Extensions of Fill's algorithm for perfect simulation

Jesper Møller* and Katja Schladitz
MaPhySto *Aalborg University*

May 14, 1998

Abstract

Fill's algorithm for perfect simulation for attractive finite state space models, unbiased for user impatience, is presented in terms of stochastic recursive sequences and extended in two ways. Repulsive discrete Markov random fields with two coding sets like the auto-Poisson distribution on a lattice with 4-neighbourhood can be treated as monotone systems if a particular partial ordering and quasi-maximal and quasi-minimal states are used. Fill's algorithm then applies directly. Combining Fill's rejection sampling with sandwiching leads to a version of the algorithm, which works for general discrete conditionally specified repulsive models. Extensions to other types of models are briefly discussed.

Keywords: MARKOV CHAIN MONTE CARLO, REPULSIVE MARKOV RANDOM FIELDS, EXACT SIMULATION, CONDITIONALLY SPECIFIED MODEL, GIBBS SAMPLER, STOCHASTIC RECURSIVE SEQUENCES, AUTO-POISSON MODEL, HARD-CORE MODEL, STRAUSS PROCESS

1 Introduction

Propp and Wilson's (1996) coupling from the past (CFTP) algorithm for perfect simulation from the stationary distribution of a Markov chain has been

*MaPhySto – Centre for Mathematical Physics and Stochastics, funded by a grant from The Danish National Research Foundation. *Address for correspondence:* Department of Mathematics, Aalborg University, Fredrik Bajers Vej 7E, 9220 Aalborg, Denmark, Email:jm@math.auc.dk

extended in several ways. Propp and Wilson assumed the state space to be finite, partially ordered so that the sampler is monotone, and to have unique maximal and minimal elements. In Häggström and Nelander (1997a,b), Häggström *et al.* (1996), Kendall (1996), Kendall and Møller (1998), Møller (1997) and Murdoch and Green (1997) CFTP algorithms are introduced allowing e.g. for uncountable state spaces (point processes) and repulsive (or anti-monotone) systems; see also Corcoran and Tweedie (1998a,b), Foss and Tweedie (1997) and Foss *et al.* (1997).

An impatient user who stops long runs of the algorithm before termination can cause biased output of the CFTP algorithm, see Fill (1997) and Thönnes (1997) for details. Fill (1997) introduced an alternative perfect simulation algorithm, based on rejection sampling and unbiased for user impatience. This algorithm can briefly be described as follows. A Markov chain is started in the minimal state and run for N transitions. Its output is accepted as a sample from the stationary distribution if and only if a second chain, coupled to the time reversed trajectory of the first one and started in the maximal state, arrives at the minimal state after N transitions (backwards in time). Else the procedure is repeated with doubled number of transitions.

Note that Fill assumes that the state space is finite, that there are unique maximal and minimal elements, and that the transition matrix of the time reversed Markov chain is stochastically monotone. So far Thönnes' (1997) simulation algorithm for some attractive point process models is the only extension of Fill's algorithm.

Møller (1997) showed how to adapt CFTP for perfect simulation of conditionally specified models. In this paper we examine the possibilities to use Fill's rejection sampling in a similar way.

In Section 2 we describe Fill's algorithm in the setting of stochastic recursive sequences (see Borovkov and Foss (1992) and Kifer (1986)), which is easier to handle than Fill's original setting. In Section 3 we show how the algorithm can be used for discrete Markov random fields with exactly two coding sets (or defined on a bipartite graph), a repulsive distribution and no maximal element. A combination of the rejection sampling algorithm with sandwiching is introduced in Section 4 to deal with repulsive models where we do not have exactly two coding sets. Some experimental results comparing CFTP and rejection sampling are presented and discussed in Section 5. Finally, Section 6 contains a discussion on extensions to non-discrete models.

2 Perfect simulation by rejection sampling

This section provides essential background material. In particular, we restate Fill's algorithm in terms of stochastic recursive sequences (SRS), using that every time-homogeneous Markov chain can be represented that way, see Borovkov and Foss (1992) and Kifer (1986).

Fill (1997) proved the correctness of the algorithm in a slightly more general setting. For easy reference later on we include a proof of correctness in our setup. The use of stochastic recursive sequences gives more insight in how the algorithm operates. Moreover, considering the proof will help us to extract the essential conditions under which the algorithm works.

Suppose we want to sample from a target distribution π with finite support Ω. We assume that Ω is equipped with some partial order $\leq$ so that Ω contains a unique minimal element $\hat{0}$ and a unique maximal element $\hat{1}$ with respect to this partial order.

As in Corcoran and Tweedie (1998b), Foss and Tweedie (1997), Foss *et al.* (1997), Häggström and Nelander (1997a,b), Häggström *et al.* (1996), Kendall (1996), Kendall and Møller (1998), Møller (1997) and Murdoch and Green (1997) assume that we can construct a time-homogeneous and ergodic Markov chain $X_0, X_1, \ldots$ with stationary distribution π by

$$X_t = \phi(R_t, X_{t-1}), \tag{1}$$

where ϕ is a deterministic function and the R_t are i.i.d. random vectors. In practise the construction (1) is often obvious and given by the way one writes the computer program for simulating π as the limit of the chain.

Let P denote the transition matrix of X. The ergodic time-homogeneous Markov chain $\widetilde{X}$, defined through its transition matrix $\widetilde{P}$ by

$$\widetilde{P}(x,y) = \frac{\pi(y)}{\pi(x)} P(y,x) \tag{2}$$

is the *time reversal* of X. We let P^N and $\widetilde{P}^N$ denote the corresponding N-step transition probabilities. The time reversal is again an SRS and thus there is an updating rule similar to (1) for $\widetilde{X}$, too,

$$\widetilde{X}_t = \widetilde{\phi}(R'_t, \widetilde{X}_{t-1}), \tag{3}$$

where the R'_t are i.i.d. on some measurable space $\mathcal{R}$ and $\widetilde{\phi}$ is a deterministic function. We assume $\widetilde{\phi}$ to be monotone in the second component:

$$\widetilde{\phi}(r,x) \leq \widetilde{\phi}(r,y) \quad \text{for all} \quad r,x,y \quad \text{with} \quad x \leq y. \tag{4}$$

Remark 1. Note that $\phi = \widetilde{\phi}$ if X is time reversible. A typical non-reversible updating scheme is cyclic Gibbs sampling as used later on in this paper. In this case $\widetilde{\phi}$ is obtained from ϕ by just reversing the order of the updating cycle. Moreover, ergodicity of the Gibbs sampler is equivalent to irreducibility, see Roberts and Smith (1994). This still holds if the support Ω is not finite anymore but countable.

Finally, for $x, x' \in \Omega$ with $P(\widetilde{\phi}(R'_t, x) = x') > 0$, let a distribution $Q(x, x')$ on $\mathcal{R}$ be defined by

$$Q(x, x')(A) := P(R'_t \in A \,|\, \widetilde{\phi}(R'_t, x) = x'), \tag{5}$$

for events A. Roughly speaking, $Q(x, x')$ describes the conditional distribution of any of the i.i.d. R'_t given that it caused the transition from x to x' in the time reversed Markov chain $\widetilde{X}$.

