Bayesian Parameter Inference for Partially Observed Stopped Processes

Ajay Jasra[1] and Nikolas Kantas[2]

[1]Department of Statistics & Applied Probability, National University of Singapore, Singapore, 117546, Sg.
E-Mail: *staja@nus.edu.sg*
[2]Department of Electrical Engineering, Imperial College London, London, SW7 2AZ, UK.
E-Mail: *n.kantas@imperial.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 are stopped or killed at the first hitting time of a known set A. Such processes occur naturally within the context of a wide variety of applications. 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) [18, 27] 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, 15] 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 be used as a proposal in PMCMC. In addition, we propose a flexible strategy that adapts 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 [24, 14], finance [7], neuroscience [5], physics [17, 22] and engineering [6, 26]. The vast majority of the papers in the literature deal with 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 data. 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 is [5], where maximum likelihood inference for the model parameters is investigated for the fully observed case.

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 to efficiently sample realisations of a process that starts in a set B_0 and terminates in the given rare target set A before returning to B_0 or getting trapped in some absorbing set. This is usually achieved using Importance Sampling (IS) or multi-level splitting; see[19, 30] and the references in those articles for an overview. Recently, sequential Monte Carlo (SMC) methods based on both these techniques have been used in [6, 17, 22]. 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 approximate distributions. The idea is to introduce a sequence of proposal densities and to sequentially simulate a collection of $N \gg 1$ 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 e.g. [18, 27].

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. In population genetics, for the coalescent model [24], this has been noted as early as in the discussion of [31] by the authors of [11]. Later, 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. The same idea appeared in parallel in [15, Section 12.2], where it was formally interpreted as an interacting particle approximation of appropriate 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 the set A and so that each level can be reached from the previous one with some reasonable probability [19]. This is usually achieved heuristically using trial simulation runs. Also more systematic techniques exist: for cases where large deviations can be applied in [13] 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 of the law of the stopped Markov process, when the process itself is a latent process and is only partially observed via some given dataset. In the context of Bayesian inference one often needs to sample from the posterior density of the model parameters, which can be very complex. Employing standard Markov chain Monte Carlo (MCMC) methods is not feasible, given the difficulty one faces to sample trajectories of the stopped process. In addition, using SMC for sequential parameter inference has been notoriously difficult; see [2, 23]. In particular, due to the successive resampling steps the simulated past of the path of each particle will be very similar to each other. This has been a long standing bottleneck when static parameters θ are estimated online using SMC methods by augmenting them with the latent state. These issues have motivated the recently introduced particle Markov chain Monte Carlo (PMCMC) [1]. 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. In the context of parameter inference for stopped process this brings up the possibility of using the multi-level SMC methodology as a proposal in MCMC. This idea to the best of our knowledge has not appeared previously in the literature. The main contributions made in this article are as follows:

- When the sequence of level sets is fixed *a priori*, the validity of using multi-level SMC within PMCMC is verified.

- To enhance performance we propose a flexible scheme where the level sets are adapted to the current parameter sample. The method is shown to produce unbiased samples from the target posterior density. In addition, we show both theoretically and via numerical examples how the mixing of the PMCMC algorithm is improved when this adaptive strategy is adopted.

This article is structured as follows: in Section 2 we formulate the problem and present the coalescent as a motivating example. In Section 3 we present multi-level SMC for stopped processes. 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 $(x_i, \dots, x_j)$, 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 countable (possibly infinite dimensional) state-space, then

$$\mathcal{S}(E) = \left\{ R = (r_{ij})_{i,j \in E} : r_{ij} \geq 0, \sum_{l \in E} r_{il} = 1 \text{ and } \nu R = \nu \text{ for some } \nu = (\nu_i)_{i \in E} \text{ with } \nu_i \geq 0, \sum_{l \in E} \nu_l = 1 \right\}$$

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 Preliminaries

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 Markov process is homogeneous. The state of the process $\{X_t\}_{t \geq 0}$ begins its evolution in a non empty set B_0 obeying an initial distribution $\nu_\theta : B_0 \to \mathscr{P}(B_0)$ and a 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 $\mathbb{P}_\theta(X_0 \in B_0 \setminus A) = 1$. The associated stopping time is defined as

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

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

In this paper we assume that we have no direct access to the state of the process. Instead the evolution of the state of the process generates a random observations' vector, which we will denote as Y. The realisation of this observations' vector is denoted as y and assume that it takes value in some non empty set F. We will also assume that there is no restriction on A depending on the observed data y, but to simplify exposition this will be omitted from the notation.

In the context of Bayesian inference we are interested in the posterior distribution:

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

where $\tau \in \mathcal{I}$ is the stopping time, π_θ is the prior distribution and γ_θ is the un-normalised complete-data likelihood with the normalising constant of this quantity being:

$$Z_\theta = \sum_{\tau \in \mathcal{I}} \int_{E^{\tau+1}} \gamma_\theta(dx_{0:\tau}, y, \tau)dx_{0:\tau}.$$

The subscript on θ will be used throughout to explicitly denote the conditional dependance on the parameter θ. Given the specific structure of the stopped processes one may write γ_θ as

$$\gamma_\theta(dx_{0:\tau}, y, \tau) = \xi_\theta(y | x_{0:\tau})\mathbb{I}_{(A^c)^\tau \times A}(x_{0:\tau})\nu_\theta(dx_0) \prod_{t=1}^{\tau} P_\theta(dx_t | x_{t-1}), \tag{2}$$

where $\xi_\theta : \Theta \times F \times \left(\bigcup_{\tau \in \mathcal{I}}\{\tau\} \times E^{\tau+1} \right) \to (0, 1)$ is the likelihood of the data given the trajectory of the process. Throughout, it will be assumed that for any $\theta \in \Theta$, $y \in F$, γ_θ admits a density $\overline{\gamma}_\theta(x_{0:\tau}, y, \tau)$ with respect to a σ−finite measure $dx_{0:\tau}$ on $\overline{E} = \left(\bigcup_{\tau \in \mathcal{I}}\{\tau\} \times E^{\tau+1} \right)$ and the posterior and prior distributions π, p admit densities $\overline{\pi}$, $\overline{p}$ respectively both defined with respect to appropriate σ−finite dominating measures.

Note that (1) is expressed as an inference problem for $(\theta, x_{0:\tau}, \tau)$ and not only θ. The overall motivation originates from being able to design a MCMC that can sample from π, which requires one to write the target (or an unbiased estimate of it) up-to a normalizing constant [3]. Still, our primary interest lies in Bayesian inference for the parameter and this can be recovered by the marginals of π with respect to θ. As it will become clear in Section 4 the numerical overhead when augmenting the argument of the posterior is necessary and we believe that the marginals with respect to $x_{0:\tau}, \tau$ might also be useful by-products.

2.2 Motivating example: the coalescent

The framework presented so far is rather abstract, so we introduce the coalescent model as a motivating example. In Figure 1 we present a particular realisation of the coalescent for two genetic types $\{A, C\}$. The process starts at epoch $t = 0$ when the most recent common ancestor (MRCA) splits into two versions of itself. In this example A is

3

Figure 1: Coalescent model example: each of $\{A, C\}$ denote the possible genetic type of observed chromosomes. In this example we have $d = 2$ and $m = 3$. The tree propagates forward in time form the MRCA downwards by a sequence of split and mutation moves. Arrows denote a mutation of one type of a chromosome to another. The name of the process originates from viewing the tree backwards in time (from bottom to top) where the points where the graph join are coalescent events.

chosen to be the MCRA and the process continues to evolve by split and mutation moves. At the stopping point (here $t = 4$) we observe some data y, which corresponds to the number of genes for each genetic type.

