Bayesian Parameter Inference for Partially Observed Stopped Processes

BY AJAY JASRA[1], NIKOLAS KANTAS[2] & ADAM PERSING[3]
[1]Department of Statistics & Applied Probability, National University of Singapore, Singapore, 117546, SG.
E-Mail: *staja@nus.edu.sg*
[2]Department of Statistical Science, University College London, London, W1CE 6BT, UK.
E-Mail: *nikolas@stats.ucl.ac.uk*
[3]Department of Mathematics, Imperial College London, London, SW7 2AZ, UK.
E-Mail: *a.persing11@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 sets from B_0 to A and then performing the resampling step only when the samples of the process reach intermediate sets in the sequence. The choice of the intermediate 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 introduce a flexible strategy that adapts the 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 a given set A. These processes appear in a wide range of applications, such as population genetics [14, 24], finance [7], neuroscience [5], physics [16, 22], and engineering [6, 26]. The vast majority of the papers in the literature deal with fully observed stopped processes and assume that the parameters of the model are known. In this paper, we address problems when this is not the case. In particular, we consider Bayesian inference for the model parameters, 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 therein. Recently, sequential Monte Carlo (SMC) methods based on both these techniques have been used in [6, 16, 22]. In [10], the authors also prove under mild conditions that SMC can achieve the same performance as popular competing methods based on traditional splitting.

Sequential Monte Carlo methods can be described as a collection of importance sampling and resampling techniques used to approximate a sequence of distributions whose densities are known point-wise up to a normalising constant and are of increasing dimension. 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; without resampling the variance typically increases exponentially as the target sequence progresses, e.g. [18, 27]. Resampling often occurs by sampling N times from the current particles with replacement, with probabilities proportional to the current importance weights. The importance weights are then reset to $1/N$.

In the context of stopped processes with fixed parameters, as remarked above, SMC provides an efficient simulation scheme for e.g. approximating integrals w.r.t. the law of the process (the process can be partially observed, but this is not a requirement). In addition, as noted by [11, 15], efficient SMC algorithms should use resampling when taking into account each sample's proximity 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 (see [11] for an explanation). 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 reached intermediate sets; this is termed multi-level SMC. These sets define a sequence from B_0 to A (for examples, see Section 3.2.1). The same idea appeared in parallel in [15, Section 12.2], where it was formally interpreted as an interacting particle approximation of multi-level Feynman-Kac formulae. The choice of the intermediate sets is critical to the performance of such a scheme. The sets (which are level sets in the rare-event literature - hence 'multi-level') should be set in a "direction" towards the set A 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, the authors in [13] use optimal control. In [8, 9], the authors use the level sets that are computed adaptively on the fly using the simulated paths of the process.

The contribution of the article is to develop computational methodolgy to infer a posterior distribution on a collection of parameters associated to a model with observed data whose likelihood depends upon an unobserved stopped Markov process. Employing standard Markov chain Monte Carlo (MCMC) methods is not feasible, given the difficulties discussed above for sampling trajectories of the stopped process. In addition, using SMC for parameter inference associated to hidden Markov models has been notoriously difficult; see [2, 23]. 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. For parameter inference associated to stopped processes, this brings up the possibility of using the multi-level SMC methodology as a proposal in MCMC. To the best of our knowledge, this idea has not previously appeared in the literature.

The main contributions made in this article are as follows:

- When the sequence of 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 sets are adapted to the current parameter sample. The method is shown to produce unbiased samples from the target posterior density. We show, via numerical examples, how the mixing of the PMCMC algorithm can be 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 sets are proposed. These ideas 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 as $\mathscr{P}(E)$. For $\mathbb{R}^n$ with $n \in \mathbb{N}$, the Borel sets are $\mathscr{B}(\mathbb{R}^n)$. For a probability measure $\gamma \in \mathscr{P}(E)$, the density with respect to an appropriate σ-finite measure dx is denoted 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 $i > j$, then $x_{i:j}$ 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 state-space and define

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

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 $d \in \mathbb{Z}^+$, $\mathbb{T}_d := \{1, \ldots, d\}$.

2 Problem Formulation

2.1 Preliminaries

Let θ be a parameter 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, 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 with 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 $A \in \mathscr{F}$, such that $\mathbb{P}_\theta(X_0 \in A) = 0$. 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 Y is denoted as y and we assume that it takes value in some non empty set F. The set A can depend upon y, but to simplify exposition, this is 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

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

Unless stated otherwise, the subscript on θ will be used throughout to explicitly denote the dependence 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 $\theta \in \Theta$ and $y \in F$, $\tau \in \mathcal{I}$ γ_θ admits a density $\overline{\gamma}_\theta(x_{0:\tau}, y, \tau)$ w.r.t. a $\sigma-$finite measure $dx_{0:\tau}$ on $E^{\tau+1}$ (using the notation $\overline{E} = \left(\bigcup_{\tau\in\mathcal{I}}\{\tau\} \times E^{\tau+1}\right))$ and the posterior and prior distributions π, p admit densities $\overline{\pi}$, $\overline{p}$ respectively both defined w.r.t. some 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 an MCMC that can sample from π, which requires one to write the target (or an unbiased estimate of it) up-to a normalising constant [3]. Still, our primary interest lies in Bayesian inference for the parameter and this can be recovered by the marginals of π w.r.t. θ.