Now Fill's algorithm for perfect simulation works as follows:

Algorithm 1 ***(Fill)***.

0. Choose an initial natural number of transitions $N > 0$.
1. Generate i.i.d. vectors $R_1, \ldots, R_N$.
2. Starting in $X_0 = \hat{0}$ and updating according to (1), create $X_1, \ldots, X_N$.
3. Given the output of the previous step, generate another stream of conditionally independent random vectors $\widetilde{R}_t$ drawn from the distributions $Q(X_{N+1-t}, X_{N-t}), t = 1, \ldots, N$.
4. Starting in $\widetilde{Y}_0 = \hat{1}$ and updating according to $\widetilde{Y}_t = \widetilde{\phi}(\widetilde{R}_t, \widetilde{Y}_{t-1})$, create $\widetilde{Y}_1, \ldots, \widetilde{Y}_N$.
5. If $\widetilde{Y}_N = \hat{0}$ then accept $Z = X_N$ as a sample from π. Else, double N and start over from step 1 (with random vectors in steps 1 and 3 which are independent of previously generated random vectors).

This is the special version of the algorithm mentioned in Fill (1997, Section 7.1). Denote by T the time to determination of the algorithm, that is the number of times steps 1-5 have to be performed till acceptance.

Proposition 1. *Algorithm 1 terminates with probability* 1, *i.e.* $T < \infty$ *almost surely, and the output* Z *follows the target distribution* π . *Further, Algorithm 1 is interruptible in the sense that the determination time* T *and the output* Z *are independent. Moreover, in step 5 of the algorithm the event of acceptance* $\{\widetilde{Y}_N = \hat{0}\}$ *occurs with probability* $\widetilde{P}^N(\hat{1}, \hat{0})/\pi(\hat{0})$.

PROOF: We verify below that

$$P(\widetilde{Y}_N = \hat{0}, X_N = z \,|\, X_0 = \hat{0}, \widetilde{Y}_0 = \hat{1}) = \frac{\pi(z)}{\pi(\hat{0})} \widetilde{P}^N(\hat{1}, \hat{0}), \tag{6}$$

which yields the acceptance probability

$$P(\widetilde{Y}_N = \hat{0} \,|\, X_0 = \hat{0}, \widetilde{Y}_0 = \hat{1}) = \frac{\widetilde{P}^N(\hat{1}, \hat{0})}{\pi(\hat{0})}. \tag{7}$$

By ergodicity, the right hand side in (7) converges to 1 as $N \to \infty$. Hence, $T < \infty$ almost surely. Moreover, by (6), Z and T are independent with distribution of the output $\mathcal{D}(Z) = \mathcal{D}(X_N \,|\, X_0 = \hat{0}, \widetilde{Y}_0 = \hat{1}, \widetilde{Y}_N = \hat{0}) = \pi$.

To show (6), let $x_N = y_N = \hat{0}, x_0 = z, y_0 = \hat{1}$. Write

$$Q_{(x,x')}(y, y') := P(\widetilde{\phi}(\widetilde{R}, y) = y'), \quad \text{where} \quad \widetilde{R} \sim Q(x, x').$$

Then we have by the definition (5) of $Q(x, x')$,

$$\widetilde{P}(x, x') Q_{(x,x')}(y, y') = P(\widetilde{\phi}(R'_t, x) = x', \widetilde{\phi}(R'_t, y) = y'). \tag{8}$$

The design of the algorithm, the definition of the time reversal (2), and (8) give

$$\begin{aligned}
&P(\widetilde{Y}_N = \hat{0}, X_N = z \,|\, X_0 = \hat{0}, \widetilde{Y}_0 = \hat{1}) \\
&= \sum_{x_1, \ldots, x_{N-1}} \Big[\prod_{t=1}^{N} P(x_t, x_{t-1}) \Big] \sum_{y_1, \ldots, y_{N-1}} \prod_{t=1}^{N} Q_{(x_{t-1}, x_t)}(y_{t-1}, y_t) \\
&= \frac{\pi(z)}{\pi(\hat{0})} \sum_{x_1, \ldots, x_{N-1}} \Big[\prod_{t=1}^{N} \widetilde{P}(x_{t-1}, x_t) \Big] \sum_{y_1, \ldots, y_{N-1}} \prod_{t=1}^{N} Q_{(x_{t-1}, x_t)}(y_{t-1}, y_t) \\
&= \frac{\pi(z)}{\pi(\hat{0})} \sum_{\substack{x_1, \ldots, x_{N-1} \\ y_1, \ldots, y_{N-1}}} \prod_{t=1}^{N} P(\widetilde{\phi}(R'_t, x_{t-1}) = x_t, \widetilde{\phi}(R'_t, y_{t-1}) = y_t).
\end{aligned} \tag{9}$$

By the monotonicity (4) of $\widetilde{\phi}$ the product in (9) is positive for paths with $x_t \le y_t, t = 1, \ldots, N-1$ only and thus

$$P(\widetilde{\phi}(R'_N, x_{N-1}) = \hat{0}, \widetilde{\phi}(R'_N, y_{N-1}) = \hat{0}) = P(\widetilde{\phi}(R'_N, y_{N-1}) = \hat{0})$$

does not depend on x_{N-1}. Summing in (9) first over x_{N-1} and then over $x_{N-2}, \ldots, x_1$ successively, finally over the y_t, we get the assertion (6).

Remark 2. To see the rejection sampling character of Algorithm 1 consider the probability of acceptance from (7). By (3) and the monotonicity of the updating rule $\widetilde{\phi}$ we have $\widetilde{P}^N(\hat{1},\hat{0}) \leq \widetilde{P}^N(z,\hat{0})$ for all z. Hence $\pi(\hat{0})/\widetilde{P}^N(\hat{1},\hat{0})$ is an upper bound of $\pi(z)/P^N(\hat{0},z) = \pi(\hat{0})/\widetilde{P}^N(z,\hat{0})$. So the rule of acceptance in step 5 of Algorithm 1 turns out to be ordinary rejection sampling with proposal distribution $P^N(\hat{0},\,.\,)$.

Remark 3. We follow Fill (1997) in doubling the number of transitions in step 5 of Algorithm 1. However, Proposition 1 holds as well if we use any unbounded non-decreasing sequence of values for N. Even bounded sequences with $P^N(\hat{0},\hat{1}) > 0$ for infinitely many Ns may be used. This observation remains true for Propositions 2 and 3 in Sections 3 and 4 concerning Algorithms 2 and 3. In fact, in Section 5 we present some empirical results, where N was incremented by 1 instead of doubled.

