Parameter Inference for Stopped Processes

Ajay Jasra[1] and Nikolas Kantas[2]

[1] *Department of Mathematics, Imperial College London, London, SW7 2AZ, UK.*

E-Mail: a.jasra@ic.ac.uk

[2] *Department of Electrical Engineering, Imperial College London, London, SW7 2AZ, UK.*

E-Mail: n.kantas@ic.ac.uk

Abstract

In this article we consider Bayesian parameter inference associated to partially-observed stochastic processes that start from a set B_0 and is stopped or killed at the first hitting time of a known set A. Such processes occur naturally within the context of population genetics [25, 15], statistical analysis of rare events [9, 17, 23], finance [7] and engineering [6, 18, 27]. The associated posterior distributions are highly complex and posterior parameter inference requires the use of advanced Markov chain Monte Carlo (MCMC) techniques. Our approach uses a recently introduced simulation methodology, particle Markov chain Monte Carlo (PMCMC) [1], where sequential Monte Carlo (SMC) [19] approximations are embedded within MCMC. However, when the parameter of interest is fixed, standard SMC algorithms are not always appropriate for many stopped processes. In [11, 16] the authors introduce SMC approximations of multi-level Feynman-Kac formulae, which can lead to more efficient algorithms. This is achieved by devising a sequence of nested sets from B_0 to A and then perform the resampling step only when the samples of the process reach intermediate level sets in the sequence. Naturally, the choice of the intermediate level sets is critical to the performance of such a scheme. In this paper, we demonstrate that multi-level SMC algorithms can easily be used as a proposal in PMCMC. In addition, we propose a flexible adaptive strategy that sets the level sets for different parameter proposals. Our methodology is illustrated on the coalescent model with migration.

Key-Words: Stopped Processes, Sequential Monte Carlo, Markov chain Monte Carlo

1 Introduction

In this article we consider Markov processes that are stopped when reaching the boundary of a given set A. These processes appear in a wide range of applications, such as population genetics [25, 15], finance [7], neuroscience [5], physics [18, 23] and engineering [6, 27]. The vast majority of the papers in the literature deal with a fully observed stopped processes and assume the parameters of the model are known. In this paper we address problems when this is not the case. In particular, Bayesian inference for the model parameters is considered, when the stopped process is observed indirectly via observations. We will propose a generic simulation method that can cope with many types of partial observations. To the best of our knowledge, there is no previous work in this direction. An exception, in the context of maximum likelihood, is [5], where inference for the model parameters in the fully observed case is investigated.

In the fully observed case, stopped processes have been studied predominantly in the area of rare event simulation. In order to estimate the probability of rare events related to stopped processes, one needs efficient methods. This is to sample realisations of the process given it starts in a set B_0 and terminates in a rare target set A before it returns to B_0 or gets trapped in some absorbing set. This is usually achieved using Importance Sampling (IS) or multi-level splitting; for an overview see [20, 31] and the references therein. Recently, sequential Monte Carlo methods based on both these techniques have been used in [6, 18, 23]. In [10] the authors also prove under mild conditions that SMC can achieve same performance as popular competing methods based on traditional splitting.

Sequential Monte Carlo methods can be described as a collection of techniques used to approximate a sequence of distributions whose densities are known point-wise up to a normalizing constant and are of increasing dimension. SMC methods combine importance sampling and resampling to sample from distributions. The idea is to introduce a sequence of proposal densities and to sequentially simulate a collection of $N > 1$ samples, termed particles, in parallel from these proposals. The success of SMC lies in incorporating a resampling operation to control the variance of the importance weights, whose value would otherwise increase exponentially as the target sequence progresses [19].

Applying SMC in the context of fully observed stopped processes requires using resampling while taking into account how close a sample is to the target set. That is, it is possible that particles close to A are likely to have very small weights, whereas particles closer to the starting set B_0 can have very high weights. As a result, the diversity of particles approximating longer paths before reaching A would be depleted by successive resampling steps. For example, the

coalescent [25] in population genetics, this has been noted as early as in the discussion of [32] by the authors of [11]. In [11] the authors used ideas from splitting and proposed to perform the resampling step only when each sample of the process reaches intermediate level sets, which define a sequence of nested sets from B_0 to A. This was formalised in [16, Section 12.2], where using multiple levels for resampling is interpreted as an interacting particle approximation of a multi-level Feynman-Kac formulae. Naturally, the choice of the intermediate level sets is critical to the performance of such a scheme. That is, the levels should be set in a "direction" towards set A and so that each level can be reached from the previous one with some reasonable probability [10, 20]. This is usually achieved heuristically using trial simulation runs. Also more systematic techniques exist: for cases where large deviations can be applied in [14] the authors use optimal control and in [8, 9] the level sets are computed adaptively on the fly using the simulated paths of the process.

The contribution of this paper is to address the issue of inferring the parameters used to model the law of the stopped process, when it it is unobserved. In the context of Bayesian inference one often needs to sample from the posterior density of these parameters, which can be very complex. Employing standard MCMC methods is not feasible, given the difficulty one faces to sample trajectories of the stopped process. On the contrary, the recently introduced PMCMC seems ideal for such latent processes. Essentially, the method constructs a Markov chain on an extended state-space in the spirit of [3], such that one may apply SMC updates for a latent process, i.e. use SMC approximations within MCMC. This brings up the possibility of using the multi-level SMC methodology as a proposal in MCMC. The main contributions made in this article are as follows:

- When the sequence of level sets is fixed *a priori*, we can use multi-level SMC within PMCMC.
- It is possible to adapt the level sets by defining an appropriate target density on an extended state-space; this adaptation can improve the mixing ability of the PMCMC algorithm.

This article is structured as follows: in Section 2 we formulate the problem and present the coalescent as a motivating example. In Section 3 multi-level SMC for stopped processes is presented. In Section 4 we detail a PMCMC algorithm which uses multi-level SMC approximations within MCMC. In addition, specific adaptive strategies for the levels are proposed, which are motivated by some theoretical results that link the convergence rate of the PMCMC algorithm to the properties of multi-level SMC approximations. In Section 5 some numerical experiments for the the

coalescent are given. The paper is concluded in Section 6. The proofs of our theoretical results can be found in the appendix.

1.1 Notations

The following notations will be used. A measurable space is written as $(E, \mathcal{E})$, with the class of probability measures on E written $\mathscr{P}(E)$. For $\mathbb{R}^n$, $n \in \mathbb{N}$ the Borel sets are $\mathscr{B}(\mathbb{R}^n)$. For a probability measure $\gamma \in \mathscr{P}(E)$ we will denote the density with respect to an appropriate σ-finite measure dx as $\overline{\gamma}(x)$. The total variation distance between two probability measures $\gamma_1, \gamma_2 \in \mathscr{P}(E)$ is written as $\|\gamma_1 - \gamma_2\| = \sup_{A \in \mathcal{E}} |\gamma_1(A) - \gamma_2(A)|$. For a vector $(x_i, \ldots, x_j)$, the compact notation $x_{i:j}$ is used; if $i > j$ $x_{i:j}$ is a null vector. For a vector $x_{1:j}$, $|x_{1:j}|_1$ is the $\mathbb{L}_1-$norm. The convention $\prod_{\emptyset} = 1$ is adopted. Also, $\min\{a, b\}$ is denoted as $a \wedge b$ and $\mathbb{I}_A(x)$ is the indicator of a set A. Let E be a countably infinite state-space, then

$$\mathcal{S}(E) = \{R = (r_{ij})_{i,j \in E} : r_{ij} \geq 0, \sum_{l \in E} r_{il} = 1 \cap \exists \nu_i \geq 0, \forall i \in E, \sum_{l \in E} \nu_l = 1, \nu R = \nu\}$$

denotes the class of stochastic matrices which possess a stationary distribution. In addition, we will denote as $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ the d-dimensional vector whose i^{th} element is 1 and is 0 everywhere else. Finally, for the discrete collection of integers we will use the notation $\mathbb{T}_d = \{1, \ldots, d\}$.

2 Problem Formulation

2.1 Posterior

Let θ be a parameter vector on $(\Theta, \mathscr{B}(\Theta))$, $\Theta \subseteq \mathbb{R}^{d_\theta}$ with an associated prior $\pi_\theta \in \mathscr{P}(\Theta)$. The stopped process $\{X_t\}_{t \geq 0}$ is a $(E, \mathcal{E})-$valued discrete-time Markov process defined on a probability space $(\Omega, \mathscr{F}, \mathbb{P}_\theta)$, where $\mathbb{P}_\theta$ is a probability measure defined for every $\theta \in \Theta$ such that for every $A \in \mathscr{F}$, $\mathbb{P}_\theta(A)$ is $\mathscr{B}(\Theta)-$measurable. For simplicity will we will assume throughout the paper that the process is homogeneous, but note that the methodology can be easily extended to the non-homogeneous case. In addition, one can extend to the scenario of a strong Markov-process, but we do not consider it here.

The process $\{X_t\}_{t \geq 0}$ begins its evolution in a non empty set B_0 with the initial distribution $\nu_\theta : B_0 \to \mathscr{P}(B_0)$ and the Markov transition kernel $p_\theta : E \times \Theta \to \mathscr{P}(E)$. The process is killed once it reaches a non-empty target set $A \subset B_0 \in \mathscr{F}$ such that $\mathbb{P}_\theta(X_0 \in B_0 \setminus A) = 1$. We define

the associated stopping time

$$T = \inf\{t \geq 0 : X_t \in A\},$$

where it is assumed that $\mathbb{P}_\theta(T < \infty) = 1$ and $T \in \mathcal{I}$, with $\mathcal{I}$ being a finite collection of positive integer values related to possible stopping times.

