent set. Thus the problem is self-partitionable. This reduction is a self-reducibility reduction if the vertex v is chosen to be the lowest numbered vertex, but not if it is chosen to be (say) the vertex of largest degree, which is a logical choice. ■

We can show that approximate integration implies good sampling under rather weaker conditions than self-partitionability. We do not develop this here, however, since we have no example of a problem satisfying these conditions which is not self-partitionable. In any case, the usual direction in applications is to go from sampling to integration.

4 Review

This section presents a brief review of developments in the areas of approximate sampling and approximate integration during the last couple of years. There are several excellent survey articles which introduce the methods and results in this area, including Welsh [78], Jerrum and Sinclair [49] and Jerrum [48]. Therefore we do not give an exhaustive survey here, but concentrate on more recent results from the theoretical computer science community. There is a very extensive body of work on the use of Markov chain Monte Carlo in statistics and statistical mechanics. We do not have space to survey that work here, but simply direct the reader to [8, 74] for further information. We discuss the closely related area of perfect sampling in a separate subsection below.

There are only a few approaches to proving rapid mixing in Markov chains. Much of this paper involves new variations of the classical method of coupling. However, we start this review section by considering results obtained by other methods.

Some of the quantities used to bound the mixing time of a Markov chain are the conductance, spectral gap and log-Sobolev constant of the chain. Using the conductance approach, Kannan, Tetali and Vempala [55] proved rapid convergence of Markov chains for generating bipartite graphs and tournaments. By bounding the spectral gap, Chung, Graham and Yau [17] proved pseudo-polynomial convergence of a Markov chain for contingency tables, when the row and column sums are large enough. This result was improved by Dyer, Kannan and Mount [31]. Their chain mixes in polynomial time when the row and column sums are large enough (the threshold being lower than in [17]). This result was also proved by bounding the spectral gap of the chain.

Diaconis and Saloff-Coste [21] proved rapid mixing of a Markov chain for sampling generating sets of abelian groups, by comparing the log-Sobolev constants of two Markov chains. They used a comparison theorem from an earlier paper [22]. Other methods for relating the mixing times of Markov chains by comparing their spectral gap or log-Sobolev constant have been proposed by Chung and Graham [16], Randall and Tetali [70] and the authors [29]. Further results on bounding the spectral gap can be found in Guattery, Leighton and Miller [40] and in Chung and Yau [18].

Another approach used to bound the mixing time is Dobrushin uniqueness [23]. This approach was used by Salas and Sokal [71] in work on graph colourings, and by Peinado and Lengauer [65] to sample structures which arise in computational chemistry. The relationship between Dobrushin uniqueness and path coupling (see below) was explored in [14].

Madras and Randall [63] gave a method for analyzing the product of two Markov chains, and applied their results to a Markov chain for colourings on a grid.

One area where the Markov chain Monte Carlo approach has been very successful is in estimating the volume of convex bodies (see [52] for an overview). Typically, the mixing rates of random walks in convex bodies are analysed using isoperimetric inequalities (see, for example [26]). The state of the art is the $O^*(n^5)$ volume approximation algorithm of Kannan, Lovász and Simonovits [54] (where $O^*(\cdot)$ is notation which hides logarithmic factors). Kannan and Li [53] showed how to sample according to the multivariate normal density, and relate this to volume computation. In [56], Kannan and Vempala determined when the volume of a convex polytope is a good estimate for the number of lattice points in the polytope.

Now let us consider results which were proved using coupling. As stated in Section 2, the problem of sampling independent sets is a running example in this paper. Let λ denote the parameter used in the weight function, and let Δ denote the maximum degree of a given graph. Luby and Vigoda [62]

described a Markov chain for independent sets which is rapidly mixing for the following values of λ, Δ: when $\Delta = 3$ the chain is rapidly mixing for $\lambda \leq 1$, and when $\Delta \geq 4$ the chain is rapidly mixing for $\lambda \leq 1/(\Delta - 3)$. Propp and Wilson [69] adapted the work of [62] to show how to sample perfectly in expected polynomial time for these values of λ, Δ.

Bubley, Dyer and Jerrum [13] used coupling to give a new approach to sampling points in a convex body. Also, Wilson [79] used coupling to prove upper and lower bounds on the mixing time for lozenge tiling and card shuffling Markov chains. Hernek [44] described a Markov chain for two-rowed contingency tables and, using coupling, showed that the chain has pseudopolynomial mixing time.

Another application of coupling is the popular area of perfect sampling. We review perfect sampling separately, in Section 4.1 below.

A variant of traditional coupling which is described in Section 6 is path coupling, introduced in [9]. (Bubley's thesis [14] contains much work on coupling and path coupling, most of which we mention here.) Using path coupling, the authors [29] proved rapid mixing of a new chain for independent sets, whenever $\lambda \leq 2/(\Delta - 2)$ and $\Delta \geq 3$. (This result is outlined in Example 6.1, and was independently discovered by Luby, Mitzenmacher and Vigoda [77].)

Another result obtained using path coupling is an $O(n^3 \log n)$ bound on the mixing time for a Markov chain for sampling linear extensions of a partial order, due to Bubley and Dyer [11]. This chain is similar to the combinatorial chain of Karzanov and Khachiyan [57], but uses a different distribution to choose the position to update in the partial order. Recently, Wilson [79] built on this result to prove an $O(n^3 \log n)$ upper bound for the mixing time of the Karzanov–Khachiyan chain. He also outlined how one may obtain an $\Omega(n^3 \log n)$ lower bound for the mixing time of the Karzanov–Khachiyan chain (or for the chain given in [11]).

A new Markov chain for two-rowed contingency tables was described by the authors in [28]. The transitions of this chain are very simple: select two columns of the table uniformly at random, and replace this 2×2 submatrix by another 2×2 matrix with the same row and column sums, chosen uniformly at random. Using path coupling, the mixing time of this chain is shown to be polynomial.

The Potts model is a generalisation of graph colouring which arises in statistical physics. The antiferromagnetic Potts model generalises proper colourings, while in the ferromagnetic Potts model configurations are favoured if many vertices are coloured with the same colour as their neighbour. The Swendsen–Wang process is a well-known method for sampling from the ferromagnetic q-state Potts model. It was mildly conjectured that this process might be rapidly mixing on any graph. This conjecture has been disproved by two results which we state below. On the other hand, Cooper and Frieze [19] proved two positive results. Using path coupling, they showed that Swendsen–Wang mixes rapidly for graphs with small maximum degree, for small enough values

of the "coupling constant" K. For the special case of trees, they showed that the mixing time is linear in the number of vertices, for any value of K. The latter result was obtained using classical coupling.

In the limit, the antiferromagentic Potts model describes proper colourings of graphs. The simple Markov chain for graph colourings described independently by [47] and [71] is rapidly mixing if there are at least 2Δ colours, where Δ is the maximum degree of the graph. This result was improved in a paper by the authors [30], which describes a Markov chain which has better bounds on the mixing time for regular graphs, and when fewer than 3Δ colours are used. The latter chain is also rapidly mixing whenever at least 2Δ colours are used. Indeed, it was thought that 2Δ colours might be needed in order to achieve rapid mixing. This was disproved by Bubley and the authors in [12] where a rapidly mixing Markov chain for 5-colourings of graphs with maximum degree three was described. The coupling was constructed by solving a large number of transportation problems. The result was extended to give a rapidly mixing Markov chain for counting 7-colourings of triangle-free 4-regular graphs.

A variation of path coupling, called delayed path coupling, was used by Czumaj et. al. [20] to prove rapid mixing for a Markov chain for generating random permutations in parallel.

We now describe some negative results which have been obtained. Gore and Jerrum [36] investigated the behaviour of the Swendsen–Wang process on the complete graph. They showed that for certain values of the "coupling constant" K, the process has exponential mixing time. Cooper and Frieze [19] extended this result to the random graph $G_{n,p}$, showing that there are critical values of K for which (with high probability) the Swendsen–Wang process requires exponential time to mix. Jerrum and Goldberg [35] proved that the so-called "Burnside process" for sampling from unlabelled structures does not always mix rapidly. Two negative results for independent sets were established by Dyer, Frieze and Jerrum [25]. These results are described in Section 8.

Finally, we mention some results in related areas, which do not directly use Markov chain Monte Carlo. There are many results concerning random walks on expander graphs, of which the paper of Broder, Frieze and Upfal [7] is an example. Gore et. al. [37] gave an algorithm for sampling words from a context-free language, while Frieze et. al. [34] presented an algorithm for generating Hamiltonian cycles in random regular graphs.

4.1 Review of perfect sampling

There has been much interest recently in algorithms for sampling perfectly from a given distribution. We now briefly review results in this area. Further aspects of perfect sampling are considered in Section 7 below.

The first paper in this area was by Propp and Wilson [67]. This paper introduced the approach known as coupling from the past (CFTP). This method is particularly efficient if the chain is *monotone*. In such a chain the state space

forms a lattice with respect to a partial order which is preserved stochastically by the transitions of the chain. For our purposes we may take this as meaning that there exists a coupling of the chain under which all transitions preserve the ordering. (See, for example, [33] for a more formal definition.)