In general we will assume there are d different genetic types. The latent state of the process x_t^i is composed of the number of genes of each type i at epoch t of the process and let also $x_t = (x_t^1, \ldots, x_t^d)$. The process begins by default when the first split occurs, so the Markov chain $\{X_t\}_{t \geq 0}$ is initialised by the density

$$\overline{\nu}_\theta(x_0) = \begin{cases} \nu_i & \text{if} \quad x_0 = 2e_i \\ 0 & \text{otherwise.} \end{cases}$$

and is propagated using the following transition density:

$$\overline{P}_\theta(x_t | x_{t-1}) = \begin{cases} \frac{x_{t-1}^i}{|x_{t-1}|_1} \frac{\mu}{|x_{t-1}|_1 - 1 + \mu} r_{il} & \text{if} \quad x_t = x_{t-1} - e_i + e_l \text{ (mutation)} \\ \frac{x_{t-1}^i}{|x_{t-1}|_1} \frac{|x_{t-1}|_1 - 1}{|x_{t-1}|_1 - 1 + \mu} & \text{if} \quad x_t = x_{t-1} + e_i \text{ (split)} \\ 0 & \text{otherwise,} \end{cases}$$

where e_i is defined in Section 1.1. Here the first transition type corresponds to individuals changing type and is called mutation, e.g. $A \to C$ at $t \in \{1, 3\}$ in Figure 1. The second transition is called a split event, e.g. $t \in \{0, 2\}$ in the example of Figure 1. To avoid any confusion we stress that in Figure 1 we present a particular realisation of the process that is composed by a sequence of alternate split and mutations, but this is not the only possible sequence. For example, the bottom of the tree could have be obtained with C being the possible MCRA and a sequence of two consecutive splits and a mutation.

The process is stopped at epoch τ when the number of individuals in the population reaches m. So for the state space we define:

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

4

and for the initial and terminal sets we have:

$$B_0 = \{x : x \in \{0, 2\}^d \text{ and } |x|_1 = 2\}$$
$$A = \{x : x \in (\mathbb{Z}^+)^d \text{ and } |x|_1 = m\}.$$

The data is generated by setting $y := y^{1:d} = x_\tau (\in A)$, which corresponds to the counts of genes that have been observed. In the example of Figure 1 this corresponds to $m = 3$. Hence for the complete likelihood we have:

$$\overline{\gamma}_\theta(x_{0:\tau}, y, \tau) = \mathbb{I}_{A \cap \{x:x=y\}}(x_\tau) \frac{\prod_{i=1}^d y^i!}{m!} \left[\overline{\nu}_\theta(x_0) \prod_{t=1}^\tau \overline{P}_\theta(x_t | x_{t-1}) \right] \tag{3}$$

As expected, the density is only non-zero if at time τ x_τ matches the data y exactly.

Our objective is to infer the genetic parameters $\theta = (\mu, R)$, where $\mu \in \mathbb{R}^+$ and $R \in \mathcal{S}(\mathbb{T}_d)$ and hence the parameter space can be written as $\Theta = \mathbb{R}^+ \times \mathcal{S}(\mathbb{T}_d)$. To facilitate Monte Carlo inference, one can reverse the time parameter and simulate backward from the data. This is now detailed in the context of importance sampling following the approach in [21].

2.2.1 Importance sampling for the coalescent model

To sample realisations of the process for a given $\theta \in \Theta$, importance sampling is adopted but with time reversed. First we introduce a time reversed Markov kernel M_θ with density $\overline{M}_\theta(x_{t-1} | x_t)$. This is used as an importance sampling proposal where sampling is performed backwards in time and the weighting forward in time. We initialise using the data and simulate the coalescent tree backward in time until two individuals remain of the same type. This procedure ensures that the data is hit when the tree is considered forward in time.

The process defined backward in time can be interpreted as a stopped Markov process with the definitions of the initial and terminal sets appropriately modified. For convenience we will consider the reverse event sequence of the previous section, i.e we posed the problem backwards in time with the reverse index being j. The proposal density for the full path starting from the bottom of the tree and stopping at its root can be written as

$$\overline{q}_\theta(x_{0:\tau}) = \mathbb{I}_{B_0 \cap \{x:x=y\}}(x_0) \left\{ \prod_{j=1}^\tau \overline{M}_\theta(x_j | x_{j-1}) \right\} \mathbb{I}_{B_0}(x_\tau).$$

With reference to (3) we have

$$\overline{\gamma}_\theta(x_{0:\tau}, y, \tau) = \frac{m-1}{m-1+\mu} \frac{\prod_{i=1}^d y^i!}{m!} \overline{\nu}_\theta(x_\tau) \left\{ \prod_{j=1}^\tau \frac{\overline{P}_\theta(x_{j-1} | x_j)}{\overline{M}_\theta(x_j | x_{j-1})} \right\} \overline{q}_\theta(x_{0:\tau}).$$

Then the marginal likelihood can be obtained

$$Z_\theta = \frac{m-1}{m-1+\mu} \frac{\prod_{i=1}^d y^i!}{m!} \sum_{\tau \in \mathcal{I}} \int_{E^{\tau+1}} \overline{\nu}_\theta(x_\tau) \left\{ \prod_{j=1}^\tau \frac{\overline{P}_\theta(x_{j-1} | x_j)}{\overline{M}_\theta(x_j | x_{j-1})} \right\} \overline{q}_\theta(x_{0:\tau}) dx_{0:\tau}.$$

In [31] the authors derive an optimal proposal $\overline{M}_\theta$ with respect to the variance of the marginal likelihood estimator. For the sake of brevity we omit any further details. In the current setup where there is only mutation and coalescences, the stopped-process can be integrated out [20], but this is not typically possible in more complex scenarios. A more complicated problem, including migration, is presented in Section 5.2. Finally, we remark that the relevance of the marginal likelihood above will become clear later in Section 4 as a crucial element in numerical algorithms for inferring θ.

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 [15, 18]. To ease exposition, when presenting generic SMC, we shall drop the dependence upon parameter θ.

Algorithm 1 Generic SMC Algorithm

Initialisation, $n = 1$:
    For $i = 1, \ldots, N$

1. Sample $u_1^{(i)} \sim \overline{M}_1$.

2. Compute weights
$$W_1^{(i)} = \frac{\overline{\gamma}_1(u_1^{(i)})}{\overline{M}_1(u_1^{(i)})}, \; \bar{W}_1^{(i)} = \frac{W_1^{(i)}}{\sum_{j=1}^N W_1^{(j)}}.$$

For $n = 2, \ldots, p$,
    For $i = 1, \ldots, N$,

1. Resampling: sample index $a_{n-1}^i \sim f(\cdot | \bar{W}_{n-1})$, where $\bar{W}_{n-1} = (\bar{W}_{n-1}^{(1)}, \ldots, \bar{W}_{n-1}^{(N)})$.

2. Sample $u_n^{(i)} \sim \overline{M}_n(\cdot | u_{1:n-1}^{(a_{n-1}^i)})$ and set $u_{1:n}^{(i)} = (u_{1:n-1}^{(a_{n-1}^i)}, u_n^{(i)})$.

3. Compute weights
$$W_n^{(i)} = w_n(u_{1:n}^{(i)}) = \frac{\overline{\gamma}_n(u_{1:n}^{(i)})}{\overline{\gamma}_{n-1}(u_{1:n-1}^{(i)})\overline{M}_n(u_n^{(i)}|u_{1:n-1}^{(i)})}, \; \bar{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^N W_n^{(j)}}.$$

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 $\{\overline{\gamma}_n\}$ and that we can evaluate each $\overline{\gamma}_n$ point-wise. In addition, we do not require knowledge of Z_n.