The evolution of process is observed through some data that consists of a random observation vector $y \in F$. Also, there is no restriction on A, such that it can depend on the observed data y, but to simplify exposition this will not be reflected explicitly in the notation. In the context of Bayesian inference we are interested in the posterior:

$$\pi(d\theta, dx_{0:t}, t|y) \propto \gamma_{\theta,y}(dx_{0:t}, t)\pi_\theta(d\theta), \tag{1}$$

with $t \in \mathcal{I}$ and

$$\gamma_{\theta,y}(dx_{0:t}, t) = \xi_{\theta,y}(x_{0:t})\mathbb{I}_{(A^c)^t \times A}(x_{0:t})\nu_\theta(dx_0)\prod_{j=1}^{t} p_\theta(dx_j|x_{j-1})\pi_\theta(d\theta),$$

where $\xi_{\theta,y} : \Theta \times F \times \left(\bigcup_{t\in\mathcal{I}}\{t\} \times E^{t+1}\right) \to \mathbb{R}_+$ is the complete-data likelihood, which in the context of Feynman-Kac models can be interpreted as a path 'potential function' [16]. Throughout, it is assumed that for any $\theta \in \Theta$, $y \in F$ we have that $\gamma_{\theta,y}$ admits a density $\overline{\gamma}_{\theta,y}(x_{0:t}, t)$ with respect to a $\sigma-$finite measure $dx_{0:t}$ on $\overline{E} = \left(\bigcup_{t\in\mathcal{I}}\{t\} \times E^{t+1}\right)$ and the posterior and prior distributions π, π_θ admit densities $\overline{\pi}$, $\overline{\pi}_\theta$ respectively both defined with respect to appropriate $\sigma-$finite dominating measures.

2.2 Motivating Example: The Coalescent

The framework presented so far is rather abstract. We will introduce the coalescent model as a motivating example, see Figure 1. This example concerns genetic data $y = y_{1:d}^m \in (\mathbb{Z}_+^d)^m$, that is, the counts of genes that have been observed. Denote the number of genes of type i at event j of the process as x_j^i, with $x_j = (x_j^1, \ldots, x_j^d)$. The objective is to find the genetic parameters $\theta = (\mu, R)$, where $\mu \in \mathbb{R}^+$ and $R \in \mathcal{S}(\mathbb{T}_d)$, so that the parameter space can be written as $\Theta = \mathbb{R}^+ \times \mathcal{S}(\mathbb{T}_d)$. For the state space we define:

$$\begin{aligned}
\overline{E} &= \bigcup_{t\in\mathcal{I}}\left(\{t\} \times E^{t+1}\right) \\
E &= \{x^{1:d} \in (\mathbb{Z}^+)^d \cap 2 \leq |x^{1:d}|_1 \leq m+1\} \\
\mathcal{I} &= \{m, m+1, \ldots\}
\end{aligned}$$

and for the likelihood we have

$$\overline{\gamma}_{\theta,y}(x_{0:t}, t) = \mathbb{I}_{\{z:|z|_1=m+1\}}(x_t)\frac{\prod_{i=1}^d y_i^m!}{m!}\mathbb{I}_{y_{1:d}^m}(x_{t-1})\left[\overline{\nu}_\theta(x_0)\prod_{j=1}^{t}\overline{p}_\theta(x_j|x_{j-1})\right]$$

Here, the process is a Markov chain and is stopped when the number of individuals in the population exceeds n. However, the density is only non-zero if at time $t-1$ the data y matches x_{t-1} exactly. The transition density is given by

$$\overline{p}_\theta(x_j|x_{j-1}) = \begin{cases} \frac{x^i_{j-1}}{|x_{j-1}|_1}\frac{\mu}{|x_{j-1}|_1-1+\mu}r_{il} & \text{if} \quad x_j = x_{j-1} - e_i + e_l \\ \frac{x^i_{j-1}}{|x_{j-1}|_1}\frac{|x_{j-1}|_1-1}{|x_{j-1}|_1-1+\mu} & \text{if} \quad x_j = x_{j-1} + e_i \\ 0 & \text{otherwise.} \end{cases}$$

In the Markov density above, the first transition type, is termed mutation where individuals change type and the second transition is a split event; see Figure 1. When we consider the process backward in time, this latter event is a coalescence. Finally, the chain is initialised by the density

$$\overline{p}_\theta(x_1) = \begin{cases} \nu_i & \text{if} \quad x_1 = 2e_i \\ 0 & \text{otherwise.} \end{cases}$$

To facilitate Monte Carlo inference, one must reverse the time parameter and simulate backward from the data. This is now detailed in the context of importance sampling, e.g. [22].

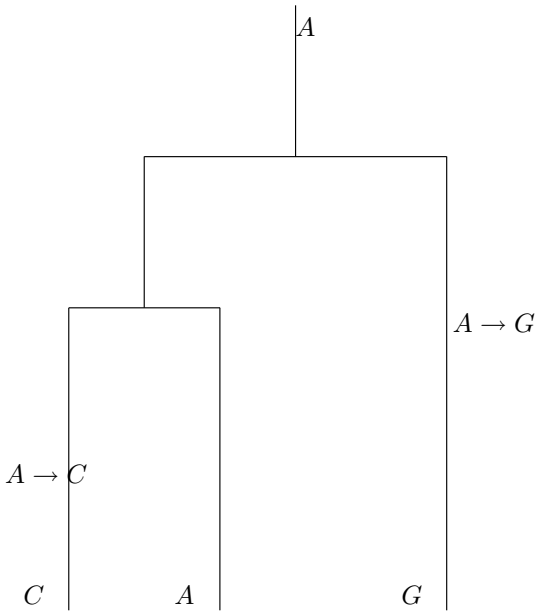


Figure 1: A Coalescent graph. The letters denote the types of the three observed chromosomes. Going up the figure (backward in time), the points where the graph join are coalescent events and the arrows denote a mutation of the type of a chromosome to another (forward in time).

2.2.1 Likelihood Computation

To compute the likelihood, for a given $\theta \in \Theta$, importance sampling is adopted. First we introduce a time reversed Markov kernel M_θ with density $\overline{M}_\theta(x_{j-1}|x_j)$. This is used as an importance sampling proposal that is initialised by the data and simulates the coalescent tree backward in time until two individuals remain of the same type and coalesce with each other; this procedure ensures that the data is hit when the tree is considered forward in time. In both directions in time $\{X_t\}$ is a stopped Markov process with A, B reversed.

In particular, we will consider the event sequence $j = t, t-1, \ldots, 1$ posed backwards in time. The proposal density for the full path starting from the bottom of the tree and stopping at its root can be written as

$$\overline{M}_\theta^{y^m_{1:d}}(x_{0:p}) \propto \mathbb{I}_{\{y^m_{1:d}\}}(x_0)\left\{\prod_{j=t}^{1}\overline{M}_\theta(x_{j-1}|x_j)\right\}\mathbb{I}_{\{x\in(\mathbb{Z}^+\cup\{0\})^d : x=e_i, i\in\mathbb{T}_d\}}(x_t).$$

Then the marginal likelihood is

$$\frac{m-1}{m-1+\mu}\frac{\prod_{i=1}^d (y_i^m)!}{m!}\sum_{t\in\mathcal{I}}\int_{E^t}\overline{p}_\theta(x_1)\left\{\prod_{j=t}^{2}\frac{\overline{p}_\theta(x_j|x_{j-1})}{\overline{M}_\theta(x_{j-1}|x_j)}\right\}\overline{M}_\theta^{y^m_{1:d}}(x_{1:t})dx_{1:t}.$$

With reference to (1) we have

$$\overline{\gamma}_{\theta,y}(x_{0:t},t) = \frac{m-1}{m-1+\mu}\frac{\prod_{i=1}^d (y_i^m)!}{m!}\overline{p}_\theta(x_1)\left\{\prod_{j=t}^{2}\frac{\overline{p}_\theta(x_j|x_{j-1})}{\overline{M}_\theta(x_{j-1}|x_j)}\right\}\overline{M}_\theta^{y^m_{1:d}}(x_{1:t})$$

Although we will not present the details here, in the context of importance sampling it is possible to derive an optimal proposal $\overline{M}_\theta$ with respect to the variance of the marginal likelihood estimator; see [32] for details.

The above scenario is used only to provide an interesting toy example. We remark, when there is only mutation, the stopped-process can be integrated out; see [21]. However, it is not typically possible to remove the stopped process in more complex scenarios. An interesting example where, to the best of our knowledge, this is the case is if we include migration events. This more complicated problem is presented in Section 5.2.

3 Multi-Level Sequential Monte Carlo Methods

In this section we shall briefly introduce generic SMC without extensive details. We refer the reader for a more detailed description to [16, 19]. SMC algorithms are designed to simulate from a sequence of probability distributions $\pi_1, \pi_2, \ldots, \pi_p$ defined on state space of increasing dimension, namely $(G_1, \mathscr{G}_1), (G_1 \times G_2, \mathscr{G}_1 \otimes \mathscr{G}_2), \ldots, (G_1 \times \cdots \times G_p, \mathscr{G}_1 \otimes \cdots \otimes \mathscr{G}_p)$. Each distribution in the

sequence is assumed to possess densities with respect to a common dominating measure:

$$\overline{\pi}_n(u_{1:n}) = \frac{\overline{\gamma}_n(u_{1:n})}{Z_n}$$

with each un-normalised density being $\overline{\gamma}_n : G_1 \times \cdots \times G_n \to \mathbb{R}_+$ and the normalizing constant being Z_n. We will assume throughout the article that there are natural choices for $\{\bar{\gamma}_n\}$ and that we can evaluate each $\overline{\gamma}_n$ pointwise. In addition, we do not require knowledge of Z_n.