3 Repulsive models with two coding sets

In this section we extend Fill's algorithm to certain discrete models. Briefly, we consider a discrete vector X, which splits into two components X_A and X_B so that the coordinates of X_A (or X_B) are conditionally independent given X_B (or X_A). Further, the conditional distributions are repulsive in a sense defined below. Moreover, the state space of X is equipped with a partial order very similar to the one introduced in Häggström *et al.* (1996) but for another setting. In particular, there exist neither minimal nor maximal element with respect to the ordering; instead quasi-minimal and quasi-maximal elements are used.

More specifically, let $X = (X_i)_{i\in I}$ be a random vector indexed by a finite set I. We suppose that the distribution π of X is specified by the conditional distributions $\mathcal{D}(X_i|X_{-i})$, where $X_{-i} := (X_k)_{k\neq i}$ denotes all but the ith component of X. Let the support of X be a (not necessarily finite) set $\Omega \subseteq \mathbb{N}^I$ containing $\hat{0} := 0^I$. This is the minimal element with respect to the natural partial ordering on Ω induced by the usual ordering on $\mathbb{N}$. (Note however, that another partial ordering is used later on). We assume that $\pi(\hat{0}) > 0$ and

$$\mathcal{D}(X_A|X_B) = \prod_{a\in A} \mathcal{D}(X_a|X_B), \quad \mathcal{D}(X_B|X_A) = \prod_{b\in B} \mathcal{D}(X_b|X_A),$$

where I is the disjoint union of A and B. In the terminology of Besag (1974), A and B are coding sets if they are maximal in the sense that they can not be replaced by another subdivision of I. Typically, if X is a Markov random field with neighbourhoods $\mathcal{G}_i, i \in I$, then we are assuming that $\mathcal{G}_a \subseteq B$ and

$\mathcal{G}_b \subseteq A$ for any $a \in A$ and $b \in B$. One example is a square grid $I = (k,l)_{k,l=1}^n$ with 4-neighbourhood ($\mathcal{G}_{(k,l)} = \{(k-1,l),(k+1,l),(k,l-1),(k,l+1)\}$ with modifications at the edges) and $A = \{(k,l) : k+l \text{ even}\}$, $B = \{(k,l) : k+l \text{ odd}\}$. Another example is given by a graph with $n+1$ vertices and n edges joining one vertex with the n others. In the sequel we let $\mathcal{G}_i$ denote the smallest subset of $I \setminus \{i\}$ so that $\mathcal{D}(X_i|X_{-i}) = \mathcal{D}(X_i|X_{\mathcal{G}_i})$.

Furthermore, for $i \in I$ and $x \in \Omega$, let $F_i(\cdot\,|x_{-i})$ denote the distribution function of $\mathcal{D}(\cdot\,|X_{-i} = x_{-i})$. Let $\Omega_{-i} = \{x_{-i} : x \in \Omega\}$ be the restriction of Ω to all but the ith component. For the simulation we need the pseudo-inverses of the distribution functions F_i defined by $F_i^-(t|x_{-i}) = \min\{s : F_i(s|x_{-i}) \geq t\}$. We assume that X is *repulsive*. That is, for any $t \in [0,1]$ the function $F_i^-(t|\cdot)$ is decreasing w.r.t. the natural partial order on Ω_{-i} induced by the order on $\mathbb{N}$. For examples of repulsive models, see Section 5 and Møller (1997, Section 2.3).

Now, we generate a Markov chain with stationary distribution π by alternately sampling from $\mathcal{D}(X_A|X_B) = \prod_{a\in A}\mathcal{D}(X_a|X_{\mathcal{G}_a})$ and $\mathcal{D}(X_B|X_A) = \prod_{b\in B}\mathcal{D}(X_b|X_{\mathcal{G}_b})$ using the componentwise pseudo-inverses of the conditional distribution functions of the A-component given the B-component,

$$F_A^-(r|x_B) := (F_a^-(r_a|X_{\mathcal{G}_a}))_{a\in A} \quad \text{for} \quad r \in [0,1]^A$$

and vice versa

$$F_B^-(r|x_A) := (F_b^-(r_b|X_{\mathcal{G}_b}))_{b\in B} \quad \text{for} \quad r \in [0,1]^B.$$

That X is repulsive now means that $F_A^-(r|x_B)$ is decreasing in x_B w.r.t. the partial order on Ω_B induced by the order on $\mathbb{N}$ and $F_B^-(r|x_A)$ is decreasing in x_A w.r.t. the partial order on Ω_A. If the B-component is large then the A-component is small and vice versa.

Presented as a stochastic recursive sequence, the two-component Gibbs sampler is given by

Gibbs sampler 1.

1. $X_{A,t} = F_A^-(R_{A,t}|X_{B,t-1})$,
2. $X_{B,t} = F_B^-(R_{B,t}|X_{A,t})$,

where $R_t \sim \text{Uniform}([0,1]^I)$, $t = 1,2,\ldots$ are mutually independent. The updating rule for the time reversed chain is given by

Gibbs sampler 2.

1. $\widetilde{X}_{B,t} = F_B^-(R'_{B,t}|\widetilde{X}_{A,t-1})$,

2. $\widetilde{X}_{A,t} = F_A^-(R'_{A,t}|\widetilde{X}_{B,t})$,

with $R'_t \sim \text{Uniform}([0,1]^I)$, $t = 1, 2, \ldots$ mutually independent. We assume that these systematic two-component Gibbs samplers are irreducible (and hence ergodic).

In order to use the special version of Fill's algorithm described in Section 2 above we need monotonicity of the "reversed" Gibbs sampler 2. To this end we define a new partial order $\preceq$ on Ω by

$$(x_A, x_B) \preceq (y_A, y_B) \iff x_A \geq y_A \quad \text{and} \quad x_B \leq y_B. \tag{10}$$

Remark 4. If Ω is symmetric around some $k^I \in \mathbb{N}^I$, this partial order is equivalent to $((2k - x)_A, x_B) \leq ((2k - y)_A, y_B)$. This was used in Häggström and Nelander (1997a) to transform anti-monotone systems into monotone ones, for instance the Ising anti-ferromagnet into the Ising ferromagnet.

It is easy to see that both Gibbs samplers preserve the new partial order (10). But besides repulsion we also allow discrete distributions, where we do not have a maximal element anymore. Consequently, in the new partial order we have neither maximal nor minimal element. To overcome this problem observe that X_t depends on X_{t-1} through $X_{B,t-1}$ only and $\widetilde{X}_t$ on $\widetilde{X}_{t-1}$ through $\widetilde{X}_{A,t-1}$ only. Hence any state $(\cdot, 0_B)$ is a *quasi-minimal* one in the sense that

$$\text{if} \quad X_{B,0} = 0_B \quad \text{then} \quad X_t \preceq Y_t \quad \text{for all} \quad t \geq 1,$$

where Y is started in an arbitrary state, updated according to Gibbs Sampler 1, and coupled to X by using the same stream of random vectors $(R_t)_{t=1}^N$. Similarly, any state $(0_A, \cdot)$ is *quasi-maximal*:

$$\text{if} \quad \widetilde{Y}_{A,0} = 0_A \quad \text{then} \quad \widetilde{X}_t \preceq \widetilde{Y}_t \quad \text{for all} \quad t \geq 1.$$

Thus the algorithm gets the following form:

Algorithm 2.