2.2 Motivating example: the coalescent

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

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 $x_t = (x_t^1, \ldots, x_t^d)$. The process begins by default when the first split occurs, so the Markov chain $\{X_t\}_{t\geq 0}$ is initialised by the density

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

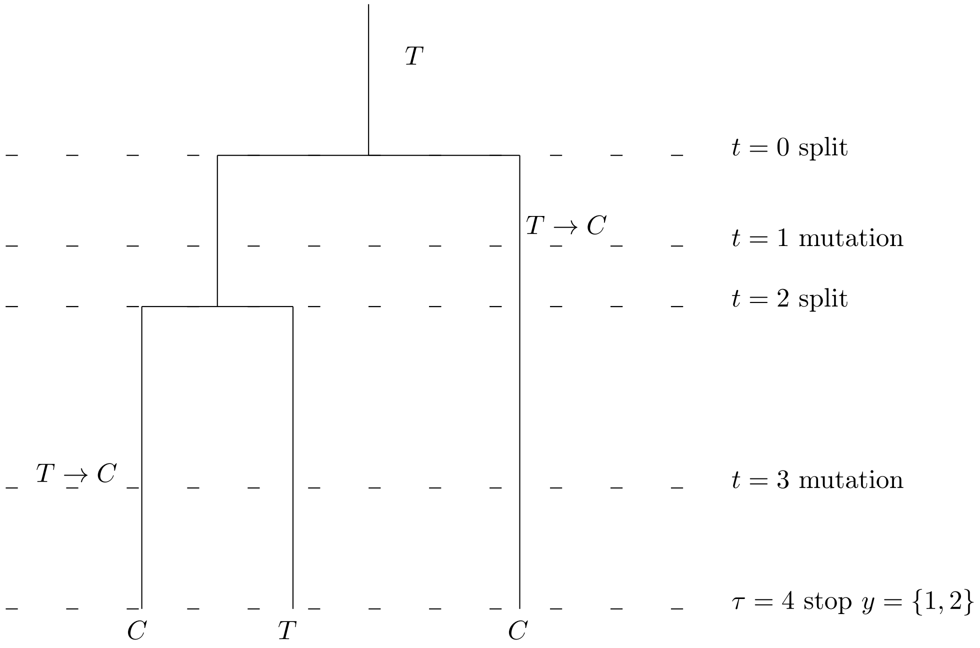

Figure 1: Coalescent model example: each of $\{C, T\}$ denotes the possible genetic type of observed chromosomes. In this example, we have $d = 2$ and $m = 3$. The tree propagates forward in time from the MRCA downwards by a sequence of split and mutation moves. Arrows denote a mutation of one type of 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.

and is propagated using the following transition density:

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

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. $T \to C$ 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 MRCA 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

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

and for the initial and terminal sets, we have

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

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

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

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 with time reversed. 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 backward 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., there we posed the problem backward 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 as

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

In [31], the authors derive an optimal proposal $\overline{M}_\theta$ w.r.t. the variance of the marginal likelihood estimator. For the sake of brevity, we omit any further details. In the current set up 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

We shall briefly introduce generic SMC in this section without extensive details; see [15, 18] for a more comprehensive account. To ease exposition, when presenting generic SMC, we drop θ from the notation.

SMC algorithms are designed to simulate from a sequence of probability distributions $\pi_1, \pi_2, \dots, \pi_p$ defined on a state space of increasing dimension, namely $(G_1, \mathscr{G}_1), (G_1 \times G_2, \mathscr{G}_1 \otimes \mathscr{G}_2), \dots, (G_1 \times \cdots \times G_p, \mathscr{G}_1 \otimes \cdots \otimes \mathscr{G}_p)$. Each distribution in the sequence is assumed to possess densities w.r.t. an appropriate 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 normalising 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

Algorithm 1 Generic SMC algorithm

```
Initialisation, $n = 1$:
    For $i = 1, \dots, N$,
    1. Sample $u_1^{(i)} \sim \overline{M}_1$ .
    2. Compute the 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 $n = 2, \dots, p$,
    For $i = 1, \dots, N$,
    1. Resampling: sample index $a_{n-1}^i \sim f(\cdot|\bar{W}_{n-1})$, where $\bar{W}_{n-1} = (\bar{W}_{n-1}^{(1)}, \dots, \bar{W}_{n-1}^{(N)})$.
    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 the weights
        $W_n^{(i)} = w_n(u_{1:n}^{(i)}) = \frac{\overline{\gamma}_n(u_{1:n}^{(i)})}{\overline{\gamma}_{n-1}(u_{1:n-1}^{(i)})\overline{M}_n(u_n^{(i)}|u_{1:n-1}^{(i)})}, \; \bar{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^N W_n^{(j)}}.$
```

of the algorithm, at each step n of the algorithm, we will use general proposal kernels M_n with densities $\overline{M}_n$. The densities possess normalising constants that do not depend on the simulated paths. Algorithm 1 provides the following SMC approximations for π_n,

$$\pi_n^N(du_{1:n}) = \sum_{j=1}^N \bar{W}_n^{(j)} \delta_{u_{1:n}^{(j)}}(du_{1:n}),$$

and for the normalising 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 (see Algorithm 1). Then the resampled index of the ancestor of particle i at time n, namely $a_{n-1}^i \in \{1, \dots, N\}$, 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 $\bar{\mathbf{a}}_n = (a_n^1, \dots, a_n^N)$ and the randomly simulated values of the state obtained during sampling (step 2 for $n \geq 2$) as $\bar{u}_n = (u_n^{(1)}, \dots, u_n^{(N)})$. We will also denote by $\bar{\mathbf{a}}_{1:p}, \bar{u}_{1:p}$ the concatenated vector of all these variables obtained during the simulations from time $n = 1, \dots, p$. Note that $\bar{u}_{1:p}$ is a vector containing all $N \times p$ simulated states and should not be confused with the particle sample of the path $(u_{1:p}^{(1)}, \dots, u_{1:p}^{(N)})$.

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

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

The complete ancestral genealogy at each time can be 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 later be 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 here a modified approach along the lines of [15, Section 12.2] which seems better suited for general stopped processes and can provably yield estimators of much lower variance relative to generic SMC.

Introduce an arbitrary sequence of $\mathscr{F}-$sets $B_1, \dots, B_p$, with $B_p = A$, with stopping times

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

such that $0 \leq \mathcal{T}_1 \leq \cdots \leq \mathcal{T}_p = \mathcal{T}$. For example, if $B_0 \supset A$

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

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 idea of these sets is that, in some way, they interpolate between B_0 and A.