The idea of monotone CFTP is to run two copies of the chain from some time in the past, one starting from the top element of the state space and one from the bottom element. If at time zero both copies of the chain have coalesced, then the state of the chain at time zero is distributed according to the stationary distribution. Otherwise, the procedure is repeated from a time twice as far into the past. In this monotone setting, the expected running time of the algorithm can be bounded in terms of the mixing time of the chain. Let T be a random variable denoting the running time of the algorithm, and suppose that the partial order has height r. Then

$$\mathbf{E}(T) \leq 2\tau(e^{-1})(1 + \log(r)), \tag{4}$$

where $\tau(\varepsilon)$ is the mixing time of the Markov chain [67]. Weaker variations of CFTP been proposed for other situations, for example antimonotone [42] and bounding chains [45, 41].

Fill [33] proposed an alternative algorithm for perfect sampling, known as Fill's algorithm. His approach is based on rejection sampling, where the coin toss used to decide whether to reject the output is performed by running the reversed chain in a clever way. Fill's algorithm is interruptible, in the sense that the impatient user does not bias the output by prematurely terminating a run. Like CFTP, Fill's algorithm works best in the monotone situation.

Monotonicity is a very strong condition. In particular, it is possible to obtain very good experimental estimates for the mixing time in the monotone situation without any use of CFTP (see Johnson [51, Section 2b]).

We now review the main results obtained in the area of perfect sampling. Again, our interest is focussed on those papers with a theoretical computer science slant. For a wider bibliography, see the web site citemed:wil.

We first consider applications of monotone CFTP. Felsner and Wernisch [32] showed how to perfectly sample random linear extensions of two-dimensional partially ordered sets, while Kim, Shor and Winkler applied monotone CFTP to independent sets in bipartite graphs, as cited in [69]. (An alternative approach to the latter problem is described in Example 7.2 below.)

Propp and Wilson [67] applied monotone CFTP to several problems, including sampling lattice paths uniformly at random, sampling permutations uniformly, sampling from the Gibbs distribution in the ferromagnetic Ising model, sampling from the ferromagnetic q-state Potts model if $q \geq 1$, and sampling lozenge tilings uniformly at random. In most applications the expected running time is not analyzed, although this is polynomial whenever the Markov chain in question is rapidly mixing, by (4).

Now let us consider *antimonotone* systems. Here there is a partial order which is reversed by transitions of the Markov chain. This concept builds

on the work of Kendall [58] on repulsive point-processes. Häggström and Nelander [42] showed how to apply CFTP to several systems, including independent sets in graphs, the antiferromagnetic Ising model, and the random cluster model when $q < 1$ (see [42] for details).

The concept of a *bounding chain* generalises both monotone and antimonotone chains. Huber [45] used bounding chains to obtain a uniform sampling algorithm for graph colourings, which has expected polynomial time if at least $(\Delta + 1)^2$ colours are used, where Δ is the maximum degree of the graph. By modifying the analysis of [10], Huber obtained an expected polynomial time uniform sampler for sink free orientations of a graph. Very recently, Huber [46] analysed the Dyer-Greenhill chain for independent sets [29] using bounding chains, to give an algorithm for perfect sampling in expected polynomial time whenever $\lambda \leq 2/(\Delta - 2)$, $\Delta \geq 3$.

There are two results on performing perfect sampling in a Markov chain where the transition probabilities are unknown. The first such result was given by Lovász and Winkler [60]. This was improved upon by Propp and Wilson [68], who presented an algorithm with universal randomized stationary stopping time which is within a constant factor of optimal. These algorithms are related to the generation of random spanning trees in a weighted directed graph. Aldous [1] and Broder [6] independently discovered an algorithm for generating random spanning trees of undirected graphs uniformly at random. Propp and Wilson [68] gave faster and more general algorithms for perfect generation of spanning trees.

We return to the subject of perfect sampling in Section 7 below.

5 Coupling

As indicated in Section 4, recent work in constructing polynomial time sampling methods has centred around Markov chain Monte Carlo algorithms, with coupling as the proof technique. We now describe this idea, as far as is relevant to finite Markov chains. For further information, see [59].

A *coupling* $\mathcal{C}(\mathcal{M})$ for $\mathcal{M}$ is a stochastic process (X_t, Y_t) on Ω^2 such that each of X_t, Y_t is marginally a copy of $\mathcal{M}$,

$$\begin{array}{rcl} \Pr(X_t = \sigma_1 \mid X_{t-1} = \omega_1) & = & P(\omega_1, \sigma_1), \\ \Pr(Y_t = \sigma_2 \mid Y_{t-1} = \omega_2) & = & P(\omega_2, \sigma_2), \end{array} \qquad (\forall t > 0). \tag{5}$$

The following simple but powerful inequality, due to Doeblin [24], then follows easily from these definitions.

Lemma 5.1 (Coupling Lemma) *Let X_t, Y_t be a coupling for $\mathcal{M}$ such that Y_0 has the stationary distribution π. Then, if X_t has distribution p_t,*

$$\mathrm{d}_{\mathrm{TV}}(p_t, \pi) \leq \Pr(X_t \neq Y_t). \tag{6}$$

Proof Suppose $A_t \subseteq \Omega$ maximizes in (2). Then, since Y_t has distribution π,

$$\begin{aligned} \mathrm{d}_{\mathrm{TV}}(p_t, \pi) &= \Pr(X_t \in A_t) - \Pr(Y_t \in A_t) \\ &\leq \Pr(X_t \in A_t, Y_t \notin A_t) \;\leq\; \Pr(X_t \neq Y_t). \quad \blacksquare \end{aligned}$$

It is important to remember that the Markov chain Y_t is simply a proof construct, and X_t the chain we actually observe. We also require that $X_t = Y_t$ implies $X_{t+1} = Y_{t+1}$, since this makes the right side of (6) nonincreasing. Then the earliest epoch T at which $X_T = Y_T$ is called *coalescence*, making T is a random variable. A *successful coupling* is such that $\lim_{t\to\infty} \Pr(X_t \neq Y_t) = 0$. Clearly we are only interested in successful couplings.

A coupling is a *Markovian coupling* if the process $\mathcal{C}(\mathcal{M})$ is a Markov chain on Ω^2. Griffeath [39] showed that there always exists a *maximal coupling*, which gives equality in (6). (An easier proof is given in [66].) This maximal coupling is in general non-Markovian, and is seemingly not constructible without knowing p_t $(t = 1, 2, \ldots)$. But coupling has little algorithmic value if we already know p_t. More generally, it seems difficult to prove mixing properties of non-Markovian couplings in our setting. Therefore we restrict attention to Markovian couplings, at the (probable) cost of sacrificing equality in (6). Burdzy and Kendall [15] investigated efficient Markovian couplings.

Let $\mathcal{C}(\mathcal{M})$ be a Markovian coupling, with Q its transition matrix, i.e. the probability of a joint transition from (ω_1, ω_2) to (σ_1, σ_2) is $Q^{\omega_1\omega_2}_{\sigma_1\sigma_2}$. The precise conditions required of Q are then

$$Q^{\omega\,\omega}_{\sigma_1\sigma_2} \neq 0 \qquad \text{implies} \qquad \sigma_1 = \sigma_2 \qquad (\forall \omega \in \Omega), \tag{7}$$

$$\sum_{\sigma_2 \in \Omega} Q^{\omega_1\omega_2}_{\sigma_1\sigma_2} = P^{\omega_1}_{\sigma_1} \quad (\forall \omega_2 \in \Omega), \qquad \sum_{\sigma_1 \in \Omega} Q^{\omega_1\omega_2}_{\sigma_1\sigma_2} = P^{\omega_2}_{\sigma_2} \quad (\forall \omega_1 \in \Omega). \tag{8}$$

Here (7) implies equality after coalescence, and (8) implies the marginals are copies of $\mathcal{M}$. Our goal is to design Q so that $\Pr(X_t \neq Y_t)$ quickly becomes small. We need only specify Q to satisfy (8) for $\omega_1 \neq \omega_2$. The other entries are completely determined by (7) and (8).

To prove rapid mixing using coupling, it is usual to map $\mathcal{C}(\mathcal{M})$ to a process on $\mathbb{N}$ by defining a function $\psi : \Omega^2 \to \mathbb{N}$ such that $\psi(\omega_1, \omega_2) = 0$ implies $\omega_1 = \omega_2$. We call this a *proximity function*. Then $\Pr(X_t \neq Y_t) \leq \mathbf{E}(\psi(X_t, Y_t))$, by Markov's inequality, and we need only show that $\mathbf{E}(\psi(X_t, Y_t))$ converges quickly to zero.

Example 5.2 Continuing Example 3.2, as an illustration we analyse the Insert/Delete chain in the case when $\Delta = 0$, i.e. $E = \emptyset$. Then the qualification in step (b) of its description becomes redundant. We use this example merely to fix ideas. There are, of course, many other (possibly simpler) ways of approaching this easy case.

At time t, let X_t, Y_t be the states occupied by our two copies of the chain. Consider the following coupling, where for convenience we drop temporarily

the subscript t:

Choose the same $v \in V$ uniformly at random in both X, Y.