3.1 Generic SMC algorithm

SMC algorithms approximate $\{\overline{\pi}_n\}_{n=1}^p$ 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 in Algorithm 1 and we obtain the following SMC approximations for π_n,
$$\pi_n^N(du_{1:n}) = \sum_{j=1}^N \bar{W}_n^{(j)} \delta_{u_{1:n}^{(i)}}(du_{1:n})$$

and for the normalizing constant Z_n:
$$\widehat{Z}_n = \prod_{k=1}^n \left\{ \frac{1}{N} \sum_{j=1}^N W_k^{(j)} \right\}. \tag{4}$$

In this paper we will use f to be the multinomial distribution. Then the resampled index of the ancestor of particle i at time n, namely $a_{n-1}^i \in \{1, \ldots, N\}$, is also a random variable with value chosen with probability $\bar{W}_{n-1}^{(a_{n-1}^i)}$. For each time n, we will denote the complete collection of ancestors obtained from the resampling step as $\bar{\mathbf{a}}_n = (a_n^1, \ldots, a_n^N)$ and the randomly simulated values of the state obtained during sampling (step 2 for $n \geq 2$) as $\bar{u}_n = (u_n^{(1)}, \ldots, u_n^{(N)})$. We will also denote $\bar{\mathbf{a}}_{1:p}, \bar{u}_{1:p}$ the concatenated vector of all these variables obtained during the simulations from time $n = 1, \ldots, p$. Note $\bar{u}_{1:p}$ the is a vector containing all $N \times p$ simulated states and should not be confused with the particle sample of the path $(u_{1:p}^{(1)}, \ldots, u_{1:p}^{(N)})$.

Furthermore, the joint density of all the sampled particles and the resampled indices is

$$\psi(\bar{u}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \left(\prod_{i=1}^{N} \overline{M}_1(u_1^{(i)}) \right) \prod_{n=2}^{p} \left(\prod_{i=1}^{N} \bar{W}_{n-1}^{(a_{n-1}^i)} \overline{M}_n(u_n^{(i)} | u_{n-1}^{(a_n^i-1)}, \dots, u_1^{(a_1^i)}) \right), \tag{5}$$

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 $n \geq 2$. In particular, we set the elements of $b_{1:n}^i$ using 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 $\bar{\mathbf{a}}_{1:n}$ and all the possible simulated state sequences that can be obtained using $\bar{u}_{1:n}$. 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

For different classes of problems one can find a variety of enhanced SMC algorithms; see e.g. [18]. In the context of stopped processes, a multi-level SMC 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 Section 12.2 of [15] which seems better suited for general stopped processes and can provably 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

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

Note that the Markov property of X_t implies $0 \leq \mathcal{T}_1 \leq \mathcal{T}_2 \leq \cdots \leq \mathcal{T}_p = \mathcal{T}$.

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 the set B_n is reached for each $j \in \mathbb{T}_N$. For a given $j \in \mathbb{T}_N$ the path $X_{0:t}^{(j)}$ is "frozen" once $X_{0:t}^{(j)} \in B_n$, until the remaining particles reach B_n and then a resampling step is performed. More formally denote for $n = 1$

$$\mathcal{X}_1 = (x_{0:\tau_1}, \tau_1) \in \{x_{0:\tau_1}, \tau_1 : x_{0:\tau_1-1} \in B_0 \setminus B_1, \, x_{\tau_1} \in B_1\}$$

where τ_1 is a realisation for the stopping time T_1 and similarly for $2 \leq n \leq p$ we have

$$\mathcal{X}_n = \left(x_{\tau_{n-1}+1:\tau_n}, \tau_n\right) \in \{x_{\tau_{n-1}+1:\tau_n}, \tau_n : x_{\tau_{n-1}+1:\tau_n-1} \in B_{n-1} \setminus B_n, \, x_{\tau_n} \in B_n\}.$$

Multi-level SMC is a SMC algorithm which ultimately targets a sequence of distributions $\{\pi_n\}$ each defined on a space

$$\overline{E}_n = \bigcup_{\tau_n \in \mathcal{I}_n} \{\tau_n\} \times E^{\tau_n+1} \tag{6}$$

where $n \in \mathbb{T}_p$, $p \geq 2$ and $\mathcal{I}_1, \dots, \mathcal{I}_p$ are finite collections of positive integer values related to the stopping times $\mathcal{T}_1, \dots, \mathcal{T}_p$ respectively. In the spirit of generic SMC define intermediate target densities $\overline{\pi}_n$ w.r.t to an appropriate σ-finite dominating measure $d\mathcal{X}_n$. We will assume there exists a natural sequence of densities $\{\overline{\pi}_n = \frac{\overline{\gamma}_n}{Z_n}\}_{1 \leq n \leq p}$ obeying the restriction $\overline{\gamma}_p \equiv \overline{\gamma}_\theta$ so that the last target density $\overline{\gamma}_p$ coincides with $\overline{\gamma}_\theta$ in (2). Note that we define a sequence of p target densities, but this time the dimension of $\overline{\gamma}_n$ compared to $\overline{\gamma}_{n-1}$ grows with a random increment of $\tau_n - \tau_{n-1}$. In addition, $\overline{\gamma}_p$ should clearly depend on the value of θ, but this suppressed in the notation. The following proposition is a direct consequence of the Markov property:

Proposition 3.1. *Assume $\mathbb{P}_\theta(\mathcal{T} < \infty) = 1$. Then the stochastic sequence defined $(\mathcal{X}_n)_{1 \leq n \leq p}$ forms a Markov chain taking values in $\overline{E}_n$ defined (6). In addition, for any bounded measurable function $h : \overline{E}_n \to \mathbb{R}$, then $\int_{\overline{E}_n} h(\mathcal{X}_p) \gamma_p(d\mathcal{X}_p) = \sum_{\tau \in \mathcal{I}} \int_{E^{\tau+1}} h(x_{0:\tau}, \tau) \gamma_\theta(dx_{0:\tau}, y, \tau).$*

The proof can be found in [15, Proposition 12.2.2, page 438], [15, Proposition 12.2.4, page 444] and the second part is due to $\overline{\gamma}_p = \overline{\gamma}_\theta$.

We will present multi-level SMC based as a particular implementation of the generic SMC algorithm. Firstly we replace $u_n, u_{1:n}$ with $\mathcal{X}_n, \mathcal{X}_{1:n}$ respectively. Contrary to the presentation of Algorithm 1 for multi-level SMC we will use a homogeneous Markov importance sampling kernel $M_\theta(dx_t|x_{t-1})$, where $M_\theta : \Theta \times E \to \mathscr{P}(E)$, $M_\theta(dx_0|x_{-1}) \equiv M_\theta(dx_0)$ by convention and $\overline{M}_\theta$ is the corresponding density w.r.t. dx. To compute the importance sampling weights of step 3 for $n \geq 2$ in Algorithm 1 we use instead:

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

and for step 2 at $n = 1$:

$$w_1(\mathcal{X}_1) = \frac{\overline{\gamma}_1(\mathcal{X}_1)}{\prod_{l=0}^{\tau_1} \overline{M}_\theta(x_l|x_{l-1})}.$$

To simplify notation from herein we write

$$\mathcal{M}_1(\mathcal{X}_1) = \prod_{l=0}^{\tau_1} \overline{M}_\theta(x_l|x_{l-1})$$

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

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

where again we have suppressed the θ-dependance of $\mathcal{M}_n$ in the notation. We present the multi-level SMC algorithm in Algorithm 2. Note here we include a procedure whereby at each stage n, particles that do not reach B_n before time t_n are rejected by assigning them a zero weight, whereas before it was hinted that resampling is performed when all particles reach B_n. Similar to (5), it is clear that the joint probability density of all the random variables used to implement a multi-level SMC algorithm with multinomial resampling is given by:

$$\psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \left(\prod_{i=1}^{N} \mathcal{M}_1(\mathcal{X}_1^{(i)}) \right) \prod_{n=2}^{p} \left(\prod_{i=1}^{N} \bar{W}_{n-1}^{(a_{n-1}^i)} \mathcal{M}_n(\mathcal{X}_n^{(i)}|\mathcal{X}_{n-1}^{(a_{n-1}^i)}) \right), \tag{7}$$

where $\bar{\mathcal{X}}_{1:p}$ is defined similarly to $\bar{u}_{1:p}$. Finally, recall by construction $Z_p = Z_\theta$ so the approximation of the normalizing constant of γ_θ for a fixed θ is

$$\widehat{Z}_\theta = \widehat{Z}_p = \prod_{n=1}^{p} \left\{ \frac{1}{N} \sum_{j=1}^{N} w_n^{(j)} \left(\mathcal{X}_{1:n}^{(j)} \right) \right\}. \tag{8}$$