3.1 Generic SMC Algorithm

To ease exposition, when presenting generic SMC, we shall drop the dependence upon parameter θ. In summary, SMC algorithms approximate $\{\overline{\pi}_n\}$ recursively by propagating a collection of properly weighted samples, called particles, using a combination of importance sampling and resampling steps. For the importance sampling part of the algorithm at each step n of the algorithm we will use general proposal kernels M_n with densities $\overline{M}_n$, which possess normalizing constants that do not depend on the simulated paths. A typical SMC algorithm is given below:

- 0. Initialisation: set $n = 1$; for $i \in \mathbb{T}_N$ sample $U_1^{(i)} \sim \overline{M}_1$ and compute

  $$w_1^{(i)} = \frac{\overline{\gamma}_1(u_1^{(i)})}{\overline{M}_1(u_1^{(i)})}$$

  with $W_1^{(i)} = w_1^{(i)}$.

- 1. Decide whether or not to resample, and if this is performed, set all weights $\{W_n^{(i)}\}_{1\leq i\leq N}$ to 1 and proceed to step 2.

- 2. Set $n = n + 1$, if $n = p + 1$ stop, else; for $i \in \mathbb{T}_N$ sample $U_n^{(i)}|u_{1:n-1}^{(i)} \sim \overline{M}_n(\cdot|u_{1:n-1}^{(i)})$, compute

  $$w_n^{(i)} = w_n(u_{1:n}^{(i)}) = \frac{\overline{\gamma}_n(u_{1:n}^{(i)})}{\overline{\gamma}_{n-1}(u_{1:n-1}^{(i)})\overline{M}_n(u_n^{(i)}|u_{1:n-1}^{(i)})}$$

  and set $W_n^{(i)} = w_n^{(i)} W_{n-1}^{(i)}$ and return to the start of step 1.

3.1.1 Some details on resampling

If one chooses to implement SMC without resampling steps, i.e. to perform sequential importance sampling, as time progresses, the variance of the weights $\{W_n^{(i)}\}_{1\leq i\leq N}$ typically increases. This has been commonly referred to as the *weight* degeneracy property. To counter this resampling is used: the particles $\{X_{1:n}^{(i)}\}_{1\leq i\leq N}$ are sampled with replacement, according to the normalised weights $\{\bar{W}_n^{(i)}\}_{1\leq i\leq N}$ given by

$$\bar{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^N W_n^{(j)}}$$

and then each $W_n^{(i)}$ is reset to 1.

If one resamples too often, the simulated past of the path of each particle will be very similar to each other. This has been documented as the *path* degeneracy problem. A common remedy was to resample only when appropriate criteria drops beneath or go above some threshold. In the former case, a common criterion is the effective sample size $\left(\sum_{j=1}^{N}\left(W_n^{(j)}\right)^2\right)^{-1}$ [28]. The path degeneracy has been a long standing bottleneck when static parameters θ are estimated online using SMC methods by augmenting them with the latent state. We refer the reader to [2, 24] for more details. These issues have motivated the development and use of offline methods like Particle MCMC detailed in Section 4.1. In addition, given path degeneracy is not critical when PMCMC methods are used, for the remainder of the article we will assume one resamples at each stage of the algorithm.

Suppose one resamples, multinomially, at every iteration, except when $n = p$. Denote the resampled index of the ancestor of particle i at time n by $a_n^i \in \mathbb{T}_N$; this is a random variable chosen with probability $\bar{W}_{n-1}^{(a_{n-1}^i)}$. Furthermore the joint density of the sampled particles and the resampled indices is

$$\psi(\bar{u}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \left(\prod_{i=1}^{N} \overline{M}_1(u_1^{(i)})\right) \prod_{n=2}^{p} \left(\prod_{i=1}^{N} \bar{W}_{n-1}^{(a_{n-1}^i)} \overline{M}_n(u_n^{(i)} | u_{n-1}^{(a_{n-1}^i)}, \ldots, u_1^{(a_1^i)})\right), \tag{2}$$

where the complete genealogy of ancestors is denoted as $\bar{\mathbf{a}}_n = (a_n^1, \ldots, a_n^N)$ and the randomly simulated values of the state as $\bar{u}_n = (u_n^{(1)}, \ldots, u_n^{(N)})$. Together they form the following SMC approximations for π_n

$$\pi_n^N(du_{1:n}) = \frac{1}{N}\sum_{j=1}^{N} \delta_{\bar{u}_{1:n}^{(a_n^i)}}(du_{1:n})$$

and an approximation of the normalizing constant of $\overline{\pi}_p$ as

$$\widehat{Z}_p = \prod_{n=1}^{p}\left\{\frac{1}{N}\sum_{j=1}^{N} W_n^{(j)}\right\}. \tag{3}$$

The complete ancestral genealogy at each time can always traced back by defining an ancestry sequence $b_{1:n}^i$ for every $i \in \mathbb{T}_N$ and $n \in \mathbb{T}_{p-1}$, whose elements are given by the backward recursion $b_n^i = a_n^{b_{n+1}^i}$ where $b_p^i = i$. In this context one can view SMC approximations as random probability measures induced by the imputed random genealogy $\bar{\mathbf{a}}_n$ and the state sequence $\bar{u}_n$. This interpretation of SMC approximations was introduced in [1] and will be later used together with $\psi(\bar{u}_{1:p}, \bar{\mathbf{a}}_{1:p-1})$ for establishing the complex extended target distribution of PMCMC.

3.2 Multi-Level SMC implementation

Applying SMC in its most generic form presented earlier might not be the best possible choice for any given problem. A multi-level implementation was proposed in [11] and the approach was illustrated for the coalescent model of Section 2.2. We consider a modified approach along the lines of [16, Section 12.2], which seems better suited for stopped processes and can yield estimators of much lower variance relative to vanilla SMC.

Introduce an arbitrary sequence of $\mathscr{F}-$nested sets

$$B_0 \supset B_1 \cdots \supset B_p = A, \quad p \geq 2$$

with the corresponding stopping times denoted as

$$T_l = \inf\{t \geq 0 : X_t \in B_l\}, \quad 1 \leq l \leq p,$$

Note that the (strong) Markov property of X_t implies $0 \leq T_1 \leq T_2 \leq \cdots \leq T_p = T$.

Similar to generic SMC we will assume there is a natural sequence of densities $\{\overline{\gamma}_{\theta,n}\}_{1\leq n\leq p}$, $\overline{\gamma}_{\theta,n} : \Theta \times F \times \left\{ \bigotimes_{j=1}^{n} \left(\bigcup_{i\in\mathcal{I}_j} \{i\} \times E^i \right) \right\} \to \mathbb{R}_+$, that obeys the restriction $\overline{\gamma}_{\theta,p} \equiv \overline{\gamma}_{\theta,y}$ so that that the last target density $\overline{\gamma}_{\theta,p}$ coincides with $\overline{\gamma}_{\theta,y}$. Note that we define a p length sequence of target densities, but this time each $\overline{\gamma}_{\theta,n}$ has a random length $t_n - t_{n-1}$. Multi-level SMC is a SMC algorithm which ultimately targets a sequence of distributions $\{\pi_{\theta,n}\}$ each defined on a space

$$\overline{E}_n = \left(\bigcup_{i\in\mathcal{I}_1} \{i\} \times E^i \right) \times \cdots \times \left(\bigcup_{i\in\mathcal{I}_n} \{i\} \times E^i \right). \tag{4}$$

where $n = 1, \ldots, p$, $p \geq 2$ and $\mathcal{I}_1, \ldots, \mathcal{I}_p$ are finite collections of positive integer values related to the stopping times $T_1, \ldots, T_p$ respectively. Note that this presentation differs significantly from [11], as the state-space of the target densities and explicit representation of the latter were not detailed there.

The implementation of multi-level SMC differs from the generic algorithm of Section 3.1 in that between successive resampling steps one proceeds by propagating in parallel trajectories of $X_{0:t}^{(j)}$ until each level B_n is reached and $X_t^{(j)} \in B_n$. The path $X_{0:t}^{(j)}$ is "frozen" until the remaining particles reach B_n and then a resampling step is performed. More formally denote for $n = 1$

$$\mathcal{X}_1 = \{x_{0:t_1}, t_1 : \, x_{0:t_1-1} \in B_0 \setminus B_1, \, x_{t_1} \in B_1\}$$

where t_1 is a realisation for the stopping time T_1 and similarly for $2 \leq n \leq p$ we have

$$\mathcal{X}_n = \{x_{t_{n-1}+1:t_n}, t_n : \, x_{t_{n-1}+1:t_n-1} \in B_{n-1} \setminus B_n, \, x_{t_n} \in B_n\}.$$

As in the generic SMC algorithm of Section 3.1, one can introduce a collection of Markov importance sampling kernels $\{M_{\theta,n}\}$, with $M_{\theta,n} : \Theta \times E \to \mathscr{P}(E)$ and denote their densities w.r.t. dx as $\{\overline{M}_{\theta,n}\}$. If this is not possible and one can simulate from the transition kernel of the Markov chain, then one may alternatively use the latter as an importance sampling kernel.