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 the following for $n = 1$:

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

where τ_1 is a realisation for the stopping time T_1. Similarly, for $2 \leq n \leq p$, we have

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

Multi-level SMC is an 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 $n \in \mathbb{T}_p$ with $p \geq 2$ 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, we define intermediate target densities $\overline{\pi}_n$ w.r.t. 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 is 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 $\overline{E}_n$, with $\overline{E}_n$ as in* (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, when $B_0 \supset B_1 \cdots \supset B_p = A$, can be found in [15, Proposition 12.2.2, page 438], [15, Proposition 12.2.4, page 444] (and is easily extended to our scenario) and the second part is due to $\overline{\gamma}_p = \overline{\gamma}_\theta$. The result simply establishes the Markov property of the sequence $(\mathcal{X}_n)_{1 \leq n \leq p}$ and an identity for moving between the $(\mathcal{X}_n)_{1 \leq n \leq p}$ representation and the original one we have defined from the start of the article; these ideas will be used later on in the paper.

We will present multi-level SMC based as a particular implementation of the generic SMC algorithm. 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

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

and for step 2 at $n = 1$, we use

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

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 $2 \leq n \leq p$,

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

where again we have suppressed the θ-dependence of $\mathcal{M}_n$ in the notation. The multi-level SMC algorithm is in Algorithm 2.

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

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

where $\bar{\mathcal{X}}_{1:p}$ is defined similarly to $\bar{u}_{1:p}$. Finally, recall that, by construction, $Z_p = Z_\theta$. So, the approximation of the normalising 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 Constructing the sets

We will begin by showing how the sets can be constructed 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 and progress towards the root. We introduce 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, as the process backward in time is a sort of death process, we have $\mathcal{T}_{n-1} \leq \mathcal{T}_n$, $1 \leq n \leq p$ and $B_p = A$. One can also write the sequence of target densities for the multi-level setting as:

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

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

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

Algorithm 2 Multi-level SMC algorithm

```
Initialisation, $n = 1$:
    For $i = 1, \dots, N$,

    1. Starting from $t = 1$, iterate until $x_t^{(i)} \in B_1$:

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

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

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

    2. Starting from $t = \tau_{n-1}^{(i)} + 1$, iterate until $x_t^{(i)} \in B_n$:

        (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 the weights
           $W_n^{(i)} = w_n(\mathcal{X}_{1:n}^{(i)}) = \frac{\overline{\gamma}_n(\mathcal{X}_{1:n}^{(i)})}{\overline{\gamma}_{n-1}(\mathcal{X}_{1:n-1}^{(i)}) \mathcal{M}_n(\mathcal{X}_n^{(i)} | \mathcal{X}_{1:n-1}^{(i)})}, \; \bar{W}_n^{(i)} = \frac{W_n^{(i)}}{\sum_{j=1}^N W_n^{(j)}}.$
```