0. Choose an initial natural number of transitions $N > 0$.

1. Generate independent random vectors $R_1, \ldots, R_N$ uniformly distributed on $[0,1]^I$.

2. Starting in $X_{B,0} = 0_B$ and updating using the forwards Gibbs sampler 1, create $X_1, \ldots, X_N$.

3. Generate random vectors $\widetilde{R}_{B,t}$ from

$$\text{Uniform}\,(F_B((X_{B,N-t})- \,|X_{A,N+1-t}), F_B(X_{B,N-t}|X_{A,N+1-t})),$$

for $t = 1, \ldots, N$, and $\widetilde{R}_{A,t}$ from

$$\text{Uniform}\,(F_A((X_{A,N-t})- \,|X_{B,N-t}), F_A(X_{A,N-t}|X_{B,N-t})),$$

for $t = 1, \ldots, N-1$. Here $f(x-) := \lim_{h \downarrow 0} f(x-h)$ denotes the left sided limit.

4. Starting in $\widetilde{Y}_{A,0} = 0_A$ and updating according to the backwards Gibbs sampler 2 create $\widetilde{Y}_1, \ldots, \widetilde{Y}_{N-1}$ using $(\widetilde{R}_t)_{t=1}^{N-1}$.
Let $\widetilde{Y}_{B,N} = F_B^-(\widetilde{R}_{B,N}|\widetilde{Y}_{A,N-1})$.

5. If $\widetilde{Y}_{B,N} = 0_B$ then accept $Z = X_N$ as a sample from π. Else, double N and start over from step 1.

The outline of this algorithm is similar to the one for simulation of attractive area interaction and related point processes by Thönnes (1997). The main difference is the use of the random number streams for both the Gibbs sampler and the coupling.

This version of the algorithm differs from Algorithm 1 in the last transitions. Consequently, we get another acceptance probability. The notation used for expressing the acceptance probability in the following proposition is intuitively clear, nevertheless it is explained in the proof below.

Proposition 2. *Algorithm 2 shares the same properties as stated for Algorithm 1 in Proposition 1, except that the probability of acceptance in step 5 now is* $\widetilde{P}^N(0_A, 0_B)/\pi(0_B)$.

PROOF: The proof follows the same outline as the proof of Proposition 1, only the last transitions in both the forwards and the backwards runs require some alterations. Therefore we only point out these differences. By the definitions of the Gibbs samplers 1 and 2 we get $P(x,y) = \pi(y_A|x_B)\pi(y_B|y_A)$ and

$$\widetilde{P}(y,x) = \pi(x_B|y_A)\pi(x_A|x_B) = \widetilde{P}(y_A, x)$$

depends on y through y_A only. Similarly, $\widetilde{P}(y, x_B) = \widetilde{P}(y_A, x_B) = \pi(x_B|y_A)$ is defined properly and clearly

$$\widetilde{P}^N(0_A, 0_B) = \sum_{x,y} \widetilde{P}(0_A, x)\widetilde{P}^{N-2}(x,y)\widetilde{P}(y, 0_B).$$

The conditional transition matrices $Q_{(x,x')}(y,y')$ decompose into A- and B-components, too. The last transition in the (coupled) backwards chains can now be described using only the B-components. For the time reversion of the forwards chain use $\pi(x_A|y_B) = \frac{\pi(x_A)}{\pi(y_B)}\pi(y_B|x_A)$ instead of (2).

4 General repulsive systems

If the assumption of two coding sets is dropped the repulsive (or anti-monotone) case can in general not be reduced to the monotone case. For CFTP, Kendall (1996) and Møller (1997) used upper and lower "chains", which condition on the other's state instead of their own previous state when updating. The pair is still a Markov chain (although the components are not) and all feasible paths are sandwiched between the upper and the lower path of this couple. In this section we describe how combining Fill's rejection sampling with this idea of sandwiching yields an interruptable algorithm for discrete conditionally specified repulsive models.

We use the notation introduced in the first two paragraphs of the previous section. The target distribution π is the distribution of a discrete random vector $(X_i)_{i\in I}$ with support $\Omega \subseteq \mathbb{N}^I$ and $\pi(\hat{0}) > 0$. As before, let $\Omega_{-i} = \{y_{-i} : y \in \Omega\}$.

In Algorithm 3 below we use Gibbs sampling with cyclic updating. For the generation of the backwards chain consisting of two components (upper and lower), we need an updating rule for a pair of current states instead of a single one. For $i \in I$, write $(x_{<i}, x'_{>i}) := (x_1, \ldots, x_{i-1}, x'_{i+1}, \ldots, x'_n)$, where n is the cardinality of I. Now the cyclic updating rule can be described by a function $\phi : [0,1]^I \times \mathbb{N}^I \times \mathbb{N}^I \to \mathbb{N}^I \times \mathbb{N}^I$ defined by

$$\phi(r,x,y) = (x',y') \quad \text{with} \quad \begin{array}{l} x'_i = F_i^-(r_i|(y'_{<i}, y_{>i})) \\ y'_i = F_i^-(r_i|(x'_{<i}, x_{>i})) \end{array} \tag{11}$$

for $i \in I$. Gibbs sampling with cyclic updating is not time reversable, but the "reversed" updating is described by $\widetilde{\phi} : [0,1]^I \times \mathbb{N}^I \times \mathbb{N}^I \to \mathbb{N}^I \times \mathbb{N}^I$ given by

$$\widetilde{\phi}(r,x,y) = (x',y') \quad \text{with} \quad \begin{array}{l} x'_i = F_i^-(r_i|(y_{<i}, y'_{>i})) \\ y'_i = F_i^-(r_i|(x_{<i}, x'_{>i})) \end{array} \tag{12}$$

for $i \in I$.

Due to the conditioning on the neighbourhood in the other component of the chain, the functions ϕ and $\widetilde{\phi}$ can leave $\Omega\times\Omega$. This can happen for instance for the hard-core model considered in Section 5. To make the definition (11) valid we have to explain $F_j^-(r|y_{-j})$ for $y_{-j} \in \mathbb{N}^{I\setminus\{j\}} \setminus \Omega_{-j}$. Since the model is repulsive, $F_j^-(r|\cdot)$ is decreasing on Ω_{-j}. We follow Häggström and Nelander (1997a) in defining

$$F_j(s|y_{-j}) = \max\{F_j(s|x_{-j}) : x_{-j} \le y_{-j} \text{ and } x_{-j} \in \Omega_{-j}\}$$

for all $j \in I$, which ensures that $F_j^-(r|\cdot)$ is decreasing on $\mathbb{N}^{I\setminus\{j\}}$.

Aditionally, we define a function $\widetilde{\phi}_0 : [0,1]^I \to \mathbb{N}^I \times \mathbb{N}^I$ describing how to start the upper component of the backwards chain. Set

$$\widetilde{\phi}_0(r) = u \quad \text{with} \quad u_i = F_i^-(r_i|\hat{0}_{-i}) \tag{13}$$

for $i \in I$.