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. We will proceed in the spirit of Section 2.2.1 and consider the backward process so that the "time" indexing is set to start from the bottom of the tree towards the root. We introduce a a collection of integers $m > l_1 > l_2 > \cdots > l_p = 2$ and define

$$B_0 = \{x \in (\mathbb{Z}^+ \cup \{0\})^d : x = y\}, \ n = 0,$$
$$B_n = \{x \in (\mathbb{Z}^+ \cup \{0\})^d : |x|_1 = l_n\}, \ 1 \leq n \leq p.$$

Clearly we have $B_{n-1} \supset B_n$, $1 \leq n \leq p$ and $B_p = A$. One can also write the sequence of target densities for the multi-level setting as:

$$\overline{\gamma}_1(x_{0:\tau_1}, \tau_1) = \frac{m-1}{m-1+\mu} \frac{\prod_{i=1}^{d}(y^i)!}{m!} \mathbb{I}_{\{y\}}(x_0) \prod_{l=1}^{\tau_1} \overline{P}_\theta(x_{l-1}|x_l) \mathbb{I}_{\{x_{t_n} \in B_n\}}(x_{t_n}),$$

$$\overline{\gamma}_n(x_{0:\tau_n}, \tau_n) = \overline{\gamma}_{n-1}(x_{0:\tau_{n-1}}, \tau_{n-1}) \prod_{l=\tau_{n-1}+1}^{\tau_n} \overline{P}_\theta(x_{l-1}|x_l) \mathbb{I}_{\{x_{t_n} \in B_n\}}(x_{t_n}), \ n = 1, \ldots, p.$$

Algorithm 2 Multi-level SMC Algorithm

Initialisation, $n = 1$:
    For $i = 1, \ldots, N$

  1. For $t = 1, \ldots, t_1$:

    (a) Sample $x_t^{(i)} \sim M_\theta(\cdot | x_{t-1}^{(i)})$.

    (b) If $x_t^{(i)} \in B_1$ set $\tau_1^{(i)} = t$, $\mathcal{X}_1^{(i)} = \left(x_{0:\tau_1^{(i)}}^{(i)}, \tau_1^{(i)} \right)$ and go to step 2.

  2. Compute weights

$$W_1^{(i)} = \frac{\overline{\gamma}_1(\mathcal{X}_1^{(i)}) \mathbb{I}_{\tau_1^{(i)} \leq t_1}}{\mathcal{M}_1(\mathcal{X}_1^{(i)})}, \ \bar{W}_1^{(i)} = \frac{W_1^{(i)}}{\sum_{j=1}^N W_1^{(j)}}.$$

For $n = 2, \ldots, p$,
    For $i = 1, \ldots, N$,

  1. Resampling: sample index $a_{n-1}^i \sim f(\cdot | \bar{W}_{n-1})$, where $\bar{W}_{n-1} = (\bar{W}_{n-1}^{(1)}, \ldots, \bar{W}_{n-1}^{(N)})$.

  2. For $t = \tau_{n-1}^{(i)} + 1, \ldots, t_n$:

    (a) Sample $x_t^{(i)} \sim M_\theta(\cdot | x_{t-1}^{(i)})$.

    (b) If $x_t^{(i)} \in B_n$ set $\tau_n^{(i)} = t$, $\mathcal{X}_n^{(i)} = \left(x_{\tau_{n-1}^{(i)}+1:\tau_n^{(i)}}^{(i)}, \tau_n^{(i)} \right)$ and go to step 3.

  3. Set $\mathcal{X}_{1:n}^{(i)} = (\mathcal{X}_{1:n-1}^{(a_{n-1}^i)}, \mathcal{X}_n^{(i)})$.

  4. Compute weights

$$W_n^{(i)} = w_n(\mathcal{X}_{1:n}^{(i)}) = \frac{\overline{\gamma}_n(\mathcal{X}_{1:n}^{(i)}) \mathbb{I}_{\tau_n^{(i)} \leq t_n}}{\overline{\gamma}_{n-1}(\mathcal{X}_{1:n-1}^{(i)}) \mathcal{M}_n(\mathcal{X}_n^{(i)} | \mathcal{X}_{1:n-1}^{(i)})}, \ \bar{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^N W_n^{(j)}}.$$

Algorithm 3 Particle independent Metropolis-algorithm (PIMH)

---

1. Sample $\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}$ from (7) using the multi-level implementation of Algorithm 1 detailed in Section (3.2) and compute $\widehat{Z}_p$. Sample $k \sim f(\cdot|\bar{W}_p)$.

2. Set $\xi(0) = \left(k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)\right) = \left(k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}\right)$ and $\widehat{Z}_p(0) = \widehat{Z}_p$.

3. For $i = 1, \ldots, K$:

   (a) Propose a new $\bar{\mathcal{X}}'_{1:p}, \bar{\mathbf{a}}'_{1:p}$ and k' as in step 1 and compute $\widehat{Z}'_p$,

   (b) Accept this as the new state of the chain with probability $1 \wedge \frac{\widehat{Z}'_p}{\widehat{Z}_p(i-1)}$. If we accept, set $\xi(i) = \left(k(i), \bar{\mathcal{X}}_{1:p}(i), \bar{\mathbf{a}}_{1:p-1}(i)\right) = \left(k', \bar{\mathcal{X}}'_{1:p}, \bar{\mathbf{a}}'_{1:p-1}\right)$ and $\widehat{Z}_p(i) = \widehat{Z}'_p$. Otherwise reject, $\xi(i) = \xi(i-1)$ and $\widehat{Z}_p(i) = \widehat{Z}_p(i-1)$.

The major design problem that remains in general is that given *any* candidates for $\{\overline{M}_{n,\theta}\}$, how to set the spacing (in some sense) of the $\{B_n\}$ 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

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

Now, in general for any $\{l_n\}_{n=1}^p$ and p it is hard to know beforehand how much better (or not) the resulting multi-level algorithm will perform relative to a vanilla SMC algorithm. Whilst [11] show empirically that in most cases one should expect a considerable improvement, there θ is considered to be fixed. In this case one could design the levels sensibly using offline heuristics or more advanced systematic methods using optimal control [13] or adaptive simulation [8, 9], e.g. by setting the next level using the median of a pre-specified rank of the particle sample. What we aim to establish in the next section is that when θ is varies as in the context of MCMC algorithms, one can both construct PMCMC algorithms based on multi-level SMC and more importantly easily design for each θ different sequences for $\{B_n\}$ based on similar ideas.

4   Multi-Level Particle Markov Chain Monte Carlo

Particle Markov Chain Monte Carlo (PMCMC) methods are MCMC algorithms, which use all the random variables generated by SMC approximations as 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 that in order use the simulated variables from SMC, one defines a complex invariant distribution for the MCMC on an extended state space. This extended target is such that a marginal of this invariant distribution is the one of interest.

This section aims on providing insight to the following questions:

1. Is it valid in general to use multi-level SMC within PMCMC?

2. Given that it is, how can we use the levels to improve the mixing of PMCMC?

The answer to the first question seems rather obvious, so we will provide some standard but rather strong conditions for which multi-level PMCMC is valid. For the second question we will propose an extension to PMMH that adapts the level sets used to θ at every iteration of PMCMC. [1] introduces three different and generic PMCMC algorithms: particle independent Metropolis Hastings algorithm (PIMH), particle marginal Metropolis Hastings (PMMH) and particle Gibbs samplers. In the remainder of the paper we will only focus on the first two of these.