Algorithm 3 Particle independent Metropolis-Hastings 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 $\xi(0) = \big(k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)\big) = \big(k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}\big)$ and $\widehat{Z}_p(0) = \widehat{Z}_p$.
3. For $i = 1, \dots, K$:
    (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 $\xi(i) = \big(k(i), \bar{\mathcal{X}}_{1:p}(i), \bar{\mathbf{a}}_{1:p-1}(i)\big) = \big(k', \bar{\mathcal{X}}'_{1:p}, \bar{\mathbf{a}}'_{1:p-1}\big)$ and $\widehat{Z}_p(i) = \widehat{Z}'_p$. Otherwise reject, $\xi(i) = \xi(i-1)$ and $\widehat{Z}_p(i) = \widehat{Z}_p(i-1)$.
```

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 generic SMC algorithm. Whilst the authors of [11] show empirically that in most cases one should expect a considerable improvement, their θ is considered to be fixed. In this case, one could design the levels sensibly using e.g. offline heuristics or more advanced systematic methods. What we aim to establish in the next section is that, when θ varies as in the context of MCMC algorithms, one can both construct PMCMC algorithms based on multi-level SMC and, more importantly, easily design different sequences for $\{B_n\}$ for each θ based upon the afore mentioned ideas.

4 Multi-level Particle Markov Chain Monte Carlo

MCMC methods can perform poorly in situations where the proposal distribution is not chosen well. When a target is complex or of high dimension, choosing an appropriate proposal distribution can be difficult. Particle Markov chain Monte Carlo (PMCMC) methods [1] attempt to alleviate this problem by combining SMC and MCMC. Essentially, one defines a target distribution with an extended state space such that a marginal of this extended target is the distribution of interest. Then one runs an SMC algorithm to approximate an appropriate proposal distribution, and this approximation of the proposal is used in an MCMC algorithm as usual. The authors of [1] introduce three different and generic PMCMC algorithms: particle independent Metropolis-Hastings (PIMH), particle marginal Metropolis-Hastings (PMMH), and particle Gibbs (PG). In the remainder of this paper, we will only focus on the first two of these samplers.

This section aims to provide insight to the following questions:

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

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

The answer to the first question seems rather obvious, and so we will provide some standard conditions for which multi-level PMCMC is valid. We will first consider a simple multi-level PIMH, both to introduce the reader to PMCMC and to begin our investigation of the first question. We will then graduate to a multi-level PMMH to be used for parameter inference and prove some convergence results. For the second question, we will propose an extension to our multi-level PMMH that adapts the sets used at every iteration of PMCMC to yield a more efficient algorithm.

4.1 Introduction to multi-level PMCMC

We will begin by presenting PIMH. PIMH is not useful for parameter inference, but it is the most basic of all of the PMCMC algorithms and offers a good introduction to PMCMC methods for the unfamiliar reader. Furthermore, the algorithm's relative simplicity makes it straightforward to analyse; insights gained on PIMH may help us to gain intuition for more complex PMCMC algorithms.

In PIMH, θ and p are fixed and the algorithm samples from the pre-specified target distribution π_p (see Algorithm 3). Recall from Section 3.1 that π_p is the ultimate target of an SMC algorithm. It can be shown, using similar arguments to [1], that the invariant density of PIMH is exactly

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

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 (see the proof of Proposition 4.2). Note that $\overline{\pi}_p^N$ admits the target density of interest, $\overline{\pi}_p$, as a marginal when k and $\bar{\mathbf{a}}_{1:p-1}$ are integrated out.

We briefly investigate some convergence properties of PIMH with multi-level SMC. We begin with posing the following mixing and regularity assumption:

(**A1**) For every $\theta \in \Theta$ and $p \in \mathcal{I}$, there exists 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 exists 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 exists a $\bar{\tau}_n < \infty$ such that

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

Assumption (A1) is rather strong, but is often used in the analysis of these kinds of SMC algorithms [1, 15]. The assumption on $\overline{\gamma}_n(\mathcal{X}_{1:n})$ is a type of growth condition that can be satisfied when E is compact. Recall that $\theta \in \Theta$ and $p \in \mathcal{I}$ are fixed. We have the following proposition, whose proof can be found in the appendix:

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 $i \geq 1$, $\xi(0) \in \mathbb{T}_N^{(p-1)N+1} \times \overline{E}$, and $\theta \in \Theta$, satisfies*

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

Remark 4.1. *Proposition 4.1 shows intrinsically that as the supremum of the sum of the stopping times w.r.t. $\{B_n\}_{n=1}^p$ gets smaller, so does the convergence rate increase. The point of the result is that for efficient algorithms, one must try to control the length of the stopping times. For variable θ one can envisage that different sequences of sets are better for different values of θ, in the sense that the expected (intermediate) stopping times are shorter. Thus, we will attempt to obtain sets $B_1, \dots, B_p$ which are adapted to the value of θ: this will potentially yield shorter stopping times and faster-rates of convergence than when there is no adaptation (as suggested by Proposition 4.1 in a simplified scenario).*

4.2 Multi-level particle marginal Metropolis-Hastings

In the remainder of this section, we will focus on using a multi-level SMC implementation within a PMMH algorithm. Given the commentary in Sections 3.2, and given our interest in drawing inference on $\theta \in \Theta$, it seems that using multi-level SMC within PMMH should be highly beneficial.

Recall that (1) can be expressed in terms of densities as

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

where we let the marginal density be given by

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

In the context of our stopped Markov process, we propose a PMMH algorithm which targets $\overline{\pi}(\theta, \mathcal{X}_{1:p})$ in Algorithm 4. For the time being, we will consider the case when p is fixed. Note that Algorithm 4 is 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 sampler [32].

We will establish the invariant density and convergence of Algorithm 4 under the following assumption:

Algorithm 4 Particle marginal Metropolis-Hastings (PMMH) 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 $\xi(0) = \big(\theta(0), k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)\big)$ 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 $\theta'$, 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
       $1 \wedge \dfrac{\widehat{Z}'_{\theta'} \bar{p}(\theta')}{\widehat{Z}_\theta(i-1) \bar{p}(\theta)} \times \dfrac{\bar{q}(\theta(i-1)|\theta')}{\bar{q}(\theta'|\theta(i-1))}.$
       If we accept, set $\xi(i) = \big(\theta(i), k(i), \bar{\mathcal{X}}_{1:p}(i), \bar{\mathbf{a}}_{1:p-1}(i)\big) = \big(\theta', k', \bar{\mathcal{X}}'_{1:p}, \bar{\mathbf{a}}'_{1:p-1}\big)$ and $\widehat{Z}_\theta(i) = \widehat{Z}'_{\theta'}$. Otherwise
       reject, $\xi(i) = \xi(i-1)$ and $\widehat{Z}_\theta(i) = \widehat{Z}_\theta(i-1)$.
```

(**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 $Q_n^\theta = \{\mathcal{X}_{1:n} \in \overline{E}_n : \gamma_{n-1}(\mathcal{X}_{1:n-1})\mathcal{M}_{n,\theta}(\mathcal{X}_n|\mathcal{X}_{n-1}) > 0\}$. For any $\theta \in \Theta$, we have that $S_n^\theta \subseteq Q_n^\theta$. In addition, the ideal Metropolis-Hastings targeting $\overline{\pi}(\theta)$ via proposal density $q(\theta'|\theta)$ is irreducible and aperiodic.

This assumption contains Assumptions 5 and 6 of [1], modified to our problem with some simple changes of notation. It allows the Hastings ratio to be a well-defined Radon-Nikodym derivative and the ergodicity of the Markov kernel to be established. 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*

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

where $\overline{\pi}$ is as in (9) *and ψ_θ is 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)*.*

This result is based on Theorem 4 of [1], and its proof can be found in the Appendix. In the next section, we propose a flexible scheme that allows us to set a different number of levels after a new θ' is proposed.

4.3 Adapting the sets

The remaining design issue for multi-level PMMH is to include a scheme for tuning 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, this is not an option when θ varies. The value of θ should dictate how small or large p should be to facilitate an efficient SMC. Hence, to obtain a more accurate estimate of the marginal likelihood (and thus an efficient PMCMC algorithm), we will consider adaptive strategies to randomly construct a different number of sets p and sequence $\{B_n\}_{n=1}^p$ for each $\theta(i)$ sampled at every PMMH iteration i. To ease exposition, we will assume that p and $\{B_n\}_{n=1}^p$ can be expressed as functions of an arbitrary auxiliary parameter, v.