Multi-level SMC can be outlined along the lines of the generic algorithm of Section 3.1, if we replace $u_{1:n}$ with $\mathcal{X}_{1:n}$. For step 1 we always resample at $n = 1, 2, \ldots, p$, where the incremental weight of step 2 of the algorithm is used, which for $n \geq 2$ is given by

$$w_n(\mathcal{X}_1, \ldots, \mathcal{X}_n) = \frac{\overline{\gamma}_{\theta,n}(\mathcal{X}_1, \ldots, \mathcal{X}_n)}{\overline{\gamma}_{\theta,n-1}(\mathcal{X}_1, \ldots, \mathcal{X}_{n-1}) \prod_{l=t_{n-1}+1}^{t_n} \overline{M}_{\theta,n}(x_l|x_{l-1})}.$$

For step 0 of the generic SMC algorithm we use instead

$$w_1(\mathcal{X}_1) = \frac{\overline{\gamma}_{\theta,1}(\mathcal{X}_1)}{\prod_{l=0}^{t_1} \overline{M}_{\theta,1}(x_l|x_{l-1})}.$$

To simplify notation from herein we write

$$\mathcal{M}_{\theta,1}(\mathcal{X}_1) = \prod_{l=0}^{t_1} \overline{M}_{\theta,1}(x_l|x_{l-1})$$

and given p, for any $2 \leq n \leq p$ we have

$$\mathcal{M}_{\theta,n}(\mathcal{X}_n|\mathcal{X}_{n-1}) = \prod_{l=t_{n-1}+1}^{t_n} \overline{M}_{\theta,n}(x_l|x_{l-1}).$$

As mentioned earlier we note that at each n once $x_{t_n} \in B_n$, then a particular particle targetting $\mathcal{X}_{1:n}$ is frozen and resampling is performed when all particles reach B_n. Similar to (2), it is clear that the joint probability density of all the random variables used to implement a particle algorithm with multinomial resampling (excluding the resampling at step p) is given by:

$$\psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \left(\prod_{i=1}^{N} \mathcal{M}_{\theta,1}(\mathcal{X}_1^{(i)}) \right) \prod_{n=2}^{p} \left(\prod_{i=1}^{N} \bar{W}_{n-1}^{(a_{n-1}^i)} \mathcal{M}_{\theta,n}(\mathcal{X}_n^{(i)}|\mathcal{X}_{n-1}^{(a_{n-1}^i)}) \right). \tag{5}$$

In this scenario, recall that an approximation of the normalizing constant is, for fixed θ

$$\widehat{Z}_{\theta,p} = \prod_{n=1}^{p} \left\{ \frac{1}{N} \sum_{j=1}^{N} w_n^{(j)} \left(\mathcal{X}_{1:n}^{(j)} \right) \right\}. \tag{6}$$

3.2.1 Setting the levels

We will begin by showing how the levels can be set for the coalescent example of Section 2.2. Recall that in the spirit of Section 2.2.1 we adopt the convention that the "time" indexing is set to start from the bottom of the tree towards the root. We introduce a a collection of integers $m > l_1 > l_2 > \cdots > l_p = 1$ and define for $1 \leq n \leq p$

$$B_n = \{x \in (\mathbb{Z}^+ \cup \{0\})^d : |x|_1 = l_n\},$$

with $B_0 = \{x \in (\mathbb{Z}^+ \cup \{0\})^d : |x|_1 = m\}$; clearly, $B_{n-1} \supset B_n$, $1 \leq n \leq p$ and $B_p = A$. One can also write the sequence of target densities for the multi-level setting as:

$$\begin{aligned}
\overline{\gamma}_{\theta,0}(x_0) &\propto \frac{m-1}{m-1+\mu}\frac{\prod_{i=1}^d (y_i^m)!}{m!}\mathbb{I}_{\{y_{1:d}^m\}}(x_0) \\
\overline{\gamma}_{\theta,n}(x_{0:t_n}, t_n) &\propto \overline{\gamma}_{\theta,n-1}(x_{0:t_{n-1}}, t_{n-1}) \prod_{l=t_{n-1}+1}^{t_n} \overline{p}_\theta(x_{l-1}|x_l)\mathbb{I}_{\{x_{t_n}\in B_n\}}(x_{t_n}),\ n = 1, \ldots, p-1 \\
\overline{\gamma}_{\theta,p}(x_{0:t_p}, t_p) &\propto \overline{\gamma}_{\theta,p-1}(x_{0:t_{p-1}}, t_{p-1}) \prod_{l=t_{p-1}+1}^{t_p-1} \overline{p}_\theta(x_{l-1}|x_l)\overline{p}_\theta(x_{t_p-1})\mathbb{I}_{\{x_{t_p}\in B_n\}}(x_{t_p}).
\end{aligned}$$

To avoid confusion, note that the direction of $l = 0, \ldots, t_p$ here is opposite than $j = 0, \ldots, t$ in Section 2.2.1. In this sense one may interpret multi-level schemes as an explicit importance sampling construction similar to $\overline{M}_\theta^{y_{1:d}^m}(x_{0:p})$ which also take into account the direction of the process towards A.

The major design problem that remains in general is that given *any* candidates for $\{\overline{M}_{n,\theta}\}$, how to set the spacing, shape of $\{B_n\}$ and how many levels are needed so that good SMC algorithms can be constructed. That is, if the $\{B_n\}$ are far apart, then one can expect that weights will degenerate very quickly and if the $\{B_n\}$ are too close that the algorithm will resample too often and hence lead to poor estimates. For instance, in the context of the coalescent example of Section 2.2, if one uses the above construction for $\{B_n\}$ the importance weight at the n-th resampling time is

$$w_n(x_{0:t_n}) = \prod_{l=t_{n-1}+1}^{t_n} \frac{\overline{p}_\theta(x_{l-1}|x_l)}{\overline{M}_{\theta,n}(x_l|x_{l-1})}\mathbb{I}_{\{x_{t_n}\in B_n\}}(x_{t_n}),$$

Now, in general for any $\{l_n\}_{n=1}^p$ and p there is no concrete reason to expect that the resulting multi-level algorithm will perform well, relative to a vanilla SMC algorithm. Whilst [11] show empirically that in most cases this can be true, one would like to guarantee this by designing the levels sensibly. This design issue becomes also more apparent when θ is varied, as it is clear that for very different parameters, one is likely to need different sequences for $\{B_n\}$.

When interpreting the algorithm as a particle approximation of a multi-level Feynman-Kac formula, such difficulties have been dealt with by [8, 9] for rare-event simulation. There the authors use adaptive techniques to determine the next set B_n. For example, for the coalescent, one might set the next level to be the median number of individuals within the sample of the particles when the effective sample size drops below some value. This idea is rather intuitive and can drastically improve the empirical performance of SMC algorithms. The intention here is to be able to use such adaptive ideas for multi-level SMC within PMCMC, which will be presented in the next section. Clearly, the ability to use the probability density (5) as a proposal in MCMC

precludes the latter idea, which is also dealt with.

4 Multi-Level Particle Markov Chain Monte Carlo

4.1 PMCMC Methods

Particle Markov Chain Monte Carlo methods are effectively MCMC algorithms, which use all the random variables generated by SMC approximations to generate proposals. As in standard MCMC the idea is to run an ergodic Markov chain to obtain samples from the distribution of interest. The difference lies in that due to using SMC approximation the invariant distribution of the simulated chain is defined on an extended state space, with marginal the distribution we are interested to sample from in the first place.

We will begin by presenting the simplest generic algorithm found in [1], namely the particle independent Metropolis-algorithm (PIMH). In this case θ and p are fixed and PIMH is designed to sample from a pre-specified target distribution π_p as the ones presented in Section 3.1. This algorithm proceeds as follows:

- 0. Sample $\bar{u}_1, \ldots, \bar{u}_p, \bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{p-1}$ from (2). Sample $k \in \mathbb{T}_N$ from $\bar{W}_p^k$ and set this as a new state. Store $\widehat{Z}(0), k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)$ (see eq. (3)). Set $i = 1$
- 1. Propose a new $\bar{u}_1', \ldots, \bar{u}_p', \bar{\mathbf{a}}_1', \ldots, \bar{\mathbf{a}}_{p-1}'$ and k' as in step 0. Accept or reject this as the new state of the chain with probability

$$1 \wedge \frac{\widehat{Z}'}{\widehat{Z}(i-1)}.$$

  If we accept, store $\left(\widehat{Z}(i), k(i), \bar{\mathcal{X}}_{1:p}(i), \bar{\mathbf{a}}_{1:p-1}(i)\right) = \left(\widehat{Z}', k', \bar{\mathcal{X}}_{1:p}'(i), \bar{\mathbf{a}}_{1:p-1}'\right)$. Set $i = i + 1$.

In [1] it is shown that the invariant density of the Markov kernel above is exactly

$$\overline{\pi}_p^N(k, \bar{u}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \frac{\overline{\pi}_p(u_{1:p}^{(k)})}{N^p} \frac{\psi(\bar{u}_{1:p}, \bar{\mathbf{a}}_{1:p-1})}{\overline{M}_1(u_1^{(b_1^k)}) \prod_{n=2}^p \left\{\bar{W}_{n-1}^{(b_{n-1}^k)} \overline{M}_n(u_n^{(b_n^k)} | u_{n-1}^{(b_{n-1}^k)}, \ldots, u_1^{(b_1^k)})\right\}}$$

where ψ is as in (5) and as before we have $b_p^k = k$ and $b_n^k = a_n^{b_{n+1}^k}$ for every $k \in \mathbb{T}_N$ and $n \in \mathbb{T}_{p-1}$. The target density of interest, $\overline{\pi}_p$, is the marginal, conditional on k and $\bar{\mathbf{a}}_{1:p-1}$.

When θ is a random variable, [1] also introduce particle marginal Metropolis (PMMH) and particle Gibbs samplers; see [1] for details. In the remainder of this article we will focus on using multi-level SMC implementation within a PMMH algorithm.