Write $\widetilde{\phi}_1(r,x,y) := x'$ and $\widetilde{\phi}_2(r,x,y) := y'$ for the components of $\widetilde{\phi}$. Note that $\widetilde{\phi}_1(r,x,x) = \widetilde{\phi}_2(r,x,x)$ for all $x \in \mathbb{N}^I$ and $\widetilde{\phi}_k(r,x,x),\ k = 1,2$ describes nothing but updating x according to the usual (one component) reversed cyclic Gibbs sampler. Throughout, this Gibbs sampler is assumed to be irreducible. We get the following monotonicity property of $\widetilde{\phi}$: If $x \le u \le v \le y$ then

$$\widetilde{\phi}_1(r,x,y) \le \widetilde{\phi}_1(r,u,v) \le \widetilde{\phi}_2(r,u,v) \le \widetilde{\phi}_2(r,x,y) \tag{14}$$

for all r. Moreover,

$$\widetilde{\phi}_1(r,x,x) \le \widetilde{\phi}_0(r) \tag{15}$$

for all r and x. The forwards updating rule ϕ is monotone as well, but for the proof of Proposition 3 below, only (14) and (15) are needed.

Rejection sampling and sandwiching are combined in the following algorithm, where we similar to (5) define a family of distributions $Q(x,x')$ for $x, x' \in \Omega$ by

$$r \sim Q(x,x') \iff r_i \sim \text{Uniform}\,(F_i(x'_i - |(x'_{<i}, x_{>i})), F_i(x'_i|(x'_{<i}, x_{>i}))) \tag{16}$$

for all $i \in I$. Note that we need $Q(x,x')$ to be defined for pairs $(x,x') \in \Omega \times \Omega$ only, because others can not occur as consecutive states of the forwards chain in step 2 below.

Algorithm 3.

0. Choose an initial natural number of transitions $N > 0$.

1. Generate independent random vectors $R_1, \ldots, R_N$ uniformly distributed on $[0,1]^I$.

2. Starting in $X_0 = \hat{0}$ and updating according to $X_t = \phi_1(R_t, X_{t-1}, X_{t-1})$, create $X_1, \ldots, X_N$.

3. For $t = 1, \ldots, N$, generate $\widetilde{R}_t \sim Q(X_{N+1-t}, X_{N-t})$.

4. Start with $L_1 = \hat{0}$ and $U_1 = \widetilde{\phi}_0(\widetilde{R}_1)$. Update the pair (L,U) according to

$$(L_t, U_t) = \widetilde{\phi}(\widetilde{R}_t, L_{t-1}, U_{t-1}) \quad \text{for} \quad t = 2, \ldots, N.$$

5. If $U_N = \hat{0}$ then accept $Z = X_N$ as a sample from π. Else, double N and start over from step 1.

Proposition 3. *Algorithm 3 shares the properties of Algorithm 1 stated in Proposition 1, except that the probability of acceptance in step 5 now is given by (22) below.*

PROOF: As in the proof of Proposition 1 we consider the joint probability of acceptance and $X_N = z$. Let $x_N = u_N = \hat{0}, x_0 = z, l_1 = \hat{0}$. Write

$$\begin{aligned} Q_{(x,x')}((l,u),(l',u')) &:= P(\widetilde{\phi}(\widetilde{R},l,u) = (l',u')) \quad \text{and} \\ Q_{0,(x,x')}(u) &:= P(\widetilde{\phi}_0(\widetilde{R}) = u), \quad \text{with} \quad \widetilde{R} \sim Q(x,x'). \end{aligned}$$

For $R \sim \text{Uniform}([0,1] \times I)$ define the transition matrices

$$P(x,x') := P(\phi(R,x,x) = (x',x')), \tag{17}$$

$$\widetilde{P}_2((x,y),(x',y')) := P(\widetilde{\phi}(R,x,y) = (x',y')), \tag{18}$$

$$\widetilde{P}_0(\hat{0},u) := P(\widetilde{\phi}_0(R) = u). \tag{19}$$

Note that, for all $x, y \in \Omega$,

$$P(x,y) = \frac{\pi(y)}{\pi(x)}\widetilde{P}(y,x) = \frac{\pi(y)}{\pi(x)}P(\widetilde{\phi}(R,y,y) = (x,x)). \tag{20}$$

The design of Algorithm 3 now gives

$$\begin{aligned} &P(U_N = \hat{0}, X_N = z \,|\, X_0 = \hat{0}, L_1 = \hat{0}) \\ &= \sum_{x_1,\ldots,x_{N-1}} \Big[\prod_{t=1}^{N} P(x_t, x_{t-1})\Big] Q_{0,(z,x_1)}(u_1) \times \\ &\qquad \times \sum_{\substack{l_2,\ldots,l_N \\ u_2,\ldots,u_{N-1}}} \prod_{t=2}^{N} Q_{(x_{t-1},x_t)}((l_{t-1},u_{t-1}),(l_t,u_t)) \\ &= \frac{\pi(z)}{\pi(\hat{0})} \sum_{\substack{x_1,\ldots,x_{N-1} \\ l_2,\ldots,l_N \\ u_2,\ldots,u_{N-1}}} P(\widetilde{\phi}(R_1,z,z) = (x_1,x_1), \widetilde{\phi}_0(R_1) = u_1) \times \\ &\quad \times \prod_{t=2}^{N} P(\widetilde{\phi}(R_t,x_{t-1},x_{t-1}) = (x_t,x_t), \widetilde{\phi}(R_t,l_{t-1},u_{t-1}) = (l_t,u_t)). \end{aligned} \tag{21}$$

By (15) and the monotonicity (14) of $\widetilde{\phi}$, the product on the right hand side is positive for paths with $l_t \le x_t \le u_t$, $t = 1,\ldots,N$ only and thus

$$\begin{aligned} &\sum_{l_N} P(\widetilde{\phi}(R_N,x_{N-1},x_{N-1}) = (\hat{0},\hat{0}), \widetilde{\phi}(R_N,l_{N-1},u_{N-1}) = (l_N,\hat{0})) \\ &\qquad = P(\widetilde{\phi}(R_N,l_{N-1},u_{N-1}) = (\hat{0},\hat{0})). \end{aligned}$$

Summing in (21) first over x_{N-1} and then $x_{N-2}, \ldots, x_1$ successively, finally over the l_t, u_t, we get

$$P(U_N = \hat{0}, X_N = z \,|\, X_0 = \hat{0}, L_1 = \hat{0}) = \frac{\pi(z)}{\pi(\hat{0})} \sum_u \widetilde{P}_0(\hat{0}, u) \widetilde{P}_2^{N-1}((\hat{0}, u), (\hat{0}, \hat{0})).$$