Given that $\theta(i)$ is a random variable, we have to determine how to perform our adaptive update consistently (that is, to provide a statistically consistent algorithm). 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 be $\overline{\pi}(\theta)$. This can be

Algorithm 5 Particle marginal Metropolis-Hastings (PMMH) using multi-level SMC with adaptive 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 $\xi(0) = \big(\theta(0), v(0), k(0), \bar{\mathcal{X}}_{1:p}(0), \bar{\mathbf{a}}_{1:p-1}(0)\big)$ 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 $\xi(i) = \big(\theta(i), v(i), k(i), \bar{\mathcal{X}}_{1:p(v(i))}(i), \bar{\mathbf{a}}_{1:p(v(i))-1}(i)\big) = \Big(\theta', v', k', \bar{\mathcal{X}}'_{1:p(v')}, \bar{\mathbf{a}}'_{1:p(v')-1}\Big)$ and
       $\widehat{Z}_\theta(i) = \widehat{Z}'_{\theta'}$. Otherwise reject, $\xi(i) = \xi(i-1)$ and $\widehat{Z}_\theta(i) = \widehat{Z}_\theta(i-1)$.
```

ensured (see Proposition 4.3) by introducing, at each PMMH iteration, the parameters that form the sets $v(i)$ as an auxiliary process. This auxiliary process must be conditionally independent of $k(i), \bar{\mathcal{X}}_{1:p}(i)$, and $\bar{\mathbf{a}}_{1:p-1}(i)$, given $\theta(i)$. Thus, we define an extended target for the PMCMC 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. In contrast, the algorithm presented here relies on an augmentation of the target space with more auxiliary variables.

At every PMMH iteration i, we will simulate the auxiliary process v defined upon an abstract state-space $(V, \mathscr{V})$. Let this auxiliary process, with associated random variable v, be distributed according to Λ_θ. Λ_θ is assumed to possess a density w.r.t. 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 sets $p(v) \in \mathcal{J} \subset \mathbb{Z}_+$ and
- a sequence of 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 will hold where p is replaced by $p(v)$. This implies that for every $\theta \in \Theta$ we have

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

where the expression holds $\Lambda_\theta-$ almost everywhere.

Now, in Algorithm 5, we can propose a multi-level PMMH algorithm that uses $\theta(i)$ to adapt the levels $\{B_n(v(i))\}_{n=1}^{p(v(i))}$ at each step i. The following proposition verifies that varying the 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

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

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]. Similar to (1), we include $x_{0:\tau}$,τ,v,$\bar{\mathcal{X}}_{1:p(v)}$, and $\bar{\mathbf{a}}_{1:p(v)-1}$ in the target posterior, even though we are primarily interested in θ. This augmentation is a consequence of using PMCMC. The disadvantage here is that as the space of the posterior increases, it is expected that the mixing of the algorithm will slow down. This could have been improved if we had opted for $x_{0:\tau}, \tau$, and v to be dependent on each other given θ, but this would require additional assumptions for the structure of γ_θ. Plus, the parameter v that determines $\{B_n\}_{n=1}^{p}$ often appears naturally and is low dimensional in many applications. Also in many applications, it might seem easier to find intuition on how to construct and tune Λ_θ than to compute the sets directly from θ. For example, for the coalescent model of Section 2.2 with fixed mutation matrix R, one can envisage that for a larger value of μ, coalescent events are less likely. So, more sets which are close together are needed, compared to the instance where a value of μ is small.

5 Numerical Examples

We will consider two examples from population genetics. The first example is a coalescent model with a low dimensional observed dataset, as presented in Section 2.2. This is a toy example for comparing different PMMH implementations. The second example is a more realistic application. It is a coalescent model that allows for the migration of individual genetic types from one sub-group to another [4, 14].