- 0: Set $\theta(0) \in \Theta$, sample a multi-level SMC algorithm (i.e. $\bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)$) using (5) for fixed $\theta = \theta(0)$ sample $k(0)$ according to $\bar{W}_p$, and compute and store the estimate of the normalizing constant $\widehat{Z}_{\theta,p}(0)$ (see eq. (6)). Set $i = 1$, $\mathsf{x}(0) = (\theta(0), k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0))$.
- 1: Sample $\theta' \sim q(\cdot|\theta(i-1))$ and then given θ', simulate $(\bar{\mathcal{X}}'_{1:p}(0), \bar{\mathbf{a}}'_{1:p-1})$ using (5), computing $\widehat{Z}_{\theta',p}$, sampling k' according to $\bar{W}_p$ and accept or reject x' with probability

$$1 \wedge \frac{\widehat{Z}_{\theta',p}}{\widehat{Z}_{\theta,p}(i-1)} \times \frac{q(\theta(i-1)|\theta')}{q(\theta'|\theta(i-1))}$$

  if accepted set $\mathsf{x}(i) = \mathsf{x}'$, with $\widehat{Z}_{\theta(i),p}(i) = \widehat{Z}_{\theta',p}$, otherwise retain the current values. Set $i = i + 1$.

Figure 2: Multi-Level SMC within MCMC.

4.2 Multi-Level SMC within MCMC

Given the commentary in Section 3.2 and our interest in drawing inference on $\theta \in \Theta$, it seems that using multi-level SMC within PMCMC should be highly beneficial. Intrinsically, one expects that any SMC algorithm which can produce good estimates of the normalizing constants should improve the mixing of the MCMC sampler. As a result, the questions that underly our subsequent analysis are:

1. Is it possible to use multi-level SMC within MCMC?
2. Given that it is, how can we use this 'optimally' in some sense.

First we will deal with the first point for the case when p is fixed. So, in the context of our stopped Markov process, we propose a PMMH algorithm in Figure 2.

We now establish the invariant density and convergence of this algorithm, under the following assumption:

(**A1**) For any $\theta \in \Theta$ and $p \in \mathcal{I}$ we define the following sets for $n = 1, \ldots, p$: $S_n^\theta = \{\mathcal{X}_{1:n} \in \overline{E}_n : \pi_{\theta,n}(\mathcal{X}_{1:n}) > 0\}$ and $Q_n^\theta = \{\mathcal{X}_{1:n} \in \overline{E}_n : \pi_{\theta,n-1}(\mathcal{X}_{1:n-1})\mathcal{M}_{\theta,n}(\mathcal{X}_n|\mathcal{X}_{n-1}) > 0\}$. For any $\theta \in \Theta$ we have that $S_n^\theta \subseteq Q_n^\theta$. In addition the ideal Metropolis Hastings sampler with target density given by

$$\bar{\pi}(\theta) = \sum_{t \in \mathcal{I}} \int_{E^t} \bar{\pi}(\theta, x_{0:t}, t|y) dx_{0:t}$$

and proposal density $q(\theta'|\theta)$ is irreducible and aperiodic.

This assumption contains assumptions 5 and 6 of [1] modified to our problem with a simple change of notations.

Proposition 4.1. *Assume (A1); for any $N \geq 1$:*

1. *The invariant density of the kernel in described in Figure 2, is on the space $\Theta \times \mathbb{T}_N^{(p-1)N+1} \times \overline{E}$ (with $\overline{E}$ as in eq. (4)) and has the representation*

$$\overline{\pi}^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \frac{\overline{\pi}(\theta, \mathcal{X}_{1:p}^{(k)})}{N^p} \frac{\psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})}{\mathcal{M}_{\theta,1}(\mathcal{X}_1^{(b_1^k)}) \prod_{n=2}^p \left\{ \bar{W}_{n-1}^{(b_{n-1}^k)} \mathcal{M}_{\theta,n}(\mathcal{X}_n^{(b_n^k)} | \mathcal{X}_{n-1}^{(b_{n-1}^k)}) \right\}} \quad (7)$$

   where

$$\overline{\pi}(\theta, \mathcal{X}_{1:p}) \propto \overline{\pi}_\theta(\theta) \overline{\gamma}_{\theta,p}(\mathcal{X}_{1:p}) \quad (8)$$

   and ψ_θ is as in (5). In addition, (7) admits $\overline{\pi}(\theta)$ (the marginal on θ of (8)) as a marginal.

2. *The kernel generates a sequence $\{\theta(i), \mathcal{X}_{1:p}(i)\}$ such that the marginal law $\{\mathcal{L}^N\}$ satisfies*

$$\lim_{i \to \infty} \| \mathcal{L}^N(\theta(i), \mathcal{X}_{1:p}(i) \in \cdot) - \pi(\cdot) \| = 0$$

   where π is as in (1).

The proof of the result is in the appendix.

Remark 4.1. *Note that, in the context of Figure 2, any enhanced strategies may be added to this simple algorithm. For example, such as updating blocks of the latent variables or backward simulation in the context of a particle Gibbs version ([35]) (the conditional SMC can be run in a similar manner to Section 4.4 of [1]).*

4.2.1 Some Convergence Analysis

Before continuing, we briefly investigate some convergence properties of the PIMH with multi-level SMC. The analysis uses the basic assumption below. Recall that $\theta \in \Theta$ and $p \in \mathcal{I}$ are fixed. We will be using the following assumption:

(**A2**) For every $\theta \in \Theta$ and $p \in \mathcal{I}$ there exist a $\varphi \in (0, 1)$ such that for $2 \leq n \leq p$ and every $(x, x') \in E \times E$:

$$\varphi \leq \overline{M}_{\theta,n}(x'|x) \leq \varphi^{-1}$$

and for $n = 1$ and every $x \in E$

$$\varphi \leq \overline{M}_{\theta,n}(x) \leq \varphi^{-1}.$$

In addition, there exist a $\rho \in (0,1)$ such that for $1 \leq n \leq p$ and every $\mathcal{X}_{1:n} \in \left(\bigcup_{i\in\mathcal{I}_1}\{i\} \times E^i\right) \times \cdots \times \left(\bigcup_{i\in\mathcal{I}_n}\{i\} \times E^i\right)$:

$$\rho^{t_n} \leq \overline{\gamma}_{\theta,n}(\mathcal{X}_{1:p}) \leq \rho^{-t_n}.$$

We note that although these assumptions are quite strong, they can be typically satisfied on compact state-spaces E.

The PIMH will generate samples from the density

$$\overline{\pi}_\theta^N(k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \frac{\overline{\pi}_{\theta,p}(\mathcal{X}_{1:p}^{(k)})}{N^p} \frac{\psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})}{\mathcal{M}_{\theta,1}(\mathcal{X}_1^{(b_1^k)}) \prod_{n=2}^p \{\bar{W}_{n-1}^{(b_{n-1}^k)} \mathcal{M}_{\theta,n}(\mathcal{X}_n^{(b_n^k)} | \mathcal{X}_{n-1}^{(b_{n-1}^k)})\}}$$

and we write expectations w.r.t. the associated probability as $\mathbb{E}_{\pi_\theta^N}$. Let $\breve{\pi}_\theta$ be the probability distribution associated with the marginal density $\overline{\pi}_\theta(\mathcal{X}_{1:p}^{(k)})$. We proceed by stating the following proposition:

Proposition 4.2. *Assume (A2). Then the PIMH with multi-level SMC, for $N \geq 1$, generates a sequence $\{\mathcal{X}_{1:p}(i)\}$ such that for any $i \geq 1$, $\mathsf{x}(0) \in \mathbb{T}_N^{(p-1)N+1} \times \overline{E}$, $\theta \in \Theta$*

$$\|\mathcal{L}(\mathcal{X}_{1:p}(i) \in \cdot | \mathsf{x}(0)) - \breve{\pi}_\theta(\cdot)\| \leq \mathbb{E}_{\pi_\theta^N}\left[\left(1 - Z_{\theta,p}\{[\rho\varphi]^{2\sum_{j=1}^p \bar{t}_j(0)} \wedge [\rho\varphi]^{2\sum_{j=1}^p \bar{t}_j}\}\right)^i\right]$$

where $\bar{t}_j(0) = \max_{1\leq l\leq N} t_j^{(l)}(0)$ and $\bar{t}_j = \max_{1\leq l\leq N} t_j^{(l)}$.

The proof of this result is in the appendix.

Remark 4.2. *The result shows that, intrinsically, as the maximum SMC algorithm time (w.r.t. π_θ^N) gets smaller, so the rate of convergence increases. This is linked to the variance of the estimate of $\widehat{Z}_{\theta,p}$; shorter algorithm runs will typically yield lower variance and hence better MCMC convergence properties. The result also indicates that longer SMC algorithm runs may reduce convergence rates of the MCMC, but one must bear in mind that the PIMH is most basic of all PMCMC algorithms. In effect, one must hope to control the SMC algorithm time for all the particles to reach A, for efficient MCMC algorithms; we attempt to do this in the next Section.*

4.3 Adaptive Strategies

As noted in Section 3.2.1, there are remaining design issues with multi-level SMC, that is, the choice of p and $\{B_n\}$. Whilst, for a *fixed* $\theta \in \Theta$, one could solve the problem with a preliminary run, once θ is a random variable, there is a key difficulty. For some θ the value of p may have to very large to facilitate an efficient SMC algorithm and conversely, may be small for other θ. For example, for the coalescent model of Section 2.2, if R (the mutation matrix) is fixed, one can

- 0: Set $\theta(0) \in \Theta$, sample $v(0)$ from $\Lambda_{\theta(0)}$ and a multi-level SMC algorithm (i.e. $\bar{\mathcal{X}}_{1:p(v(0))}(0), \bar{\mathbf{a}}_{1:p(v(0))-1}(0)$) using (5) for fixed $\theta = \theta(0)$ and compute and store the estimate of the normalizing constant $\widehat{Z}_{\theta,p}(0)$ (see eq. (6)). Set $i = 1$, $\check{\mathrm{x}}(0) = (\theta(0), k(0), v(0), \bar{\mathcal{X}}_{1:p(v(0))}(0), \bar{\mathbf{a}}_{1:p(v)-1}(0))$.

