On exact simulation of Markov random fields using coupling from the past

Olle Häggström* Karin Nelander

Chalmers University of Technology and University of Göteborg

November 12, 1997

Abstract

A general framework for exact simulation of Markov random fields using the Propp–Wilson coupling from the past approach is proposed. Our emphasis is on situations lacking the monotonicity properties that have been exploited in previous studies. A critical aspect is the convergence time of the algorithm; this we study both theoretically and experimentally. Our main theoretical result in this direction says, roughly, that if interactions are sufficiently weak, then the expected running time of a carefully designed implementation is $O(N \log N)$, where N is the number of interacting components of the system. Computer experiments are carried out for random q-colourings and for the Widom–Rowlinson lattice gas model.

1. Introduction

A major trend in probability and statistics in the 1990's is the extensive use of **Markov chain Monte Carlo** (MCMC) methods to sample from complicated multivariate probability distributions. This is particularly the case in image analysis, spatial statistics, and a wide variety of Bayesian contexts; see e.g. Gilks *et al.* [5] for a broad introduction to the theory and applications of MCMC.

The idea, which dates back at least to the 1953 paper by Metropolis *et al.* [15], is to define some ergodic reversible Markov chain whose unique stationary distribution equals the desired probability measure π. Starting from an arbitrary initial state, one runs the chain until it is close to equilibrium, and outputs the final state whose distribution then is close to π. The construction and computer

*Research supported by a grant from the Swedish Natural Science Research Council.

implementation of such a Markov chain is often straightforward, but the approach has the following drawbacks:

(i) Typically, the Markov chain never actually reaches equilibrium (it only comes arbitrarily close). Therefore, the output will have a nonzero bias no matter how long the chain is run.

(ii) More often than not, it is very difficult to establish rigorous upper bounds on the time taken to come close to equilibrium which are good enough to be of any practical use.

In itself, (i) is not a disastrous problem. Much more serious is (ii), which as a matter of fact challenges the rigour of much of today's MCMC practice. [Actually, there is a third problem (possibly the most serious of all) in that MCMC algorithms require the use of random number generators, whereas in practice of course only pseudo-random number generators are available. Like most researchers in this field, we shall simply ignore this problem, pretending it doesn't exist.]

In one of the truly important MCMC papers in recent years, Propp and Wilson [18] devised an algorithm which simultaneously solves problems (i) and (ii) above by

(i') producing a *completely unbiased* sample from the target distribution π, and

(ii') determining automatically how long it needs to run.

They furthermore demonstrated experimentally that their algorithm is computationally feasible on important examples including the Ising model and certain random tilings. The algorithm is based on so-called **coupling from the past** (CFTP). Loosely speaking, the idea of CFTP is to couple copies of the Markov chain starting in all possible states at some time in the distant past, and to run them until time 0. If all copies have coalesced by time 0, then the value at time 0 does not depend on the starting value, and the output can be shown to be unbiased. The algorithm involves trying earlier and earlier starting times until eventually coalescence has occured. (A more careful description will be given in Sections 3 and 4.)

The work of Propp and Wilson has been followed up and extended in various directions e.g. by Kendall [12], Häggström *et al.* [7], Häggström and Nelander [8], Møller [16] and Murdoch and Green [17]. A completely different algorithm for exact sampling, based on so-called strong uniform times rather than on coupling, has been developed by Fill [1]. In most of these papers (including [18] but with the notable exception of [17]) some monotonicity property of the target distribution π is required in order to get efficient exact sampling.

In the present paper, we will discuss the possibility of exact sampling using CFTP in the context of **Markov random fields**, particularly in the case where

no obvious monotonicity structure is present. Markov random fields are widely used as stochastic models e.g. in image analysis [2, 23], spatial statistics [9], and statistical mechanics [3, 6]. The algorithm we present is a generalization of the monotone CFTP algorithm of Propp and Wilson. In our general setup, the exact way in which the chains are coupled will not be specified, thus leaving plenty of room for adapting the algorithm to best suit particular contexts. It is probably not realistic to expect that a general rule can be devised for how the coupling should be constructed in order to be as efficient as possible, but we shall try to give some reasonable advise in particular cases.

Here is a quick outline of the rest of this paper. In Section 2, we provide the context by recalling the definition of a Markov random field, and giving some examples. The CFTP algorithm is presented in Section 3, where we also give conditions which guarantee its convergence. In Section 4 we point out how this algorithm generalizes those of [18] and [8], and in Section 5 we give a simple example which demonstrates the necessity of running the chains "from the past" rather than "to the future". Section 6 gives a brief discussion of the possibility of replacing the Gibbs sampler by Metropolis–Hastings chains in the CFTP setup, and the last two sections treat the issue of computational complexity, first via a rigorous result (Section 7), and then experimentally (Section 8). Some final remarks are made in Section 9.

2. Markov random fields

We here give a very brief introduction to Markov random fields; the reader may turn e.g. to [13, 23, 6] for more extensive discussions. We consider stochastic systems living on a finite graph G with vertex set V and edge set E. For two vertices $u, v \in V$ we write $u \sim v$ to indicate the existence of an edge connecting u and v. Let S be some finite set. Each vertex $v \in V$ can be in any state $s \in S$, chosen in some random fashion, so the random objects we are interested in take their values in S^V. We write $\xi, \omega, \ldots$ for fixed elements of S^W, where W can be any subset of V, and we furthermore write $X, X_1, \ldots$ for S^V-valued random objects. For $W \subseteq V$, we define the **boundary** ∂W of W as the set

$$\partial W = \{v \in V \setminus W : \exists w \in W \text{ such that } v \sim w\}.$$

Definition 2.1. *An S^V-valued random element X with distribution π is said to be a **Markov random field** on G if for each $W \subseteq V$ the conditional distribution of $X(W)$ given $X(V \setminus W)$ depends on $X(V \setminus W)$ only through its values on ∂W. In other words, X is a Markov random field if for all $W \subseteq V$, $\omega \in S^W$, $\omega' \in S^{V \setminus W}$ and all $\omega'' \in S^{\partial W}$ such that ω'' is the restriction of ω' to ∂W and $\pi(X(V \setminus W) = \omega') > 0$, we have*

$$\pi(X(W) = \omega \,|\, X(V \setminus W) = \omega') \quad = \quad \pi(X(W) = \omega \,|\, X(\partial W) = \omega'').$$

Loosely speaking, X is a Markov random field if the values of X at different vertices can only depend on each other via nearest neighbour interactions.

Mathematically, there is of course no loss of generality in assuming that X is a Markov random field, since if S and V are finite sets and X is an arbitrary S^V-valued random element, we can trivially turn X into a Markov random field by defining G to be the graph with vertex set V and edge set consisting of all pairs of vertices in V. The point of discussing CFTP in a Markov random field context is more of a practical than of a mathematical nature: In typical applications, each vertex will have relatively few neighbours in G even if the whole system is quite large (for instance, in image analysis, G may be a portion of the square lattice of size (say) 512×512, in which each vertex has at most four neighbours). The algorithms will involve updating the value at a single vertex $v \in V$ according to its conditional distribution (under the target distribution π) given $X(V \setminus v)$, and this is obviously computationally more feasible if the conditional distribution only depends on neighbours of v. The advantage of this is even greater in the CFTP context (i.e. when coupling many copies of the chain) than when only a single chain is involved.

We now go on to describe some examples of Markov random fields.

Example 2.1: Random q-colourings. Let $q \geq 2$ be an integer, and let $S = \{1, \ldots, q\}$. A configuration $\omega \in S^V$ is said to be a q-colouring of G if no two adjacent vertices have the same "colour", i.e. if $\omega(v) \neq \omega(w)$ whenever $v \sim w$. A random q-colouring X is simply a q-colouring of G chosen uniformly at random. For each $W \subseteq V$ and each $\omega' \in S^{V \setminus W}$, the conditional distribution of $X(W)$ given $X(V \setminus W) = \omega'$ is uniform over the set of all $\omega \in S^W$ that do not produce any pairs of neighbouring vertices having the same value (either within W or between W and $V \setminus W$). This conditional distribution depends on ω' only via $\omega'(\partial W)$, so X is a Markov random field. The main interest in random q-colourings is combinatorial, but they have also been studied in statistical mechanics where they are thought of as "zero temperature limits" of antiferromagnetic Potts models (see Example 2.3). An MCMC algorithm for generating random q-colourings is discussed by Jerrum [10].

