EXACT SAMPLING USING BRANCHING PARTICLE SIMULATION

Christophe Andrieu, Nicolas Chopin, Arnaud Doucet, Sylvain Rubenthaler

EXACT SAMPLING USING BRANCHING PARTICLE SIMULATION

CHRISTOPHE ANDRIEU, NICOLAS CHOPIN, ARNAUD DOUCET, AND SYLVAIN RUBENTHALER

Abstract. Particle methods, also known as Sequential Monte Carlo methods, are a popular set of computational tools used to sample approximately from non-standard probability distributions. A variety of convergence results ensure that, under weak assumptions, the distribution of the particles converges to the target probability distribution of interest as the number of particles increases to infinity. Unfortunately it can be difficult to determine practically how large this number needs to be to obtain a reliable approximation. We propose here a procedure which allows us to return exact samples. The proposed algorithm relies on the combination of an original branching variant of particle Markov chain Monte Carlo methods and dominated coupling from the past.

1. Introduction

We are given spaces E_1, E_2, ..., M_1 a probability measure on E_1, for each $k \geq 2$ a transitition kernel M_k from E_{k-1} to E_k and potentials $(G_k)_{k\geq 1}$ ($G_k : E_k \to \mathbb{R}_+$). All densities and kernels are supposed to have a density with respect to some reference measures μ_k on E_k ($k = 1, 2, \ldots, T$). Moreover, in the following, densities on enumerable sets will always be taken with respect to the counting measure. In the case we write a density on a space defined as a product of spaces E_i and enumerable spaces, the reference measure will be the product of the measures mentionned above. We want to draw samples according to the law (on paths of length T) defined for any measurable function f by

$$\pi(f) = \frac{\mathbb{E}\left(f(X_1, \ldots, X_T) \prod_{k=1}^{T-1} G_k(X_k)\right)}{\mathbb{E}\left(\prod_{k=1}^{T-1} G_k(X_k)\right)} \tag{1.1}$$

where $(X_k)_{k\geq 1}$ is Markov with initial law M_1 and transitions $(M_k)_{k\geq 2}$ (for all $k \geq 1$, X_k takes values in E_k). For all $n \in \mathbb{N}^*$, we note $[n] = \{1, \ldots, n\}$. We set $Z_T = \mathbb{E}\left(\prod_{i=1}^{T-1} G_i(X_i)\right)$. Then π has the following density at $(x_1, \ldots, x_T) \in E_1 \times \cdots \times E_T$:

$$\pi(x_1, \ldots, x_T) = \frac{1}{Z_T} M_1(x_1) \prod_{k=1}^{T-1} G_k(x_k) M_{k+1}(x_k, x_{k+1}) \,. \tag{1.2}$$

2. Densities of branching processes

2.1. **Branching process.** We first introduce some definitions concerning the elements of $(\mathbb{N}^*)^k$ for $k = 1, 2, \ldots$. If $1 \leq q \leq n$ and $\mathbf{i} = (i_1, \ldots, i_n) \in (\mathbb{N}^*)^n$, we define $\mathbf{i}(q) = (i_1, \ldots, i_q)$ and we say that $\mathbf{i}(q)$ is an ancestor of $\mathbf{i}$ and we denote this relation by $\mathbf{i}(q) \prec \mathbf{i}$; we will also say that $\mathbf{i}$ is a descendant of $\mathbf{i}(q)$. We introduce the notation: if $\mathbf{i} = (i_1, \ldots, i_k) \in (\mathbb{N}^*)^k$ and $j \in \mathbb{N}^*$, $(\mathbf{i}, j) = (i_1, \ldots, i_k, j) \in (\mathbb{N}^*)^{k+1}$.

We now define a branching process. We start with n_1 particles, i.i.d. with law M_1 (n_1 is a fixed number). We then proceed recursively through time to build a genealogical tree. At time 1, the particles are denoted by $X_1^1, \ldots, X_1^{n_1}$. We set

$$S_1 = \{1, 2, \ldots, n_1\} \,. \tag{2.1}$$

Date: 5th November 2014.

Key words and phrases. Coupling from the past, Exact simulation, Particle methods, Particle Markov chain Monte Carlo Methods.

At time k, the particles are denoted by $(X_k^{\mathbf{i}})_{\mathbf{i}\in S_k}$, where S_k is a finite subset of $(\mathbb{N}^*)^k$, and the number of particles is $N_k = \#S_k$ (the cardinality of S_k). For $\mathbf{i} \in S_k$, we say that $X_k^{\mathbf{i}}$ is the position of the particle indexed by $\mathbf{i}$, or, in a shorter way, the position of $\mathbf{i}$. In an abuse of notation, in the case $\mathbf{j} \prec \mathbf{i}$ ($\mathbf{j} \in S_q$), we also say that $X_q^{\mathbf{j}}$ is an ancestor of $X_k^{\mathbf{i}}$ and that $X_k^{\mathbf{i}}$ is a descendant of $X_k^{\mathbf{j}}$. Starting from the particles at time $k \leq T-1$ with particles $(X_k^{\mathbf{i}})_{\mathbf{i}\in S_k}$ ($S_k \subset (\mathbb{N}^*)^k$), the system evolves in the following manner:

- For each $\mathbf{i} \in S_k$, the number of children of $\mathbf{i}$ is a random variable $N_{k+1}^{\mathbf{i}}$ with law f_{k+1} such that :
$$\mathbb{P}(\, N_{k+1}^{\mathbf{i}} = j \big|\, X_k^{\mathbf{i}}) = f_{k+1}(G_k(X_k^{\mathbf{i}}), j)\,.$$
Here, f_{k+1} is a law with a parameter $G_k(X_k^{\mathbf{i}})$, we will define this law precisely later. We suppose that
$$f_{k+1}(G_k(x), 0) = 1 - \alpha_{k+1} G_k(x) \tag{2.2}$$
for some $\alpha_{k+1} \geq 0$. This will remain true trough all the paper. The variables $N_{k+1}^{\mathbf{i}}$ ($1 \leq i \leq N_k$) are independent. The total number of particles at time $k+1$ is then $N_{k+1} = \sum_{\mathbf{i}\in S_k} N_k^{\mathbf{i}}$.
- If $N_{k+1} \neq 0$, we draw σ_{k+1} uniformly in $\mathfrak{S}_{N_{k+1}}$ (the N_{k+1}-th symmetric group). If $N_{k+1} = 0$, we use the convention $\mathfrak{S}_{N_{k+1}} = \emptyset$ and the system stops here.
- We order S_k alphanumerically: $S_k = \{\mathbf{i}_1, \ldots, \mathbf{i}_{N_k}\}$. For $r \in [N_k]$, we set $C_{k+1}^r = \{1 + \sum_{l=1}^{r-1} N_k^{\mathbf{i}_l}, \ldots, \sum_{l=1}^{r} N_k^{\mathbf{i}_l}\}$. We set
$$S_{k+1} = \cup_{r=1}^{N_k} \cup_{j\in\sigma_{k+1}(C_{k+1}^r)} (\mathbf{i}_r, j)\,.$$
For $r \in [N_k]$, $j \in \sigma_{k+1}(C_{k+1}^r)$, we draw $X_{k+1}^{(\mathbf{i}_r,j)} \sim M_{k+1}(X_k^{\mathbf{i}_r}, .)$ in E_{k+1}. To simplify the notation, we write $C(\mathbf{i}_r) = C_{k+1}^r$ for all $r \in [N_k]$ and $C(\mathbf{i}) = C(\mathbf{i}_r)$ if $\mathbf{i} = \mathbf{i}_r \in S_k$. We can then write that, for $\mathbf{i} \in S_k$, the descendants of $\mathbf{i}$ at time $k+1$ are the $(\mathbf{i}, j)$ for $j \in \sigma_{k+1}(C(\mathbf{i}))$.