We will illustrate the performance of PMMH on these two models. In both cases, we will implement PMMH using adaptive multi-level SMC and generic SMC. By generic SMC, we mean an SMC algorithm that does not take into account the structure of the latent process and will resample after every genetic event (time-point of the algorithm). For the multi-level SMC algorithms, we will utilize a simple intuitive strategy for adapting the sets. Additionally, for the first example, we will implement PMMH using fixed multi-level SMC (i.e. not adapting the sets).

5.1 The coalescent model

We will use a known stochastic matrix R with all entries equal to $1/d$. In this example, $d = 4$ and the dataset is $y = (10, 5, 9, 5)$. We will attempt to infer $\mu \in \Theta = [0, 1.5]$, and μ will have a uniform prior.

5.1.1 Implementation details

We use the approach for simulation detailed in Section 2.2.1; that is, simulating the hidden process backward in time. For M_θ, the optimal proposal of [31] is used.

We implemented several different multi-level SMC schemes within PMMH to thoroughly compare Algorithms 4 and 5. We began by employing four versions of Algorithm 4, where p is fixed and can lie, for different implementations, in the set $\{8, 14, 21, 28\}$. We then employed two versions of Algorithm 5. We allowed p to be chosen from the range of four values $\{8, 14, 21, 28\}$ and from the range of twenty-one values $\{8, 9, \ldots, 28\}$. In an effort to compare Algorithms 4 and 5 to a standard method, we also implemented a PMMH that used generic SMC. This version of SMC used no interpolating sets and resampled after every genetic event.

In both the fixed and adaptive multi-level SMC, we always placed the levels (recall section 3.2.1) almost equally spaced apart for a given value of p. When an adaptive strategy was employed, we sampled p directly using a multinomial distribution defined on its range, with probabilities proportional to μ^p. In all of the algorithms described in this section, the PMMH proposal was a log normal random walk (i.e., we used $\zeta' = \zeta(i-1) + 0.4\mathcal{N}(0, 1)$ with $\zeta = \log(\mu)$).

We ran all algorithms with $N \in \{50, 100, 200\}$ for 10^5 iterations. All simulations were repeated twenty times to verify the reproducibility of the new algorithms' output. For $N = 50$, the algorithms required approximately two to four hours to complete, while for $N = 100$ and 200, the algorithms required approximately four to eight hours and eight to 16 hours, respectively. Those algorithms that resampled the least fell on the shorter end of the spectrum. The algorithms were implemented in MATLAB 7.14 R2012a and run on a Windows desktop using an Intel Core i7-2600 CPU at 3.40 GHz.

5.1.2 Numerical results

The ACFs for the simulations of Algorithms 4 and 5 illustrate fast mixing (see Figures 2, 3, and 4). Algorithm 5 exhibited slightly faster mixing when p was chosen from a smaller range, and this is expected because expanding the range of p also increases the size of the state space of the extended target of the PMMH. Furthermore, the trace plots show that both algorithms yield low auto-correlations of μ. Increasing values of N did yield slightly faster convergence rates in Algorithm 4. In Algorithm 5, the rate of mixing was approximately constant given N.

Examining the algorithms' estimates of the density of μ given the data highlights their more material differences (see Figures 2, 3, 4, 5, and 6). Algorithm 4 gave a consistent density of the likelihood for all fixed values of p, regardless of the value of N. In all cases, the density appears to have a single mode close to zero. However, Algorithm 5 revealed a different mode when p was allowed to be chosen from the range $\{8, 14, 21, 28\}$. In that case, the density appears to have a single mode close to one. When we allowed p to be chosen from the larger range $\{8, 9, \ldots, 28\}$, Algorithm 5 was able to find both modes simultaneously (see Figure 4). This interpretation is supported by Figure 5 as a finer distribution on p (for the case $p \in \{8, 9, ..., 28\}$) seems to facilitate the movement between these afore-mentioned modes. It is also remarked that these results are reasonably consistent across the repititions.