Example 2.2: The Ising model. Possibly the most studied of all Markov random fields is the Ising model, which was first introduced in statistical mechanics as a model for spontaneous magnetization in ferromagnetic materials. The vertices are thought of as atoms, and the state space $S = \{-1, 1\}$ is thought of as representing two different spin orientations. The Ising measure ν_G^J for G with coupling constant $J \in \mathbb{R}$ is the probability measure on $\{-1, 1\}^V$ which to each configuration $\xi \in \{-1, 1\}^V$ assigns probability

$$\nu_G^J(\xi) = \frac{1}{Z_G^J} \exp\Big(-2J \sum_{\substack{x,y \in V \\ x \sim y}} \mathbf{1}_{\xi(x) \neq \xi(y)} \Big) \tag{2.1}$$

where Z_G^J is a normalizing constant. That this defines a Markov random field follows easily from the fact that the right hand side of (2.1) can be factorized into factors involving only the states of neighbouring vertices. The case $J > 0$, in which neighbouring vertices tend to agree (take the same value), is referred to as the ferromagnetic Ising model, and the case $J < 0$, where they tend to disagree, is called antiferromagnetic. See e.g. [3] or [6] for further general discussion. The Ising model was among the first models to be successfully simulated using CFTP; see [18].

Example 2.3: The Potts model. A natural generalization of the previous two examples is the q-state Potts model. Here $S = \{1, \ldots, q\}$ for some $q \geq 2$. The Potts measure $\nu_G^{q,J}$ with coupling constant $J \in \mathbb{R}$ assigns, to each configuration $\xi \in S^V$, probability

$$\nu_G^{q,J}(\xi) = \frac{1}{Z_G^{q,J}} \exp\Big(- 2J \sum_{\substack{x,y \in V \\ x \sim y}} \mathbf{1}_{\xi(x) \neq \xi(y)} \Big).$$

For the same reason as in Example 2.2, this gives rise to a Markov random field. Taking $q = 2$ and identifying $\{1, 2\}$ with $\{-1, 1\}$, we retrieve the Ising model. Fixing q and letting $J \to -\infty$, we get the random q-colouring in the limit. We refer to [3] or [6] for general discussions and to [19] for a discussion of the $J \to -\infty$ limit.

Example 2.4: The hard-core model. Let $S = \{0, 1\}$, and consider for $a > 0$ the $\{0, 1\}^V$-valued random object which arises by letting the vertices independently be in state 1 (resp. 0) with probability $\frac{a}{a+1}$ (resp. $\frac{1}{a+1}$), and then conditioning on the event that no two adjacent vertices both have value 1. This is the so called hard-core model, which is of interest both in statistical mechanics [3] and in operations research [11]. Again, we get a Markov random field. The statistical mechanics interpretation of the model is that the 1's are non-penetrable gas particles which would overlap if they were located at adjacent vertices. CFTP simulations of the hard-core model appear in [8].

Example 2.5: The Widom–Rowlinson lattice gas model. In 1970, Widom and Rowlinson [21] introduced a model for a gas with two different types of particles living in $\mathbb{R}^d$. Lebowitz and Gallavotti [14] soon thereafter introduced a discrete variant, a multi-type version of which is defined as follows. Fix $a > 0$ and an integer $q \geq 2$, and let $S = \{0, 1, \ldots, q\}$. Consider the S^V-valued random configuration X which arises by letting each vertex independently pick a value from $\{0, 1, \ldots, q\}$ according to the probability distribution $(\frac{1}{1+qa}, \frac{a}{1+qa}, \ldots, \frac{a}{1+qa})$, and conditioning on the event that for each pair of neighbouring vertices $x, y \in V$ we have either $X(x) = X(y)$ or $X(x)X(y) = 0$. The value 0 at a vertex should be thought of as "empty", and $1, \ldots, q$ should be thought of as q different types

of particles. With this interpretation, the conditioning amounts to saying that no two particles of different type are allowed to sit next to each other. As in the previous examples, only nearest neighbours interact directly, so we have a Markov random field. Most studies of this and related models concentrate on the binary gas which arises by taking $q = 2$; see [4] for a treatment of the (continuum) Widom–Rowlinson model with arbitrary q.

3. The algorithm

3.1. The Gibbs sampler

We start this section with a description of the Gibbs sampler, as it is a key ingredient in CFTP. Suppose we want to simulate a random element X taking its values in S^V according to the probability measure π. Suppose further that the conditional probabilities $\pi(X(v) = i \mid X(V \setminus \{v\}) = \xi)$ are easily available for all possible combinations of $v \in V$, $i \in S$ and $\xi \in S^{V \setminus \{v\}}$. The Gibbs sampler is a discrete time Markov chain $\{X_t\}$ with state space S^V and the following evolution. At each integer time t, a location $v \in V$ is chosen to be updated. For definiteness, we take the mechanism for picking locations to be to pick v at random (uniformly) from V, independently for each t. (Other mechanisms, such as cycling deterministically through V, are sometimes preferred elsewhere in the MCMC literature.) The values at vertices $w \in V \setminus \{v\}$ are left unchanged and the new value at vertex v is chosen according to the correct conditional probability:

$$\begin{array}{rcll} X_{t+1}(w) & = & X_t(w), & w \in V \setminus \{v\} \\ X_{t+1}(v) & \stackrel{\mathcal{D}}{=} & \pi(\cdot | X_t(V \setminus \{v\})). & \end{array} \tag{3.1}$$

If it is the case that the set of elements of S^V with positive π-measure is connected (where two elements of S^V are thought of as adjacent if they differ only at one vertex), then the resulting Markov chain $\{X_t\}$ is irreducible and aperiodic, hence ergodic. Clearly the Markov chain has π as its stationary distribution, and therefore the distribution of X_t converges to π as $t \to \infty$, regardless of the starting value X_0.

In practice (i.e. in computer implementations) the updating is realized using a random number which is uniformly distributed on $[0, 1]$, and a deterministic function

$$\phi : S^V \times [0, 1] \times V \to S^V$$

such that $X_{t+1} = \phi(X_t, u_t, w_t)$ leaves X_t intact on $V \setminus \{w_t\}$ and updates the value on w_t depending on u_t in such a way that (3.1) holds. There is an enormous choice of such functions; all that is needed is that

$$\int_0^1 \mathbf{1}_{\{[\phi(\xi,u,v)](v)=s\}} du \;=\; \pi(\, X(v) = s \mid X(V \setminus \{v\}) = \xi(V \setminus \{v\})\,) \tag{3.2}$$

for all $v \in V$, $s \in S$ and all possible configurations $\xi \in S^V$. In standard Gibbs sampling, the choice of ϕ-function is of no importance (and is therefore usually implicit) as long as (3.2) holds, whereas in contrast this choice plays an important role in CFTP. A common choice is the so called monotone ϕ-function for which

$$[\phi(\xi, u, v)](v) = \max\{s \in S : \pi(X(v) \geq s | X_t(V \setminus \{v\}) = \xi(V \setminus \{v\})) \geq u\} \quad (3.3)$$

(see Section 4) but sometimes there will be reason to pick ϕ differently.

When X is a Markov random field, ϕ can always be chosen in such a way that $[\phi(\xi, u, v)](v)$ only depends on the values of ξ at the neighbours of v; we will henceforth assume that ϕ is chosen in such a way.