Such a system has a density on the space

$$\begin{aligned} E = E_1^{n_1} \times \cup_{n_2^{\mathbf{i}}, \mathbf{i}\in S_1} \{n_2^{\mathbf{i}}, \mathbf{i} \in S_1\} \times E_2^{n_2} \times \mathfrak{S}_{n_2} \\ \times \cup_{n_3^{\mathbf{i}}, \mathbf{i}\in S_2} \{n_3^{\mathbf{i}}, \mathbf{i} \in S_2\} \times E_3^{n_3} \times \mathfrak{S}_{n_3} \times \ldots \\ \cdots \times \cup_{n_T^{\mathbf{i}}, \mathbf{i}\in S_{T-1}} \{n_T^{\mathbf{i}}, \mathbf{i} \in S_{T-1}\} \times E_T^{n_T} \times \mathfrak{S}_{n_T}. \end{aligned}$$

where we recall that n_1 is given, $S_1 = [n_1]$ and for $k \in \{1, \ldots, T-1\}$, $n_k = \sum_{\mathbf{i}\in S_{k-1}} n_k^{\mathbf{i}}$, $S_k = \{\mathbf{i}_1, \ldots, \mathbf{i}_{n_k}\}$, $C_{k+1}^r = \{1 + \sum_{l=1}^{r-1} n_k^{\mathbf{i}_l}, \ldots, \sum_{l=1}^{r} n_k^{\mathbf{i}_l}\}$, $S_{k+1} = \cup_{r=1}^{n_k} \cup_{j\in\sigma_{k+1}(C_{k+1}^r)} (\mathbf{i}_r, j)$. A random variable on the space E will be called a branching process. At a point $(x_k^{\mathbf{i}}, n_k^{\mathbf{i}}, \sigma_k) \in E$, $x_k^{\mathbf{i}} \in E_k$ ($1 \leq k \leq T, \mathbf{i} \in S_k$), $n_k^{\mathbf{i}} \in \mathbb{N}$ ($2 \leq k \leq T, \mathbf{i} \in S_{k-1}$), $\sigma_k \in \mathfrak{S}_{n_k} (2 \leq k \leq T)$, this density is given by

$$\begin{aligned} q((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k}, (n_k^{\mathbf{i}})_{k\in\{2,\ldots,T\},\mathbf{i}\in S_k}, (\sigma_k)_{k\in\{2,\ldots,T\}}) = \\ \prod_{\mathbf{i}\in S_1} M_1(x_1^{\mathbf{i}}) \prod_{k=1}^{T-1} \left\{ \prod_{\mathbf{i}\in S_k} f_{k+1}(G_k(x_k^{\mathbf{i}}), n_{k+1}^{\mathbf{i}}) \times \frac{1}{n_{k+1}!} \times \prod_{\mathbf{i}\in S_k} \prod_{j\in\sigma_{k+1}(C(\mathbf{i}))} M_{k+1}(x_k^{\mathbf{i}}, x_{k+1}^{(\mathbf{i},j)}) \right\}. \end{aligned} \tag{2.3}$$

In the following, we write the shortcut notation $((x_k^{\mathbf{i}}), (n_k^{\mathbf{i}}), (\sigma_k))$ for a point in E. For $\chi \in E$, we write $S_k(\chi)$, $k \in [T]$ for the corresponding subsets of $(\mathbb{N}^*)^k$, $k \in [T]$, we write $N_k(\chi)$ for the number of particles at each timestep $k \in [T]$. We will use the same notations for a point $\chi \in E'$ (E' defined below).

2.2. **Proposal distribution.** We introduce the space

$$E' = E \times (\mathbb{N}^*)^T\,.$$

Suppose we draw a branching process $((X_k^{\mathbf{i}}), (N_k^{\mathbf{i}}), (\sigma_k))$ with law q and then draw $\mathbf{B}$ uniformly in S_T if $N_T \geq 1$, and set $\mathbf{B} = (1, 1, \ldots, 1)$ if $N_T = 0$. This random variable takes values in E' and, at

a point $((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_k},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b})$ with $((x_k^{\mathbf{i}}),(n_k^{\mathbf{i}}),(\sigma_k))\in E$ such that $n_T\geq 1$, $\mathbf{b}\in S_T$ ($S_T\subset(\mathbb{N}^*)^T$ deduced from $((x_k^{\mathbf{i}}),(n_k^{\mathbf{i}}),(\sigma_k))\in E$), it has the density

$$\widehat{q}((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_k},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b}) = q((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_k},(\sigma_k)_{k\in\{2,\dots,T\}})\times\frac{1}{n_T}\,. \tag{2.4}$$

We call this the proposal distribution. One can view the space E' as the space of branching processes where a particular trajectory is singled out, we will call it the colored trajectory. At a point $((x_k^{\mathbf{i}}),(n_k^{\mathbf{i}}),(\sigma_k),\mathbf{b})$, what we call the colored trajectory is $(x_1^{\mathbf{b}(1)},\dots,x_T^{\mathbf{b}(T)})$ in the case $n_T\geq 1$; in the case $n_T=0$, the colored trajectory is $(x_1^1,x_2^{(1,1)},\dots,x_q^{(1,\dots,1)})$ where $q=\max\{k:n_k\geq 1\}$.

2.3. **Target distribution.** Let us denote by $\widehat{f}(g,.)$ the law $f(g,.)$ conditionned to be ≥ 1, that is: for all $g\geq 0$, $i\geq 1$, $k\geq 1$,

$$\widehat{f}_k(g,i)=\frac{f_k(g,i)}{1-f_k(g,0)}\,. \tag{2.5}$$

This quantity is not defined in the case $g=0$ but we will no need it in this case. An alternative way of building a branching process with a colored trajectory is to draw a trajectory with law π, say that is is the colored trajectory and then build a branching process conditionned to contain this trajectory. The indexes of the colored trajectory embeded in the branching process are denoted by a random variable $\mathbf{B}$. The first coordinate $\mathbf{B}(1)$ is chosen uniformly in $[n_1]$. The other coordinates are deduced from the branching process in the following way: suppose that, at time $k+1$, the random permutation of the branching process is σ_{k+1} and the numbers of children are $(N_{k+1}^{\mathbf{i}})_{\mathbf{i}\in S_k}$, we set $\mathbf{B}_{k+1}=\sigma_{k+1}(\min\{C(\mathbf{B}(k))\})$ and $\mathbf{B}(k+1)=(\mathbf{B}(k),\mathbf{B}_{k+1})$. We thus introduce what we call the target distribution. Its support is contained in $\{((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_k},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b})\in E' : n_T\geq 1\}$ and it has the density:

$$\begin{aligned}&\widehat{\pi}((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_{k-1}},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b})\\ &\qquad=\frac{1}{n_1}\pi\left(x_1^{\mathbf{b}(1)},\dots,x_T^{\mathbf{b}(T)}\right)\prod_{\mathbf{i}\in S_1}M_1(x_1^{\mathbf{i}})\\ &\qquad\times\prod_{k=1}^{T-1}\left\{\left(\prod_{\mathbf{i}\in S_k,\mathbf{i}\neq\mathbf{b}(k)}f_{k+1}(G_k(x_k^{\mathbf{i}}),n_{k+1}^{\mathbf{i}})\right)\times\widehat{f}_{k+1}(G_k(x_k^{\mathbf{b}(k)}),n_{k+1}^{\mathbf{b}(k)})\times\frac{1}{n_{k+1}!}\right.\\ &\left.\times\left(\prod_{\mathbf{i}\in S_k,\mathbf{i}\neq\mathbf{b}(k)}\prod_{j\in\sigma_{k+1}(C(\mathbf{i}))}M_{k+1}(x_k^{\mathbf{i}},x_{k+1}^{(\mathbf{i},j)})\right)\left(\prod_{j\in\sigma_{k+1}(C(\mathbf{b}(k))\setminus\min\{C(\mathbf{b}(k))\}}M_{k+1}(x_k^{\mathbf{b}(k)},x_{k+1}^{(\mathbf{i},j)})\right)\right\},\end{aligned} \tag{2.6}$$

where the term $\widehat{f}_{k+1}(G_k(x_k^{\mathbf{b}(k)}),n_{k+1}^{\mathbf{b}(k)})$ corresponds to the simulation of the number of offsprings of the particle $x_k^{\mathbf{b}(k)}$. Using (1.2). (2.2) and (2.5), we can rewrite $\widehat{\pi}$ into

$$\begin{aligned}&\widehat{\pi}((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_{k-1}},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b})\\ &=\frac{1}{n_1}\times\prod_{i=1}^{n_1}M_1(x_1^i)\times\prod_{k=2}^{T}\left\{\prod_{i\in S_{k-1}}\left(f_k(G_{k-1}(x_{k-1}^{\mathbf{i}}),n_k^{\mathbf{i}})\times\prod_{j\in[n_k^{\mathbf{i}}]}M_k(x_{k-1}^{\mathbf{i}},x_k^{(\mathbf{i},j)})\right\}\right)\\ &\qquad\times\frac{1}{Z_T\alpha_1\dots\alpha_{T-1}}\,.\end{aligned} \tag{2.7}$$

2.4. **Ratio of densities.** We deduce from (2.3), (2.4) and (2.7), that at a point in the support of $\widehat{\pi}$, the ratio of $\widehat{\pi}$ and $\widehat{q}$ is equal to

$$\frac{\widehat{\pi}((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_{k-1}},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b})}{\widehat{q}((x_k^{\mathbf{i}})_{k\in[T],\mathbf{i}\in S_k},(n_k^{\mathbf{i}})_{k\in\{2,\dots,T\},\mathbf{i}\in S_{k-1}},(\sigma_k)_{k\in\{2,\dots,T\}},\mathbf{b})}=\frac{n_T}{n_1Z_T\alpha_1\dots\alpha_{T-1}}\,. \tag{2.8}$$

3. A Markov chain on $E_1 \times \cdots \times E_T$

We now define a Markov kernel Q on $E_1 \times \cdots \times E_T$. We start from a path $(x_1, \ldots, x_T) \in E_1 \times \cdots \times E_T$. We will move to a new path in several steps.