(i) If $v \in X \cap Y$, delete in both X, Y with probability $1/(1+\lambda)$,
no move in both X, Y with probability $\lambda/(1+\lambda)$.

(ii) If $v \notin X \cup Y$, insert in both X, Y with probability $\lambda/(1+\lambda)$,
no move in both X, Y with probability $1/(1+\lambda)$.

(iii) If $v \in X \setminus Y$, delete in X, no move in Y, with probability $1/(1+\lambda)$,
insert in Y, no move in X, with probability $\lambda/(1+\lambda)$.

(iv) If $v \in Y \setminus X$, insert in X, no move in Y, with probability $\lambda/(1+\lambda)$,
delete in Y, no move in X, with probability $1/(1+\lambda)$.

It is easy to see that this coupling has the correct marginals for both X_t, Y_t, though they are far from independent. To monitor convergence, let $H(X, Y)$ be the *Hamming distance* between X, Y, i.e. $H(X, Y) = |X \oplus Y|$, where $\oplus$ denotes symmetric difference. Hamming distance clearly satisfies the requirements of a proximity function, and $H(X, Y) \leq n$.

In cases (i) and (ii) of the coupling we have $H(X_{t+1}, Y_{t+1}) = H(X_t, Y_t)$, whereas in cases (iii) and (iv) we have $H(X_{t+1}, Y_{t+1}) = H(X_t, Y_t) - 1$. However, the probability that case (iii) or (iv) occurs is simply $H(X_t, Y_t)/n$. Hence

$$\mathbf{E}(H(X_{t+1}, Y_{t+1}) \mid (X_t, Y_t)) = H(X_t, Y_t) - H(X_t, Y_t)/n = (1 - 1/n)H(X_t, Y_t),$$

from which it follows by induction that

$$\mathbf{E}(H(X_t, Y_t)) = (1 - 1/n)^t H(X_0, Y_0) \leq n(1 - 1/n)^t \leq ne^{-t/n}.$$

But now, by (6) and the non-negativity and integrality of H,

$$\mathrm{d_{TV}}(p_t, \pi) \leq \Pr(X_t \neq Y_t) \leq \mathbf{E}(H(X_t, Y_t)) \leq ne^{-t/n},$$

from which it follows easily that $\tau(\varepsilon) \leq n \log(n/\varepsilon)$. Thus we have established rapid mixing of the chain for this simple case.

Using a more clever coupling, the mixing time of this chain can be improved by a constant factor. We omit the details since our interest is purely illustrative. ∎

6 Path coupling

A major difficulty with coupling is that we are obliged to to specify it, and show improvement in the proximity function, for every pair of states. The idea of *path coupling* [9], where applicable, can be a major saving in this respect. We describe the approach below.

Recall that a *quasi-metric* satisfies the conditions for a metric except possibly the symmetry condition. Any metric is a quasi-metric, but a simple example of a quasi-metric which is not a metric is directed edge distance in a digraph.

Suppose we have a relation $S \subseteq \Omega^2$ such that S has transitive closure Ω^2, and suppose that we have a proximity function defined for all pairs in S, i.e. $\psi : S \to \mathbb{N}$. Then we may lift ψ to a quasi-metric $\delta(\omega, \omega')$ on Ω as follows. For each pair $(\omega, \omega') \in \Omega^2$, consider the set $\mathcal{P}(\omega, \omega')$ of all sequences

$$\omega = \omega_1, \omega_2, \ldots, \omega_{r-1}, \omega_r = \omega' \quad \text{with} \quad (\omega_i, \omega_{i+1}) \in S \quad (i = 1, \ldots, r-1). \tag{9}$$

Then we set

$$\delta(\omega, \omega') \;=\; \min_{\mathcal{P}(\omega,\omega')} \sum_{i=1}^{r-1} \psi(\omega_i, \omega_{i+1}). \tag{10}$$

It is easy to prove that δ is a quasi-metric. We call a sequence minimizing (10) *geodesic*. We now show that, without any real loss, we may define the (Markovian) coupling only on pairs in S. Such a coupling is a called a path coupling. We give a detailed development below. Clearly $S = \Omega^2$ is always a relation whose transitive closure is Ω^2, but path coupling is only useful when we can define a suitable S which is "much smaller" than Ω^2. A relation of particular interest is $\mathcal{R}_\sigma$ from Section 3, but this is not always the best choice.

As in Section 5, we use σ (or σ_i) to denote a state obtained by performing a single transition of the chain from the state ω (or ω_i). Let P_σ^ω denote the probability of a transition from state ω to state σ in the Markov chain, and let $Q_{\sigma\sigma'}^{\omega\omega'}$ denote the probability of a joint transition from (ω, ω') to (σ, σ'), where $(\omega, \omega') \in S$, as specified by the path coupling. Since this coupling has the correct marginals, we have

$$\sum_{\sigma' \in \Omega} Q_{\sigma\sigma'}^{\omega\omega'} = P_\sigma^\omega, \qquad \sum_{\sigma \in \Omega} Q_{\sigma\sigma'}^{\omega\omega'} = P_{\sigma'}^{\omega'} \qquad (\forall (\omega, \omega') \in S). \tag{11}$$

We extend this to all pairs $(\omega, \omega') \in \Omega^2$, as follows. For each pair, fix a sequence $(\omega_1, \omega_2, \ldots, \omega_r) \in \mathcal{P}(\omega, \omega')$. We do not assume the sequence is geodesic here, or indeed the existence of any proximity function, but this is our eventual purpose. The implied global coupling $\bar{Q}_{\sigma_1\sigma_r}^{\omega_1\omega_r}$ is then defined along this sequence by successively conditioning on the previous choice. Using (11), this can be written explicitly as

$$\bar{Q}_{\sigma_1\sigma_r}^{\omega_1\omega_r} = \sum_{\sigma_2 \in \Omega} \sum_{\sigma_3 \in \Omega} \cdots \sum_{\sigma_{r-1} \in \Omega} Q_{\sigma_1\sigma_2}^{\omega_1\omega_2} \frac{Q_{\sigma_2\sigma_3}^{\omega_2\omega_3}}{P_{\sigma_2}^{\omega_2}} \cdots \frac{Q_{\sigma_{r-1}\sigma_r}^{\omega_{r-1}\omega_r}}{P_{\sigma_{r-1}}^{\omega_{r-1}}}. \tag{12}$$

Summing (12) over σ_r or σ_1, and again applying (11), causes the right side to successively simplify, giving

$$\sum_{\sigma_r \in \Omega} \bar{Q}_{\sigma_1\sigma_r}^{\omega_1\omega_r} = P_{\sigma_1}^{\omega_1} \quad (\forall \omega_r \in \Omega), \qquad \sum_{\sigma_1 \in \Omega} \bar{Q}_{\sigma_1\sigma_r}^{\omega_1\omega_r} = P_{\sigma_r}^{\omega_r} \quad (\forall \omega_1 \in \Omega). \tag{13}$$

Hence the global coupling satisfies (8), as we would anticipate from the properties of conditional probabilities.

Now suppose the global coupling is determined by geodesic sequences. We bound the expected value of $\delta(\sigma_1, \sigma_r)$. This is

$$\begin{aligned}
\mathbf{E}(\delta(\sigma_1,\sigma_r)) &= \sum_{\sigma_1}\cdots\sum_{\sigma_r}\delta(\sigma_1,\sigma_r)\frac{Q^{\omega_1\omega_2}_{\sigma_1\sigma_2}Q^{\omega_2\omega_3}_{\sigma_2\sigma_3}\cdots Q^{\omega_{r-1}\omega_r}_{\sigma_{r-1}\sigma_r}}{P^{\omega_2}_{\sigma_2}\cdots P^{\omega_{r-1}}_{\sigma_{r-1}}} \\
&\le \sum_{\sigma_1}\cdots\sum_{\sigma_r}\sum_{i=1}^{r-1}\delta(\sigma_i,\sigma_{i+1})\frac{Q^{\omega_1\omega_2}_{\sigma_1\sigma_2}Q^{\omega_2\omega_3}_{\sigma_2\sigma_3}\cdots Q^{\omega_{r-1}\omega_r}_{\sigma_{r-1}\sigma_r}}{P^{\omega_2}_{\sigma_2}\cdots P^{\omega_{r-1}}_{\sigma_{r-1}}} \\
&= \sum_{i=1}^{r-1}\sum_{\sigma_1}\cdots\sum_{\sigma_r}\delta(\sigma_i,\sigma_{i+1})\frac{Q^{\omega_1\omega_2}_{\sigma_1\sigma_2}Q^{\omega_2\omega_3}_{\sigma_2\sigma_3}\cdots Q^{\omega_{r-1}\omega_r}_{\sigma_{r-1}\sigma_r}}{P^{\omega_2}_{\sigma_2}\cdots P^{\omega_{r-1}}_{\sigma_{r-1}}} \\
&= \sum_{i=1}^{r-1}\sum_{\sigma_i}\sum_{\sigma_{i+1}}\delta(\sigma_i,\sigma_{i+1})Q^{\omega_i\omega_{i+1}}_{\sigma_i\sigma_{i+1}}, \qquad (14)
\end{aligned}$$

where we have used the triangle inequality for a quasi-metric and the same observation as that leading from (12) to (13).