3.2. CFTP

The idea behind the Propp–Wilson coupling from the past algorithm was touched upon in the introduction. Now we will give a precise description. To that end we introduce some notation. Let $\{W_t\}_{t=-1,-2,...}$ and $\{U_t\}_{t=-1,-2,...}$ be independent i.i.d. sequences such that U_t is uniformly distributed on $[0, 1]$ and W_t is uniformly distributed on the vertex set V. For $t_1 < t_2 \leq 0$, let us define

$$\Phi_{t_1}^{t_2}(\xi, (\boldsymbol{u}, \boldsymbol{w})) = \phi(\phi(...(\phi(\xi, u_{t_1}, w_{t_1}), u_{t_1+1}, w_{t_1+1}), ..., u_{t_2-2}, w_{t_2-2}), u_{t_2-1}, w_{t_2-1}),$$

where $(\boldsymbol{u}, \boldsymbol{w})$ is short for $((..., u_{-2}, u_{-1}), (..., w_{-2}, w_{-1}))$. The CFTP algorithm can be expressed in terms of this function: Successively try larger and larger values of t until $\Phi_{-t}^{0}(\xi, (\boldsymbol{u}, \boldsymbol{w}))$ equals one common value for all $\xi \in S^V$. In words this means that one copy of the above described Gibbs sampler is started in each of the possible configurations of S^V at time $-t$ and run until time 0. During this evolution they are coupled in the sense that they are all subject to the same set of random variables $\{U_t\}$ and $\{W_t\}$. The time t needed for coalescence at time 0 is determined by the algorithm itself via successive doublings of the starting times. At first the Gibbs samplers are started at time -1 and run for one step. Their states are compared. If they are not all equal, they are restarted at time -2 and updated twice, and so on. It is of vital importance that for each t, the same realisation of (u_t, w_t) is used every time that time t is reached. The algorithm returns states according to the desired distribution π; this is Theorem 1 of [18], where it is also shown that the technique of successively doubling the starting times to determine the time needed to coalescence, is close to optimal. When π satisfies the monotonicity assumptions of [18], the algorithm can be simplified in such a way that only two Gibbs samplers are needed, one is started in the minimal element of S^V and the other in the maximal element (see Section 4).

In general, it is too demanding (in terms of time and computer memory) to keep exact track of the states of all the Gibbs samplers. We propose instead to keep track of the possible values at each vertex separately, in the following

manner. Let $\mathcal{S}$ be the set of all subsets of S. We will now introduce a $\mathcal{S}^V$-valued Markov chain $\{\mathcal{X}_t\}$. An element Ξ in $\mathcal{S}^V$ is interpreted as the set of all configurations $\eta \in S^V$ such that $\eta(v) \in \Xi(v)$ for every vertex v. This means that for every vertex v, $\Xi(v)$ gives the set of possible values at that location. In the beginning all values in S are possible, so at each starting point $-T$ we have $\mathcal{X}_{-T} = S^V$. The further evolution is as follows. At time t a location w_t is chosen to be updated and we draw a value u_t from the uniform distribution on the interval $[0, 1]$. We let

$$\mathcal{X}_{t+1}(w) = \mathcal{X}_t(w), \quad w \neq w_t$$

and

$$\mathcal{X}_{t+1}(w_t) \;=\; \{s \in S : \exists \eta \in \mathcal{X}_t \text{ such that } [\phi(\eta, u_t, w_t)](w_t) = s\}. \tag{3.4}$$

The new value at w_t is the set of all values the Gibbs sampler would have given us for all the possible values of the environment to w_t consistent with $\mathcal{X}_t$. This updating is usually computationally not all that heavy, due to the assumption made in the final paragraph of Section 3.1. More generally, we allow any updating rule for which (3.4) is replaced by

$$\mathcal{X}_{t+1}(w_t) \;\supseteq\; \{s \in S : \exists \eta \in \mathcal{X}_t \text{ such that } [\phi(\eta, u_t, w_t)](w_t) = s\}. \tag{3.5}$$

This may be more demanding in terms of the starting time $-T$, but the point is that in certain cases it may save time on each iteration, compared to (3.4).

The idea is now to run the $\{\mathcal{X}_t\}$-chain from earlier and earlier starting times (using the successive doublings as above) until there is, by time 0, only one value in every vertex, that is we only have one possible configuration $\eta \in S^V$, which is then taken to be the output of the algorithm. It must then be the case that, regardless of its starting configuration, the Gibbs sampler started sufficiently early and subject to the same W_t and U_t variables would have ended in this same configuration.

In analogy with the Φ-function for the simple Gibbs sampler, we now introduce the function Ψ. $\Psi_{t_1}^{t_2}(\Xi, (\boldsymbol{u}, \boldsymbol{w}))$ will give the state at time t_2 of the $\{\mathcal{X}_t\}$-chain, if it at time t_1 was in state $\Xi \in \mathcal{S}^V$ and during the evolution from t_1 to t_2 was subject to $(\boldsymbol{u}, \boldsymbol{w})$. Note that

$$\cup_{\xi \in S^V} \Phi_{t_1}^{t_2}(\xi, (\boldsymbol{u}, \boldsymbol{w})) \;\subseteq\; \Psi_{t_1}^{t_2}(S^V, (\boldsymbol{u}, \boldsymbol{w})) \tag{3.6}$$

and that there may be strict inclusion even if (3.4) is used. For $\Xi \in \mathcal{S}^V$, we write Card$[\Xi]$ for the number of elements $\xi \in S^V$ that are consistent with Ξ.

Theorem 3.1. *Suppose that there exits an $n < \infty$ such that*

$$P(\mathit{Card}\,[\mathcal{X}_n] = 1 \mid \mathcal{X}_0 = S^V) \;>\; 0. \tag{3.7}$$

Then the above described algorithm terminates a.s. and produces an unbiased sample from π.

The proof of this theorem follows very closely the pattern of the proofs of Theorems 2.1 and 2.2 of [8], however the setup of the present theorem is considerably more general. We first recall the concept of distance in total variation $||\mu - \mu'||$ between two probability measures μ and μ' on S^V, defined as

$$||\mu - \mu'|| \quad = \quad \max_{E \subseteq S^V} |\mu(E) - \mu'(E)|.$$

Proof of Theorem 3.1. The events $\{\text{Card}[\Psi_{-kn}^{-(k-1)n}(S^V, (\boldsymbol{U}, \boldsymbol{W}))] = 1\}$ with $k = 1, 2, ...$ are independent and by assumption all have the same positive probability of occurring. Hence with probability 1, at least one of them will occur. This in turn implies that the algorithm terminates almost surely.

Let T_* be the smallest t for which $\text{Card}[\Psi_{-t}^0(S^V, (\boldsymbol{U}, \boldsymbol{W}))] = 1$ and let π' be the distribution of the output of the algorithm. Fix $\epsilon > 0$ and take t so large that $P(T_* > t) < \epsilon$. Suppose that X is picked according to π, independently of $(\boldsymbol{U}, \boldsymbol{W})$. It must be the case that $\Phi_{-t}^0(X, (\boldsymbol{U}, \boldsymbol{W}))$ is distributed according to π, since π is invariant for the Gibbs sampler. Furthermore, (3.6) implies that $\Phi_{-t}^0(X, (\boldsymbol{U}, \boldsymbol{W})) = \Psi_{-T_*}^0(S^V, (\boldsymbol{U}, \boldsymbol{W}))$ for t greater than or equal to T_*. Hence,

$$\begin{aligned} ||\pi - \pi'|| &\leq P(\Phi_{-t}^0(X, (\boldsymbol{U}, \boldsymbol{W})) \neq \Psi_{-T_*}^0(S^V, (\boldsymbol{U}, \boldsymbol{W})) \\ &\leq P(T_* > t) \\ &< \epsilon. \end{aligned}$$

Since $\epsilon > 0$ was arbitrary, we are done. ■

In some situations, we might have that the set of attainable values varies from location to location, i.e. that $\pi(X(v) = s) = 0$ for some choices of v and s. The algorithm can then be improved by the following minor modification. For $v \in V$, define

$$S_v \quad = \quad \{s \in S : \pi(X(v) = s) > 0\}.$$

Each time the $\mathcal{X}_t$-chain is restarted, we may use $\prod_{v \in V} S_v$ rather than S^V as starting value. The analogue of Theorem 3.1 for this variant of the algorithm goes through in the same way as the above proof.