(1) Draw of a conditionnal forest. We sample a random variable χ with law $\widehat{\pi}$ conditionnaly on $(X_1^{\mathbf{B}(1)}, \ldots, X_T^{\mathbf{B}(T)}) = (x_1, \ldots, x_T)$. We use for this the expression (2.6). Such a sampling can be done recursively in $k \in [T]$ in the following way.
- We take $\mathbf{B}(1) = \mathbf{B}_1$ uniformly in $[n_1]$. We take $(X_1^{\mathbf{i}})_{\mathbf{i} \in S, \mathbf{i} \neq \mathbf{B}_1}$ i.i.d. with law M_1.
- Suppose we have sampled $((X_q^{\mathbf{i}})_{1 \leq q \leq k, \mathbf{i} \in S_q}, (N_q^{\mathbf{i}})_{2 \leq q \leq k, \mathbf{i} \in S_{q-1}}, S_k)$ for $k \leq T-1$. For $\mathbf{i} \in S_k$, we take $N_{k+1}^{\mathbf{i}}$ with law $f_{k+1}(G_k(X_k^{\mathbf{i}}), .)$ if $\mathbf{i} \neq \mathbf{B}(k)$ and $N_{k+1}^{\mathbf{B}(k)}$ with law $\widehat{f}_{k+1}(G_k(X_k^{\mathbf{B}(k)}), .)$. We set $N_{k+1} = \sum_{\mathbf{i} \in S_k} N_k^{\mathbf{i}}$. We draw σ_{k+1} uniformly in $\mathfrak{S}_{N_{k+1}}$. We set $\mathbf{B}_{k+1} = \sigma_{k+1}(\min\{C(\mathbf{B}(k))\})$, $\mathbf{B}(k+1) = (\mathbf{B}(k), \mathbf{B}_{k+1})$. We set $S_{k+1} = \cup_{\mathbf{i} \in S_k} \cup_{j \in \sigma_{k+1}(C(\mathbf{i}))} (\mathbf{i}, j)$.
  - For $\mathbf{i} \in S_k$, $\mathbf{i} \neq \mathbf{B}(k)$, $j \in \sigma_{k+1}(C(\mathbf{i}))$, we take $X_{k+1}^{(\mathbf{i},j)}$ with law $M_{k+1}(X_k^{\mathbf{i}}, .)$.
  - For $\mathbf{i} = \mathbf{B}(k)$, $j \in \sigma_{k+1}(C(\mathbf{i}))$, $j \neq \mathbf{B}_{k+1}$, we take $X_{k+1}^{(\mathbf{i},j)}$ with law $M_{k+1}(X_k^{\mathbf{i}}, .)$.

(2) Proposal. We draw a proposal $\overline{\chi} = ((\overline{X}_k^i)_{1 \leq k \leq T, \mathbf{i} \in S_k}, (\overline{N}_k^{\mathbf{i}})_{k \in \{2,\ldots,T\}, \mathbf{i} \in S_{k-1}}, (\overline{\sigma}_k)_{k \in \{2,\ldots,T\}},)$ with law $\widehat{q}$. It contains a colored trajectory $(\overline{X}_1^{\overline{\mathbf{B}}(1)}, \ldots, \overline{X}_T^{\overline{\mathbf{B}}(T)})$.

(3) Accept/reject step. We move to $(\overline{X}_1^{\overline{\mathbf{B}}(1)}, \ldots, \overline{X}_T^{\overline{\mathbf{B}}(T)})$ with probability $\min\left(1, \frac{N_T(\overline{\chi})}{N_T(\chi)}\right)$ and we stay in $(x_1, \ldots, x_T)$ with probability $1 - \min\left(1, \frac{N_T(\overline{\chi})}{N_T(\chi)}\right)$.

Theorem 3.1. *The law π is invariant for the kernel Q.*

Proof. Suppose we start with a random variable $(X_1, \ldots, X_T)$ with law π. Going through step 1 of the construction above, we get a random variable

$$\chi = ((X_k^{\mathbf{i}})_{k \in [T], \mathbf{i} \in S_k}, (N_k^{\mathbf{i}})_{k \in \{2,\ldots,T\}, \mathbf{i} \in S_{k-1}}, (\sigma_k)_{k \in \{2,\ldots,T\}}, \mathbf{B})$$

in E' such that $(X_1^{\mathbf{B}(1)}, \ldots, X_T^{\mathbf{B}(T)}) = (X_1, \ldots, X_T)$. By (2.6), χ has the law $\widehat{\pi}$. We draw a random variable $\overline{\chi}$ with law $\widehat{q}$ as in step 2 above. We then proceed to the step 3 above. Let U be a uniform variable in $[0, 1]$. We set

$$\widehat{\chi} = \begin{cases} \overline{\chi} & \text{if } U \leq \min\left(1, \frac{N_T(\overline{\chi})}{N_T(\chi)}\right) , \\ \chi & \text{otherwise.} \end{cases}$$

The result of the random move by the Markov kernel Q is the colored trajectory of χ. By (2.8), we have that

$$\frac{N_T(\overline{\chi})}{N_T(\chi)} = \frac{\widehat{\pi}(\overline{\chi})\widehat{q}(\chi)}{\widehat{q}(\overline{\chi})\widehat{\pi}(\chi)} ,$$

and so $\widehat{\chi}$ is of law $\widehat{\pi}$, which finishes the proof. □

4. Algorithms

4.1. **Simulation of a branching process.** One might be worried wether a process of law q_0 might be such that N_P is very big or equal to 0 with a high probability. Such events are undesirable in our simulations, as it will be seen later. For all $k \geq 1$, $g \in [0; \|G_k\|_\infty]$, we want $f_{k+1}(g, 0) = 1 - \alpha_{k+1} g$ (for some parameter $\alpha_{k+1} \leq 1/\|G_k\|_\infty$). Suppose we take the following simple law:

$$f_{k+1}(g, i) = \frac{\alpha_{k+1} g}{q_{k+1}} \text{ for } i \in \{1, 2, \ldots, q_{k+1}\} \tag{4.1}$$

and $f_{k+1}(g, i) = 0$ for $i \geq q_{k+1} + 1$, for some integer q_{k+1} to be chosen. Suppose we have built a branching process up to time k. We define a measure π_k on E_k by its action on test functions:

$$\pi_k(f) = \frac{\mathbb{E}(f(X_k) \prod_{i=1}^{k-1} G_i(X_i))}{\mathbb{E}(\prod_{i=1}^{k-1} G_i(X_i))} ,$$

where $(X_1, \dots, X_T)$ is a non-homogeneous Markov chain with initial law M_1 and transitions $M_2, M_3, \dots, M_T$. Suppose we make a simulation of a branching process up to time k. The particles at time k are denoted by $(X_k^{\mathbf{i}})_{\mathbf{i}\in S_k}$. One could show that for N_k big enough, the empirical measure $\frac{1}{N_k}\sum_{i=1}^{N_k}\delta_{X_k^i}$ is a fairly good approximation π_k. We want the expected number of children of the $k-$th generation to be approximatively N_k, that is:

$$\frac{1}{N_k}\sum_{i=1}^{N_k}\sum_{j=1}^{q_{k+1}} j\frac{\alpha_{k+1}G_k(X_k^i)}{q_{k+1}} = \frac{1}{N_k}\sum_{i=1}^{N_k}\frac{(q_{k+1}+1)}{2}\alpha_{k+1}G_k(X_k^i) \approx 1\,.$$

This should be the case if $\frac{(q_{k+1}+1)}{2}\alpha_{k+1}\pi_k(G_k) = 1$. Suppose now we have approximated π_k by an empirical measure $\frac{1}{N}\sum_{i=1}^{N} Y_k^i$ with some particles $(Y_k^i)_{1\le i\le N}$ coming from a SMC scheme (with N particles). Then we achieve our goal by taking q_k to be the closest integer to

$$\frac{2N}{\alpha_{k+1}\sum_{i=1}^{N} G_k(\overline{X}_k^i)} - 1\,.$$

We have here described a way of choosing the integers $(q_k)_{k\in\{2,\dots,T\}}$ before making a simulation of the branching process. The arguments given a purely heuristic. Other ways of calibrating $(q_k)_{k\in\{2,\dots,T\}}$ are possible, the only important thing is that these integers should be fixed before running the perfect simulation algorithm described below. No proof is needed as we only need that in practice, the number of particles in the branching process remains stable in time.