Suppose we can find $\beta \le 1$, such that, for all $(\omega, \omega') \in S$,

$$\mathbf{E}(\delta(\sigma,\sigma')) \;=\; \sum_{\sigma}\sum_{\sigma'}\delta(\sigma,\sigma')\,Q^{\omega\omega'}_{\sigma\sigma'} \;\le\; \beta\,\delta(\omega,\omega'). \qquad (15)$$

Then, from (14), (15) and (10) we have

$$\mathbf{E}(\delta(\sigma_1,\sigma_r)) \;\le\; \sum_{i=1}^{r-1}\beta\,\delta(\omega_i,\omega_{i+1}) \;=\; \beta\sum_{i=1}^{r-1}\delta(\omega_i,\omega_{i+1}) \;=\; \beta\,\delta(\omega_1,\omega_r). \qquad (16)$$

Thus we can show (15) for every pair, merely by showing that this holds for all pairs in S. To apply path coupling to a particular problem, we must find a relation S and proximity function ψ so that this is possible. In particular we need $\delta(\omega, \omega')$ for $(\omega, \omega') \in S$ to be easily deducible from ψ.

Suppose that Ω has *diameter* D, i.e. $\delta(\omega, \omega') \le D$ for all $\omega, \omega' \in \Omega$. Then, if $\beta < 1$, a similar calculation to that at the end of Example 5.2 gives

$$\mathrm{d_{TV}}(p_t, \pi) \le \varepsilon \quad \text{for} \quad t \ge \ln(D\varepsilon^{-1})/(1-\beta). \qquad (17)$$

This bound is polynomial even when D is exponential in the problem size. It is also possible to prove a bound when $\beta = 1$, provided we know the quasi-metric cannot "get stuck". Specifically, we need an $\alpha > 0$ (inversely polynomial in the problem size) such that, in the above notation,

$$\Pr(\delta(\sigma,\sigma') \ne \delta(\omega,\omega')) \ge \alpha \qquad (\forall \omega, \omega' \in \Omega). \qquad (18)$$

Observe that it is not sufficient simply to establish (18) for pairs in S. However, the structure of the path coupling can usually help in proving it. In this case, we can show that

$$\mathrm{d_{TV}}(p_t, \pi) \le \varepsilon \quad \text{for} \quad t \ge \lceil eD^2/\alpha \rceil \lceil \ln(\varepsilon^{-1}) \rceil. \qquad (19)$$

This is most easily shown using a martingale argument. We omit the proof, but see [61] for details. Here we need D to be polynomial in the problem size.

Example 6.1 We continue Example 5.2, by analysing a modification of the INSERT/DELETE chain. We show that this is rapidly mixing if $\lambda \leq 2/(\Delta - 2)$. The reader may observe that the analysis would be difficult to tackle directly using coupling.

INSERT/DELETE/DRAG CHAIN

Let X_t be the current independent set.
(a) With probability $\frac{1}{2}$, go to (c),
otherwise choose $v \in V$ uniformly at random.
(b) (Delete) If $v \in X_t$, $X_{t+1} \leftarrow X_t \setminus \{v\}$ with probability $1/(1+\lambda)$.
(Insert) If $v \notin X_t$ and $X_t \cup \{v\}$ is an independent set,
$X_{t+1} \leftarrow X_t \cup \{v\}$ with probability $\lambda/(1+\lambda)$.
(Drag) If $v \notin X_t$, but v has a *unique* neighbour $u \in X_t$,
$X_{t+1} \leftarrow X_t \cup \{v\} \setminus \{u\}$ with probability $\lambda/4(1+\lambda)$.
(c) Set $t \leftarrow t+1$, go to (a).

This chain has the same stationary distribution as the INSERT/DELETE chain, since the DRAG move is symmetric. The set S contains all pairs in $\mathcal{I}(G)$ at unit Hamming distance. The proximity function on S is Hamming distance. This lifts to Hamming distance globally, which is a metric.

For a pair $(X, Y) \in S$, let w denote the unique vertex in which they differ, and without loss assume $X \setminus Y = \{w\}$. Let X', Y' denote the evolved states after one step. We then use the following path coupling. We interleave the analysis with its statement.

(a) Choose the same v uniformly at random in both X, Y.

(b) If $v \neq w$ or a neighbour of w, perform the same step in both X, Y.
Clearly then $\delta(X', Y') = \delta(X, Y) = 1$.

(c) If $v = w$, then
INSERT in Y, do nothing in X with probability $\lambda/(1+\lambda)$.
DELETE in X, do nothing in Y with probability $1/(1+\lambda)$.
In either case we have $\delta(X', Y') = 0$.

(d) If v is a neighbour of w which has no neighbour in Y,
DRAG in X, INSERT in Y with probability $\lambda/4(1+\lambda)$.
Do nothing in X, INSERT in Y with probability $3\lambda/4(1+\lambda)$.
Do nothing in X or Y with probability $1/(1+\lambda)$.
In the first event, we have $X' = Y'$, but in the second, the Hamming distance increases by 1. Thus (conditional on v)

$$\mathbf{E}(\delta(X', Y')) = 1 + \frac{-1 \times \lambda}{4(1+\lambda)} + \frac{1 \times 3\lambda}{4(1+\lambda)} = 1 + \frac{\lambda}{2(1+\lambda)}.$$

(e) If v is a neighbour of w which more than one neighbour in Y, do nothing in both X, Y. Clearly $\delta(X', Y') = 1$.

(f) If v is a neighbour of w which has a unique neighbour in Y,
DRAG in Y, do nothing in X with probability $\lambda/4(1+\lambda)$.
Do nothing in either X or Y with probability $1 - \lambda/4(1+\lambda)$.
In the first event, the Hamming distance increases by 2. Thus (conditional on v)

$$\mathbf{E}(\delta(X', Y')) = 1 + \frac{2 \times \lambda}{4(1+\lambda)} = 1 + \frac{\lambda}{2(1+\lambda)}.$$

Now suppose w has d_1, d_2, d_3 neighbours as in (d), (e) and (f) above. Then, since $d_1 + d_2 + d_3 \leq \Delta$,

$$\mathbf{E}(\delta(X', Y')) \;=\; 1 - \frac{1}{n} + \frac{d_1\lambda}{2n(1+\lambda)} + \frac{d_3\lambda}{2n(1+\lambda)} \;\leq\; 1 + \frac{1}{n}\left(\frac{\Delta\lambda}{2(1+\lambda)} - 1\right),$$

and $\mathbf{E}(\delta(X', Y')) \leq \delta(X, Y) = 1$ when $\Delta\lambda/2(1+\lambda) \leq 1$, i.e. $\lambda \leq 2/(\Delta - 2)$. ∎

7 Perfect sampling

Perfect sampling was reviewed in Section 4.1 above. We now collect some results and observations on this idea. First we present an easy application of path coupling to monotone couplings. This is illustrated using independent sets in bipartite graphs.

We then attempt to relate the notions of perfect sampling and good sampling, by showing how a rapidly mixing Markov chain can be transformed into an expected polynomial time perfect sampler. This shows that good sampling is at least as hard as expected polynomial-time perfect sampling.

Before presenting these results, we give two definitions of perfect sampling. These definitions seem to encompass all reasonable possibilities. Let Ω be a state space and π a distribution on Ω. We say that an algorithm $\mathcal{A}$ is a *weak perfect sampler* for π if $\mathcal{A}$ outputs $i \in \Omega$ with probability π_i, and runs in expected polynomial time. The well known coupling from the past (CFTP) algorithm [67] is an example of a weak perfect sampler. Here we cannot exclude the possibility that there exists a state $i \in \Omega$ which the algorithm $\mathcal{A}$ cannot output in polynomial time. There are results where a weak perfect sampler has the additional property that

$$\Pr(T > k\,\mathbf{E}(T)) < (1/4)^k \qquad (k \in \mathbb{N}), \tag{20}$$

or similar (see, for example, [45]). This guarantees that the perfect sampler could be turned into a good sampler, and may be of some comfort, but it does not imply that perfect sampling is achievable within any fixed polynomial time. If perfect sampling is the goal here, there seems little to be gained by showing

that something like (20) holds. Even if the algorithm terminates in an acceptable time, we can only claim that the sample produced is perfectly distributed if we were truly prepared to wait forever for it to terminate. Otherwise the sample is only approximate, and the variation distance is determined by exactly how long we would have been prepared to wait. This point is illustrated by the following example.