The above output illustrates, for this example, that adapting p allows PMMH using multi-level SMC to better traverse the state space compared to when a fixed number of levels is used. To obtain further confidence that a non-adaptive algorithm is insufficient for the example at hand, we implemented seven more versions of Algorithm 4 with p fixed at various other values in the range of eight to 28. Each algorithm was only run once and with $N = 50$. In all instances, the same single mode estimate of the likelihood was obtained.

Furthermore, the size of the range of p has a clear impact on the performance of Algorithm 5. The two implementations of Algorithm 5 seem to suggest that larger ranges for p yield more efficient PMMH, possibly because it is easier to tailor SMC to changing values of θ through a more refined range for p.

PMMH using generic SMC failed in this example for $N = 50, 100,$ and 200, and so we do not present the output here. After 10^5 iterations, PMMH using generic SMC does not come close to converging. The ACFs never drop below 0.8, and the average acceptance rates were on the order of 0.02% to 0.4%. The trace plots also confirm these former points.

5.2 The coalescent model with migration

This model is similar to the one described in Section 2.2. The major difference is that, in this model, genetic types are classified into sub-groups within which most activity occurs. Additionally, individuals are allowed to migrate from one sub-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. We assign g to be the number of sub-groups, and the state at time t is composed as the concatenation of g sub-groups of different genetic types:

$$x_t = (x^1_{1,t}, \ldots, x^d_{1,t}, \ldots, x^1_{g,t}, \ldots, x^d_{g,t}).$$

The process undergoes 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 $\alpha, \beta \in \{1, \ldots, g\}$, $\alpha \neq \beta$, and $e_{\alpha,i}$ is a vector with a zero in every element except the $(\alpha - 1)g + i$ -th one. Similar to the simpler model of Section 2.2, the transition probabilities are parameterised by the mutation parameter μ, the mutation matrix R, and the 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 at time τ, where the number of individuals in the population reaches m (i.e., $y = y^{1:gd} = x_\tau$).

In our example, we generated data with $m = 100$, $d = 64$, and $g = 3$ (see Figure 7). This is quite a challenging set-up. We set the mutation matrix R to be known and uniform, and we will concentrate on inferring $\theta = (\mu, G)$. Independent gamma priors with shape and scale parameters equal to one are adopted for each of the parameters.

5.2.1 Implementation details

As for the model described in Section 2.2, one can reverse time and employ a backward-sampling importance sampling method; see [14] for the particular implementation details.

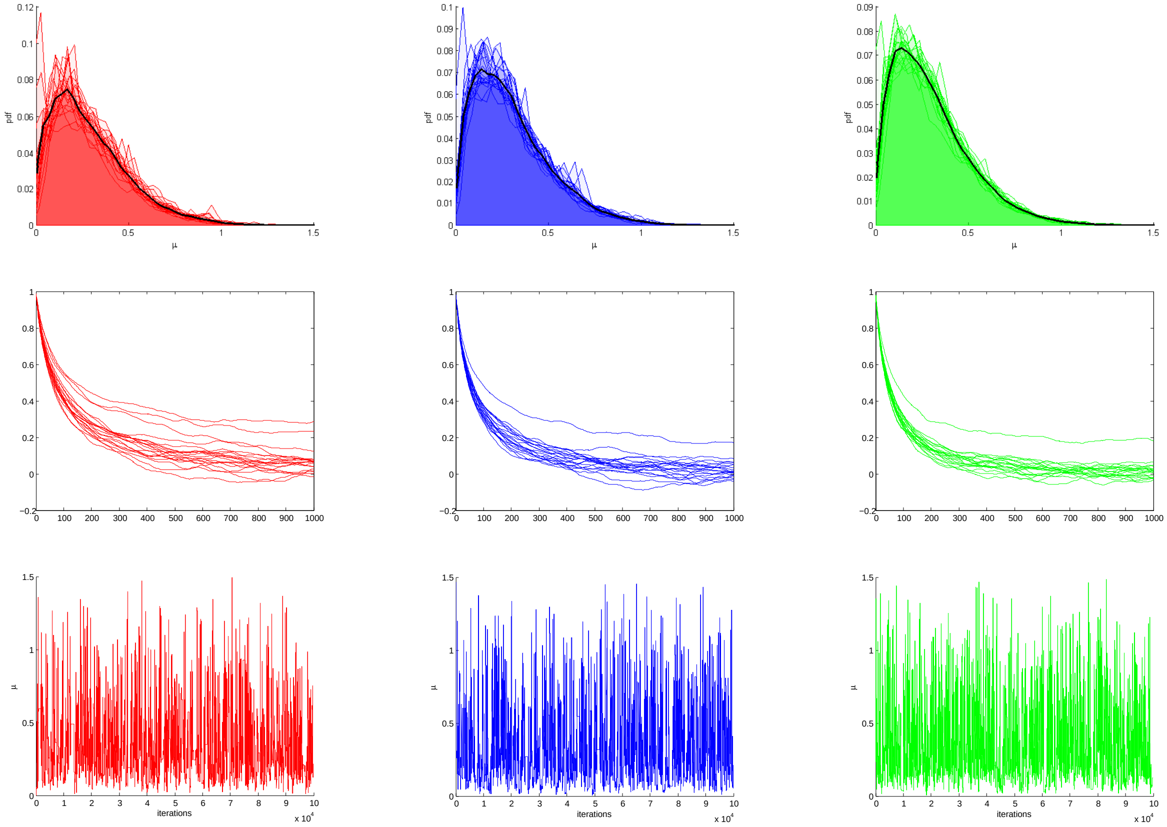

Figure 2: PMMH output for the coalescent model: Algorithm 4. Due to space constraints, we only present the $p = 8$ output for $N = 50$ (red), 100 (blue), 200 (green) from left to right. Top: estimated pdf of μ. We plot 20 repetitions with the average estimate printed in black. Middle: autocorrelation functions. Bottom: trace plots for one repetition. For $N = 50, 100,$ and 200, the average acceptance ratio is on the order of 0.07, 0.09, and 0.10, respectively.

We consider three versions of PMMH that used adaptive multi-level SMC (i.e. Algorithm 5). For the three versions we had $p \in \{10, 20, 33\}$, $p \in \{10, 16, 21, 27, 33\}$ and $p \in \{10, 13, 16, 19, 21, 24, 27, 30, 33\}$. In all versions, we used approximately equal spacing between the levels and we chose p with probability proportional to $p^{\log\{\mu+\sum_{i>j} G_{ij}+1\}}$. We also implemented a PMMH that used generic SMC (i.e., we resampled after every genetic event). For all four versions of PMMH, we set the proposals for the parameters to be Gaussian random-walks on the log-scale.

We also varied the number of particles used in the multi-level and generic SMC; we ran each version of PMMH for $N \in \{10, 50, 100, 200\}$ for 10^5 iterations. For PMMH using generic SMC, we additionally ran a batch of simulations that each iterated through 5×10^5 steps. To get a sense of the reproducability of the algorithms' results, every simulation was repeated ten times.