4.2. **Representation of the Markov transition.** Suppose we have a trajectory $(x_1, \dots, x_T) \in E_1 \times \cdots \times E_T$. We want to sample a random variable of law $Q((x_1, \dots, x_T), .)$. In practice, we do not have to make a simulation of the random permutations appearing in the Markov transition Q. We can simply run a simulation of the positions of the particles and forget about their indexes. The permutations are only here to make the proof of Lemma 3.1 easier. Having said this, we can change the way we index the particles.

We take functions $(m_k)_{1\le k\le T}$, $(\varphi_k)_{k\le 2\le T}$, $(\widehat{\varphi}_k)_{2\le k\le T}$ such that $m_1 : [0,1] \to E_1$ for all $k \in \{2, \dots, T\}$, $n \in \mathbb{N}^*$, $m_k : E_{k-1} \times [0,1] \to E_k$, $\varphi_k : \mathbb{R}_+ \times [0,1] \to \mathbb{N}$, $\widehat{\varphi}_k : \mathbb{R}_+ \times [0,1] \to \mathbb{N}$ and for a random variable U of uniform law on $[0,1]$,

$$m_1(U) \text{ is of law } M_1\,,$$

and for any $k \in [T-1]$, $x \in E_k$, $g \in \mathbb{R}_+$, $j \in \mathbb{N}$,

$$\begin{aligned} m_{k+1}(x,U) &\text{ is of law } M_{k+1}(x,.)\,,\\ \mathbb{P}(\varphi_{k+1}(G_k(x),U) = j) &= f_{k+1}(G_k(x), j)\,,\\ \mathbb{P}(\widehat{\varphi}_{k+1}(G_k(x),U) = j) &= \widehat{f}_{k+1}(G_k(x), i)\,. \end{aligned} \tag{4.2}$$

For $k \in [T-1]$, $x \in E_k$, $u \in [0,1] \mapsto \varphi_{k+1}(G_k(x), u)$ is the pseudo-inverse of the cumulative distribution function of the random variable of law $f_{k+1}(G_{k+1}(x), .)$, and $u \in [0,1] \mapsto \widehat{\varphi}_{k+1}(G_k(x), u)$ is the pseudo-inverse of the cumulative distribution function of the random variable of law $\widehat{f}_{k+1}(G_{k+1}(x), .)$.

Suppose now we are given a random variable $\Theta = (U_{\mathbf{i}}, U'_{\mathbf{i}}, V_{\mathbf{i}}, V'_{\mathbf{i}}, W_1, W_2)_{\mathbf{i}\in\mathbb{N}^n, n\ge 0}$ made of a family of i.i.d. random variables of uniform law on $[0,1]$. We denote by $\mathcal{O}$ the space in which Θ takes its value ($\mathcal{O}$ is in bijection with $[0,1]^{\mathbb{N}}$). Using these variables and the functions above, we can build a random variable of law $Q((x_1, \dots, x_T), .)$. We start with a recursive construction of the conditionnal forest.

- We set $\mathbf{B} = (1, 1, \dots, 1) \in \mathbb{N}^T$. We set $X_1^{\mathbf{B}(1)} = x_1$. We set $X_1^i = m_1(U_i)$ for $i \in \{2, \dots, n_1\}$. We set $S_1 = [n_1]$.
- Suppose we have $S_k \subset (\mathbb{N}^*)^k$ of cardinality N_k , containing $\mathbf{B}(k)$, and particles $(X_k^{\mathbf{i}})_{\mathbf{i}\in S_k}$. For all $\mathbf{i} \in S_k \backslash \mathbf{B}(k)$, we set $N_{k+1}^{\mathbf{i}} = \varphi_{k+1}(G_k(X_k^{\mathbf{i}}), V_{\mathbf{i}})$, and for $j \in [N_k^{\mathbf{i}}]$, we set $X_{k+1}^{(\mathbf{i},j)} = m_{k+1}(X_k^{\mathbf{i}}, U_{(\mathbf{i},j)})$. We set $N_{k+1}^{\mathbf{B}(k)} = \widehat{\varphi}_{k+1}(G_k(X_k^{\mathbf{B}(k)}), V_{\mathbf{i}})$, $X_{k+1}^{\mathbf{B}(k+1)} = x_{k+1}$, and for $j \in$

$\{2,\dots,N_{k+1}^{\mathbf{B}(k)}\}$, we set $X_{k+1}^{(\mathbf{B}(k),j)} = m_{k+1}(X_k^{\mathbf{B}(k)}, U_{(\mathbf{B}(k),j)})$. We set $S_{k+1} = \cup_{\mathbf{i}\in S_k} \cup_{j\in[N_{k+1}^{\mathbf{i}}]} (\mathbf{i},j)$. We set $N_{k+1} = \#S_{k+1}$.

Hypothesis 1. At each time-step $k \in [T-1]$, there exists a dominating potential $\widetilde{G}_k : \mathbb{N}^k \times E_k \times \mathcal{O} \to \mathbb{R}$ such that, for all $x \in E_k$, $\mathbf{i} \in \mathbb{N}^k$, $\Theta \in \mathcal{O}$, $G_k(x) \le \widetilde{G}_k(\mathbf{i}, x, \Theta) \le \|G_k\|_\infty$.

Using the same random variables as the ones we used above, we can build a bigger conditionnel forest, again recursively.

- We take the same $\mathbf{B}$ as above. For $i \in [n_1]$, we set $\widetilde{X}_1^i = X_1^i$. We set $\widetilde{S}_1 = S_1$.
- Suppose we have $\widetilde{S}_k \subset (\mathbb{N}^*)^k$ of cardinality $\widetilde{N}_k$, containing $\mathbf{B}(k)$, and particles $(\widetilde{X}_k^{\mathbf{i}})_{\mathbf{i}\in S_k}$. For all $\mathbf{i} \in \widetilde{S}_k \backslash \mathbf{B}(k)$, we set

$$\widetilde{N}_{k+1}^{\mathbf{i}} = \begin{cases} \varphi_{k+1}(\widetilde{G}_k(\mathbf{i}, \widetilde{X}_k^{\mathbf{i}}), V_{\mathbf{i}}, \Theta) & \text{if } \mathbf{i} \text{ is a descendant of } \mathbf{B}(1), \mathbf{B}(2), \dots \text{or } \mathbf{B}(k), \\ \varphi_{k+1}(G_k(\widetilde{X}_k^{\mathbf{i}}), V_{\mathbf{i}}) & \text{otherwise,} \end{cases} \tag{4.3}$$

and for $j \in [\widetilde{N}_k^{\mathbf{i}}]$, we set $\widetilde{X}_{k+1}^{(\mathbf{i},j)} = m_{k+1}(\widetilde{X}_k^{\mathbf{i}}, U_{(\mathbf{i},j)})$. We set $\widetilde{N}_{k+1}^{\mathbf{B}(k)} = \widehat{\varphi}_{k+1}(\widetilde{G}_k(\mathbf{B}(k), \widetilde{X}_k^{\mathbf{B}(k)}, \Theta), V_{\mathbf{i}})$, $\widetilde{X}_{k+1}^{\mathbf{B}(k+1)} = x_{k+1}$, and for $j \in \{2,\dots,\widetilde{N}_{k+1}^{\mathbf{B}(k)}\}$, we set $\widetilde{X}_{k+1}^{(\mathbf{B}(k),j)} = m_{k+1}(\widetilde{X}_k^{\mathbf{B}(k)}, U_{(\mathbf{B}(k),j)})$. We set $\widetilde{S}_{k+1} = \cup_{\mathbf{i}\in\widetilde{S}_k} \cup_{j\in[N_{k+1}^{\mathbf{i}}]} (\mathbf{i},j)$. We set $\widetilde{N}_{k+1} = \#\widetilde{S}_{k+1}$.

One could show recursively on k that for all $k \in [T]$, $S_k \subset \widetilde{S}_k$, for all $\mathbf{i} \in S_k$, $X_k^{\mathbf{i}} = \widetilde{X}_k^{\mathbf{i}}$, $N_{k+1}^{\mathbf{i}} \le \widetilde{N}_{k+1}^{\mathbf{i}}$ (almost surely in ω). We then build a proposal forest in a similar way, recursively on k.