Example 7.1 Consider a random walk on the integers $[n]$ with transitions from state i given as follows: with probability $1/2$ move to state $\min(i+1, n)$, otherwise move to state $\max(i-1, 1)$. (This generalises an example used by Propp and Wilson [67] for a different purpose.) This chain is monotone under the obvious linear ordering of the states, and the stationary distribution is uniform. A monotone coupling for this chain can be defined as follows. From a given state (i, j), move to $(\min(i+1, n), \min(j+1, n))$ with probability $1/2$, otherwise move to $(\max(i-1, 1), \max(j-1, 1))$. The expected time for coalescence to occur, from initial state $(1, n)$, is $O(n^2)$. However, for all states i, j and times t, the probability that the coupling is in state (i, j) at time t is of the form $x2^{-t}$, for some integer x. If n is not a power of 2, therefore, we cannot guarantee perfect sampling within *any* bounded amount of time, using monotone CFTP. The difficulty here is not due to the use of the (possibly restrictive) PTM model of computation. Even if we move to the more powerful oracle coin model [73, p. 18], the conclusion still holds. ∎

Propp and Wilson [67] referred to this phenomenon as "user impatience", but we emphasise that how impatient *we might have been* is still relevant even when the algorithm terminates in an acceptable time. Since it is unlikely that we were really prepared to wait forever, this seems an immense difficulty for this form of perfect sampling. The situation here is in marked contrast to an expected polynomial time *optimization* algorithm where, if we wish, we may stop and restart the algorithm at any point without adverse consequences.

By contrast, an algorithm $\mathcal{A}$ is called a *strong perfect sampler* for π if the following conditions hold:

(a) The output of $\mathcal{A}$ is either the symbol $\perp$ or an element of Ω.

(b) The probability that $i \in \Omega$ is output, conditional on an element of Ω being output, is equal to π_i.

(c) The probability that $\perp$ is output is at most $\frac{1}{2}$.

(d) The running time of the algorithm is polynomial.

Fill's algorithm [33] is an example of a strong perfect sampler. (Fill used the term "interruptible".) Our definition of a strong perfect sampler for a distribution π generalises the definition of a uniform generator, given in [50]. Here we can truly claim that the output is a perfect sample, provided it is not $\perp$. Conditions (c) and (d) guarantee that it is likely we do not have to wait

too long for this to occur. We do not, as with the weak perfect sampler, have to reason about what we might have done on other possible executions of the algorithm to assert this. It seems therefore that, where available, this form of perfect sampling is markedly superior. Huber [45, 46] used bounding chains to turn rapidly mixing Markov chains into weak perfect samplers, by applying CFTP. With a little modification, one can instead turn these algorithms into strong perfect samplers by applying Fill's algorithm [33].

A strong perfect sampler can easily be turned into a good sampler. Simply run the strong perfect sampler to completion. If the output is an element of Ω, then return this element. If the output is the symbol $\perp$, then return a fixed element of Ω, chosen arbitrarily beforehand. The variation distance between this distribution and the desired distribution is bounded above by the probability that the strong perfect sampler outputs the symbol $\perp$. We can ensure that the variation distance is at most ε, in time increased only by a factor $O(\log \varepsilon^{-1})$, simply by running the sampler repeatedly. Therefore strong perfect sampling is at least as hard as good sampling.

Good samplers exist for many problems, but few are known to possess strong perfect samplers. Can strong perfect sampling algorithms generally be found where good samplers exist? We conjecture that this is not generally possible, and that strong perfect sampling is harder than good sampling. Less ambitiously, can we devise a strong perfect sampler for all matchings in a graph, or for the linear extensions of a partial order? These questions are all open.

In Section 7.2, we show that good sampling is at least as hard as weak perfect sampling. One may also ask whether good samplers can be constructed whenever weak perfect sampling is possible. We present a very restrictive model of computation where weak perfect sampling is possible, but with high probability good sampling is not achievable. This shows that good sampling is harder than weak perfect sampling. Ideally, we would like to prove a similar result in a more realistic model of computation.

Bellare, Goldreich and Petrank [3] showed how to perform strong perfect sampling of NP-witnesses, using an NP-oracle. It follows that, if $\mathsf{P} = \mathsf{NP}$, we can perform strong perfect sampling for any relation in NP. Conversely, if strong perfect sampling is impossible for any problem in NP, then $\mathsf{P} \neq \mathsf{NP}$.

7.1 Monotone path coupling

Suppose we have a Markov chain $\mathcal{M}$ on a state space Ω. Further suppose that there exists a partial order $\leq$ on Ω, possessing unique top and bottom elements. Let a path coupling $(\omega, \omega') \mapsto (\sigma, \sigma')$ be defined with respect to a subset S of Ω^2 and a proximity function ψ. We call this a *monotone path coupling* if the following conditions hold.

(a) Whenever $(\omega, \omega') \in \Omega^2$ such that $\omega < \omega'$, we can form a strictly increasing geodesic sequence $\omega = \omega_1, \omega_2, \ldots, \omega_r = \omega'$ between ω and ω'.

(b) If $(\omega, \omega') \in S$ and $\omega \leq \omega'$ then $\sigma \leq \sigma'$.

If these conditions are satisfied, then the implied global coupling is also monotone. An example of a monotone path coupling is given below.

Example 7.2 Consider the INSERT/DELETE/DRAG chain of Example 6.1, restricted to bipartite graphs. Let $G = (V, E)$ be a bipartite graph with vertex bipartition $V = V_1 \cup V_2$. A natural partial order on $\mathcal{I}(G)$ is defined by $X \leq Y$ if and only if

$$X \cap V_1 \subseteq Y \cap V_1 \qquad \text{and} \qquad Y \cap V_2 \subseteq X \cap V_2.$$

This partial order has a top element V_1 and a bottom element V_2. We now show that the coupling described in Example 6.1 is monotonic with respect to this partial order.

Suppose $X, Y \in \mathcal{I}(G)$ satisfy $X \leq Y$. Then $X \setminus Y \subseteq V_2$ and $Y \setminus X \subseteq V_1$. We can easily form a strictly increasing geodesic sequence between X and Y. Thus it suffices to show that the coupling is monotonic for pairs at distance one apart.

Let $X, Y \in \mathcal{I}(G)$ satisfy $H(X, Y) = 1$ and $X \leq Y$. Without loss of generality, we can assume that $X = Y \cup \{w\}$ for some $w \in V_2$. Let v be the vertex chosen uniformly at random in the step (a) of the coupling. If a vertex is either added to or deleted from both X and Y then the resulting pair (X', Y') satisfies $X' \leq Y'$. Therefore we need only check monotonicity when v satisfies condition (c), (d) or (f) of Example 6.1. However, it is easy to see that the coupling defined in each of these cases is monotone. For example, suppose that v satisfies condition (f). Then v is a neighbour of w which has a unique neighbour u in Y. Since $w \in V_2$ we have $v \in V_1$ and $u \in V_2$. Here $(X', Y') = (X, (Y \setminus \{u\}) \cup \{v\})$ or $(X', Y') = (X, Y)$. In the second case there is nothing to prove, while in the first case we have

$$X' = X < Y < Y \setminus \{u\} < (Y \setminus \{u\}) \cup \{v\} = Y'.$$

This shows that the coupling is monotone, and the other cases are proved similarly.

Suppose now that $\lambda \leq 2/(\Delta - 2)$ and $\Delta \geq 3$. Since the coupling is monotone, we can use a slight modification of Fill's algorithm [33] to give a strong perfect sampler for independent sets in bipartite graphs. For the necessary details on Fill's algorithm, see [33].

7.2 Transforming an approximate sampler into a weak perfect sampler

This section shows that weak perfect sampling is not difficult to achieve if we have a good MCMC sampler. The idea is simple: with very high probability, we output the result of running the Markov chain for a polynomial number of steps, just as we would in approximate sampling. Then with exponentially

small probability, we perform an exponential amount of computation and output some state. The computation ensures that the output is distributed exactly according to the stationary distribution. The probability is chosen small enough that the expected cost of computation is polynomial.

Let $\mathcal{M}$ be a combinatorial Markov chain with transition matrix P, and state space Ω. Let $N = |\Omega|$ and suppose that $n = \lceil \log_2 N \rceil$. We assume that the transitions of $\mathcal{M}$ from state i are performed by a strong perfect sampler, for each i. Let δ be a parameter, the value of which is fixed later, such that $0 < \delta \leq \frac{1}{2}$. Let f, g be polynomial-time computable integers such that $0 < f/g \leq \pi_i$ for all $i \in \Omega$. We can certainly find such f, g: for example, let $f = 1$ and $g = 2^{|\sigma|^c}$, in the notation of Section 2. (We do not assume that the exact values of the π_i are known, although in almost all applications they will be known up to some normalizing constant.) Define ε by

$$\varepsilon = \frac{\delta f}{(1-\delta)g},$$

for this particular value of δ. Let $t = \tau(\varepsilon)$, where $\tau(\cdot)$ is the mixing time of $\mathcal{M}$. Fix some start state, which we denote by 1, and let p_t denote the distribution of the Markov chain after t steps, starting from initial state 1.

Let r be the map defined by

$$\delta r_i = \pi_i - (1-\delta)p_t(i).$$

Then

$$\delta r_i \;=\; (1-\delta)(\pi_i - p_t(i)) + \delta\pi_i \;\geq\; -\varepsilon(1-\delta) + \delta\pi_i \;=\; \delta(\pi_i - f/g) \;\geq\; 0.$$

Hence r_i is nonnegative for all $i \in \Omega$. Since $\sum_{i\in\Omega} r_i = 1$, the map r is a probability distribution on Ω. Consider the algorithm given in pseudocode in Figure 1. The probability that this algorithm outputs $i \in \Omega$ is given by

$$(1-\delta)p_t(i) + \delta r_i = \pi_i.$$

Hence the algorithm performs perfect sampling.