Let us end this section by giving an example of a situation where the condition (3.7) in Theorem 3.1 is easy to check. Suppose that there exists an $s \in S$ and a $\delta > 0$ such that

$$\pi(X(v) = s \mid X(V \setminus \{v\}) = \omega') \quad \geq \quad \delta \tag{3.8}$$

for all $v \in V$ and $\omega' \in S^{V \setminus \{v\}}$ (this holds for all examples in Section 2 except for Example 2.1). We can then define the ϕ-function in such a way that

$$[\phi(\xi, u, w)](w) \quad = \quad s$$

for every $\xi \in S^V$ whenever $u < \delta$. Then (3.7) holds with $n = \text{Card } [V]$, because there is a positive (although usually remote) probability that $w_1, ..., w_n$ will cycle through V and that $u_1, ..., u_n$ will all be smaller than δ, in which case $\mathcal{X}_n$ will equal s^V. Theorem 3.1 then ensures convergence and unbiasedness of the algorithm. On the other hand, the convergence may of course in some cases be too slow for practical usefulness. This issue will be dealt with in Sections 7 and 8.

4. Monotone cases

As mentioned previously, exact simulation becomes particularly simple when certain monotonicity properties of the Markov random field hold. In this section, we shall point out how the CFTP algorithm described in Section 3 generalize the monotone CFTP algorithm of Propp and Wilson [18] and the anti-monotone variant of Häggström and Nelander [8]. In order to define monotonicity and anti-monotonicity, we introduce, for any $W \subseteq V$, the pointwise partial order $\preceq$ on S^W: $\xi \preceq \eta$ if $\xi(v) \leq \eta(v)$ for each $v \in W$.

Definition 4.1. *An S^V-valued random element X with distribution π is said to be **monotone** if for each $v \in V$ and each $s \in S$ we have*

$$\pi(X(v) \leq s \,|\, X(V \setminus \{v\}) = \xi) \;\; \geq \;\; \pi(X(v) \leq s \,|\, X(V \setminus \{v\}) = \eta)$$

whenever $\xi \preceq \eta$, $\pi(X(V \setminus \{v\}) = \xi) > 0$ and $\pi(X(V \setminus \{v\}) = \eta) > 0$. If on the other hand

$$\pi(X(v) \leq s \,|\, X(V \setminus \{v\}) = \xi) \;\; \leq \;\; \pi(X(v) \leq s \,|\, X(V \setminus \{v\}) = \eta)$$

*for all such v, s, η and ξ, then X is said to be **anti-monotone**.*

An example of a monotone Markov random field is the Ising ferromagnet. The Widom–Rowlinson model with $q = 2$ is also monotone if we equip the set $\{0, 1, 2\}$ with the (somewhat unusual!) ordering $1 < 0 < 2$. Examples of anti-monotone systems are the Ising antiferromagnet, the hard-core model, and the somewhat trivial random 2-colouring. Examples of systems which are neither monotone, nor anti-monotone, are the $q \geq 3$ random q-colourings, the $q \geq 3$ Potts models, and the $q \geq 3$ Widom–Rowlinson models.

Let us now assume that X is monotone. Then the monotone ϕ-function defined in (3.3) is particularly suitable for CFTP. Suppose that we are about to update the location $v \in V$ using the variable $u \in [0, 1]$ and the "environment" $\Xi \in \mathcal{S}^{V \setminus \{v\}}$. Write $\xi_{min}(\Xi)$ for the minimal (with respect to $\preceq$) element of $S^{V \setminus \{v\}}$ which is compatible with Ξ, and define $\xi_{max}(\Xi)$ similarly. Since X is monotone, we have

$$[\phi(\xi_{min}(\Xi), u, v)](v) \;\leq\; [\phi(\xi, u, v)](v) \;\leq\; [\phi(\xi_{max}(\Xi), u, v)](v)$$

for all $\xi \in S^{V\setminus\{v\}}$ that are compatible with Ξ. We can therefore update v by assigning it the element

$$\{s \in S : [\phi(\xi_{min}(\Xi), u, v)](v) \leq s \leq [\phi(\xi_{max}(\Xi), u, v)](v)\}$$

of $\mathcal{S}$. Write $\hat{0}$ resp. $\hat{1}$ for the minimal resp. maximal elements of S^V. Starting this version of CFTP at time t_1 with the starting value $\mathcal{S}^V$, we get for each $v \in V$ that

$$[\Psi_{t_1}^{t_2}(\mathcal{S}^V, (\boldsymbol{u}, \boldsymbol{w}))](v) \quad = \quad \{s \in S : [\Phi_{t_1}^{t_2}(\hat{0}, (\boldsymbol{u}, \boldsymbol{w}))](v) \leq s \leq [\Phi_{t_1}^{t_2}(\hat{1}, (\boldsymbol{u}, \boldsymbol{w}))](v)\}.$$

In effect, what this algorithm does is to run just two copies of the Gibbs sampler, one starting with the minimal state $\hat{0}$ and one starting with the maximal state $\hat{1}$ (the value of the $\mathcal{X}_t$-chain at each time is the set of states "sandwiched" between these two chains). This is exactly what the monotone CFTP algorithm of Propp and Wilson [18] does, and we conclude that the Propp–Wilson monotone CFTP algorithm can be seen as a special case of the general algorithm in Section 3. Similar arguments show that the anti-monotone CFTP algorithm (which also uses the monotone ϕ-function (3.3)) described in [8] fits into the setup of Section 3 as well.

The monotone CFTP algorithm has turned out so successful in terms of rapid convergence (see e.g. [18]) that there seems to be little reason to look for other ϕ-functions than (3.3) in monotone cases. An illustration of this can be found in Section 8, where the monotone algorithm turns out to be faster than a certain other attempt for the two-type Widom–Rowlinson lattice gas model. In contrast, running times for anti-monotone CFTP have proved to be somewhat less satisfactory, and it is quite conceivable that it may pay off to look beyond (3.3) in some anti-monotone situations.

5. A counterexample to the forward algorithm

In our experience, one of the most frequently asked questions in audiences that have not previously been confronted with CFTP is the following: Instead of rerunning the chain from earlier and earlier starting times, why not run the chain from time 0 forwards in time until we reach coalescence, and output the state at the time of coalescence? The answer is that this naive "coupling to the future" algorithm produces (in general) biased samples. This, however, is not immediately obvious, so we feel that it is an important pedagogical issue to come up with simple counterexamples that demonstrate biasedness of samples generated by this approach. The counterexamples that have appeared in the literature so far (see [18] and [1]) either require tedious calculation, or do not fit into the Gibbs sampling context in which CFTP is usually formulated. Here we shall present an example which is both extremely simple to analyse and which fits into the general context of Section 3.