- We set $\overline{S}_1 = [n_1]$. For $\mathbf{i} \in S_1$, We set $\overline{X}_1^{\mathbf{i}} = m_1(U_{\mathbf{i}}')$.
- Suppose we have $\overline{S}_k \subset (\mathbb{N}^*)^k$ of cardinality $\overline{N}_k$ and particles $(\overline{X}_k^{\mathbf{i}})_{\mathbf{i}\in S_k}$. For all $\mathbf{i} \in \overline{S}_k$, we set $\overline{N}_{k+1}^{\mathbf{i}} = \varphi_{k+1}(G_k(\overline{X}_k^{\mathbf{i}}), V_{\mathbf{i}}')$, and for $j \in [\overline{N}_k^{\mathbf{i}}]$, we set $\overline{X}_{k+1}^{(\mathbf{i},j)} = m_{k+1}(\overline{X}_k^{\mathbf{i}}, U_{(\mathbf{i},j)}')$. We set $\overline{S}_{k+1} = \cup_{\mathbf{i}\in S_k} \cup_{j\in[\overline{N}_{k+1}^{\mathbf{i}}]} (\mathbf{i},j)$. We set $\overline{N}_{k+1} = \#\overline{S}_{k+1}$.

Then we order $\overline{S}_T$ alphanumerically: $\overline{S}_T = \{\mathbf{i}_1,\dots,\mathbf{i}_{\overline{N}_T}\}$. We set

$$\mathbf{B}_T^\star = \mathbf{i}_r \text{ if } W_1 \in \left[\frac{r-1}{\overline{N}_T}, \frac{r}{\overline{N}_T}\right) .$$

The accep/reject step then goes in the following way:

$$\text{if } W_2 \le \min\left(1, \frac{\overline{N}_T}{\widetilde{N}_T}\right), \text{ move to } (\overline{X}_1^{\mathbf{B}^\star(1)}, \dots, \overline{X}_T^{\mathbf{B}^\star(T)}),$$

otherwise, stay in $(x_1,\dots,x_T)$. We will sometimes insist on the dependance of the variables on Θ, $x_1,\dots,x_T$ by writing $\widetilde{N}_T = \widetilde{N}_T(\Theta,(x_1,\dots,x_T))$, $N_T = N_T(\Theta)$, $W_2 = W_2(\Theta)$, ...

Remark 4.1. For each of the branching processes in this subsection, for all $k,q \in [T]$, $\mathbf{i} \in (\mathbb{N}^*)^k$, $\mathbf{j}$ in $(\mathbb{N}^*)^q$, if $\mathbf{i}$ is not an ancestor of $\mathbf{j}$ and $\mathbf{j}$ is not an ancestor of $\mathbf{i}$, then, conditionnaly on $(X_k^{\mathbf{i}}, X_q^{\mathbf{j}})$ (resp. $(\widetilde{X}_k^{\mathbf{i}}, \widetilde{X}_q^{\mathbf{j}})$, $(\overline{X}_k^{\mathbf{i}}, \overline{X}_q^{\mathbf{j}})$), the descendants of $X_j^{\mathbf{i}}$ are independant of the descendants of $X_q^{\mathbf{j}}$. In the same way, when we sample a conditionnal forest (resp. a bigger conditionnal forest) conditionnaly to $(x_1,\dots,x_T)$, for all $k \in [T-1]$, the descendants of $X_k^{\mathbf{B}(k)}$ (resp. $\widetilde{X}_k^{\mathbf{B}(k)}$) with indexes $\mathbf{i} \notin \{\mathbf{B}(k+1),\dots,\mathbf{B}(T)\}$ depend only on Θ and x_k.

4.3. **Backward coupling.** Suppose we are given i.i.d. random variables $(\Theta_0,\Theta_1,\Theta_2,\dots)$ having the same law as Θ (all of them are defined on a probability space $(\Omega,\mathcal{F},\mathbb{P})$. Any of these random variables is sufficient to perform a simulation of the Markov transition Q. The following result is a consequence of Theorem 3.1 of [FT98] (the original result can be found in [PW96]).

Theorem 4.2. *If τ is a stopping time with respect to the filtration $(\sigma(\Theta_0,\dots,\Theta_n))_{n\ge0}$ such that for all $(x_1,\dots,x_T)$, $(x_1',\dots,x_T')$ in $E_1 \times \cdots \times E_T$,*

$$F_{\Theta_0} \circ F_{\Theta_1} \circ \cdots \circ F_{\Theta_\tau}(x_1,\dots,x_T) = F_{\Theta_0} \circ F_{\Theta_1} \circ \cdots \circ F_{\Theta_\tau}(x_1',\dots,x_T') ,$$

Algorithm 1 Perfect simulation

for n in $\mathbb{N}$ repeat until $n = \tau$
  draw Θ_n, and store it
  test wether $n = \tau$ or not
pick any trajectory $(x_1, \dots, x_T) \in E_1 \times \cdots \times E_T$
set $\xi(-1) = (x_1, \dots, x_T)$
for $n = 0$ to τ repeat
  $\xi(n) = F_{\Theta_{\tau-n}}(\xi(n-1))$
return $\xi(\tau)$

then, for any $(x_1, \dots, x_T)$ *in* $E_1 \times \cdots \times E_T$,

$$F_{\Theta_0} \circ F_{\Theta_1} \circ \cdots \circ F_{\Theta_\tau}(x_1, \dots, x_T) \text{ is of law } \pi\,.$$

We suppose we have dominating potentials $\widetilde{G}_k$, $k \in [T-1]$ as in Subsection 4.2 above. We write $\Theta_n = (U_{n,\mathbf{i}}, U'_{n,\mathbf{i}}, V'_{n,\mathbf{i}}, V_{n,\mathbf{i}}, W_{n,1}, W_{n,2})$. In our simulations, we will use a stopping time of the following kind

$$\begin{aligned}(4.4)\quad \tau &= \min\left\{ n \,:\, W_{n,2} \le \min\left(1, \frac{\overline{N}_T(\Theta_n)}{\widetilde{N}_T(\Theta_n, (x_1, \dots, x_T))}\right), \forall (x_1, \dots, x_T) \in E_1 \times \cdots \times E_T \right\} \\ &= \min\left\{ n \,:\, W_{2,n} \le \min\left(1, \frac{\overline{N}_T(\Theta_n)}{\sup_{(x_1,\dots,x_T)\in E_1\times\cdots\times E_T} \widetilde{N}_T(\Theta_n, (x_1, \dots, x_T))}\right)\right\}\end{aligned}$$

This stopping time satisfies the assumptions of the above Proposition. Algorithm 1 is thus a perfect simulation of the law π. At this point, this algorithm is merely theoretical. The following two remarks will make it implementable, at least in some cases.

- We need to be able to compute $\max\{\widetilde{N}_T(\Theta(\omega), (x_1, \dots, x_T)), (x_1, \dots, x_T) \in E_1 \times \cdots \times E_T\}$ for a fixed ω. The easiest case is where E_1, E_2, …, E_T are finite. We will see below how to reduce the problem to this case in cases where E_1, E_2 … are not finite. If E_1 is finite, we can look for $x_1 \in E_1$ maximizing the descendants of $\widetilde{X}_1^1$ at time T (using $\Theta(\omega)$ to make the simulation), and so on. As we said in Remark 4.1, once $\widetilde{X}_1^1, \dots, \widetilde{X}_T^{(1,\dots,1)}$ are fixed, their descendants are independant, this is what makes the use of branching processes interesting.
- In Algorithm 1, we first sample $\Theta_0(\omega), \Theta_1(\omega), \dots$ until τ. And then we need the same realisations of the variables $(\Theta_0(\omega), \Theta_1(\omega), \dots)$ to compute $\xi(0)$, …, $\xi(\tau)$. The obejct $\Theta_0(\omega)$ is an infinite collection of numbers so it is impossible to store. We set $\mathcal{E}(\omega)$ to be the subset of indexes $\mathbf{i} \in \cup_{n\in[T]}(\mathbb{N}^*)^n$ such that $U_{n,\mathbf{i}}(\omega)$ or $V_{n,\mathbf{i}}(\omega)$ is used when computing $\sup_{(x_1,\dots,x_T)\in E_1\times\cdots\times E_T} \widetilde{N}_T(\Theta_n(\omega), (x_1, \dots, x_T))$. We notice that, for all n, we do not need to store the whole $\Theta_n(\omega)$; having stored the number of descendants of $\widetilde{X}_1^2(\omega), \dots, \widetilde{X}_1^{n_1}(\omega)$ (these are the starting points in the building of the "bigger conditionnal forest above"), $\overline{N}_T(\Theta_n(\omega))$, $(\overline{X}_1^{\mathbf{B}^\star(1)}, \dots, \overline{X}_T^{\mathbf{B}^\star(T)})(\omega)$ (this is the colored trajectory in the proposal above) and

$$(4.5)\qquad \{(U_{\mathbf{i}}(\omega), V_{\mathbf{i}}(\omega), W_2(\omega))_{\mathbf{i}\in(\mathbb{N}^*)^n, n\in[T]} : \mathbf{i}(1) = 1, \mathbf{i} \in \mathcal{E}(\omega)\}$$

  is enough to compute $F_{\Theta_n(\omega)}(\xi)$ for any ξ in $E_1 \times \cdots \times E_T$. The collection of numbers in (4.5) contains the number which might be needed when we compute the descendance of $X_1^1(\omega)$, $X_2^1(\omega)$, …, $X_{T-1}^1(\omega)$ in what is called above the "bigger conditionnal forest", and we do not need any other numbers. Another point is that we can code the simulation in such a way that we sample these variables in the alphanumerical order at each time step.