We now describe an implementation of this algorithm with expected running time which is polynomial in n. We also check that the algorithm only requires a polynomial amount of space.

In order to calculate r_i exactly, we must be able to calculate the t-step probabilities $p_t(i)$. We now show how to do this. Without loss of generality, assume that t is a power of two, by rounding up if necessary. For all $\ell \geq 0$ and i, $j \in \Omega$ let $q^{(\ell)}(i,j)$ be defined by

$$q^{(\ell)}(i,j) = \Pr\left(X_{2^\ell} = j \mid X_0 = i\right).$$

```
Begin.
   with probability (1 − δ) do
     run M for t steps from initial state 1 and output the final state;
   otherwise
     choose i ∈ Ω with probability r_i and output i;
   end;
End.
```

Figure 1: A perfect sampling algorithm

Then $q^{(0)}(i,j) = P_{ij}$ for all i, $j \in \Omega$. Moreover,

$$\begin{aligned} q^{(\ell)}(i,j) &= \Pr\left(X_{2^\ell} = j \mid X_0 = i\right) \\ &= \sum_{k\in\Omega} \Pr\left(X_{2^{(\ell-1)}} = k \mid X_0 = i\right) \cdot \Pr\left(X_{2^{(\ell-1)}} = j \mid X_0 = k\right) \\ &= \sum_{k\in\Omega} q^{(\ell-1)}(i,k) \cdot q^{(\ell-1)}(k,j) \end{aligned}$$

for all i, $j \in \Omega$ and $\ell \geq 0$. Now $2^s P$ is integral, for some polynomially bounded computable integer s, by definition of a weight matrix. Let $b_\ell = 2^\ell s$ for $\ell \geq 0$. Using induction, it is not difficult to show that $2^{b_\ell} q^{(\ell)}(i,j)$ is integral for all i, $j \in \Omega$ and $\ell \geq 0$.

```
Begin.
   if ℓ = 0 then
     return 2^s P_ij;
   else
     sum := 0;
     for k := 1 to N do
       sum := sum + Q(i, k, ℓ − 1) · Q(k, j, ℓ − 1);
     endfor;
     return sum;
   endif;
End.
```

Figure 2: The procedure $Q(i,j,\ell)$

This leads to the recursive procedure $Q(i,j,\ell)$ for calculating $2^{b_\ell} q^{(\ell)}(i,j)$, given in Figure 2. Let $R(t)$ denote the running time of $Q(i,j,\log_2(t))$. We now bound $R(t)$ from above. Let c be a constant which is equal to the maximum of $R(1)$ and the cost of the overheads involved in a call to the procedure Q.

Then $R(2t) \leq c + 2^{n+1}R(t)$ for all t. Using induction we can show that

$$R(t) \leq c\,\frac{(2t)^{n+1} - 1}{2^{n+1} - 1} < 2c\,t^{n+1}$$

for $t = 2^\ell$, $\ell \geq 0$. Therefore $R(t) = O(t^{n+1})$ for all t, proving that the running time is polynomial.

Now consider the storage requirement of this procedure. It takes at most $b_\ell + 1$ bits to store $2^{b_\ell} q^{(\ell)}(i,j)$. At the ℓth level, the procedure needs $2(b_{\ell-1}+1)$ bits to store $Q(i,k,\ell-1)Q(k,j,\ell-1)$ and at most $n + b_\ell + 2$ bits to store the cumulative sum of these. Hence the total storage requirement is at most

$$\sum_{\ell=1}^{\log_2(t)} (2_b\ell + n + 4) < (n+4)\log_2(t) + 4st.$$

This is polynomial in n, as required.

Finally, we show how the procedure $Q(i,j,\ell)$ can be used to sample from the distribution r. Let $b = st = b_{\log_2(t)}$, so $b = b_\ell$ where $\ell = \log_2(t)$. Suppose that $\delta = 2^{-k}$ for some polynomially bounded integer k. We divide the interval $\{0, \ldots, 2^b Z - 1\}$ into N subintervals, where the ith subinterval has length $2^b Z r_i$. We can express $2^b Z\, r_i$ as

$$2^b Z\, r_i = 2^{k+b}\, M_i - Z(2^k - 1)\, Q(1,i,\ell).$$

This shows that $2^b Z\, r_i$ is an integer. Let $d = \lceil \log_2(Z) \rceil$. To sample from the distribution r, we repeatedly choose $X \in \{0, \ldots, 2^{d+b} - 1\}$ as a string of $d + b$ random bits, until $X < 2^b Z$. We output i if and only if X lies in the ith subinterval. The procedure is given in pseudocode in Figure 3 below.

The space requirements of this procedure are polynomial, as is easily checked. With overheads ignored, the expected value of the running time is

$$R(t) \sum_{i=1}^{N} i\, r_i \leq NR(t).$$

This completes the description of the implementation of the perfect sampling algorithm. It remains only to show that the expected running time is polynomial. Let T be a random variable which is the running time of the perfect sampler. Then the expected value of T satisfies

$$\mathbf{E}(T) \leq (1-\delta)t + \delta NR(t).$$

Clearly this expression is polynomial if and only if $\delta NR(t)$ is polynomial. Since $\mathcal{M}$ is rapidly mixing, there are known polynomials p, q such that

$$t \leq p(n)q(\log(\varepsilon^{-1})).$$

```
Begin.
   repeat
      generate X ∈ {0, ..., 2^{d+b} − 1} as a string of d + b random bits;
   until X < 2^b Z;
   c_M := 2^{b+k};
   c_Q := (2^k − 1)Z;
   Msum := 0;
   Qsum := 0;
   i := 0;
   while Msum − Qsum ≤ X do
      i := i + 1;
      Msum := Msum + c_M · M_i;
      Qsum := Qsum + c_Q · Q(1, i, t);
   endwhile;
   return i;
End.
```

Figure 3: A procedure for sampling from the distribution r

Therefore

$$\delta NR(t) \leq \frac{g}{f\varepsilon^{-1}}(2p(n))^{n+1}q(\log(\varepsilon^{-1}))^{n+1},$$

writing δ in terms of ε. Now

$$\lim_{\varepsilon \to 0} \frac{\log(\varepsilon^{-1})^c}{\varepsilon^{-1}} = 0$$

for any constant c. Therefore, by choosing ε small enough, we can make the quantity $\delta NR(t)$ as small as we like. In particular, let $\varepsilon = n^{-Kn}$ where $K = \deg(p) + 2\deg(q) + 1$. Then it is not difficult to show that $\delta NR(t) \leq g/fn^K$, which is polynomial.

One technical point remains. If $\varepsilon = n^{-Kn}$ then δ is unlikely to equal to some negative power of two. This can be easily addressed. Let $\delta = 2^{-k}$, where

$$k = \lceil \log_2(f/(\varepsilon g) + 1) \rceil,$$

and let $\varepsilon' = f\delta/(g(1-\delta))$. Then

$$\varepsilon' = \frac{f}{g(2^k - 1)} \leq \varepsilon,$$

so ε' is small enough to ensure that $\delta NR(t)$ is polynomial. Moreover, it is not difficult to show that

$$\varepsilon' \geq (2\varepsilon^{-1} + g/f)^{-1},$$

which is not doubly exponentially small. So all requirements are satisfied if we use ε' instead of ε.

We have shown how to transform an efficient approximate sampler into a weak perfect sampler. Moreover, by increasing t if necessary we can assume that $p_t(i) > 0$ for all $i \in \Omega$. This ensures that there does not exist a state i which cannot be output in polynomial time.

A possible criticism of our algorithm is that, in order to claim that the sample is perfect, the user must really have been prepared to carry out a huge amount of computation in the second phase, even though this occurs with exponentially small probability. However, other perfect sampling methods such as CFTP suffer from precisely this problem, even in the monotone case, as illustrated in Example 7.1 above. On the other hand, this algorithm does have one advantage over CFTP. It runs in deterministic polynomial space, whereas CFTP may require unbounded space to be available if truly perfect samples are to be output.

We make one final remark on the topic of perfect sampling. Consider the following highly restrictive model of computation. An algorithm has access to a tape, which stores the details of a distribution ρ on the set $[2^n]$. The distribution has the form

$$\rho_1 = 1 - 2^{-n}, \qquad \rho_j = 2^{-n}, \qquad \rho_i = 0 \text{ otherwise},$$

where $j \in \{2^{n-1} + 1, \ldots, 2^n\}$ is selected uniformly at random beforehand. The tape head is positioned at cell 1, and the tape can only be read sequentially. In this model, weak perfect sampling is possible, as follows. With probability $1 - 2^{-n}$, output 1. Otherwise, read through the tape and output j when you find it. This procedure has expected running time $\mathbf{E}(T) \leq 2$. However, no algorithm can be guaranteed to perform good sampling, since otherwise we may demand variation distance at most 3^{-n}. But in polynomial time the algorithm can only check a polynomial number of values of ρ_i, and so the algorithm is not be able to find j. The best the algorithm can do is guess. The probability that it guesses correctly is $2^{-(n-1)}$. Whenever it guesses incorrectly, the total variation distance between its output and ρ is greater than $2^{-(n+1)}$, which is greater than 3^{-n}. Hence with probability $1 - 2^{-(n-1)}$ the algorithm fails, uniformly over all choices of j.