- 1: Sample $\theta' \sim q(\cdot|\theta(i-1))$ and v' from $\Lambda_{\theta'}$ and then given θ', simulate $\bar{\mathcal{X}}_{1:p(v')}, \bar{\mathbf{a}}_{1:p(v')-1}$ using (5), computing $\widehat{Z}_{\theta',p}$, sampling k' according to $\bar{W}_{p(v')}$ and accept or reject $\check{\mathrm{x}}'$ with probability
$$1 \wedge \frac{\widehat{Z}_{\theta',p}}{\widehat{Z}_{\theta,p}(i-1)} \times \frac{q(\theta(i-1)|\theta')}{q(\theta'|\theta(i-1))}$$
if accepted set $\check{\mathrm{x}}(i) = \check{\mathrm{x}}'$, with $\widehat{Z}_{\theta(i),p}(i) = \widehat{Z}_{\theta',p}$, otherwise retain the current values. Set $i = i + 1$.

Figure 3: Adaptive Multi-Level SMC within MCMC.

envisage a 'large' value of μ may need many sets that are close together, whilst smaller values less so. Hence, to obtain a low variance, accurate estimate of the marginal likelihood, and thus an efficient MCMC algorithm, we will consider an adaptive strategy to propose randomly a different number of levels p and levels' sequence $\{B_n\}$ for every SMC run within each MCMC iteration.

One of the main questions we wish to address is how to perform such an adaptive strategy consistently. An important point, is the fact that since we are interested in parameter inference, it is required that the marginal of the target density is $\overline{\pi}(\theta)$. This can be ensured by introducing a conditionally independent process (on θ) and defining an extended target for the MCMC algorithm, which includes p and $\{B_n\}$ in the target variables. It should be noted that this is explicitly different from Proposition 1 of [29]; there the MCMC kernel is dependent upon an auxiliary process.

4.3.1 Approach

Consider now that it is possible at every MCMC iteration to simulate an auxiliary process v defined upon an abstract state-space $(V, \mathscr{V})$. Let this with associated random variable v, be distributed according to Λ_θ, which is assumed to possess a density with respect to a. $\sigma-$finite measure dv written as $\overline{\Lambda}_\theta$. As hinted by the notation Λ_θ should depend on θ and v is meant be used to determine the sequence of levels levels $\{B_n\}$ for each θ in the PMMH iteration. This auxiliary variable will, for every $\theta \in \Theta$, $\Lambda_\theta-$almost everywhere:

- Induce a path from B_0 to A which is random in length.

- The number of sets induced by v is written $p(v) \in \mathcal{J} \subset \mathbb{Z}_+$.

The advantage of using an auxiliary variable lies that the parameters that determine $\{B_n\}$ can be easily forced to depend on v within this framework. Also, in most applications it might seem easier to find intuition on how to construct and tune Λ_θ than computing the level sets directly from θ.

We will naturally assume that for any $\theta \in \Theta$, (4) will hold $\Lambda_\theta-$almost everywhere (where with p is replaced by $p(v)$). This implies, for every $\theta \in \Theta$, that $\Lambda_\theta-$ almost everywhere

$$\sum_{(t_1,\ldots,t_{p(v)})\in\mathcal{I}_1\times\cdots\times\mathcal{I}_{p(v)}} \int_{E^{t_1+\cdots+t_{p(v)}}} \overline{\gamma}_{\theta,y}(\mathcal{X}_{1:p(v)})d\mathcal{X}_{1:p(v)} \quad = \quad \sum_{t\in\mathcal{I}} \int_{E^t} \overline{\gamma}_{\theta,y}(x_{0:t})dx_{0:t}. \tag{9}$$

In Figure 3 we propose a PMMH algorithm, which at each steps uses adapts the sequence of the levels $\{B_n\}$. For this algorithm we present also the following proposition, whose proof is contained in the appendix.

Proposition 4.3. *Assume (A1). For any $N \geq 1$:*

1. *The invariant density of the kernel in described in Figure 3, is on the space*

$$\Theta \times V \times \bigcup_{j\in\mathcal{J}} \{j\} \times \left[\mathbb{T}_N^{j(N-1)+1} \times \left\{ \left(\bigcup_{i\in\mathcal{I}_1} \{i\} \times E^i \right) \times \cdots \times \left(\bigcup_{i\in\mathcal{I}_{p(j)}} \{i\} \times E^i \right) \right\}^N \right]$$

and has the representation

$$\overline{\pi}_a^N(\theta, k, v, \bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}) = \tag{10}$$

$$\frac{\overline{\pi}(\theta, \mathcal{X}_{1:p(v)}^{(k)})}{N^{p(v)}} \frac{\psi_{\theta,v}(\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}))\overline{\Lambda}_\theta(v)}{\mathcal{M}_{\theta,1}(\mathcal{X}_1^{(b_1^k)}) \prod_{n=2}^{p(v)} \{\bar{W}_{n-1}^{(b_{n-1}^k)} \mathcal{M}_{\theta,n}(\mathcal{X}_n^{(b_n^k)} | \mathcal{X}_{n-1}^{(b_{n-1}^k)})\}}$$

where ψ_θ is as in (5)*. In addition,* (10) *admits $\overline{\pi}(\theta)$ as a marginal.*

2. *The kernel generates a sequence $\{\theta(i), \mathcal{X}_{1:p}(i)\}$ such that the marginal law $\{\mathcal{L}_a^N\}$ satisfies*

$$\lim_{i\to\infty} \|\mathcal{L}_a^N(\theta(i), \mathcal{X}_{1:p}(i) \in \cdot) - \pi(\cdot)\| = 0$$

where π is as in (1)*.*

5 Numerical Examples

Our first example is a toy version of the coalescent, where we use the Wang-Landau ([33]) algorithm within our adaptive scheme. We also consider the application of our algorithm to the coalescent with migration (see [4, 15]).

5.1 The Coalescent

We return to the example of Section 2.2 and it is assumed that the stochastic matrix R is known and has all entries equal to $1/d$. In this scenario, and as noted in Section 4.3, it may be necessary, for good mixing of the PMCMC, to have a large number of levels for μ large and vice-versa. We are to compare multi-level SMC within PMCMC against a particular adaptive multi-level SMC algorithm within PMCMC.

5.1.1 Sampling Strategy

One of the challenges of parameter inference associated to the coalescent model, is the tree structure of the latent variable; this has lead to significant research effort to increase the state-space to facilitate parameter estimation e.g. [34]. The tree makes it very difficult to sample blocks (e.g. up-to the first set B_1) as the tree particles may not 'match-up'. We introduce the following, simple, strategy that is suitable for low or finite dimensional parameter spaces.

We propose to use the Wang-Landau algorithm, which allows one to traverse the entire state-space, via selection of a partition. The algorithm uses an adaptive MCMC kernel, where the target density is changed on the fly. In short, we take the target density, at time i of the MCMC algorithm as

$$\overline{\pi}^N_{w,i}(\theta, k, v, \bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}) \propto \sum_{l=1}^{\bar{l}} \left[\frac{\mathbb{I}_{\Theta_l}(\theta)}{\zeta_i(l)} \overline{\pi}^N_a(\theta, k, v, \bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}) \right]$$

where $\zeta_0(l) = 1$. $\zeta_i(l) > 0$, $l \in \mathbb{T}_{\bar{l}}$, $\Theta = \bigcup_{l=1}^{\bar{l}} \Theta_l$ and for $i \neq j$, $\Theta_i \cap \Theta_j = \emptyset$. The sequence $\{\zeta(l)\}$ is updated using the stochastic approximation at iteration i:

$$\zeta_i(l) = \zeta_{i-1}(l)(1 + \mathbb{I}_{\Theta_l}(\theta(i))\varpi_i)$$

where $\{\varpi_i\}$ is a sequence of step-sizes (see [33] for details) and we adopt the algorithm in Figure 3 to update the rest of the states. It is possible to draw inference from $\overline{\pi}^N_a$, using importance sampling; see e.g. [12]. We refer the reader to [33] for the details on the Wang-Landau algorithm; its usage here is relatively standard.

For the adaptation of the levels (see Section 2.2.1) we propose the following simple idea. Sample the number of levels with probability proportional to μ^p and then, given p place each level an (almost) equal distance apart. For example, if there are 20 individuals and a 4 is sampled, then levels $l_1, \ldots, l_p$ are placed at $19, 14, 9, 4$ with 1 at the end (i.e. we have 5 levels).

The SMC uses approximations of optimal proposal distribution, that are detailed in [32].

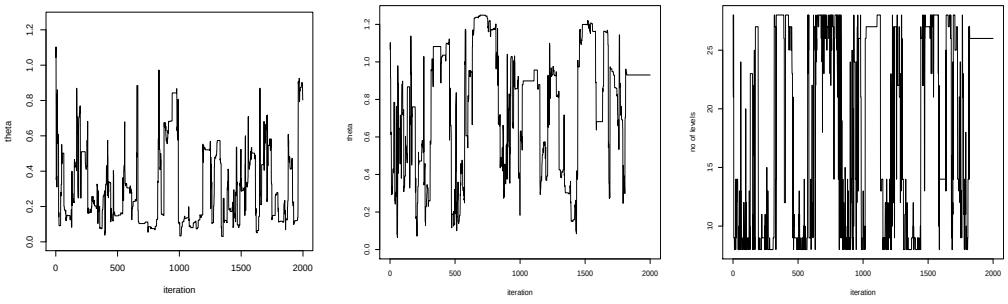
N	50	100	150
Relative $\mathbb{L}_1$	1.71 (0.75)	1.89 (0.67)	2.20 (0.57)

Table 1: The Relative $\mathbb{L}_1-$distance jumped for the adaptive PMCMC against the ordinary. The brackets are the variance across the runs.