So, instead of storing these variables, we can store random seeds. For example, instead of storing $U^{(1,1,1)}(\omega)$, $U^{(1,1,2)}(\omega)$, $U^{(1,1,3)}(\omega)$, ..., we can store a single random seed[1].

We are now able to explain the purpose of Subsection 4.1. It is clear that when simulating a branching process, wether it is a conditionnal forest or a proposal forest, we do not want the number of particles to grow up. Such a growth would be exponential in T, which would be very bad for the complexity of our algorithm. On the other hand, if our branching processes become extinct before time T, it will be often the base that $\overline{N}_T(\Theta_n) = 0$, leading to $\tau \neq n$, and thus the first loop of Algorithm 1 could go on for a very long time. Again, this would be very bad for the complexity of our algorithm.

5. Examples

5.1. **Self-avoiding random walks in $\mathbb{Z}^2$.**

5.1.1. *Description of the model.* We take $E_1 = \mathbb{Z}^2$, $E_2 = \mathbb{Z}^2 \times \mathbb{Z}^2$, ..., $E_T = (\mathbb{Z}^2)^T$ and for all $n \in [T]$, $(z_1, \dots, z_n) \in (\mathbb{Z}^2)^n$,

$$G_n(z_1, \dots, z_n) = \begin{cases} 1 & \text{if } z_i \neq z_j \text{ for all } i, j \in [n],\ i \neq j, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

We take $M_1 = \delta_{(0,0)}$ (the Dirac mass at the origin of $\mathbb{Z}^2$). For all $n \in [T-1]$, $(z_1, \dots, z_n) \in (\mathbb{Z}^2)^n$, we take

$$M_{n+1}(z_1, \dots, z_n), (z_1, \dots, z_n, z_n + (0,1)) = M_{n+1}(z_1, \dots, z_n), (z_1, \dots, z_n, z_n + (0,-1)) \tag{5.2}$$
$$= M_{n+1}(z_1, \dots, z_n), (z_1, \dots, z_n, z_n + (1,0)) = M_{n+1}(z_1, \dots, z_n), (z_1, \dots, z_n, z_n + (-1,0)) = \frac{1}{4}\,.$$

Then the marginal of π on E_{T-1} is the uniform law on the set of paths $(z_1, z_2, \dots, z_{T-1}) \in \mathbb{Z}^2$ such that $z_1 = (0,0)$, $\|z_i - z_{i+1}\| = 1$ for all i ($\|\dots\|$ being the Euclidean norm), for $i, j \in [T-1]$ with $i \neq j$, $z_i \neq z_j$ (the path does not intersect with itself, one also says that it is self-avoiding).

5.1.2. *Stopping time.* We set $\mathbf{B} = (1, 1, \dots, 1) \in \mathbb{N}^T$. For $k \in [T]$, $\mathbf{i} \in \mathbb{N}^k$, $q \in [k]$ such that $\mathbf{i}(q) = \mathbf{B}(q)$ and $\mathbf{i}(q+1) \neq \mathbf{B}(q+1)$, we set, for all $x = (z_1, \dots, z_k) \in E_k$, $\Theta \in \mathcal{O}$, $\widetilde{G}_k(\mathbf{i}, x) = G_{k-q}(z_q, z_{q+1}, \dots, z_k)$ in other words

$$\widetilde{G}_k(\mathbf{i}, x) = \begin{cases} 1 & \text{if } (z_q, \dots, z_k) \text{ is self-avoiding,} \\ 0 & \text{otherwise} \end{cases}$$

(as $\widetilde{G}_k$ does not depend on Θ in this example, we replace $\widetilde{G}_k(\mathbf{i}, x, \Theta)$ by $\widetilde{G}_k(\mathbf{i}, x)$). We do not need to define $\widetilde{G}$ in the remaining cases. As we said in Subsection 4.3, we sample variables Θ_0, Θ_1, ... and we look for the stopping time τ defined in (4.4). For fixed n, Θ_n, and $k \in [T-1]$, $x_k \in E_k$, if we sample a bigger conditionnal forest with $\widetilde{X}_k^{\mathbf{B}(k)} = x_k$, we introduce the following notation:

$$\widetilde{N}_T(\Theta_n, x_k) = \#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \mathbf{i} \neq \mathbf{B}(T), \mathbf{B}(k) \prec \mathbf{i}\}\,.$$

We do not need the values $\widetilde{X}_q^{\mathbf{B}(q)}$ for $q \neq k$ to compute the above quantity. Due to the fact that the descendants of $\mathbf{B}(k)$ are generated using the potentials $\widetilde{G}_\cdot$ (see formula (4.3)), the set $\#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \mathbf{i} \neq \mathbf{B}(T), \mathbf{B}(k) \prec \mathbf{i}\}$ depends only on Θ_n. So

$$\text{for all } x_k \in (\mathbb{Z}^2)^k,\ \widetilde{N}_T(\Theta_n, x_k) = \widetilde{N}_T(\Theta_n, ((0,1), \dots, (0,k)))\,.$$

We set $\widetilde{\widetilde{N}}_T(\Theta_n, k) = \widetilde{N}_T(\Theta_n, ((0,1), \dots, (0,k)))$. Now we have, for all $(x_1, \dots, x_T) \in E_1 \times \dots \times E_T$,

$$\widetilde{N}_T(\Theta_n, (x_1, \dots, x_T)) = \widetilde{\widetilde{N}}_T(\Theta_n) := \#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \nexists k : \mathbf{B}(k) \prec \mathbf{i}\} + 1 + \sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta_n, k)\,. \tag{5.3}$$

[1] We recall here that when the user asks for random variables $U^{(1,1,1)}(\omega)$, $U^{(1,1,2)}(\omega)$, $U^{(1,1,3)}(\omega)$, ..., a computer will return numbers picked in a non-random list. So instead of storing theses random variables, we can store only the starting point in the list (the so-called "random seed")

FIGURE 5.1. Self-avoiding random walk in $\mathbb{Z}^2$.

T	100	200	300	350	400
$C(T)$	14.90	28.45	19.89	23.28	30.48

TABLE 1. $C(T)$ for the self-avoiding random walk

FIGURE 5.2. Log-log graph of complexity versus time in the case of the self-avoiding random walk.

This equation means that the supremum in (4.4) is easy to find. And so, by Remark 4.1

$$\tau = \min\left\{ n \geq 0 : W_{2,n} \leq \min\left(1, \frac{\overline{N}_T(\Theta_n)}{\widetilde{\widetilde{N}}_T(\Theta_n)}\right)\right\}.$$

5.1.3. *Complexity of the algorithm.* We take here the following law for the simulation of the number of children

$$\text{for all } k,\ f_{k+1}(0,0) = 1\,,\ f_{k+1}(1,1) = p_{k+1}\,,\ f_{k+1}(1,2) = p_{k+1}\,,$$

for some sequences $(p_k)_{2\leq k\leq T}$ taking values in $(0,1)$. We now look at a branching process in E_1, E_2, ...,E_T based on the potential defined in (5.1), the transitions defined in (5.2) and the above reproduction law. A sensible way of choosing the constants (p_k)'s is to choose them such that a branching process starting with n_1 particles will have a random number N_k of descendants of the same order of magnitude as n_1 (this requires some pre-processing). These numbers N_k are random but the law of large numbers makes them not too fluctuant. It turns that a good tuning is to have the (p_k)'s almost constant. By doing so, we are able to draw a trajectory with law π by a matter of minutes if $T \leq 1000$ and by a matter of one hour if $T \leq 5000$ (see Figure 5.1). Here, we ran a program in `C`. We used parallelization to make it faster (with the `OpenMP` library). The program uses around five cores simulteanously. Laptops are multi core nowadays, so the limiting factor is not the number of cores but the management of the memory. Indeed, the genealogies we build in our algorithm can take a lot of space, if the code is not written properly. An basic calculation shows that n_1 should be chosen as $n_1 = C(T) \times T$, with $C(T)$ depending on T (see the Appendix for details). We estimate $C(T)$ by Monte-Carlo for $T \in \{100, 200, 300, 350, 400\}$ (see the Appendix for details, we use 1000 samples for each expectation and variance we have to estimate). We can then compare T and $C(T)$ (see Table 1). A simple least square regression in log-log scale gives a slope of 0.27. So it seems sensible to take n_1 proportionnal to T or $T^{3/2}$.