4.1   Particle independent Metropolis Hastings (PIMH)

We will begin by presenting the simplest generic algorithm found in [1], namely the particle independent Metropolis Hastings algorithm (PIMH). In this case θ and p are fixed and PIMH is designed to sample from the pre-specified

target distribution π_p also considered in Section 3.1. Although PIMH is not useful for parameter inference it is included for pedagogic purposes. One must bear in mind that PIMH is the most basic of all PMCMC algorithms. As such it is easier to analyse but still can provide useful intuition that can be used later in the context of PMMH and varying θ.

PIMH is presented in Algorithm 3. It can be shown, using similar arguments to [1], that the invariant density of the Markov kernel above is exactly (see the proof of Proposition 4.2)

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

where ψ is as in (7) and as before we have $b_p^k = k$ and $b_n^k = a_n^{b_{n+1}^k}$ for every k, n. Note that $\overline{\pi}_p^N$ admits the target density of interest, $\overline{\pi}_p$ as the marginal, when k and $\bar{\mathbf{a}}_{1:p-1}$ are integrated out.

We commence by briefly investigating some convergence properties of PIMH with multi-level SMC. Even though the scope of PIMH is not parameter inference, one can use insight on what properties are desired by multi-level SMC for PIMH when designing other PMCMC algorithms used for parameter inference. We begin with posing the following mixing and regularity assumption:

(**A1**) For every $\theta \in \Theta$ and $p \in \mathcal{I}$ there exist a $\varphi \in (0,1)$ such that for every $(x, x') \in E \times E$:

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

There exist a $\rho \in (0,1)$ such that for $1 \leq n \leq p$ and every $\mathcal{X}_{1:n} \in \bar{E}_n$:

$$\rho^{\tau_n} \leq \overline{\gamma}_n(\mathcal{X}_{1:n}) \leq \rho^{-\tau_n}.$$

The stopping times are finite, that is for $1 \leq n \leq p$ there exist a $\bar{\tau}_n < \infty$ such that

$$\tau_n \leq \bar{\tau}_n.$$

Assumption (A1) is rather strong, but are often used in the analysis of these kind of algorithms [1, 15] because they simplify the proofs to a large extent. Recall that $\theta \in \Theta$ and $p \in \mathcal{I}$ are fixed. We proceed by stating the following proposition:

Proposition 4.1. *Assume (A1). Then for $N \geq 1$ Algorithm 3 generates a sequence $(\mathcal{X}_{1:p}(i))_{i \geq 0}$ that for any $i \geq 1$, $\xi(0) \in \mathbb{T}_N^{(p-1)N+1} \times \overline{E}$, $\theta \in \Theta$ satisfies:*

$$\|\mathcal{L}aw(\mathcal{X}_{1:p}(i) \in \cdot | \xi(0)) - \pi_p(\cdot)\| \leq \left(1 - Z_p \left((\rho\varphi)^{2 \sum_{j=1}^{p} \bar{\tau}_j} \right) \right)^i.$$

The proof can be found in the appendix. The following remarks are generalised and do not always hold, but provide some intuition for the ideas that follow. The result shows intrinsically that as the supremum of the sum of the stopping times with respect to $\{B_n\}_{n=1}^p$ gets smaller, so does the convergence rate increase. This can be also linked to the variance of the estimator of $\widehat{Z}_p$, which is well known to increase linearly with p [15, Theorem 12.2.2, pages 451-453]. Shorter stopping times will typically yield lower variance and hence better MCMC convergence properties. On the other hand often γ_p will be larger for longer p and longer stopping times (Proposition 4.1 is derived for a fixed p). In addition, sampling a stopped process is easier using a higher number of levels. In summary, the tradeoff is that although it is more convenient to use more auxiliary variables for simulating the process, these will slow down the mixing of PMCMC. In practice one balances this by trying to use a moderate number of levels for which most the particles to reach A. This tradeoff serves as a motivation for developing flexible schemes to vary $p, \{B_n\}_{n=1}^p$ with θ in the PMCMC algorithm presented later in Section 4.3.

4.2 Particle marginal Metropolis Hastings (PMMH)

In the remainder of this section we will focus on using a multi-level SMC implementation within a PMMH algorithm. 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. Recall (1) can be expressed in terms of densities as:

$$\overline{\pi}(\theta, \mathcal{X}_{1:p}) \propto \overline{\gamma}_p(\mathcal{X}_{1:p}) \overline{p}(\theta) \tag{9}$$

Algorithm 4 Particle marginal Metropolis Hastings using multi-Level SMC.

---

1. Sample $\theta(0) \sim p(\cdot)$. Given $\theta(0)$ sample $\bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)$ using multi-level SMC and compute $\widehat{Z}_{\theta(0)}$. Sample $k \sim f(\cdot|\bar{W}_p)$.

2. Set $\xi(0) = \left(\theta(0), k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)\right)$ and $\widehat{Z}_\theta(0) = \widehat{Z}_{\theta(0)}$.

3. For $i = 1, \ldots, K$:

   (a) Sample $\theta' \sim q(\cdot|\theta(i-1))$; given θ' propose a new $\bar{\mathcal{X}}'_{1:p}, \bar{\mathbf{a}}'_{1:p-1}$ and k' as in step 1 and compute $\widehat{Z}'_{\theta'}$.

   (b) Accept this as the new state of the chain with probability

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

   If we accept, set $\xi(i) = \left(\theta(i), k(i), \bar{\mathcal{X}}_{1:p}(i), \bar{\mathbf{a}}_{1:p-1}(i)\right) = \left(\theta', k', \bar{\mathcal{X}}'_{1:p}, \bar{\mathbf{a}}'_{1:p-1}\right)$ and $\widehat{Z}_\theta(i) = \widehat{Z}'_{\theta'}$. Otherwise reject, $\xi(i) = \xi(i-1)$ and $\widehat{Z}_\theta(i) = \widehat{Z}_\theta(i-1)$.

---

and let the marginal density given by

$$\bar{\pi}(\theta) = \sum_{\tau \in \mathcal{I}} \int_{E^{\tau+1}} \bar{\pi}(\theta, x_{0:\tau}, \tau|y) dx_{0:\tau}.$$

For the time being we will consider the case when p is fixed. In the context of our stopped Markov process, we propose a PMMH algorithm targeting $\bar{\pi}(\theta, \mathcal{X}_{1:p})$ in Algorithm 4.

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

(A2) For any $\theta \in \Theta$ and $p \in \mathcal{I}$ we define the following sets for $n = 1, \ldots, p$: $S_n^\theta = \{\mathcal{X}_{1:n} \in \overline{E}_n : \gamma_n(\mathcal{X}_{1:n}) > 0\}$ and $Q_n^\theta = \{\mathcal{X}_{1:n} \in \overline{E}_n : \gamma_{n-1}(\mathcal{X}_{1:n-1})\mathcal{M}_{\theta,n}(\mathcal{X}_n|\mathcal{X}_{n-1}) > 0\}$. For any $\theta \in \Theta$ we have that $S_n^\theta \subseteq Q_n^\theta$. In addition the ideal Metropolis Hastings targeting $\overline{\pi}(\theta)$ using 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. We proceed with the following result:

Proposition 4.2. *Assume (A2); then for any $N \geq 1$:*

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

   $$\overline{\pi}_p^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \frac{\overline{\pi}(\theta, \mathcal{X}_{1:p}^{(k)})}{N^p} \frac{\psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})}{\mathcal{M}_1(\mathcal{X}_1^{(b_1^k)}) \prod_{n=2}^p \left\{ \bar{W}_{n-1}^{(b_{n-1}^k)} \mathcal{M}_n(\mathcal{X}_n^{(b_n^k)}|\mathcal{X}_{n-1}^{(b_{n-1}^k)}) \right\}} \tag{10}$$

   where $\overline{\pi}$ is as in (9) and ψ_θ as in (7). In addition, (10) admits $\overline{\pi}(\theta)$ as a marginal.

2. *Algorithm 4 generates a sequence $(\theta(i), \mathcal{X}_{1:p}(i))_{i \geq 0}$ such that*

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

   where π is as in (1).