5.1.2 Numerical Results

For this example, $d = 4$ with the data-set being $(10, 5, 9, 5)$. The parameter-space is $\Theta = [0, 1.25]$ (with a uniform prior) and we take 50 partitions equally spaced. For the adaptations $p \in \{10, \ldots, 27\}$. The standard SMC had 14 sets with equally spaced levels. The kernel $q(\cdot|\mu)$ in Figure 3 is a normal random walk on the scale $\log([1.25 - \mu]/\mu)$. We included two kernels with proposal variance 0.08 (picked with probability 0.75) and 1. The adaptive and normal versions were run with $N = 50, 100, 150$ for 10000 iterations 10 times (the CPU time, around 800 seconds for $N = 50$ in MATLAB, is approximately the same, but results have been standardized for any differences). All coincidental simulation parameters are the same for both samplers.

The average $\mathbb{L}_1-$distance jumped is in Table 1. In this Table, it can be observed that the adaptive algorithm moves across the space more than its standard counter-part. This is supported by the plots in Figure 4 (every 5 steps, $N = 50$) . Here one can see that the variation in levels, allows the sampler to move across the state-space. For this example, we found the adaptive method to work better as N increased. This is attributed to the fact that the SMC algorithms improve, but also that one has a better sequence of levels for the adaptive method, which leads to better estimates of the marginal likelihoods and superior mixing MCMC algorithms.



(a) μ Normal. (b) μ Adaptive. (c) Levels.

Figure 4: Some PMCMC Plots. In (a) and (b) are the sampled μ. In (c) is the variation of the levels of the PMCMC.

5.2 Coalescent Model with Migration

For our second example we consider the coalescent model with migration.

5.2.1 Model

The model is essentially the same as described in Section 2.2. The difference is the addition of sub-groups and the possibility of individuals in a given group migrating to another.

More formally, we consider g groups, with each observation lying in one of the g groups and having a type in the set $\{1, \ldots, d\}$ as for the coalescent model. That is,

$$x_j = (x_{1,j}^1, \ldots, x_{1,j}^d, \ldots, x_{g,j}^1, \ldots, x_{g,j}^d)$$

where j is the time-index in the Markov chain and

$$E = \{x_{1:g}^{1:d} \in (\mathbb{Z}^+)^{dg} \cap 2 \leq |x_{1:g}^{1:d}|_1 \leq m+1\}.$$

Forward in time, the process under-goes 3 transitions; split, mutation and migration. That is, one can have any of the following 3 transitions:

$$\begin{aligned} X_j &= X_{j-1} + e_{\alpha,i} \\ X_j &= X_{j-1} - e_{\alpha,i} + e_{\alpha,l} \\ X_j &= X_{j-1} - e_{\alpha,i} + e_{\beta,i} \end{aligned}$$

where $e_{\alpha,i}$ has a zero in every position, except the $(\alpha-1)g+i$, $\alpha, \beta \in \{1, \ldots, g\}$, $\alpha \neq \beta$. The transition probabilities are all available, except much more complex than in the scenario with only splits and mutation; we refer the reader to [15] for details. We remark, that the transition probabilities are parameterized by the mutation parameter μ, mutation matrix R and migration matrix Ξ. The latter matrix is symmetric, with zero values on the diagonal, and positive values on the off-diagonals.

As for the model described in Section 2.2, time can be reversed and an IS method introduced. In this article we use the approach described in [15] and refer the reader to that paper for the details. In particular, the $\hat{\pi}$ in [15, pp. 440] is taken as the mutation matrix, which facilitates fast computation, with the possibility of an inefficient SMC procedure.

5.2.2 Results

In our example we generated data with $m = 100$, $d = 256$ and $g = 3$; this is quite a challenging set-up. Throughout, we set the mutation matrix as uniform and concentrate on inferring the

N	10	25	50
Acceptance Rate	0.22 (0.05)	0.26 (0.06)	0.35
CPU Time	18810 sec (480)	43290 sec (529)	3 days

Table 2: Results for the Coalescent with Migration. For the case $N = 10, 25$ the algorithms were run 10 times. The standard errors across the runs are given in brackets.

$\theta = (\mu, \Xi)$. Independent gamma priors with shape and scale parameters equal to 1/5 were adopted for each of the parameters.

For the PMCMC, we used 50 particles and the adaptive scheme for the levels (we use the same set-up as for the previous example) was as follows. We allow either 10, 20 or 33 levels which are equally spaced. The choice of the number of levels is made by taking the number of levels to the power

$$\log\{\mu + \sum_{i>j} \Xi_{ij} + 1\}.$$

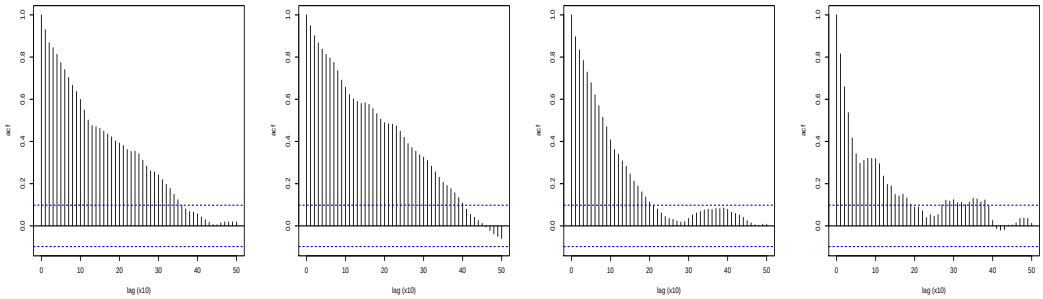
We found this simple adaptive scheme to work well in practice. The proposals for the parameters were random-walks on the log-scale.

The algorithm was run for 4000 iterations on a pentium 2.4 Ghz machine (coded in C), which took around 3 days. Whilst the run-time is quite long, it can easily be made faster by more efficient coding. Some results can be seen in Figure 5 and Table 2.

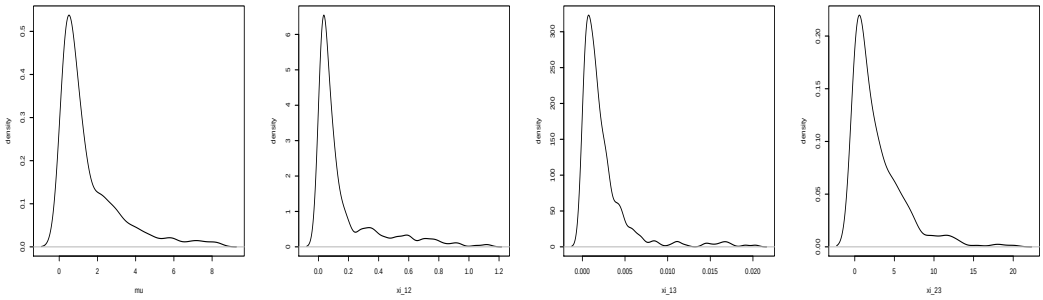
The plots in Figure 5 indicate that the sampler has performed reasonably well, w.r.t. the auto-correlation of the sampler; the acceptance rate in Table 2 is 0.35. In addition, in Table 2, the variability of the acceptance rates and CPU time w.r.t. N is displayed (note that the proposal variances are the same for each N). Here the algorithm seems to perform quite well even for smaller values of N (at least in terms of exploration, if not convergence). The results in this example are encouraging; to our knowledge full and exact Bayesian inference has not been attempted for this class of problem. We expect, with more sophisticated moves the MCMC algorithm can work for even more complex problems.

6 Conclusions

In this article we have presented a multi-level PMCMC algorithm which allows one to perform parameter inference with latent stopped processes. The proposed algorithm requires considerable amount of computation, but to the authors best knowledge for such problems there seems to be



(a) ACF μ. (b) ACF Ξ_{12}. (c) ACF Ξ_{13}. (d) ACF Ξ_{23}.



(e) Density μ. (f) Density Ξ_{12}. (g) Density Ξ_{13}. (h) Density Ξ_{23}.

Figure 5: Some PMCMC Plots for the Coalescent with Migration.

a lack of alternative approaches. Also recent developments in [26] can be adopted to speed up computations.

There are several extensions to the work here, which may be considered. Firstly, the scheme that is used to adapt the level sets relies mainly on intuition. In the rare events literature one may find more systematic techniques to designed level sets, based upon optimal control [14] or simulation [9]. Although these methods are not examined here, they can be characterised using alternative auxiliary variables similar to the ones in Proposition 4.3, so the auxiliary variable framework is quite generic. It is remarked that one can also use multi-level splitting within MCMC. It may be easier for practitioners familiar with splitting to applying the afore mentioned adaptive schemes. In addition the general structure of the auxiliary variable used to adapt the level sets allows the use of independent SMC runs with less particles to set the levels to the ones used for computing the acceptance ratio.

Secondly, one could seek to use the algorithm within an SMC sampler framework [17] as in [13]. As noted in that latter article, the SMC element can improve the sampling scheme, sometimes at a computational complexity that is the same as the original MCMC algorithm. In

addition, this article focuses on the PMMH algorithm. Following Remark 4.1, extensions using particle Gibbs and block updates might prove valuable for many applications.