Let G be the complete graph on just two nodes v_1 and v_2, and let $S = \{0, 1, 2\}$. Let X be the S^V-valued random object whose distribution π is given by

$$\pi[(X(v_1), X(v_2)) = (i, j)] = \begin{cases} \frac{1}{4}, & (i, j) = (0, 0) \\ \frac{1}{4}, & (i, j) = (0, 1) \\ \frac{1}{4}, & (i, j) = (2, 1) \\ \frac{1}{4}, & (i, j) = (2, 2) \\ 0, & \text{otherwise.} \end{cases}$$

In words, X is uniformly distributed over those elements of $\{0, 2\} \times \{0, 1, 2\}$ in which the two coordinates differ by at most 1. A natural way to implement the Gibbs sampler for this system is as follows. If $X(v_1)$ is to be updated using the random variable U, uniformly distributed on $[0, 1]$, and $X_2(v) = j$, we let

$$X(v_1) = \begin{cases} \min\{i \in \{0, 2\} : \pi(i, j) > 0\} & \text{if } U < \frac{1}{2} \\ \max\{i \in \{0, 2\} : \pi(i, j) > 0\} & \text{if } U \geq \frac{1}{2}. \end{cases}$$

If instead $X(v_2)$ is to be updated, and $X(v_1) = i$, we let

$$X(v_2) = \begin{cases} i & \text{if } U < \frac{1}{2} \\ 1 & \text{if } U \geq \frac{1}{2}. \end{cases}$$

If we run CFTP based on this Gibbs sampler, it is easy to see that Theorem 3.1 applies to show that the algorithm terminates a.s. and produces an unbiased sample from π. If, on the other hand, the naive forward variant of the algorithm is used (i.e. if we start at time 0 and run until coalescence), then $\{\mathcal{X}_t\}_{t=0}^{\infty}$ becomes a stopped $\mathcal{S}^V$-valued Markov chain with the transition probabilities indicated in Figure 1. A quick glance at Figure 1 reveals that the output of this algorithm equals $(0, 1)$ or $(2, 1)$ with probability $\frac{1}{2}$ each. Thus, the distribution of the output differs (drastically) from the target distribution π.

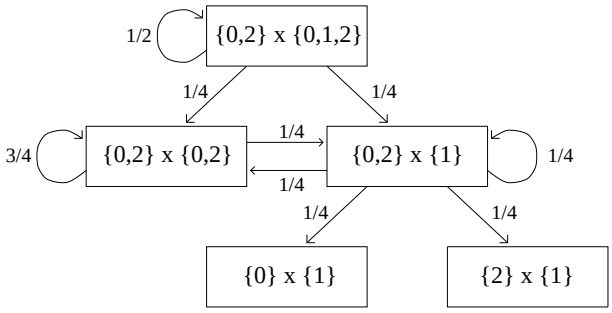

Figure 1: Transition probabilities until coalescence in the $\mathcal{X}_t$-chain.

6. Metropolis–Hastings alternatives

The Gibbs sampler is by no means the only Markov chain which is useful for standard (non-exact) MCMC simulation. A popular alternative is the Metropolis–Hastings algorithm, one version of which has the following evolution. Suppose the chain is in state $X_t = \xi \in S^V$ at time t. Pick $v \in V$ and $s \in S$ independently and uniformly at random, and update the location v by letting

$$X_{t+1}(v) = \begin{cases} s & \text{w.p.} \quad \min\left\{1, \frac{\pi(X(v)=s \mid X(V\setminus\{v\})=\xi(V\setminus\{v\}))}{\pi(X(v)=\xi(v) \mid X(V\setminus\{v\})=\xi(V\setminus\{v\}))}\right\} \\ \xi(v) & \text{w.p.} \quad 1 - \min\left\{1, \frac{\pi(X(v)=s \mid X(V\setminus\{v\})=\xi(V\setminus\{v\}))}{\pi(X(v)=\xi(v) \mid X(V\setminus\{v\})=\xi(V\setminus\{v\}))}\right\}. \end{cases} \tag{6.1}$$

Many variants of this chain have been proposed, and the MCMC literature contains plenty of discussion on what particular chain (the Gibbs sampler, some Metropolis–Hastings variant, or some hybrid algorithm) is most suitable in various settings; see e.g. the discussion paper by Tierney [20].

It is straightforward to modify the CFTP algorithm in Section 3 in such a way that it becomes a coupling of Metropolis–Hastings chains rather than of Gibbs samplers (there will still, however, be an enormous freedom in defining the joint behaviour of the chain, corresponding to the freedom in defining the ϕ-function for the coupled Gibbs samplers). It is very well possible that such Metropolis–Hastings alternatives may turn out to be important in the future development and application of CFTP algorithms. Here, however, we shall merely point out some reasons (specific to the CFTP context) why it makes sense to stick to the Gibbs sampler as opposed to using one of the alternative chains:

(*i*) An essential property (in order for the Propp–Wilson monotone CFTP algorithm to work) of monotone Markov random fields is that if two Gibbs samplers $\{X_t\}_{t=0}^{\infty}$ and $\{X'_t\}_{t=0}^{\infty}$ are started at time 0 with $X_0 \preceq X'_0$, then we have for each $t \geq 0$ that X_t is dominated (in the usual sense of stochastic ordering) by X'_t. This fails for most alternative Markov chains including the one defined by (6.1), and therefore monotone CFTP cannot be based on such alternative Markov chains.

(*ii*) If we update the location v in the Gibbs sampler-based CFTP algorithm, then the new value $\mathcal{X}_{t+1}(v)$ depends only on the random variable U_t and on $\mathcal{X}_t(w)$ for nearest neighbours w of v. For alternative CFTP algorithms based on Metropolis–Hastings chains, this new value may in addition depend on $\mathcal{X}_t(v)$. This may increase the amount of computation needed in each update.

(*iii*) The CFTP algorithm converges when $\mathrm{Card}[\mathcal{X}_t] = 1$, and this requires that the set of locations $v \in V$ for which $\mathcal{X}_t(v)$ has not yet coalesced (i.e. consists of more than one element $s \in S$) to become extinct. In the Gibbs sampler-based CFTP algorithm, the speed of convergence is facilitated by the fact that if we update a location v all of whose nearest neighbours have already

coalesced, then v is bound to coalesce as well. For the reason indicated in (ii), this need not be the case for Metropolis–Hastings-based CFTP algorithms.

7. Convergence rates

In practice, the CFTP algorithm is useless unless the time taken for convergence is sufficiently small. An obviously necessary (but by no means sufficient) condition for CFTP to converge rapidly is that the individual Gibbs sampler mixes rapidly (i.e. comes close to the equilibrium distribution relatively fast). One situation where one can expect relatively rapid convergence is when the interactions of the Markov random field are weak, i.e. if for each $v \in V$ the conditional distribution of $X(v)$ does not depend too strongly on $X(V \setminus \{v\})$. Here we shall give a more precise meaning to this concept of weak dependence, and show how it can be exploited to implement a fast CFTP algorithm, using a so-called multigamma coupler; the multigamma coupler was first discussed by Murdoch and Green [17].

To begin with, we need yet another piece of notation. For two measures (not necessarily probability measures) P and Q on S, we write $P \sqsupseteq Q$ if $P(s) \geq Q(s)$ for every $s \in S$.

Assume that X is an S^V-valued Markov random field whose distribution π has the property that for some $\gamma > 0$ and every $v \in V$, there exists a subprobability measure Q_v on S of total mass γ, such that

$$\pi(X(v) \in \cdot \,|\, X(V \setminus \{v\}) = \xi) \quad \sqsupseteq \quad Q_v$$

for every $\xi \in S^{V \setminus \{v\}}$ that has positive π-probability. (A typical situation where this holds is when (3.8) is in force, in which case we can take $\gamma = \delta$ and for each v let Q_v put all of its mass on the element s.) It is then possible to define a ϕ-function for the Gibbs sampler for X, with the property that

$$[\phi(\xi, u, v)](v) \quad = \quad [\phi(\xi', u, v)](v) \tag{7.1}$$

for any $v \in V$ and any $\xi, \xi' \in S^V$ as long as $u \leq \gamma$. In other words, if the uniform random variable $U \in [0, 1]$ used for updating v turns out to be less than γ, then the new value of $X(v)$ does not depend on the previous value of $X(V \setminus \{v\})$. We call a CFTP implementation whose ϕ-function satisfies (7.1) a multigamma coupler with parameter γ. It is possible to find multigamma couplers for all the examples in Section 2 except for the random q-colourings. For instance, it is easy to see that the hard-core model admits a multigamma coupler with $\gamma = \frac{1}{a+1}$, and that the Widom–Rowlinson model admits a multigamma coupler with $\gamma = \frac{1}{1+qa}$.