The proof of the result is in the Appendix. The result is based on Theorem 4 of [1]. Note that Algorithm 4 presented in a generic form of a "vanilla" PMMH algorithm, so it can be enhanced using various strategies. For example, it is possible to add block updating of the latent variables or backward simulation in the context of a particle Gibbs version [33]. In the next section, we propose a flexible scheme that allows to set a different number of levels after a new θ' is proposed.

Algorithm 5 Particle marginal Metropolis Hastings using multi-Level SMC with adaptive level sets.

---

1. Sample $\theta(0) \sim p(\cdot)$. Given $\theta(0)$: sample $v(0)$ from $\Lambda_{\theta(0)}$, then $\bar{\mathcal{X}}_{1:p(v(0))}(0), \bar{\mathbf{a}}_{1:p(v(0))-1}(0)$ using multi-level SMC and compute $\widehat{Z}_{\theta(0)}$. Sample $k \sim f(\cdot|\bar{W}_p)$.

2. Set $\xi(0) = \left(\theta(0), v(0), k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)\right)$ and $\widehat{Z}_\theta(0) = \widehat{Z}_{\theta(0)}$.

3. For $i = 1, \ldots, K$:

   (a) Sample $\theta' \sim q(\cdot|\theta(i-1))$; sample v' from $\Lambda_{\theta'}$ and $\bar{\mathcal{X}}'_{1:p(v')}, \bar{\mathbf{a}}'_{1:p(v')-1}$, k' as in step 1 and compute $\widehat{Z}'_{\theta'}$.

   (b) Accept this as the new state of the chain with probability

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

   If we accept, set $\xi(i) = \left(\theta(i), v(i), k(i), \bar{\mathcal{X}}_{1:p(v(i))}(i), \bar{\mathbf{a}}_{1:p(v(i))-1}(i)\right) = \left(\theta', v', k', \bar{\mathcal{X}}'_{1:p(v')}, \bar{\mathbf{a}}'_{1:p(v')-1}\right)$ and $\widehat{Z}_\theta(i) = \widehat{Z}'_{\theta'}$. Otherwise reject, $\xi(i) = \xi(i-1)$ and $\widehat{Z}_\theta(i) = \widehat{Z}_\theta(i-1)$.

4.3 Adapting the level sets

The remaining design issue for PMMH is how to tune multi-level SMC by choosing p and $\{B_n\}_{n=1}^p$. Whilst, for a fixed $\theta \in \Theta$, one could solve the problem with preliminary runs, when θ varies this is not an option. In general the value of θ should dictate how small or large p should be to facilitate an efficient SMC algorithm. Hence, to obtain a more accurate estimate of the marginal likelihood and thus an efficient MCMC algorithm, we need to consider adaptive strategies to propose randomly a different number of levels p and levels' sequence $\{B_n\}_{n=1}^p$ for each $\theta(i)$ sampled at every PMMH iteration i. To ease exposition we will assume that $p, \{B_n\}_{n=1}^p$ can be expressed as functions of an arbitrary auxiliary parameter v.

Given $\theta(i)$ is a random variable, 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 PMMH invariant density is $\overline{\pi}(\theta)$. This can be ensured (see Proposition 4.3) by introducing at each PMMH iteration, the parameters that form the level sets $v(i)$ as an auxiliary process, which given $\theta(i)$ is conditionally independent of $k(i), \bar{\mathcal{X}}_{1:p}(i), \bar{\mathbf{a}}_{1:p-1}(i)$. This way we define an extended target for the MCMC algorithm, which includes p and $\{B_n\}_{n=1}^p$ in the target variables. It should be noted that this scheme is explicitly different from Proposition 1 of [28], where the MCMC transition kernel at iteration i is dependent upon an auxiliary process. Here one just augments the target space with more auxiliary variables.

Consider now that it is possible at every PMMH iteration i to simulate the 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 $\{B_n\}_{n=1}^p$ for each $\theta(i)$ in PMMH. This auxiliary variable will induce for every $\theta \in \Theta$:

- a random number of level sets $p(v) \in \mathcal{J} \subset \mathbb{Z}_+$.

- a sequence of level sets $\{B_n(v)\}_{n=1}^{p(v)}$ with $B_{p(v)} = A$.

We will assume that for any $\theta \in \Theta$, Proposition 3.1 and(6) will hold $\Lambda_\theta-$almost everywhere, where this time p should be replaced by $p(v)$. This implies that for every $\theta \in \Theta$ we have:

$$\sum_{\tau_{p(v)} \in \mathcal{I}_{p(v)}} \int_{E^{1+\tau_{p(v)}}} \overline{\gamma}_\theta(x_{0:\tau_{p(v)}}, y, \tau_{p(v)}) dx_{0:\tau_{p(v)}} = \sum_{\tau \in \mathcal{I}} \int_{E^{\tau+1}} \overline{\gamma}_\theta(x_{0:\tau}, y, \tau) dx_{0:\tau}, \tag{11}$$

where the expression holds $\Lambda_\theta-$ almost everywhere. In Algorithm 5 we propose a PMMH algorithm, which at each step i uses $\theta(i)$ to adapt the levels $\{B_n(v(i))\}_{n=1}^{p(v(i))}$. For Algorithm 5 we present the following proposition that verifies varying the level sets in this way is theoretically valid:

Proposition 4.3. *Assume (A2) and* (11) *hold. Then, for any* $N \geq 1$:

1. The invariant density of the procedure in Algorithm 5 is defined on the space

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

and has the representation

$$\overline{\pi}^N(\theta, k, v, \bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}) = \frac{\overline{\pi}(\theta, \mathcal{X}_{1:p(v)}^{(k)})}{N^{p(v)}} \frac{\psi_\theta(\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1})\overline{\Lambda}_\theta(v)}{\mathcal{M}_1(\mathcal{X}_1^{(b_1^k)}) \prod_{n=2}^{p(v)} \{\bar{W}_{n-1}^{(b_{n-1}^k)} \mathcal{M}_n(\mathcal{X}_n^{(b_n^k)}|\mathcal{X}_{n-1}^{(b_{n-1}^k)})\}} \qquad (12)$$

where $\overline{\pi}$ is as in (9) and ψ_θ is as in (7). In addition, (12) admits $\overline{\pi}(\theta)$ as a marginal.

2. The generated sequence $\big(\theta(i), \mathcal{X}_{1:p(v(i))}(i)\big)_{i \geq 0}$ satisfies

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

where π is as in (1).

The proof is contained in the Appendix. We are essentially using an auxiliary framework similar to [3]. As in (1) we included $x_{0:\tau}, \tau$ in the target posterior, when we were primarily interested in θ, this time we augment the target posterior with v and the SMC variables $\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}$, which is a consequence of using PMCMC. The disadvantage is that as the space of the posterior increases it is expected that the mixing of the algorithm will be slower. This could be improved if we have opted $x_{0:\tau}, \tau$ and v to be dependent on each other given θ, but this would need additional assumptions for the structure of γ_θ. In addition, in many applications the parameters v that determine $\{B_n\}_{n=1}^p$ appear naturally and v often is low dimensional. 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. For example, for the coalescent model of Section 2.2 with the mutation matrix R is fixed, one can envisage for a larger value of μ coalescent events are less likely and more level sets closer together are needed compared to smaller values of μ.

5 Numerical Examples

We will illustrate the performance of PMMH using numerical examples on two models from population genetics. The first one deals with the coalescent model of Section 2.2 when a low dimensional dataset is observed. This is meant as an academic/toy example suitable for comparing different PMMH implementations. The second example is a more realistic application and deals with a coalescent model that allows migration of individual genetic types from one sub-group to another [4, 14]. In both cases we will illustrate the performance of PMMH implemented with a simple intuitive strategy for adapting the level sets.