Hence,

$$P(U_N = \hat{0} \,|\, X_0 = \hat{0}, L_1 = \hat{0}) = \frac{1}{\pi(\hat{0})} \sum_u \widetilde{P}_0(\hat{0}, u) \widetilde{P}_2^{N-1}((\hat{0}, u), (\hat{0}, \hat{0})). \qquad (22)$$

As in the proof of Proposition 1, it suffices now to verify that Algorithm 3 terminates with probability 1. We proceed in a way similar to the proof of Møller (1997, Theorem 1). Consider a "target chain" $\widetilde{X}_t$, a pair of lower and upper chains (L_t, U_t), and a "dominating chain" D_t coupled via a sequence R_t, $t = 0, 1, \ldots$ of independent vectors uniformly distributed on $[0,1]^I$: We set $L_0 = \widetilde{X}_0 = \hat{0}$, $U_0 = D_0 = u$, and for $t = 1, 2, \ldots$, $\widetilde{X}_t = \widetilde{\phi}_1(R_t, \widetilde{X}_{t-1}, \widetilde{X}_{t-1})$, $(L_t, U_t) = \widetilde{\phi}(R_t, L_{t-1}, U_{t-1})$, and $D_t = \widetilde{\phi}_0(R_t)$. Then, for any $t \geq 0$, we have by (14) and (15) that $L_t \leq \widetilde{X}_t \leq U_t \leq D_t$. Clearly, with probability 1, $D_t = \hat{0}$ for some $t \geq 0$, and for the first such $T = t$ we get coalescence so that $L_s = \widetilde{X}_s = U_s$ for all $s \geq T$. Since $\widetilde{P}_2$ is the transition matrix for (L_t, U_t) and $\widetilde{X}_t$ converges in distribution to π, we conclude

$$\lim_{N\to\infty} \widetilde{P}_2^N((\hat{0}, u), (\hat{0}, \hat{0})) = \lim_{N\to\infty} \widetilde{P}^N(\hat{0}, \hat{0}) = \pi(\hat{0}).$$

Combining this with (22) gives $\lim_{N\to\infty} P(U_N = \hat{0} \,|\, \widetilde{X}_0 = \hat{0}, L_1 = \hat{0}) = 1$, so Algorithm 3 terminates almost surely.

Remark 5. The monotonicity properties (14) and (15) ensure that, for all z, the acceptance probability (22) is bounded from above by $\widetilde{P}^N(z, \hat{0})/\pi(\hat{0})$. (To see this, consider the Markov chains $\widetilde{X}$, defined by $\widetilde{\phi}$ and started in (z, z), and (L, U), defined by $(\hat{0}, \widetilde{\phi}_0)$ for the first transition and $\widetilde{\phi}$ for all further transitions, coupled via the same stream of random vectors.) Hence $\pi(\hat{0}) / \sum_u \widetilde{P}_0(\hat{0}, u) \widetilde{P}_2^{N-1}((\hat{0}, u), (\hat{0}, \hat{0}))$ is an upper bound of $\pi(z)/P^N(\hat{0}, z)$ and thus Algorithm 3 can be interpreted as rejection sampling in the same way as Algorithm 1.

Remark 6. Algorithm 3 can be used with random updating, too. The updating rule is now $\phi : ([0,1] \times I) \times \mathbb{N}^I \times \mathbb{N}^I \to \mathbb{N}^I \times \mathbb{N}^I$ defined by

$$\phi((r, j), x, y) = (x', y') \quad \text{with} \quad x'_i = \begin{cases} x_i & i \neq j \\ F_j^-(r | y_{-j}) & i = j \end{cases} \quad y'_i = \begin{cases} y_i & i \neq j \\ F_j^-(r | x_{-j}) & i = j \end{cases}$$

for $i \in I$. As Gibbs sampling with random site updating is time reversible, we can use ϕ for both the forwards and the backwards chains, that is $\widetilde{\phi} = \phi$. The equivalent of the monotonicity (14) still holds. However, it is more difficult than in the cyclic updating case to get a starting state for the upper component of the backwards chain. Instead of conditioning on $\hat{0}$ just for the first cycle, one has to wait now (for a random number of transitions) till each site has been updated at least once (in the backwards chain). This complication does not occur for models with a maximum $\hat{1} \in \Omega$. In this case we can extend Algorithm 3 and Proposition 3 to random updating by a few replacements: For (16) substitute

$$(r, j) \sim Q(x, x') \iff \begin{array}{l} x'_i = x_i,\ i \neq j \quad \text{and} \\ r \sim \text{Uniform}\,(F_j(x'_j - |x_{-j}), F_j(x'_j|x_{-j})). \end{array}$$

Change step 4 in Algorithm 3 in the following way: Start with $L_0 = \hat{0}$ and $U_0 = \hat{1}$. Update the pair (L, U) according to

$$(L_t, U_t) = \phi(\widetilde{R}_t, L_{t-1}, U_{t-1}) \quad \text{for} \quad t = 1, \ldots, N.$$

The corresponding changes in the proof are obvious.

5 Experimental results

So far, experimental results for Fill's algorithm are restricted to the Ising model by Fismen (1997) and the penetrable spheres model by Thönnes (1997). We conducted a small scale simulation study in order to compare the running times of Algorithms 2 and 3 and their CFTP equivalents, see Møller (1997). Moreover, we consider below two models with two coding sets in order to compare the two methods – the anti-monotone Gibbs sampler and sandwiching as in Algorithm 3 and the monotone two-step Gibbs sampler as in Algorithm 2. Here we interpret running time as the number of loops through the whole algorithm needed till acceptance or coalescence, respectively.

We considered the auto-Poisson and the hard-core model on a square lattice $I = (k, l)_{k,l=1}^{n}$ with 4-neighbourhood and periodic boundaries. That means the neighbourhood relation $\sim$ on I is defined by

$$(k_1, l_1) \sim (k_2, l_2) \text{ if and only if } \begin{array}{l} k_1 = k_2 \text{ and } |l_1 - l_2| \in \{1, n-1\} \text{ or} \\ l_1 = l_2 \text{ and } |k_1 - k_2| \in \{1, n-1\}. \end{array}$$

We will refer to n as the size of the lattice in the following. Clearly, even n allows a decomposition of I into two coding sets as assumed for Algorithm 2.

First, we introduce shortly the hard-core model. For more details, see for example Georgii (1988). A configuration $x \in \{0,1\}^I$ is called *feasible* if and only if $x_i x_j = 0$ for all $i \sim j$. Now the hard-core model with activity β is defined by

$$\pi_\beta(x) \propto \mathbb{1}\{x \text{ feasible}\} \prod_{i \in I} \beta^{x_i},$$

where $\propto$ denotes proportionality. For lattice size 2 the normalizing constant and thus the probabilities of all feasible configurations can be easily determined. This allowed the used C/Splus-programs to be checked by a χ^2-test. The code is available via anonymous ftp at
`ftp://ftp.math.auc.dk/pub/jm/`.
All simulation results in the following are based on 20 samples for each (even) lattice size and parameter value.