Let $d = d(G)$ denote the maximum degree in G, i.e. d is the maximum number of edges incident to the same vertex. Also let $N = N(G)$ be the number of vertices in G. We recall from the proof of Theorem 3.1 the definition of the

random variable $T_* = \inf\{t : \text{Card}[\Psi^0_{-t}(S^V, (\boldsymbol{U}, \boldsymbol{W}))] = 1\}$. A sensible quantity for describing the speed of convergence of CFTP algorithms is the expectation $E[T_*]$. The following theorem tells us that if d and $\gamma > \frac{d}{d+1}$ are fixed, then $E[T_*]$ grows no faster than $O(N \log N)$ in the size N of G. This makes it practically feasible to simulate Markov random fields living on fairly large graphs.

Theorem 7.1. *Let X be a Markov random field on a graph G with maximum degree d and size N, and suppose that X admits a multigamma coupler with parameter $\gamma > \frac{d}{d+1}$. Then such a CFTP implementation has*

$$E[T_*] \;\leq\; \frac{N(1 + \ln N)}{\gamma - (1-\gamma)d}. \tag{7.2}$$

Proof. Define

$$T^* \;=\; \inf\{t : \text{Card}[\Psi^t_0(S^V, (\boldsymbol{U}, \boldsymbol{W}))] = 1\}.$$

It is easy to see that T^* and T_* have the same distribution, so it suffices to show that (7.2) holds with T_* replaced by the conceptually slightly simpler quantity T^*. We thus run the chain $\{\mathcal{X}_t\}_{t=0}^{\infty}$ from time 0 onwards, starting with $\mathcal{X}_0 = S^V$. Define the auxiliary random process $\{Y_t\}_{t=0}^{\infty}$ by letting Y_t be the number of vertices $v \in V$ for which $\mathcal{X}_t(v)$ consists of more than one element $s \in S$. Also define a third process $\{Z_t\}_{t=0}^{\infty}$ by letting

$$Z_t \;=\; t + \frac{N \sum_{j=1}^{Y_t} \frac{1}{j}}{\gamma - (1-\gamma)d}$$

for $t \leq T^*$ (note that $Y_{T^*} = 0$ so that $Z_{T^*} = T^*$) and letting $Z_t = T^*$ for $t > T^*$. The main part of the proof consists of showing that $\{Z_t\}_{t=0}^{\infty}$ is a supermartingale with respect to the filtration $\mathcal{F}_t = \sigma(\mathcal{X}_0, \ldots, \mathcal{X}_t)$ (note that T^* is a stopping time for this filtration). Once this is done, we get

$$\begin{aligned} E[T^*] \;&=\; E[Z_{T^*}] \leq Z_0 \\ &=\; \frac{N \sum_{j=1}^{N} \frac{1}{j}}{\gamma - (1-\gamma)d} \leq \frac{N(1 + \log N)}{\gamma - (1-\gamma)d} \end{aligned}$$

where the first inequality is an application of Doob's Optional-Stopping Theorem (see e.g. Williams [22]).

It remains to prove the supermartingale property of $\{Z_t\}_{t=0}^{\infty}$, i.e. that

$$E[Z_{t+1} - Z_t \,|\, \mathcal{F}_t] \leq 0. \tag{7.3}$$

Suppose that $Y_t = n$. For $n = 0$, we have $T^* \leq t$, so that $Z_{t+1} = Z_t = T^*$, and (7.3) becomes trivial. Hence, we can assume that $n \in \{1, \ldots, N\}$. Obviously, the

increments $(Y_{t+1} - Y_t)$ of the $\{Y_t\}_{t=0}^{\infty}$ process can only take the values -1, 0 and 1, so the definition of Z_t gives

$$E[Z_{t+1} - Z_t \,|\, \mathcal{F}_t,\, Y_t = n] = 1 + \tfrac{N}{\gamma-(1-\gamma)d}\left(\tfrac{P[Y_{t+1}=n+1\,|\,\mathcal{F}_t]}{n+1} - \tfrac{P[Y_{t+1}=n-1\,|\,\mathcal{F}_t]}{n}\right). \quad (7.4)$$

We go on to estimate the probabilities in (7.4). In order for $Y_{t+1} = n+1$ to happen, we need to choose a vertex with "coalescence" to update, and "uncoalesce" it. For this, it is necessary (i) that $\mathcal{X}_t(w)$ consists of more than one $s \in S$ for some nearest neighbour w of the chosen location V_t (this has probability at most $\frac{dn}{N}$), and (ii) that $U_t \geq \gamma$ (this has probability $1-\gamma$). By independence of V_t and U_t, we thus have

$$P[Y_{t+1} = n+1 \,|\, \mathcal{F}_t] \;\leq\; \frac{dn(1-\gamma)}{N}. \quad (7.5)$$

On the other hand, for $Y_{t+1} = n-1$ to happen, it is sufficient that (i) the location V_t chosen to update does not have coalescence (this has probability $\frac{n}{N}$) and (ii) $U_t \leq \gamma$ (this has probability γ). We get

$$P[Y_{t+1} = n-1 \,|\, \mathcal{F}_t] \;\geq \tfrac{n\gamma}{N}. \quad (7.6)$$

By inserting (7.5) and (7.6) in (7.4), we get

$$\begin{aligned} E[Z_{t+1} - Z_t \,|\, \mathcal{F}_t] &\leq 1 + \frac{N}{\gamma-(1-\gamma)d}\left(\frac{dn(1-\gamma)}{(n+1)N} - \frac{n\gamma}{nN}\right) \\ &\leq 1 + \frac{N}{\gamma-(1-\gamma)d}\left(\frac{d(1-\gamma)}{N} - \frac{\gamma}{N}\right) \\ &= 0. \end{aligned}$$

Hence, (7.3) holds for any $n \in \{0, \ldots, N\}$, so the proof is complete. ∎

8. Simulation experiments

In this section, we complement the theoretical results in the previous section with some computer experiments. We have studied the performance of the algorithm both on the Widom–Rowlinson lattice gas model, for which Theorem 7.1 is applicable, and on random q-colourings, for which the theorem does not apply.

As our graph G we have used square portions of $\mathbb{Z}^2$, with free boundary. Most of our interest is concentrated on the running time of the algorithm, which we measure in number of iterations needed to reach coalescence. When we study running time, the algorithm is run forward using the naive coupling to the future (as in Section 5). This is justified by the observation made in the proof of Theorem 7.1, that T_* and T^* are both governed by the same probability distribution. Our estimates of the running time are based on 20 replicates in the case of the random q-colourings and on 50 replicates in the case of the Widom–Rowlinson lattice gas model. All programming was done in the programming language *C*.

8.1. Widom–Rowlinson lattice gas model

One of the models for which Theorem 7.1 is applicable is the Widom–Rowlinson lattice gas model. We have studied the effect on the running time when varying the value of the activity parameter, a, keeping constant the number of different particles, q, and the size of the system. This we have done for systems of size 50×50 and for two values of q, namely 2 and 3. The results for the case $q = 3$ is shown in Figure 2, while Figure 3 contains the results for the case $q = 2$.

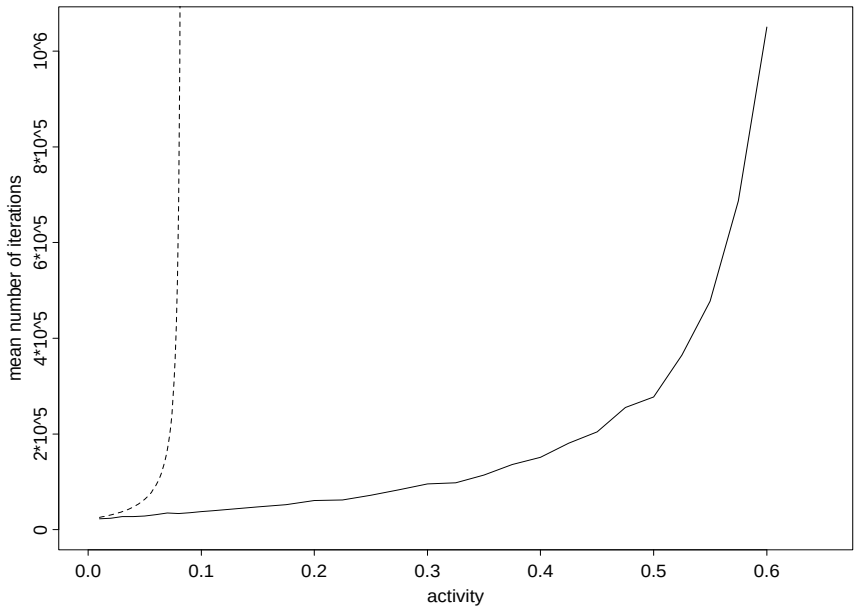

Figure 2: Mean number of iterations versus activity for $q = 3$ on a 50×50 system, drawn in a solid line. The upper bound of Theorem 7.1 is drawn with a dashed line.