This shows that good sampling is harder than weak perfect sampling, at least in this very severe model of computation. It would be very interesting to know whether a similar result can be proved for some more realistic model of computation.

8 Negative results

There are many problems for which good sampling or integration do not seem possible. For these problems, we seek proofs of impossibility. If the decision problem of determining whether there is an ω at which $W(\sigma, \omega) > 0$

is NP-complete, then clearly this is immediate since (presumably) we cannot access the sample space in polynomial time. Otherwise, the "magnification" technique from [50] exploits the existence of an embedded NP-complete problem to show that we can only access a negligible portion of the sample space. See [78, pp. 295–296] for a discussion. In more complex situations, we may still be able to show that good sampling or integration implies a (randomized or deterministic) polynomial time algorithm for some NP-complete problem. These are the most satisfactory types of results, but in other situations we are less ambitious, and simply focus on proving that the MCMC approach is unlikely to succeed. Here we attempt to show that some natural MCMC algorithm, or class of algorithms has mixing time superpolynomial in problem size.

We do not review all results in this area. These may be found in the work cited in Section 4 or in [78, pp. 295–296]. Instead, we examine two typical results which have been obtained recently for the problem of sampling or counting in $\mathcal{I}(G)$. These correspond to taking $\lambda = 1$ in the measure defined in Example 3.1. It follows from Example 6.1 that a good sampler is available for $\lambda = 1$ if $\Delta \leq 4$. This implies the existence of a good approximate counter for $\mathcal{I}(G)$. The question then arises as to whether $\Delta \leq 4$ is an artefact of the analysis, or is there some constant Δ for which sampling (or approximate counting) is impossible? This question was partially answered by Luby and Vigoda [62], who showed that there *exists* a constant value for which approximate counting is impossible, unless $\mathsf{P} = \mathsf{NP}$. This could be made quantitative, using more recent results of Berman and Karpinski [5], but the best Δ obtainable in this way seems to be around 8000. However, Dyer, Frieze and Jerrum [25] adopted a different (though related) approach, and proved the following stronger result.

Theorem 8.1 *If $\Delta \geq 25$ then there is no polynomial time algorithm which can approximate the logarithm of the number of independent sets in a Δ-regular graph to within relative error 10^{-6} unless $\mathsf{P} = \mathsf{NP}$ (under randomized reductions).*

(By "randomized reductions", we mean that the algorithm used to transform an instance of problem A to problem B may be randomized.)

Theorem 8.1 clearly also rules out the possibility of any good sampler when $\Delta \geq 25$. We omit the proof, but simply remark that the result rests on recent deep work in the complexity of approximate optimization. In particular, it relies on work of Håstad [43].

However, there is a clear gap between what is known to be possible ($\Delta \leq 4$) and this impossibility result ($\Delta \geq 25$). In this region, the following weaker result is proved in [25].

Theorem 8.2 Let $\Delta \geq 6$ and $b(n) \leq 0.35n$. There exists a family of bipartite graphs $G_0(n)$ on $n + n$ vertices, and maximum degree Δ, with the following property. Any Markov chain $\mathcal{M}$ on $\mathcal{I}(G_0)$ which changes the status of at most $b(n)$ vertices at each step has mixing time $\Omega(e^{\gamma n})$, for some constant $\gamma \geq 0$.

Again we omit the proof, remarking only that it relies principally on random graph methods.

If $\Delta \geq 6$, Theorem 8.2 rules out any MCMC good sampler which extends (say) INSERT/DELETE/DRAG, but changes any constant number of vertices at one step. It rules out even faster moving chains. The only possibility is a chain which could change the status of very many vertices at a single step. However, it is difficult to see how a step of such a chain could be implemented in polynomial time.

The reader may observe that Theorem 8.2 still leaves an intriguing gap. What can be said for $\Delta = 5$? At the time of writing this remains open, but we conjecture that a good sampler does exist in this case.

Acknowledgements

We thank Mark Jerrum, Mark Huber and Russ Bubley for commenting on a draft of this paper and suggesting some improvements. This work was sponsored by the ESPRIT Working Group RAND2.

References

[1] D. Aldous, A random walk construction of uniform spanning trees and uniform labelled trees, *SIAM Journal on Discrete Mathematics*, **3** (1990), 450–465.

[2] D. Aldous & J. Fill, *Reversible Markov chains and random walks on graphs*, Monograph in preparation, available from `http://www.stat.berkeley.edu/users/aldous/book.html`, Department of Statistics, University of California, Berkeley.

[3] M. Bellare, O. Goldreich & E. Petrank, Uniform generation of NP-witnesses using an NP-oracle, *Electronic Colloquium on Computational Complexity*, (1994), Report number TR98-032.

[4] J. van den Berg & J. E. Steif, Percolation and the hard-core lattice gas model, *Stochastic Processes and their Applications*, **49** (1994), 179–197.

[5] P. Berman & M. Karpinski, On some tighter inapproximability results, preprint, Department of Computer Science, University of Bonn, 1998.

[6] A. Broder, Generating random spanning trees, in *30th Annual Symposium on Foundations of Computer Science,* IEEE, Los Alimitos (1989), pp. 442–447.

[7] A. Broder, A. Frieze & E. Upfal, Static and dynamic path selection on expander graphs: a random walk approach, in *29th Annual Symposium on Theory of Computing,* ACM, New York (1997), pp. 531–539.

[8] S. Brooks (Administrator), MCMC Preprint Service, Web site, `http://www.stats.bris.ac.uk/MCMC`.

[9] R. Bubley & M. Dyer, Path coupling: a technique for proving rapid mixing in Markov chains, in *38th Annual Symposium on Foundations of Computer Science,* IEEE, Los Alimitos (1997), pp. 223–231.

[10] R. Bubley & M. Dyer, Graph orientations with no sink and an approximation for a hard case of #SAT, in *8th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1997), pp. 248–257.

[11] R. Bubley & M. Dyer, Faster random generation of linear extensions, in *9th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1998), pp. 350–354.

[12] R. Bubley, M. Dyer & C. Greenhill, Beating the 2Δ bound for approximately counting colourings: a computer-assisted proof of rapid mixing, in *9th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1998), pp. 355–363.

[13] R. Bubley, M. Dyer & M. Jerrum, An elementary analysis of a procedure for sampling points in a convex body, *Random Structures and Algorithms*, **12** (1998), 213–235.

[14] R. Bubley, Randomized algorithms: approximation, generation and counting, Ph D thesis, University of Leeds, 1998.

[15] K. Burdzy & W. Kendall, Efficient Markovian couplings: examples and counterexamples, Technical Report 331, Department of Statistics, University of Warwick, 1998.

[16] F. Chung & R. Graham, Random walks on generating sets for finite groups, *Electronic Journal of Combinatorics*, **4** (1997), Research paper 7.

[17] F. Chung, R. Graham & S. Yau, On sampling with Markov chains, *Random Structures and Algorithms*, **9** (1996), 55–77.

[18] F. Chung & S. Yau, Eigenvalue inequalities for graphs and convex subgraphs, *Communications in Analysis and Geometry*, **5** (1997), 575–623.

[19] C. Cooper & A. Frieze, Mixing properties of the Swendsen–Wang process on classes of graphs, (Preprint), Department of Mathematical Sciences, University of North London, 1998.

[20] A. Czumaj, P. Kanarek, M. Kutyłowski & K. Loryś, Delayed path coupling and generating random permutations via distributed stochastic processes, in *10th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1999), pp. 271–280.

[21] P. Diaconis & L. Saloff-Coste, Walks on generating sets of abelian groups, *Probability Theory and Related Fields*, **105** (1996), 393–421.

[22] P. Diaconis & L. Saloff-Coste, Logarithmic Sobolev inequalities for finite Markov chains, *The Annals of Applied Probability*, **6** (1996), 695–750.

[23] R. L. Dobrushin, The description of a random field by means of conditional probabilities and conditions of its regularity, *Theory of Probability and its Applications*, **13** (1968), 197–224.

[24] W. Doeblin, Exposé de la théorie des chaines simples constantes de Markov à un nombre fini d'états, *Revue Mathématique de l'Union Interbalkanique*, **2** (1933), 77–105.

[25] M. Dyer, A. Frieze & M. Jerrum, On counting independent sets in sparse graphs, (Preprint), School of Computer Studies, University of Leeds, 1998.

[26] M. Dyer, A. Frieze & R. Kannan, A random polynomial-time algorithm for approximating the volume of convex bodies, *Journal of the ACM*, **38** (1991), 1–17.