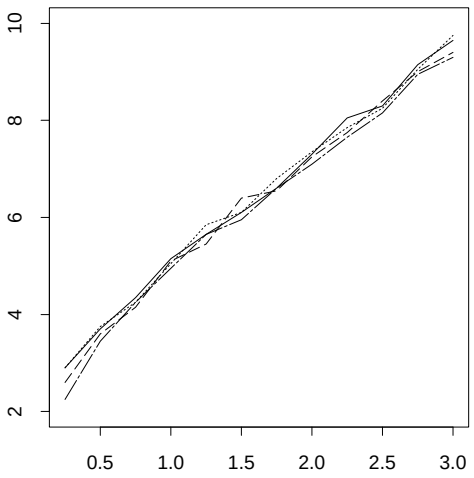
A drawback of Algorithms 2 is the need to store the complete forwards trajectory. Fill (1997) addresses this problem, suggesting variations of his algorithm, where time is traded off against storage. However, we implemented Algorithm 2 and 3 as given in Sections 3 and 4, respectively. Hence the simulations were restricted by storage rather than by time. When the number of transitions N was doubled after each loop, all four algorithms showed the same behaviour. See Figure 1 for details. However, due to the much higher need for storage, the Fill style algorithms both needed up to twice (size 50, activity 3) as much real computing time as their CFTP equivalents. The picture changes slightly in the practically rather irrelevant case of increasing N by 1 instead of doubling, see Figure 2. The real computing times too are in favour of the Fill style algorithms in this case. This might be a hint that doubling N is not the ideal strategy for these algorithms and a compromise like adding a fixed number (> 1) of iterations in each loop could enhance the performance.

Our second example is the auto-Poisson model, which is specified by the conditional distributions of each component given the values of its neighbours. We chose the special version

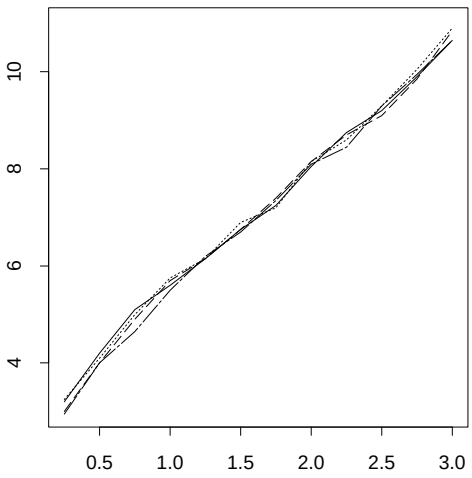
$$\mathcal{D}(X_i | X_{-i}) = \text{Poisson}(\exp(a + b \sum_{j : i \sim j} X_j)).$$

This model is well defined for $a \in \mathbb{R}, b < 0$, see Besag (1974).

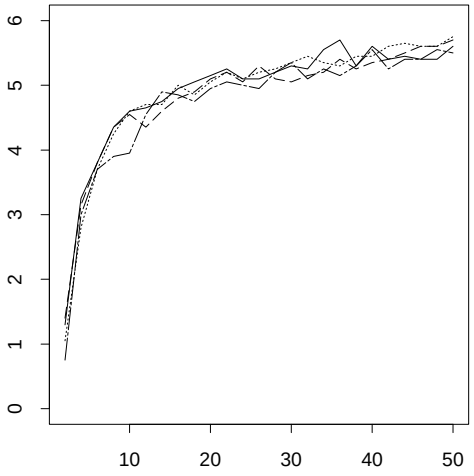
For small b the auto-Poisson model approaches the hard-core model with a acting as an activity parameter. As for the hard-core model, the simulations showed no significant differences between the four algorithms. For $a = 0.4$ and $b = -2.5$ the algorithms needed as many loops as for the hard-core model with activity 3.



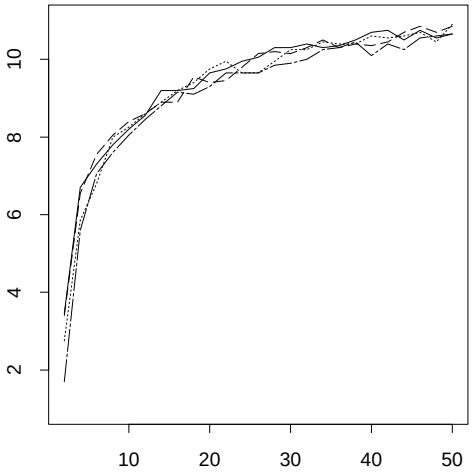
(a) Size 20, mean number of loops versus activity



(b) Size 50, mean number of loops versus activity

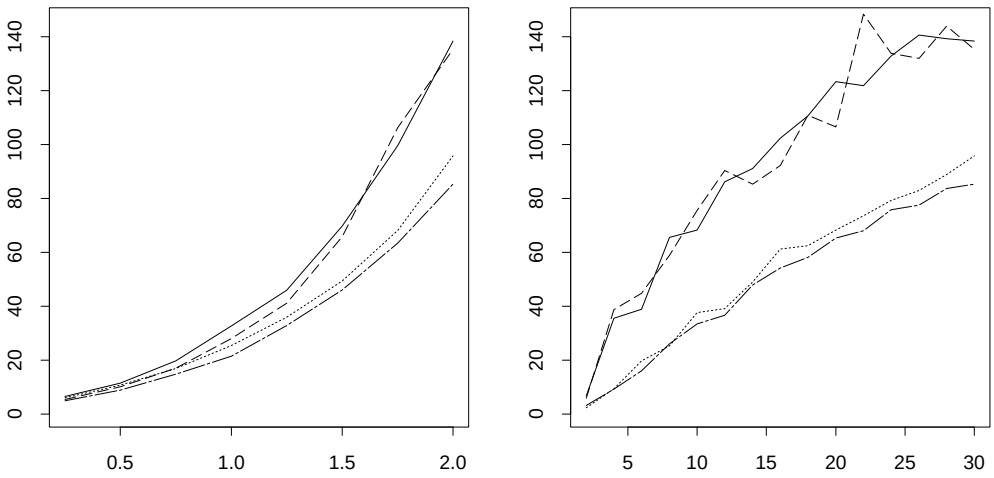


(c) Activity 1, mean number of loops versus size



(d) Activity 3, mean number of loops versus size

Figure 1: Hard-core model, mean number of loops of the algorithms CFTP, anti-monotone (solid line); Fill, anti-monotone (dotted line); CFTP, monotone (dashed line); Fill, monotone (dashed dotted line) versus activity and lattice size, N doubled.



(a) Size 30, mean number of loops versus activity

(b) Activity 2, mean number of loops versus size

Figure 2: Hard-core model, mean number of loops of the algorithms CFTP, anti-monotone (solid line); Fill, anti-monotone (dotted line); CFTP, monotone (dashed line); Fill, monotone (dashed dotted line) versus activity and lattice size, N increased by 1.