Thirdly, from a modelling perspective, it may be of interest to apply our methodology in the context of hidden Markov models. In this context, one has

$$\xi_{\theta,y_{0:t}} = \prod_{i=0}^{t} g_\theta(y_i|x_i)$$

with $g_\theta(\cdot|x)$ a conditional density of the observations, given the hidden chain. It would be important to understand, given a range of real applications, the feasibility of statistical inference, combined with the development of our methodology.

Acknowledgement

We thank Arnaud Doucet and Maria De Iorio for many conversations on this work. The second author was kindly supported by the EPSRC programme grant on Control For Energy and Sustainability EP/G066477/1.

Appendix

Proof. [Proof of Proposition 4.1] The proof of (1) and (2) is as in [1], with the exception of the marginal. This latter property relies on proposition 12.2.3 of [16] (the R in that proposition is ∞ in our context), coupled with eq. (7.17) of Theorem 7.4.2 of [16] (the τ^N in that result is always bigger than n in our case). The first result, which uses the strong Markov property, ensures the Feynman-Kac representation of interest. Then, Theorem 7.4.2 demonstrates that any particle approximation of the type in this paper and of a Feynman-Kac formula admits an unbiased estimate of the normalizing constant. In other-words, we have that

$$\overline{\pi}^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \frac{\widehat{Z}_{\theta,p}}{Z_p}\psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})\bar{W}_p^k$$

where $Z_p = \int_\Theta Z_{p,\theta}\overline{\pi}_\theta(\theta)d\theta$. Summing over k and using the unbiased property of the SMC algorithm discussed above, the result is complete. □

Proof. [Proof of Proposition 4.2] Our proof concentrates on the stochastic upper bound of the ratio of target to proposal, which is

$$\frac{\prod_{n=1}^{p} \frac{1}{N}\sum_{j=1}^{N} w_n^{(j)}}{Z_{p,\theta}}.$$

Now, clearly via (A2)

$$w_n \leq \frac{\rho^{t_n}}{\rho^{t_{n-1}}\varphi^{t_n - t_{n-1}}} \leq \left[\frac{1}{\rho\varphi}\right]^{t_n + t_{n-1}}$$

with the convention that $t_0 = 0$. Thus, it follows that

$$\prod_{n=1}^{p} \frac{1}{N} \sum_{j=1}^{N} w_n^{(j)} \leq \prod_{n=1}^{p} \left[\frac{1}{\rho\varphi}\right]^{\bar{t}_n + \bar{t}_{n-1}} \leq \left[\frac{1}{\rho\varphi}\right]^{2\sum_{n=1}^{p} \bar{t}_n}.$$

The result is completed by applying Theorem 6 of [30] and using the upper-bound above. □

Proof. [Proof of Proposition 4.3] The proof is much the same as that of Proposition 4.1, the only issue is the marginal and (2). To find the marginal of $\overline{\pi}_a^N$, re-write the target as:

$$\overline{\pi}_a^N(\theta, k, v, \bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}) = \frac{\widehat{Z}_{\theta,p}}{Z_p} \psi_{\theta,z}(\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}) \bar{W}_{p(v)}^k \overline{\Lambda}_\theta(v).$$

then summing w.r.t. k, integrating w.r.t. $\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}$ and using (9), we have via the unbiasedness of the estimate of the normalizing constant and integrating v

$$\overline{\pi}_a^N(\theta) = \int_V \overline{\pi}(\theta) \overline{\Lambda}_\theta(v) dv$$

from which the result follows.

Now the marginal conditional, given k and v and θ of $\mathcal{X}_{1:p(v)}^{(k)}$ is

$$\frac{\overline{\pi}(\theta, \mathcal{X}_{1:p(v)}^{(k)}) \overline{\Lambda}_\theta(v)}{\overline{\pi}(\theta) \overline{\Lambda}_\theta(v)} = \overline{\pi}(\mathcal{X}_{1:p(v)}^{(k)} | \theta)$$

hence the sequence $\{\theta(i), \mathcal{X}_{1:p(v)}^{(k)}(i)\}$ satisfies the property required. □

References

[1] Andrieu, C., Doucet, A. & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods (with discussion). *J. R. Statist. Soc. Ser. B*, **72**, 269–342.

[2] Andrieu, C., Doucet, A. & Tadic, V. (2009). On-line parameter estimation in general state-space models using pseudo-likelihood. Technical Report, University of Bristol.

[3] Andrieu, C. and Roberts, G.O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Statist,* **37**, 697–725.

[4] Bahlo, M. & Griffiths, R. C. (2000). Coalescence time for two genes from a subdivided population. *J. Math. Biol*, **43**, 397–410.

[5] Bibbona, E. & Ditlevsen, S. (2010). Estimation in discretely observed Markov processes killed at a threshold. Technical Report, University of Torino.

[6] Blom, H.A.P., Bakker, G.J. & Krystul, J. (2007) Probabilistic Reachability Analysis for Large Scale Stochastic Hybrid Systems. *In Proc. 46th IEEE Conf. Dec. Contr.*, New Orleans, USA.

[7] Casella, B.& Roberts, G.O. (2008). Exact Monte Carlo simulation of killed diffusions. *Adv. Appl. Probab.*, **40**, 273–291.

[8] Cerou, F., & Guyader, A. (2007). Adaptive multilevel splitting for rare-events analysis. *Stoch. Anal.*, **25**, 417–433.

[9] Cerou, F., Del Moral, P., Furon, T. & Guyader, A. (2009). Rare event simulation for a static distribution, Technical Report, HAL-INRIA RR-6792.

[10] Cerou, F., Del Moral, P. & Guyader, A. (2011). A non asymptotic variance theorem for unnormalized Feynman-Kac particle models. *Ann. Inst. Henri Poincare*, (to appear).

[11] Chen, Y., Xie, J. & Liu, J.S. (2005). Stopping-time resampling for sequential Monte Carlo methods. *J. R. Statist. Soc. Ser. B*, **67**, 199–217.

[12] Chopin, N., Lelievre, T. & Stoltz, G. (2010). Free-energy methods for efficient exploration of mixture posterior densities. Technical Report, ENSAE.

[13] Chopin, N., Jacob, P. & Papaspiliopoulos, O. (2011). SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. Technical Report, ENSAE.

[14] Dean, T. & Dupuis, P. (2009). Splitting for rare event simulation: A large deviations approach to design and analysis. *Stoch. Proc. Appl.*, **119**, 562–587.

[15] De Iorio, M. & Griffiths, R. C. (2004). Importance sampling on coalescent histories. II: Subdivided population models. *Adv. Appl. Probab.* **36**, 434–454.

[16] Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer: New York.

[17] Del Moral, P., Doucet, A. & Jasra, A. (2006). Sequential Monte Carlo samplers. *J. R. Statist. Soc. B*, **68**, 411–436.

[18] Del Moral, P. & Garnier, J. (2005). Genealogical Particle Analysis of Rare events. *Ann. Appl. Prob.*, **15**, 2496–2534.

[19] Doucet, A., De Freitas, J. F. G. & Gordon, N. J. (2001). *Sequential Monte Carlo Methods in Practice*. Springer: New York.

[20] Glasserman, P., Heidelberger, P., Shahabuddin, P. & Zajic, T. (1999). Multilevel splitting for estimating rare event probabilities. *Oper. Res.*, **47**, 585–600.

[21] Gorur, D. & Teh, Y. W. (2009). An efficient sequential Monte-Carlo algorithm for coalescent clustering. *Adv. Neur. Infor. Proc. Sys.*.

[22] Griffiths, R. C. & Tavare, S. (1994). Simulating probability distributions in the coalescent. *Theoret. Pop. Biol.*, **46**, 131–159.

[23] Johansen, A. M., Del Moral P. & Doucet, A. (2006). Sequential Monte Carlo Samplers for Rare Events, *In Proc. 6th Interl. Works. Rare Event Simul.*, Bamberg, Germany.

[24] Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J. M. & Chopin, N. (2011). On particle methods for parameter estimation in general state-space models, Technical Report, Imperial College London.

[25] Kingman, J. F. C. (1982). On the genealogy of large populations. *J. Appl. Probab.*, **19**, 27–43.

[26] Lee, A., Yau, C., Giles, M., Doucet, A. & Holmes C.C. (2010) On the Utility of Graphics Cards to Perform Massively Parallel Implementation of Advanced Monte Carlo Methods, *J. Comp. Graph. Statist.*, to appear.

[27] Lezaud, P., Krystul, J. & Le Gland, F. (2010) Sampling per mode simulation for switching diffusions. *In Proc. 8th Internl. Works. Rare-Event Simul.* RESIM, Cambridge, UK.

[28] Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer: New York.

[29] Roberts, G. O. & Rosenthal J. S. (2007). Coupling and ergodicity of adaptive MCMC. *J. Appl. Prob.*, **44**, 458–475.

[30] Roberts, G. O. & Rosenthal J. S. (2011). Quantitative Non-Geometric convergence bounds for independence samplers. *Meth. Comp. Appl. Prob.*, (to appear).

[31] Sadowsky, J. S. & Bucklew, J. A. (1990). On large deviation theory and asymptotically efficient Monte Carlo estimation. *IEEE Trans. Inf. Theor.* **36**, 579–588.

[32] Stephens, M. & Donelly, P. (2000). Inference in molecular population genetics (with discussion). *J. R. Statist. Soc. Ser. B*, **62**, 605–655.

[33] Wang, F., & Landau, D. P. (2001). Efficient, multiple range random-walk algorithm to calculate the density of states. *Phys. Rev. Lett.*, **86**, 2050–2053.

[34] Wilson, I., Weale, M. & Balding, D. J. (2003). Inferences from DNA data: population histories, evolutionary processes and forensic match probabilities. *J. R. Statist. Soc. Ser. A*, **166**, 155–188.

[35] Whiteley, N. (2010). Discussion of Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. Ser. B*, **72**, 306–307.