In the graphs we have added a curve representing the upper bound for the expected value of the number of iterations to coalescence presented in Theorem 7.1. As can be seen from the pictures, this is not a very sharp upper bound. It is also the case that this upper bound gives information only when $\gamma = \frac{1}{1+qa}$ is greater than $\frac{d}{d+1}$, where d is is the maximum degree in G. We see that it is possible to perform simulations in reasonable time even for γ quite a bit less than $\frac{d}{d+1}$, that is to say for a significantly greater than $\frac{1}{dq}$. The value of d is of course 4 in our case, so that Theorem 7.1 guarantees rapid convergence for $a < \frac{1}{8}$ when $q = 2$ and for $a < \frac{1}{12}$ when $q = 3$. That running times increase drastically as a increases is

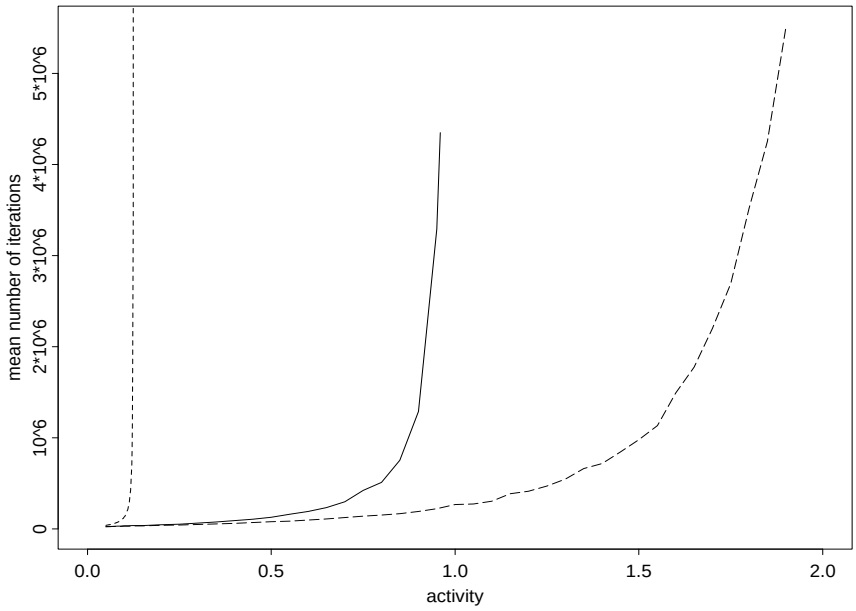

Figure 3: Mean number of iterations versus activity for $q = 2$ on a 50×50 system using the multigamma coupler (solid line), and using the monotone algorithm (long-dashed line). The upper bound of Theorem 7.1 is drawn with a short-dashed line.

an inevitable consequence of the phase transition phenomenon exhibited by the Widom–Rowlinson model (see e.g. [14]). We see in Figure 4 that, despite the symmetry of the model with respect to the two particle types, one of the types dominates the picture. This is a clear indication that a is in or near the phase transition regime of the parameter space.

Let us now introduce a ϕ-function which implements a multigamma coupler for this model. In the following *Neighbours* will be a function that, given the current configuration ξ and the chosen vertex v, returns the number of different values from $1, 2, ..., q$ that can be found among the neighbours of v. $Exist(\xi, v, j)$ will be a Boolean function that returns TRUE if at least one of v's neighbours in the current configuration ξ has the value j, while the rest then has value 0. It

Figure 4: The Widom–Rowlinson lattice gas model on a 50×50 lattice with $q = 2$ and $a = 1.9$. White balls correspond to particles of type 1, and black balls to particles of type 2.

returns FALSE otherwise.

$$[\phi(\xi,u,v)](v) = \begin{cases} 0 & \text{if } [u < \frac{1}{1+qa}] \quad \text{OR} \\ & \quad [\mathit{Neighbours}(\xi,v) \geq 2] \quad \text{OR} \\ & \quad [(\mathit{Neighbours}(\xi,v) = 1) \text{ AND } (u < \frac{1}{1+a})] \\ j & \text{if } [(\mathit{Neighbours}(\xi,v) = 1) \text{ AND } \mathit{Exist}(\xi,v,j) \text{ AND} \\ & \quad (u \geq \frac{1}{1+a})] \quad \text{OR} \\ & \quad [(\mathit{Neighbours}(\xi,v) = 0) \text{ AND } (u \in [\frac{1+(j-1)a}{1+qa}, \frac{1+ja}{1+qa}))] \\ & \qquad\qquad\qquad\qquad\qquad\qquad j = 1,2,...,q \end{cases} \tag{8.1}$$

It is easy to check that (3.2) is satisfied for this choice of ϕ. Furthermore, since $[\phi(\xi,u,v)](v) = 0$ whenever $u < \frac{1}{1+qa}$, this defines a multigamma coupler with $\gamma = \frac{1}{1+qa}$. The CFTP implementation, using (3.4), is straightforward but somewhat tedious; we omit the details.

The Widom–Rowlinson lattice gas model with $q = 2$ can be viewed as monotone if we equip the state space with the ordering: $1 < 0 < 2$. For this case we have compared the performance of the multigamma coupler to the performance of the monotone algorithm. As could be expected, and can be seen in Figure 3,

making use of the monotonicity property pays off in terms of lower running times, especially for large a.

We remark that a similar comparison of monotone and multigamma couplers would be less interesting for the Ising model, because in that case it turns out in that monotone ϕ-function in (3.3) actually is a multigmma coupler with optimal γ.

8.2. Random q-colourings

In the case of random q-colourings, it is easy to see that no multigamma coupler can be found. This means that we can not rely on Theorem 7.1 to give us an upper bound for the expected time needed for coalescence. We have nevertheless been able to make simulations for various values of q, and various sizes of the square graph, results of which can be seen in Figure 5. Intuitively, the value

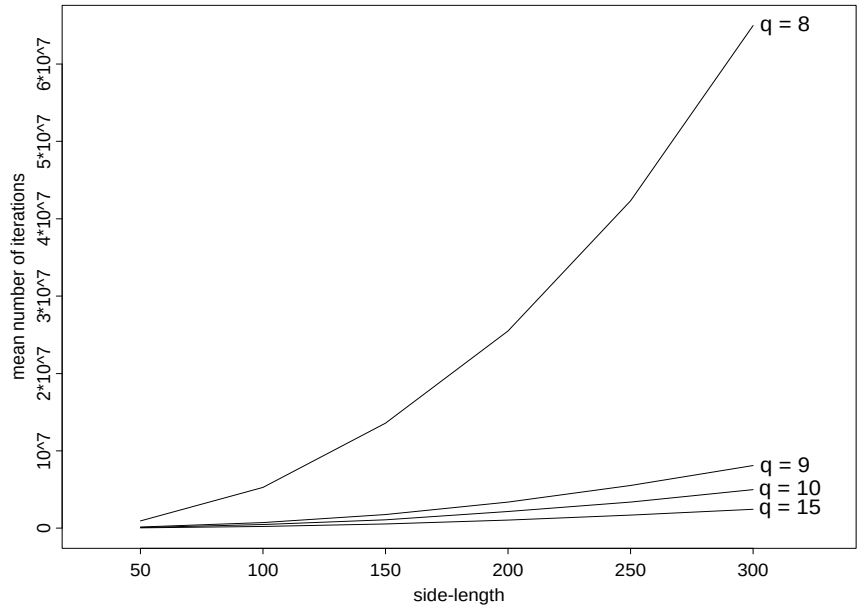

Figure 5: Mean number of iterations versus size for different values of q.

of the random q-colouring at a vertex v given everything else depends less on its neighbours the larger q is, and therefore it should be easier to get rapid CFTP convergence the larger q is. Our experience supports this intuition and is illustrated in Figure 5.