We then estimate the average number of particles at each generation (on a 1000 sample) when we run a simulation of a bigger conditionnal forest, assuming, we take $n_1 = T^{3/2}$. For a fixed T, the average complexity of drawing one sample of a bigger conditionnal forest is the sum on all generations of the average number of particles at each generation times the number of the generation (this is the cost of computing the potential). With this choice of dependency between n_1 and T, this complexity is the average complexity of our algorithm (see the end of the Appendix and (5.3)).

Using again a Monte-Carlo method with 1000 samples for each expectation we have to estimate, we are able to draw in Figure 5.2 the log of the expected complexity against $\log(T)$ ($T \in [400]$) with $n_1 = T^{3/2}$. We draw a linear regression on the same graph. The estimated slope is 3.85. So the complexity seems to be polynomial in T.

5.2. **Filter in $\mathbb{R}^3$.**

5.2.1. *Description of the model.* We are given the following signal/state $((X_n)_{n\geq 1}$) and observations $((Y_n)_{n\geq 1})$ in $\mathbb{R}^3$:

$$\left\{ \begin{array}{rcll} X_{n+1} & = & AX_n + V_{n+1}\,, & \forall n \geq 1\,, \\ Y_n & = & X_n + W_n\,, & \forall n \geq 1\,, \end{array} \right.$$

with X_1 following a law M_1 and $(V_n)_{n\geq 2}$ independant of $(W_n)_{n\geq 1}$, the V_n's are i.i.d. with a law of density f and the W_n's are i.i.d. with a law of density g (with respect to the Lebesgue measure). The coefficient A is a 3×3 real matrice. We suppose we have functions F and G such that, for all $U\in[0,1]$, $F(U)$ is a random variable in $\mathbb{R}^3$ of law of density f, $G(U)$ is a random variable in $\mathbb{R}^3$ of law of density g.

We are interested in $\mathcal{L}(X_T|Y_1,\ldots,Y_{T-1})$ for some $T\in\mathbb{N}^*$. From now on, we will suppose that the sequence Y_1, Y_2, ... is fixed. We set, for all $k\in\mathbb{N}^*$, $G_k(x)=g(Y_k-x)$. We denote by $M_2=M_3=\cdots=M$ the transition kernel of the Markov chain $(X_n)_{n\geq 1}$. We set $E_1=E_2=\cdots=\mathbb{R}^3$. Then $\mathcal{L}(X_T|Y_1,\ldots,Y_{T-1})$ coincides with π defined in (1.1). We make the following hypotheses.

Hypothesis 2. The matrix A is invertible. For all $x,y\in\mathbb{R}^3$, $\|Ax-Ay\|\leq\alpha\|x-y\|$ ($\|\ldots\|$ is the Euclidean norm) with $\alpha\in[0,1)$.

Hypothesis 3. We have $g(x)\underset{\|x\|\to+\infty}{\longrightarrow}0$.

5.2.2. *Computing the stopping time.* We take m_k introduced in (4.2) to be, for all $x\in\mathbb{R}^3$, $U\in[0,1]$,

$$m_k(x,U)=Ax+F(U)\,.$$

We fix $\delta>0$. For $x=(x_1,x_2,x_3)\in\mathbb{R}^3$, we set

$$L_\delta(x)=\left[\delta\left\lfloor\frac{x_1}{\delta}\right\rfloor,\delta\left\lfloor\frac{x_1}{\delta}\right\rfloor+\delta\right)\times\left[\delta\left\lfloor\frac{x_2}{\delta}\right\rfloor,\delta\left\lfloor\frac{x_2}{\delta}\right\rfloor+\delta\right)\times\left[\delta\left\lfloor\frac{x_3}{\delta}\right\rfloor,\delta\left\lfloor\frac{x_3}{\delta}\right\rfloor+\delta\right)\,.$$

We set $\mathbf{B}=(1,1,\ldots,1)\in\mathbb{N}^T$. We suppose we are given a random variable Θ as in Subsection 4.2. We consider $k\in[T]$, $x\in\mathbb{R}^3$, $\mathbf{i}\in\mathbb{N}^k$ such that there exists $q\in[k]$ satisfying $\mathbf{i}(q)=\mathbf{B}(q)$, $\mathbf{i}(q+1)\neq\mathbf{B}(q+1)$. There exists one and only one sequence $(x_q,x_{q+1},\ldots,x_k)$ such that

$$\begin{array}{rcl} x_{q+1} & = & m_{q+1}(x_q,U_{\mathbf{i}(q+1)})\,, \\ x_{q+2} & = & m_{q+2}(x_{q+1},U_{\mathbf{i}(q+2)})\,, \\ & \ldots & \\ x=x_k & = & m_k(x_{k-1},U_{\mathbf{i}})\,. \end{array}$$

For $y\in\mathbb{R}^3$, we introduce the notations defined recursively:

$$\begin{array}{rcl} m_{q,q+1}(y) & = & m_{q+1}(y,U_{\mathbf{i}(q+1)})\,, \\ m_{q,q+2}(y) & = & m_{q+2}(m_{q,q+1}(y),U_{\mathbf{i}(q+2)})\,, \\ & \ldots & \\ m_{q,k}(y) & = & m_k(m_{q,k-1}(y),U_{\mathbf{i}})\,. \end{array}$$

We set

$$\widetilde{G}_k(\mathbf{i},x,\Theta)=\sup_{y\in L_\delta(x_q)}G_k(m_{q,k}(y))\,. \tag{5.4}$$

This implies that $G_k(x)\leq\widetilde{G}_k(\mathbf{i},x,\Theta)$. The idea here is to bound the potential $G_k(x)$ by its sup on subset of $\mathbb{R}^3$ containing x. Due to Hypothesis 2, the diameter of $\{m_{q,k}(y):y\in L_\delta(x_q)\}$ in (5.4) is bounded by $(\delta\sqrt{3})^{k-q}$. Under the additional assumption that g is continuous, it will make that $G_k(x)$ is not too far from $\widetilde{G}_k(\mathbf{i},x,\Theta)$ in the above bound. And so, the number of descendants of $\widetilde{X}_k^{\mathbf{B}(k)}$ should not explode when $T-k$ becomes big. Of course, these are only heuristics and we will study the complexity of the algorithm based on these $\widetilde{G}_\cdot$ below.

As we said in Subsection 4.3 and in the previous example, we sample variables Θ_0, Θ_1, ... and we look for the stopping time τ defined in (4.4). For fixed n, Θ_n and $k\in[T-1]$, $x_k\in\mathbb{R}^3$, we sample a bigger conditionnal forest with $\widetilde{X}_k^{\mathbf{B}(k)}=x_k$, we introduce the following notation

$$\widetilde{N}_T(\Theta_n,x_k)=\#\{\mathbf{i}\in\widetilde{S}_T(\Theta_n),\mathbf{i}\neq\mathbf{B}(T),\mathbf{B}(k)\prec\mathbf{i}\}\,.$$

T	5	10	15	20
$C(T)$	165.6	466.4	634.1	766.5

TABLE 2. $C(T)$ in a filtering case

We do not need the values $\widetilde{X}_q^{\mathbf{B}(q)}$, $q \neq k$ to compute the above quantity. We define

$$\widetilde{\widetilde{N}}_T(\Theta_n, k) = \sup_{x_k \in \mathbb{R}^3} \widetilde{N}_T(\Theta_n, x_k)\,. \tag{5.5}$$

Because of Remark 4.1, for all n, $\widetilde{\widetilde{N}}_T(\Theta_n)$ defined by $\widetilde{\widetilde{N}}_T(\Theta_n) := \sup_{(x_1,\dots,x_T)\in(\mathbb{R}^3)^T} \widetilde{N}_T(\Theta_n, (x_1,\dots,x_T))$ satisfies

$$\widetilde{\widetilde{N}}_T(\Theta_n) = \#\{i \in \widetilde{S}_T(\Theta_n),\, \nexists k : \mathbf{B}(k) \prec \mathbf{i}\} + 1 + \sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta_n, k)\,.$$

For fixed n, Θ_n and $k \in [T-1]$, suppose $x_k, x_k' \in \mathbb{R}^3$ are such that $L_\delta(x_k) = L_\delta(x_k')$ then the descendants of $\widetilde{X}_k^{\mathbf{B}(k)}$ in the bigger conditionnal forest are the same wether $\widetilde{X}_k^{\mathbf{B}(k)} = x_k$ or $\widetilde{X}_k^{\mathbf{B}(k)} = x_k'$. Suppose $\Theta_n = (U_{n,\mathbf{i}}, U'_{n,\mathbf{i}}, V_{n,\mathbf{i}}, V'_{n,\mathbf{i}}, W_{n,1}, W_{n,2})_{\mathbf{i}\in\mathbb{N}^n, n\geq 1}$. The number of children of $\widetilde{X}_k^{\mathbf{B}(k)}$ is $\widetilde{N}_{k+1}^{\mathbf{B}(k)} = \widehat{\varphi}_{k+1}(\widetilde{G}_k(\widetilde{X}_k^{\mathbf{B}(k)}, V_{n,\mathbf{i}})$, which is equal to zero if $\|\widetilde{X}_k^{\mathbf{B}(k)} - Y_k\|$ is big enough under Hypothesis 3. So, the number of operations needed to compute $\widetilde{N}_T(\Theta_n, k)$ is finite for all k, Θ_n. So, once we are given Θ_0, Θ_1, ... the stopping time τ can be computed in finite time.