5.1 The coalescent model

We will use a known stochastic matrix R with all entries equal to $1/d$. In this example $d = 4$ with and the dataset is $y = (10, 5, 9, 5)$. The parameter-space is set as $\Theta = [0, 1.5]$ and a uniform prior will be used. For M_θ we will use the optimal proposal distributions provided by [31]. The PMMH proposal $q(\cdot|\mu(i-1))$ in Algorithm 5 is a log normal random walk, i.e. we use $\zeta' = \zeta(i-1) + 0.4\mathcal{N}(0, 1)$ with $\zeta = \log(\mu)$.

We will compare PMMH when implemented with a simple adaptive scheme for p and when p is fixed. In the latter case we set $p = 14$. When an adaptive strategy is employed we will sample each time p directly using a multinomial distribution defined on $\{8, \ldots, 28\}$ with weights proportional to μ^p. In both cases given p we place the levels almost equally spaced apart.

5.1.1 Numerical results

The adaptive and normal versions were run with $N = 50, 100, 200$ for 10^5 iterations. In each case the algorithm took approximately 2.5, 5, 10 hours to complete when implemented in Matlab and run on a Linux workstation using a Intel Core 2 Quad Q9550 CPU at 2.83 GHz. The results are shown in Figure 2 and 3. We observed that when we varied the number of levels, this allowed the sampler to traverse through a bigger part of the state space compared to when a fixed number of levels is used. As a result the estimated pdf of the adaptive case manages to include a second mode that is not seen in the non adaptive case. In the fixed levels case we see a clear improvement with increasing N, although the difference in the mixing between $N = 100$ and 200 is marginal. In the adaptive case the sampler performed well even with lower values of N.

Figure 2: PMMH for the coalescent without adaptation. A fixed number of 14 level sets is used. Left: estimated pdf of μ for $N = 50, 100, 200$. Centre: the trace plot for $N = 100$. Right: autocorrelation plots for $N = 50, 100, 200$. The average acceptance ratio was 0.07, 0.08 and 0.10 respectively.

Figure 3: PMMH for the coalescent adaptation. The number of levels sampled is proportional to μ^p. Far left: estimated pdf of μ for $N = 50, 100, 200$. Central left: the trace plot for $N = 100$. Central right: histogram of number of levels in the posterior for $N = 100$. Far right: autocorrelation function plots for $N = 50, 100, 200$. The average acceptance ratio was 0.10, 0.11 and 0.13 respectively.

5.2   The coalescent model with migration

The model is similar to the one as described in Section 2.2. The major difference is that this time genetic types are of classified into sub-groups within which most activity happens. In addition, individuals are allowed to migrate from one group to another. We commence with a brief description of the model and refer the interested reader to [4, 14] for more details. As in Section 2.2 we will consider the process forward in time. Let g be the number of groups and the state at time t be composed as the concatenation of g groups of different genetic types as:

$$x_t = (x_{1,t}^1, \ldots, x_{1,t}^d, \ldots, x_{g,t}^1, \ldots, x_{g,t}^d)$$

The process under-goes split, mutation and migration transitions as follows:

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

where $\alpha, \beta \in \{1, \ldots, g\}$ with $\alpha \neq \beta$ and $e_{\alpha,i}$ is a vector with a zero in every element except the $(\alpha - 1)g + i$ -th one. Similarly to the simpler model of Section 2.2 the transition probabilities are parameterised by the mutation parameter μ, mutation matrix R and a migration matrix G. The latter is a symmetric matrix with zero values on the diagonal and positive values on the off-diagonals. Finally the data is generated when at time τ the number of individuals in the population reaches m, and $y = y^{1:gd} = x_\tau$.

As for the model described in Section 2.2 one can reverse time and employ an backward sampling forward weighting importance sampling method; see [14] for the particular implementation details. In our example we generated data with $m = 100$, $d = 64$ and $g = 3$. This is quite a challenging set-up. As in the previous example we set the mutation matrix R to be known and uniform and we will concentrate on inferring the $\theta = (\mu, G)$. Independent gamma priors with shape and scale parameters equal to 1 were adopted for each of the parameters.

5.2.1 Numerical results

We implemented PMMH using $N = 50, 100, 200$ and a simple adaptive scheme for p. We allow $p \in \{10, 20, 33\}$ and use approximately equal spacing between the levels. We choose each p with probability proportional to $p^{\log\{\mu + \sum_{i>j} G_{ij} + 1\}}$. The proposals for the parameters were Gaussian random-walks on the log-scale. The algorithm was implemented in C/C++ was run for 10^5 iterations, which took approximately 3, 6 and 12 hours to complete. Whilst the run-time is quite long it can be improved by at least one order of magnitude if the SMC is implemented on Graphical Processing Units (GPU) as in [25].

For the dataset plotted in Figure 4 (left) the results are plotted in Figures 4 (right) and 5. The auto-correlation and trace plots indicate that the sampler mixes reasonably well for every N. These results in this example are encouraging as to the best of our knowledge Bayesian inference has not been attempted for this class of problems. We expect that practitioners with insight in the field of population genetics can come with more sophisticated MCMC proposals or adaptive schemes for the level sets, so that the methodology can be extended to realistic applications.

Figure 4: Left: Dataset for the Coalescent with Migration. Right: Histogram of number of levels p in the resulting posterior for $N = 100$.

6   Discussion

In this article we have presented a multi-level PMCMC algorithm which allows one to perform Bayesian inference for the parameters of a latent stopped processes. In terms of methodology the main novelty of the approach is that uses auxiliary variables to adaptively compute the level sets with θ. The general structure of this auxiliary variable allows it to incorporate the use of independent SMC runs with less particles to set the levels. In the numerical examples we demonstrated that the addition auxiliary variables slow down the convergence of PMCMC, but this seemed a reasonable compromise in terms of performance compared when fixed number of level sets were used. The proposed algorithm requires considerable amount of computation, but to the authors best knowledge for such problems there seems to be a lack of alternative approaches. Also, recent developments GPU hardware can be adopted to speed up the computations even by orders of magnitude as in [25].

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. We found simple adaptive implementations to work well in practice. In the rare events literature one may find more systematic techniques to design the level sets, based upon optimal control [13] 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 we use is quite generic. In addition, we emphasise that within a PMCMC framework one may also include multi-level splitting algorithms instead of SMC, which might appeal practitioners familiar with multi-level splitting.

Secondly, one could seek to use these ideas within a SMC sampler framework of [16] as done in [12]. As noted in the latter article, a sequential formulation can improve the sampling scheme, sometimes at a computational complexity that is the same as the original PMCMC algorithm. In addition, this article focuses on the PMMH algorithm, so clearly extensions using particle Gibbs and block updates might prove valuable for many applications.

Finally, 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

$$\xi(y|x_{0:\tau}) = \prod_{i=0}^{\tau} g_\theta(y_i|x_i)$$

Figure 5: PMMH for for the Coalescent with Migration for $N = 50, 100, 200$. Top row: estimated pdfs for μ, G_{12}, G_{13}, G_{23} (from left to right). Middle: trace plots for $N = 100$. Bottom: autocorrelation function plots. The acceptance rate was $0.34, 0.37, 0.4$ respectively.

with $g_\theta(\cdot|x)$ being the conditional likelihood of the observations. It would be important to understand, given a range of real applications, the feasibility of statistical inference, combined with the development of our methodology. An investigation of the effectiveness of such a scheme when applied to queuing networks is currently underway.

Acknowledgement

Appendix

Proof. [Proof of Proposition 4.1] The result is a straight forward application of Theorem 6 of [29] which adapted to our notation states:

$$\|\mathcal{L}aw(\mathcal{X}_{1:p}(i) \in \cdot|\xi(0)) - \check{\pi}_\theta(\cdot)\| \leq \mathbb{E}_{\pi_p^N}\left[\left(1 - \left(\mathbb{E}_{\psi_\theta}\left[1 \wedge \frac{\hat{Z}_p(\Xi)}{\hat{Z}_p(\xi(0))}\bigg|\xi(0)\right] \wedge \mathbb{E}_{\psi_\theta}\left[1 \wedge \frac{\hat{Z}_p(\Xi)}{\hat{Z}_p(\xi)}\bigg|\xi\right]\right)\right)^i\right],$$

where the conditional expectation is the expectation w.r.t. the SMC algorithm (i.e. $\Xi \sim \psi_\theta$) and the outer expectation is w.r.t. the PIMH target (i.e. $\xi \sim \pi_p^N$). We also denote the estimate of the normalizing constant as $\hat{Z}_p(\cdot)$ with $\cdot$ denoting which random variables generate the estimate.