Jerrum [10] describes an algorithm for estimating the number of q-colourings of a low degree graph using the Gibbs sampler. He finds an upper bound of the order $N \log N$ (where N as usual is the number of vertices) for the time for the chain to come close to equilibrium provided that q is at least $2d + 1$, with d as above. In our case this corresponds to $q \geq 9$. Rapid convergence to equilibrium of the Gibbs sampler is, however, not sufficient to guarantee fast CFTP convergence. Nevertheless, as we can see in Figure 5, our CFTP algorithm exhibits a marked growth in running time between the cases $q = 9$ and $q = 8$. We were not able to get results for the case $q = 7$ in reasonable time.

The ϕ-function used in our CFTP implementations is defined as follows. First note that the conditional distribution in the right hand side of (3.2) is simply uniform over all colours that are permissible (i.e. not attained by any neighbour). As before w_t will be the vertex chosen to be updated at time t. The meaning of u_t will however be slightly different. It could be taken as a permutation of the colours $1, 2, ..., q$ chosen uniformly at random. This still fits into the framework of Section 3, because the unit interval can be partitioned into $q!$ pieces of equal length, each piece representing one of the possible permutations. The new value at location w_t is the first value from the vector u_t that is permissible. Since in our case w_t has at most 4 neighbours, at most 5 proposal values will be needed. This means that we can simplify somewhat by not letting u_t be a permutation of all colours but rather 5 draws without replacement of colours from $1, 2, ..., q$.

The CFTP implementation using (3.4) of this ϕ-function is, in principle, straightforward. However, a simple-minded implementation may lead to a computational complexity which grows rapidly with q, so we shall describe our implementation in some detail. The 5 draws in u_t are examined in order. The first thing to do is to see if any of the neighbours of w_t have the examined value in their respective lists of possible values. If that is not the case the examined value is added to w_t's list of values and the update is complete. This is so because for any $\eta \in \mathcal{X}_t$ the single Gibbs sampler described above would have accepted the examined value. Here $\{\mathcal{X}_t\}$ is the $\mathcal{S}^V$-valued Markov chain, defined in Section 3, that for all vertices has a list of the, at present, possible values at the respective locations. If on the other hand the examined value was found somewhere among the possible values of the neighbours of w_t, we look to see if it is the only value for one or more neighbours. Being an only value implies that all $\eta \in \mathcal{X}_t$ contain this value at the same neighbour, or neighbours, of w_t. This in turn means that a single Gibbs sampler acting on any of $\eta \in \mathcal{X}_t$ would have rejected the value. The CFTP algorithm will also reject the value and will then go on to examine the next value in u_t. If the examined value is found among the neighbouring values of w_t, but not as an only value, the value is added to w_t's list of new values. We now have to decide whether or not to look at more values from u_t. To see that the decision should not always be yes, let us look at an example. Suppose that the neighbourhood of w_t in a $q = 9$ system looks as in Figure 6 and that $u_t = (5, 4, 8, 3, 1)$. We see that we can not have any neighbourhood of

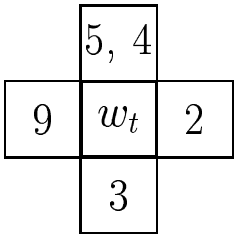

Figure 6: Example of a neighbourhood of the chosen vertex w_t at time t in a 9-colouring system.

w_t containing both the value 5 and the value 4. This means that we do not have to look any further into u_t after having decided to keep the values 5 and 4. Had we now followed the lazy approach of not considering whether it is necessary or not to look at new values, the new list of values at w_t would have consisted of 5, 4 and 8, and we would have made it unnecessary hard for us to finally coalesce. One way to make this decision is to examine if any permutation of any subset of the values already kept occur at *different* neighbour lists of w_t. If so, we have to go on to examine the next value of u_t, otherwise the update is complete.

9. Concluding remarks

As we already said in the introduction, a serious problem with most MCMC algorithms is the lack of useful bounds for the rate of convergence to equilibrium. CFTP provides one approach to solving this problem. We think that there is a fair chance that CFTP methods will dominate the entire MCMC field within a few years from now. However, for this to happen the domain of applicability of CFTP must be greatly expanded. The present paper is an attempt to make a few small steps in this direction.

Rapid convergence of the algorithm is obviously of crucial importance. Theorem 7.1 provides a sufficient condition for this to be feasible. Section 8 shows that this condition is far from necessary. This is not surprising, since a very general result seldom is sharp in special cases. We nevertheless feel that Theorem 7.1 may be quite useful, because it indicates what kind of properties one can look for in designing efficient CFTP algorithms. We also think it may be a worthwhile project to try to improve Theorem 7.1 or to look for other general criteria in the same spirit.

Acknowledgement. The idea behind Theorem 7.1 grew out of a discussion with Robin Pemantle.

References

[1] Fill, J.A. (1997) An interruptible algorithm for perfect sampling via Markov chains, *Ann. Appl. Probab.*, to appear.

[2] Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Machine Intelligence* **6**, 721–741.

[3] Georgii, H-O. (1988) *Gibbs Measures and Phase Transitions*, de Gruyter, New York.

[4] Georgii, H-O. and Häggström, O. (1996) Phase transition in continuum Potts models, *Commun. Math. Phys.* **181**, 507–528.

[5] Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. (1996) *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London.

[6] Häggström, O. (1996) Random-cluster representations in the study of phase transitions, preprint.

[7] Häggström, O., van Lieshout, M.N.M. and Møller, J. (1996) Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes, preprint.

[8] Häggström, O. and Nelander, K. (1997) Exact sampling from anti-monotone systems, preprint.

[9] Hjort, N.L. and Omre, H. (1994) Topics in spatial statistics (with discussion), *Scand. J. Statist.* **21**, 289–357.

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

[11] Kelly, F.P. (1985) Stochastic models of computer communication systems, *J. Roy. Statist. Soc. Ser. B* **47**, 379–395.

[12] Kendall, W. (1996) Perfect simulation for the area-interaction point process, *Probability Perspective*, (L. Accardi and C. Heyde, eds), Springer, to appear.

[13] Kindermann, R. and Snell J.L. (1980) *Markov Random Fields and their Applications*, American Mathematical Society, Providence, Rhode Island.

[14] Lebowitz, J.L. and Gallavotti, G. (1971) Phase transitions in binary lattice gases, *J. Math. Phys.* **12**, 1129–1133.

[15] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953) Equations of state calculations by fast computing machine, *J. Chem. Phys.* **21**, 1087–1091.

[16] Møller, J. (1997) A note on perfect simulation of conditionally specified models, preprint.

[17] Murdoch, D.J. and Green, P.J. (1997) Exact sampling from a continuous state space, preprint.

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

[19] Salas, J. and Sokal, A.D. (1997) Absence of phase transition for antiferromagnetic Potts models via the Dobrushin uniqueness theorem, *J. Statist. Phys.* **86**, 551–579.

[20] Tierney, L. (1994) Markov chains for exploring posterior distributions (with discussion), *Ann. Statist.* **22**, 1701–1762.

[21] Widom, B. and Rowlinson, J.S. (1970) New model for the study of liquid-vapor phase transition, *J. Chem. Phys.* **52**, 1670–1684.

[22] Williams, D. (1991) *Probability with Martingales*, Cambridge University Press.

[23] Winkler, G. (1995) *Image Analysis, Random Fields and Dynamic Monte Carlo Methods. A Mathematical Introduction*, Springer, New York.