However, the main motivation for considering the auto-Poisson model is to approximate point processes, see Besag *et al.* (1982). Consider the Strauss process on the unit square (Strauss (1975), Kelly and Ripley (1976)) defined by its density

$$f(x) \propto \beta^{\#x}\gamma^{s(x)}$$

with respect to the unit rate homogeneous Poisson process. Here $x \subseteq [0,1]^2$ is a finite point configuration, # denotes cardinality, and $s(x) = \frac{1}{2}\#\{(\xi,\eta) \in x \times x : 0 < ||\xi - \eta|| \leq R\}$ is the number of point pairs in x at distance less than the interaction radius R from each other. The parameters $\beta > 0$ and $0 \leq \gamma \leq 1$ control the intensity and the strength of the interaction.

To approximate a Strauss process, typically much bigger neighbourhoods are needed. If we want to approximate a Strauss process with parameters $\beta = 100, \gamma = 0.5$ and interaction radius $R = 0.05$ by the auto-Poisson model on a lattice of size 50, we have to chose $a = -3.2$ and $b = -0.7$, and a circular neighbourhood consisting of 20 sites. A small experiment with these parameters showed, that the size of the neighbourhood is less critical than the strength of the repulsion. The two anti-monotone algorithms both needed

around 2 loops.

6 Discussion

We presented a class of examples where Fill type algorithms work without direct monotonicity of the sampler and without maximal element. Nevertheless, CFTP, in particular if combined with dominating chain and sandwiching, seems to have a wider scope of possible applications.

One restriction of rejection sampling arises from the requirement $\pi(\hat{0}) > 0$, which prevents extension to continuous state space models such as the auto-gamma model, for which Møller (1997) introduced a CFTP algorithm. However, perfect rejection sampling is possible for distributions on $\Omega \subseteq [0,\infty)^I$, where π now is a density w.r.t. the product measure on Ω obtained from the product of the Dirac measure in 0 and the Lebesgue measure on $(0,\infty)$. Currently, Murdoch and Rosenthal (1998) are investigating other ways of perfect simulation for continuous state space models combining CFTP and Fill's algorithm. Duncan Murdoch pointed out to us, that we could start the forwards run in Algorithm 3 in an arbitray state x with $\pi(x) > 0$ and then check whether the lower and the upper backwards "chains" coalesce to x at time N (that is $L_N = U_N = x$) in step 5 of Algorithm 3.

Kendall (1996) combined CFTP and spatial birth-and-death processes, yielding in particular a perfect simulation algorithm for repulsive area interaction processes. (See Kendall and Møller (1998) for further extensions.) We could not find an equivalent using rejection sampling. There exists a $\hat{0}$ – the empty point configuration – with $\pi(\hat{0}) > 0$. A trivial possibility is to accept if and only if the proposal structure at time 0 was empty. This yields of course the right distribution for the output, but the acceptance probability is impractically small. An algorithm in the spirit of Algorithm 3 fails due to the intricate dependence structure. That means the output of the forwards run and the event of acceptance are not independent. Metropolis-Hastings alternatives as suggested in Kendall and Møller (1998) would not overcome this problem either.

Acknowledgement

This research was supported by the European Union's research network "Statistical and Computational Methods for the Analysis of Spatial Data. ERB-FMRX-CT96-0096" and by MaPhySto – Centre for Mathematical Physics and Stochastics, funded by a grant from The Danish National Research Foun-

dation. We thank Martin B. Hansen for programming assistence and Duncan J. Murdoch and Elke Thönnes for useful comments.

References

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society* **B 36**, 192–236.

Besag, J., Milne, R. and Zachary, S. (1982). Point process limits of lattice processes. *Journal of Applied Probability* **19**, 210–216.

Borovkov, A. A. and Foss, S. G. (1992). Stochastically recursive sequences and their generalizations. *Siberian Advances in Mathematics* **2**, 16–81, translated from PRIM.

Corcoran, J. and Tweedie, R. (1998a). Perfect sampling from independent Metropolis-Hastings chains. Preprint.

Corcoran, J. and Tweedie, R. (1998b). Perfect sampling of Harris recurrent Markov chains. Preprint.

Fill, J. A. (1997). An interruptible algorithm for perfect sampling via Markov chains. To appear in *Annals of Applied Probability*. Available at `http://www.mts.jhu.edu/~fill/`. Preprint.

Fismen, M. (1997). *Exact sampling using Markov chains*. Diploma thesis, Department of Mathematical Sciences, Norwegian University of Technology and Science, Trondheim.

Foss, S. G. and Tweedie, R. L. (1997). Perfect simulation and backward coupling. To appear in *Stochastic Models*. Preprint.

Foss, S. G., Tweedie, R. L. and Corcoran, J. N. (1997). Simulating the invariant measures of Markov chains using backward coupling at regeneration times. To appear in *Probability in the Engineering and Informational Sciences*. Preprint.

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

Häggström, O. and Nelander, K. (1997a). Exact sampling from anti-monotone systems. *Research Report 1997-03*, Department of Mathematics, Chalmers University, to appear in *Statistica Neerlandica*.

Häggström, O. and Nelander, K. (1997b). On exact simulation of Markov random fields using coupling from the past. Submitted. Preprint.

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. *Technical Report R-96-2040*, Aalborg University, submitted.

Kelly, F. and Ripley, B. (1976). On Strauss's model for clustering. *Biometrika* **63**, 357–360.

Kendall, W. S. (1996). Perfect simulation for the area-interaction point process. *Research Report 292*, Department of Statistics, University of Warwick, to appear in C.C. Heyde and L. Accardi, editors, *Probability Perspective*, World Scientific Press.

Kendall, W. S. and Møller, J. (1998). Perfect Metropolis-Hastings simulation of locally stable spatial point processes. In preparation.

Kifer, Y. (1986). *Ergodic Theory of Random Transformations*. Birkhäuser, Basel.

Møller, J. (1997). Perfect simulation of conditionally specified models. *Research Report R-97-2006*, Aalborg University, to appear in *Journal of Royal Statistical Society, Ser. B.*

Murdoch, D. J. and Green, P. J. (1997). Exact sampling from a continuous state space. *Research Report S-97-01*, Department of Mathematics, University of Bristol, to appear in *Scandinavian Journal of Statistics.*

Murdoch, D. J. and Rosenthal, J. S. (1998). Personal communication.

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

Roberts, G. O. and Smith, A. F. M. (1994). Simple conditions for the convergence of the Gibbs sampler and the Metropolis-Hastings algorithms. *Stochastic Processes and their Applications* **49**, 207–216.

Strauss, D. (1975). A model for clustering. *Biometrika* **63**, 467–475.

Thönnes, E. (1997). Perfect simulation of some point processes for the impatient user. Preprint, Department of Statistics, University of Warwick.

The preprints Corcoran and Tweedie (1998a,b), Foss and Tweedie (1997), Foss *et al.* (1997), Häggström *et al.* (1996), Kendall (1996), Møller (1997), Murdoch and Green (1997) and Thönnes (1997) are available from the MCMC preprint service at `http://www.stats.bris.ac.uk/MCMC/`.