Now, clearly via (A1)

$$w_n(X_{0:\tau_n}) \leq \frac{\rho^{\tau_n}}{\rho^{\tau_{n-1}}\varphi^{\tau_n - \tau_{n-1}}} \leq \left[\frac{1}{\rho\varphi}\right]^{\tau_n + \tau_{n-1}}$$

with the convention that $\tau_0 = 0$. Thus, it follows that

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

and we obtain:

$$\frac{Z_p(\Xi)}{\hat{Z}_p(\cdot)} \geq Z_p(\Xi) \left(\rho\varphi \right)^{2 \sum_{n=1}^{p} \bar{\tau}_n}.$$

Note that by assumption $Z_p(\Xi) \left(\rho\varphi \right)^{2 \sum_{n=1}^{p} \bar{\tau}_n} \leq 1$ and thus we have

$$\| \mathcal{L}aw(\mathcal{X}_{1:p}(i) \in \cdot | \xi(0)) - \check{\pi}_\theta(\cdot) \| \leq \left(1 - \mathbb{E}_{\psi_\theta} [Z_p(\Xi) \left(\rho\varphi \right)^{2 \sum_{n=1}^{p} \bar{\tau}_n}] \right)^i$$

Given [15, Theorem 7.4.2, Equation (7.17), page 239] and the fact that γ_θ is defined to be strictly positive in (A1) we have that the SMC approximation $\hat{Z}_p(\cdot)$ is an unbiased estimate of the normalizing constant Z_p

$$\mathbb{E}_{\psi_\theta} [Z_p(\Xi)] = Z_p, \tag{13}$$

and we can easily conclude. $\qquad\qquad\square$

Proof. [Proof of Proposition 4.2] The proof of parts 1. and 2. follows the line of arguments used in Theorem 4 of [1], which we will adapt to our set-up. The main difference lies in the multi-level construction and second statement regarding the marginal of $\overline{\pi}^N$. For the validity of the multi-level set-up we will rely on Proposition 3.1.

Suppose we design a Metropolis Hastings kernel with invariant density $\overline{\pi}^N$ and use a proposal $q^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) f(k|W_p) \bar{q}(\theta(i-1)|\theta') = \psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) \bar{W}_p^{(k)} \bar{q}(\theta|\theta')$. Then

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

$$= \frac{N^{-p} \overline{\gamma}_p(\mathcal{X}_{1:p}^{(k)}) \overline{p}(\theta) / Z_p}{\mathcal{M}_1(\mathcal{X}_1^{(b_1^k)}) \prod_{n=2}^{p} \mathcal{M}_n(\mathcal{X}_n^{(b_n^k)} | \mathcal{X}_{n-1}^{(b_{n-1}^k)}) \left(\prod_{n=1}^{p} \bar{W}_n^{(b_n^k)} \right) \bar{q}(\theta(i-1)|\theta')}$$

$$= \frac{\overline{\gamma}_p(\mathcal{X}_{1:p}^{(k)}) \left(\prod_{n=1}^{p} N^{-1} \left(\sum_{j=1}^{N} w_n(\mathcal{X}_n^{(j)}) \right) \right) \overline{p}(\theta)}{Z \mathcal{M}_1(\mathcal{X}_1^{(b_1^k)}) \left(\prod_{n=2}^{p} \mathcal{M}_n(\mathcal{X}_n^{(b_n^k)} | \mathcal{X}_{n-1}^{(b_{n-1}^k)}) \right) \left(\prod_{n=1}^{p} w(\mathcal{X}_n^{(b_n^k)}) \right) \bar{q}(\theta|\theta')}$$

$$= \frac{\hat{Z}_p}{Z} \times \frac{\overline{p}(\theta)}{\bar{q}(\theta|\theta')},$$

where we denote the normalising constant of the posterior in (1) as:

$$Z = \int_\Theta Z_p \overline{p}(\theta) d\theta$$

Therefore the Metropolis-Hastings procedure to sample from $\bar{\pi}_p^N$ will be as in Algorithm 4.

Alternatively using similar arguments one we may write

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

Summing over k and using the unbiased property of the SMC algorithm in Equation (13) it follows that $\bar{\pi}_p^N(\cdot)$ admits $\bar{\pi}(\theta)$ as a marginal, so the proof of part 1. is complete.

Part 2. is a direct consequence of Theorem 1 in [3] and Assumption (A2).

$\qquad\qquad\square$

Proof. [Proof of Proposition 4.3] The proof is the similar as that of Proposition 4.2. For the proof of the first statement of part 1. one repeats the same arguments as for Proposition 4.2 with difference being in the inclusion of $\overline{\Lambda}_\theta(v)$ for $\bar{\pi}^N$ and $\bar{q}^N$. For the second statement, to get the marginal of $\overline{\pi}^N$, re-write the target as:

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

Let $\overline{\pi}_p^N(\theta)$ denote the marginal of $\overline{\pi}_p^N(\cdot)$ obtained in Proposition 4.2. Using (11) and the conditional independence of v and $\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}$, then for the marginal of $\overline{\pi}^N(\cdot)$ w.r.t v, $\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}$, k we have that

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

where the summing over k and integrating w.r.t. $\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}$ is as in Proposition 4.2.

For part 2. note that the conditional density 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 $\left(\theta(i), \mathcal{X}_{1:p(v)}^{(k)}(i)\right)_{i \geq 0}$ satisfies the required property as direct consequence Theorem 1 in [3] and Assumption (A2). $\qquad\square$

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. & 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, arXiv:1011.1356v1.

[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. Appl.*, **25**, 417–433.

[9] CEROU, F., DEL MORAL, P., FURON, T. & GUYADER, A. (2011). Rare event simulation for a static distribution, *Stat. Comput.*, to appear.

[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., JACOB, P. & PAPASPILIOPOULOS, O. (2011). SMC^2: A sequential Monte Carlo algorithm with particle Markov chain Monte Carlo updates. Technical Report, ENSAE, arXiv:1101.1528v2.

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

[14] DE IORIO, M. & GRIFFITHS, R. C. (2004). Importance sampling on coalescent histories. II: Subdivided population models. *Adv. Appl. Probab.* **36**, 434–454.

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

[16] DEL MORAL, P., DOUCET, A. & JASRA, A. (2006). Sequential Monte Carlo samplers. *J. R. Statist. Soc. B*, **68**, 411–436.

[17] DEL MORAL, P. & GARNIER, J. (2005). Genealogical Particle Analysis of Rare events. *Ann. Appl. Prob.*, **15**, 2496–2534.

[18] DOUCET, A., DE FREITAS, J. F. G. & GORDON, N. J. (2001). *Sequential Monte Carlo Methods in Practice.* Springer: New York.

[19] GLASSERMAN, P., HEIDELBERGER, P., SHAHABUDDIN, P. & ZAJIC, T. (1999). Multilevel splitting for estimating rare event probabilities. *Oper. Res.*, **47**, 585–600.

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

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

[22] 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.

[23] KANTAS, N., DOUCET, A., SINGH, S. S., MACIEJOWSKI, J. M. & CHOPIN, N. (2011). On particle methods for parameter estimation in general state-space models, submitted.

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

[25] 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.*, **19**, 769–789.

[26] 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.

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

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

[29] ROBERTS, G. O. & ROSENTHAL J. S. (2011). Quantitative Non-Geometric convergence bounds for independence samplers. *Meth. Comp. Appl. Prob.*, **13**, 391–403.

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

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

[32] 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.

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