[27] M. E. Dyer & A. M. Frieze, Computing the volume–a case where randomization provably helps, in *Probabilistic Combinatorics and its Applications* (ed. B. Bollobás), *Proceedings of Symposia in Applied Mathematics*, 44, American Mathematical Society, Providence, RI (1991), pp. 123–170.

[28] M. Dyer & C. Greenhill, A genuinely polynomial-time algorithm for sampling two-rowed contingency tables, in *25th International Colloquium on Automata, Languages and Programming,* Springer, Berlin (1998), pp. 339–350.

[29] M. Dyer & C. Greenhill, On Markov chains for independent sets, (Preprint), School of Computer Studies, University of Leeds, 1997.

[30] M. Dyer & C. Greenhill, A more rapidly mixing Markov chain for graph colourings, *Random Structures and Algorithms*, **13** (1998), 285–317.

[31] M. Dyer, R. Kannan & J. Mount, Sampling contingency tables, *Random Structures and Algorithms*, **10** (1997), 487–506.

[32] S. Felsner & L. Wernisch, Markov chains for linear extensions: the two-dimensional case, in *8th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1997), pp. 239–247.

[33] J. A. Fill, An interruptible algorithm for perfect sampling via Markov chains, *Annals of Applied Probability*, **8** (1998), 131–162.

[34] A. Frieze, M. Jerrum, M. Molloy, R. Robinson, N. Wormald, Generating and counting Hamiltonian cycles in random regular graphs, *Journal of Algorithms*, **21** (1996), 176–198.

[35] L. Goldberg & M. Jerrum, The "Burnside process" converges slowly, (Preprint), Department of Computer Science, University of Edinburgh, 1998.

[36] V. Gore & M. Jerrum, The Swendsen–Wang process does not always mix rapidly, in *29th Annual Symposium on Theory of Computing,* ACM, New York (1997), pp. 674–681.

[37] V. Gore, M. Jerrum, S. Kannan, Z. Sweedyk & S. Mahaney, A quasi-polynomial-time algorithm for sampling words from a context-free language, *Information and Computation*, **134** (1997), 59–74.

[38] C. Greenhill, The complexity of counting colourings and independent sets in sparse graphs and hypergraphs, (Preprint), School of Computer Studies, University of Leeds, 1998.

[39] D. Griffeath, A maximal coupling for Markov chains, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebeite*, **31** (1975), 95–106.

[40] S. Guattery, T. Leighton & G. Miller, The path resistance method for bounding λ_2 of a Laplacian, in *8th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1997), pp. 201–210.

[41] O. Häggström & K. Nelander, On exact simulation of Markov random fields, using coupling from the past, (Preprint), Department of Mathematical Statistics, Chalmers University of Technology, 1997.

[42] O. Häggström & K. Nelander, Exact sampling from anti-monotone systems, *Statistica Neerlandica*, in press.

[43] J. Håstad, Some optimal inapproximability results, in *29th Annual Symposium on Theory of Computing,* ACM, New York (1997), pp. 1–10.

[44] D. Hernek, Random generation of $2 \times n$ contingency tables, *Random Structures and Algorithms*, **13** (1998), 71–79.

[45] M. Huber, Exact sampling and approximate counting techniques, in *30th Annual Symposium on Theory of Computing,* ACM, New York (1998), pp. 31–40.

[46] M. Huber, Exact random sampling from independent sets, (Preprint), School of Operations Research and Industrial Engineering, Cornell University, 1998.

[47] M. Jerrum, A very simple algorithm for estimating the number of k-colorings of a low-degree graph, *Random Structures and Algorithms*, **7** (1995), 157–165.

[48] M. Jerrum, Mathematical foundations of the Markov chain Monte Carlo method, in *Probabilistic Methods for Algorithmic Discrete Mathematics* (eds. M. Habib, C. McDiarmid, J. Ramirez-Alfosin & B. Reed), Springer-Verlag, Berlin (1998), pp. 116–165.

[49] M. Jerrum & A. Sinclair, The Markov chain Monte Carlo method: an approach to approximate counting and integration, in *Approximation Algorithms* (ed. D. S. Hochbaum), PWS Publishing Company, Boston (1996), pp. 482–520.

[50] M. R. Jerrum, L. G. Valiant & V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, *Theoretical Computer Science*, **43** (1986), 169–188.

[51] V. E. Johnson, Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths, *Journal of the American Statistical Association*, **91** (1996), 154–166.

[52] R. Kannan, Markov chains and polynomial time algorithms, in *35th Annual Symposium on Foundations of Computer Science,* IEEE, Los Alimitos (1994), pp. 656–671.

[53] R. Kannan & G. Li, Sampling according to the multivariate normal density, in *36th Annual Symposium on Foundations of Computer Science,* IEEE, Los Alimitos (1996), pp. 204–212.

[54] R. Kannan, L. Lovász & M.Simonovits, Random walks and an $O^*(n^5)$ volume algorithm, *Random Structures and Algorithms*, **11** (1997), 1–50.

[55] R. Kannan, P. Tetali & S. Vempala, Simple Markov chain algorithm for generating bipartite graphs and tournaments, in *8th Annual Symposium on Discrete Algorithms,* ACM–SIAM, New York–Philadelphia (1997), pp. 193–200.

[56] R. Kannan & S. Vempala, Sampling lattice points, in *29th Annual Symposium on Theory of Computing,* ACM, New York (1997), pp. 696–700.

[57] A. Karzanov & L. Khachiyan, On the conductance of order Markov chains, *Order*, **8** (1991), 7–15.

[58] W. Kendall, Perfect simulation for the area-interaction point process, in *Probability towards 2000*, Springer, Berlin (1998), pp. 218–234.

[59] T. Lindvall, *Lectures on the Coupling Method*, Wiley-Interscience, New York (1992).

[60] L. Lovász & P. Winkler, Exact mixing in an unknown Markov chain, *Electronic Journal of Combinatorics*, **2** (1995), Research paper 15.

[61] M. Luby, D. Randall & A. Sinclair, Markov chain algorithms for planar lattice structures, in *36th Annual Symposium on Foundations of Computer Science*, IEEE, Los Alimitos (1995), pp. 150–159.

[62] M. Luby & E. Vigoda, Approximately counting up to four, in *29th Annual ACM Symposium on Theory of Computing*, ACM, New York (1997), pp. 682–687.

[63] N. Madras & D. Randall, Factoring graphs to bound mixing rates, in *37th Annual Symposium on Foundations of Computer Science*, IEEE, Los Alimitos (1996), pp. 194–203.

[64] C. H. Papadimitriou, *Computational Complexity*, Addison–Wesley, Reading, MA (1994).

[65] M. Peinado & T. Lengauer, Random generation of embedded graphs and an extension to Dobrushin uniqueness, in *30th Annual Symposium on Theory of Computing,* ACM, New York (1998), pp. 176–185.

[66] J. W. Pitman, On coupling of Markov chains, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebeite*, **35** (1976), 315–322.

[67] J. G. Propp & D. B. Wilson, Exact sampling with coupled Markov chains and applications to statistical mechanics, *Random Structures and Algorithms*, **9** (1996), 223–252.

[68] J. G. Propp & D. B. Wilson, How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph, *Journal of Algorithms*, **27** (1998), 170–217.

[69] J. Propp & D. Wilson, Coupling from the past: a user's guide, in *Microsurveys in Discrete Probability* (eds. D. Aldous & J. Propp), (To appear), American Mathematical Society (1998), pp. 181–192.

[70] D. Randall & P. Tetali, Analyzing Glauber dynamics by comparison of Markov chains, in *3rd Latin American Symposium on Theoretical Informatics*, Springer, Campinas, Brazil (1998), pp. 292–304.

[71] J. Salas & A. Sokal, Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem, *Journal of Statistical Physics*, **86** (1997), 551–579.

[72] C. Schnorr, Optimal algorithms for self-reducible problems, in *3rd International Colloquium on Automata, Languages and Programming*, Edinburgh University Press, Edinburgh (1976), pp. 322–337.

[73] A. J. Sinclair, *Algorithms for Random Generation and Counting*, Birkhäuser, Boston (1993).

[74] A. Sokal, Introduction to Monte Carlo, Postscript file, available from `http://math.nyu.edu/faculty/goodman/teaching/Monte\_Carlo/Sokal.ps`

[75] S. Vadhan, The complexity of counting in sparse, regular and planar graphs, (Preprint), Laboratory for Computer Science, MIT, 1997.

[76] L. G. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, **8** (1979), 189–201.

[77] E. Vigoda, personal communication, University of Berkeley, 1997.

[78] D. Welsh, Approximate counting, in *Surveys in Combinatorics 1997* (ed. R. A. Bailey), Cambridge University Press, Cambridge (1997), pp. 287–323.

[79] D. B. Wilson, Mixing times of lozenge tiling and card shuffling Markov chains, (Preprint), Department of Mathematics, MIT, 1997.

[80] D. B. Wilson (Maintainer), Perfectly random sampling with Markov chains, Web site, `http://dimacs.rutgers.edu/~dbwilson/exact.html`.

School of Computer Studies
University of Leeds
Leeds
LS2 9JT
United Kingdom
`dyer@scs.leeds.ac.uk`
`csg@scs.leeds.ac.uk`