5.2.3. *Complexity of the algorithm: a case study.* We suppose $A = 0.5 \times \mathrm{Id}(3)$, the (V_n)'s follow the law $\mathcal{N}(0, (0.2)^2)$, the $(W_n)'s$ follow the law $\mathcal{N}(0, (0.5)^2)$, $\delta = 0.1$. We take here the laws of the number of children of an individual to be the same as in Subsection 4.1, (4.1) (for all k, $\alpha_{k+1} = 1/\|G_k\|_\infty$). We made a simulation of a sequence $(Y_k)_{1\leq k\leq T}$ and stored it. Here the codes were written in `python` and are thus relatively slow to execute. Nevertheless, it takes a few minutes to sample a trajectory of length 50. Using the calculations of the Appendix, we see that we should choose $n_1 = C(T) \times T$, with $C(T)$ depending on T. We estimate $C(T)$ for $T \in \{1, 2, \dots, 20\}$ using Monte-Carlo (for each T, we used 10000 samples for the estimation of μ_1, σ_1^2 defined in the Appendix, we used 100 samples for the estimation of each $\mathbb{E}(\widetilde{N}_T(\Theta, k))$ appearing in the definition of μ_2, μ_2 begin defined in the Appendix). We can then compare T and $C(T)$ see Table 2 for $T \in \{5, 10, 15, 20\}$). A simple least square regression in log-log scale gives a slope of 1.21. So it seems sensible to take n_1 proportionnal to T or $T^{3/2}$.

We now want to estimate the complexity of the whole algorithm. Due to the remark at the end of the Appendix, this complexity is of the same order of the complexity of sampling a branching process and finding $\widetilde{N}_T(\Theta, k)$ for each k. Let us fix k. When we compute $\widetilde{N}_T(\Theta, k)$, we need to compute $\widetilde{N}_T(\Theta, x_k)$ for a finite number of x_k in a ball around Y_k (see equation (5.5)). This number is, in expectation, proportionnal to δ^{-3}. Taking $n_1 = \lfloor T^{3/2} \rfloor$, the complexity of the algorithm for a fixed T is of order less than $T^{3/2}\mu_1(T) + \delta^{-3}\sum_{i=1}^T \mu_2(T)$. We use the above estimates of $\mu_1(T)$, $\mu_2(T)$ for $T \in \{1, 2, \dots, 20\}$. We compare $T^{3/2}\mu_1(T)$ and $\sum_{i=1}^T \mu_2(T)$ to T. A least square regression in log-log scale gives us a slope less than 2 in both cases. This means the complexity, as a function of T, grows like T^2. We know that the coefficient of proportionnality between the complexity and T^2 includes δ^{-3}, so the algorithm is dimension dependent (it will be δ^{-d} in dimension d). What is the optimal choice of δ with respect to the complexity is not clear.

6. Appendix

We show here how to choose n_1 as a function of T (it applies to the examples of Subsection 5.1 and 5.2). We set $\mu_1 = \mathbb{E}(\overline{N}_T(\Theta))$ and $\sigma_1^2 = \mathrm{V}(\overline{N}_T(\Theta))$ in the case $n_1 = 1$ and we set

$\mu_2 = \frac{\mathbb{E}(1+\sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta,k))}{T}$. We then choose n_1 such that

$$n_1 - 1 \geq \frac{16\sigma_1^2}{\mu_1^2}\,. \tag{6.1}$$

It implies that (remember that $\overline{N}_T(\Theta)$ is a sum of n_1 i.i.d. variables)

$$\begin{aligned} \mathbb{P}\left(\overline{N}_T(\Theta) \leq \frac{n_1\mu_1}{2}\right) &\leq \mathbb{P}\left(\overline{N}_T(\Theta) - n_1\mu_1 \leq -\frac{n_1\mu_1}{2}\right) \\ &\leq \frac{4\sigma_1^2}{n_1\mu_1^2} \leq \frac{1}{4}\,, \end{aligned} \tag{6.2}$$

and (remember that $\#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \nexists k : \mathbf{B}(k) \prec \mathbf{i}\}$ is a sum of $n_1 - 1$ i.i.d. variables

$$\mathbb{P}\left(\#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \nexists k : \mathbf{B}(k) \prec \mathbf{i}\} \geq 2\mu_1(n_1-1)\right) \leq \frac{\sigma_1^2}{(n_1-1)\mu_1^2} \leq \frac{1}{4}\,. \tag{6.3}$$

We have

$$\mathbb{P}\left(1 + \sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta,k) \geq 4T\mu_2\right) \leq \frac{1}{4}\,. \tag{6.4}$$

We choose n_1 such that it also satisfies

$$n_1\mu_1 \geq T\mu_2\,. \tag{6.5}$$

So, $\overline{N}_T(\Theta) \geq \frac{n_1\mu_1}{2}$ and $\#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \nexists k : \mathbf{B}(k) \prec \mathbf{i}\} \leq 2\mu_1(n_1-1)$ and $1 + \sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta,k) \leq 4T\mu_2$ implies

$$\begin{aligned} \frac{\overline{N}_T(\Theta)}{\widetilde{\widetilde{N}}_T(\Theta)} &\geq \frac{\left(\frac{n_1\mu_1}{2}\right)}{2\mu_1(n_1-1) + 4T\mu_2} \\ &\geq \frac{n_1\mu_1}{8T\mu_2} \geq \frac{1}{8}\,. \end{aligned}$$

We have

$$\begin{aligned} \mathbb{P}\left(\overline{N}_T(\Theta) \geq \frac{n_1\mu_1}{2}, \#\{\mathbf{i} \in \widetilde{S}_T(\Theta_n), \nexists k : \mathbf{B}(k) \prec \mathbf{i}\} \leq 2\mu_1(n_1-1), 1 + \sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta,k) \leq 4T\mu_2\right) \\ \geq 1 - \mathbb{P}\left(\overline{N}_T(\Theta) \geq \frac{n_1\mu_1}{2}\right) - \mathbb{P}\left(\nexists k : \mathbf{B}(k) \prec \mathbf{i}\} \leq 2\mu_1(n_1-1)\right) \\ - \mathbb{P}\left(1 + \sum_{k=1}^{T-1} \widetilde{\widetilde{N}}_T(\Theta,k) \leq 4T\mu_2\right), \end{aligned}$$

and using (using (6.2), (6.3), (6.4)), we see this last quantity is bigger than $\frac{1}{4}$. So

$$\mathbb{P}\left(\frac{\overline{N}_T(\Theta)}{\widetilde{\widetilde{N}}_T(\Theta)} \geq \frac{1}{8}\right) \geq \frac{1}{4}\,.$$

This means that the expected number of steps in the "repeat" loop of Algorithm 1 is bounded independently of T, provided n_1 satisfies (6.1) and (6.5).

References

[FT98] S. G. Foss and R. L. Tweedie, *Perfect simulation and backward coupling*, Comm. Statist. Stochastic Models **14** (1998), no. 1-2, 187–203, Special issue in honor of Marcel F. Neuts. MR 1617572 (99f:60123)

[PW96] James Gary Propp and David Bruce Wilson, *Exact sampling with coupled Markov chains and applications to statistical mechanics*, Proceedings of the Seventh International Conference on Random Structures and Algorithms (Atlanta, GA, 1995), vol. 9, 1996, pp. 223–252. MR 1611693 (99k:60176)

C. Andrieu, Department of Mathematics, University of Bristol, University Walk, Bristol BS8 1TW, United Kingdom.
E-mail address: c.andrieu@bris.ac.uk

N. Chopin, C.R.E.S.T., Timbre J350 3, Avenue Pierre Larousse, 92240 MALAKOFF, France.
E-mail address: Nicolas.Chopin@ensae.fr

A. Doucet, Department of Statistics, 1 South Parks Road Oxford, OX1 3TG, United Kingdom.
E-mail address: doucet@stats.ox.ac.uk

S. Rubenthaler, Laboratoire J. A. Dieudonné, Université Nice Sophia Antipolis, Parc Valrose, 06108 Nice cedex 2, France.
E-mail address: rubentha@unice.fr