The algorithms were implemented in C++, and we ran the simulations on a Linux workstation that used an Intel Core 2 Quad Q9550 CPU at 2.83 GHz. When using multi-level SMC with $p \in \{10, 20, 33\}$, the running time for each algorithm was approximately 36, 180, 360, and 720 minutes for $N = 10, 50, 100,$ and 200, respectively. When we allowed $p \in \{10, 16, 21, 27, 33\}$, the running times increased by about 50% in each case, as opposed to a 100% increase when we allowed $p \in \{10, 13, 16, 19, 21, 24, 27, 30, 33\}$. Whilst the running times are quite long, they can be improved by at least one order of magnitude if the SMC is implemented on Graphical Processing Units (GPUs), as in [25].

5.2.2 Numerical results

In Figure 7 one can see the histograms of p for each of the three versions of PMMH, which indicates that a larger value of p is preferable (this was consistent across the repititions). However, even due to the differences in the adaptive schemes, they gave consistent output regardless of the way in which we chose p. Across each of the three

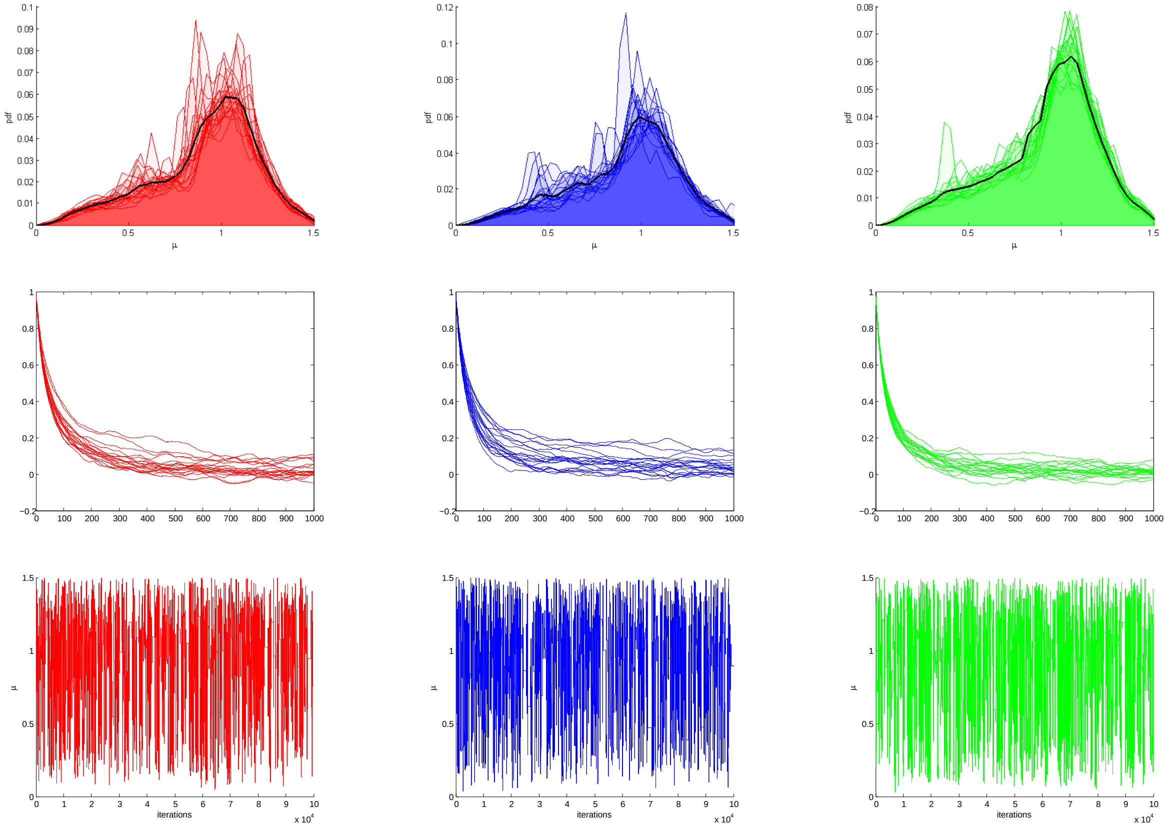

Figure 3: PMMH output for the coalescent model: Algorithm 5. We allow $p \in \{8, 14, 21, 28\}$ and present the output for $N = 50$ (red), 100 (blue), 200 (green) from left to right. Top: estimated pdf of μ. We plot 20 repetitions with the average estimate printed in black. Middle: autocorrelation functions. Bottom: trace plots for one repetition. For $N = 50, 100,$ and 200, the average acceptance ratio is on the order of 0.07, 0.07, and 0.08, respectively.

versions of PMMH using multi-level SMC, the estimated density of θ appears to be constant across the algorithms (see Figure 8), and the ACF and trace plots demonstrate quick and efficient mixing (see Figures 9 and 10). The algorithm converged a bit more slowly as we increased the size of the range of p, much like in the first example in Section 5.1. Given the algorithms' ability to mix so well, we find the results to be consistent for $N \in \{10, 50, 100, 200\}$ and to be reproducible amongst the ten repeated simulations.

PMMH using generic SMC performed quite poorly in this example. We do not present the output here, but the key issues are as follows. We find that after 10^5 iterations, and even after 5×10^5 iterations, PMMH using generic SMC completely failed to converge. The ACFs never drop below 0.7, and the trace plots exemplify this property. In fact, the average acceptance rate is on the order of 0.01% to 0.1%.

Our results in this example are encouraging. 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 develop more appropriate MCMC proposals and more sophisticated adaptive schemes for the sets, so that the methodology can be extended to even more realistic applications.

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 it uses auxiliary variables to adaptively compute the 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 a significant increase in precision when adaptively computing the sets compared to when a fixed number of sets is used. The proposed algorithm requires considerable amount of computation, but to the authors' best knowledge, there seems to be a lack of alternative approaches for such problems. Also, recent

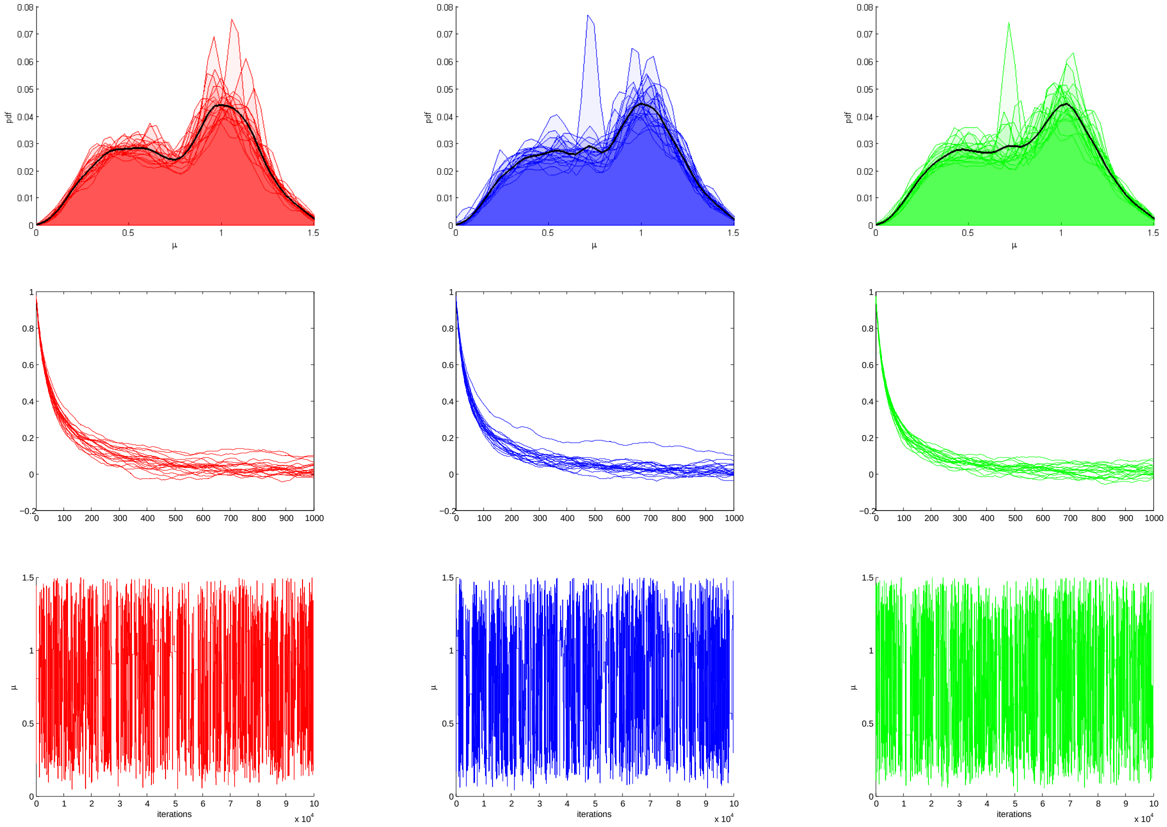

Figure 4: PMMH output or the coalescent model: Algorithm 5. We allow $p \in \{8, 9, ..., 28\}$ and present the output for $N = 50$ (red), 100 (blue), 200 (green) from left to right. Top: estimated pdf of μ. We plot 20 repetitions with the average estimate printed in black. Middle: autocorrelation functions. Bottom: trace plots for one repetition. For $N = 50, 100,$ and 200, the average acceptance ratio is on the order of 0.09, 0.11, and 0.12, respectively.

developments in GPU hardware can be adopted to speed up the computations 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 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 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. This might appeal to practitioners more familiar with multi-level splitting.

Secondly, one could seek to use these ideas within an SMC sampler framework of [17], as is 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

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

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 under way.

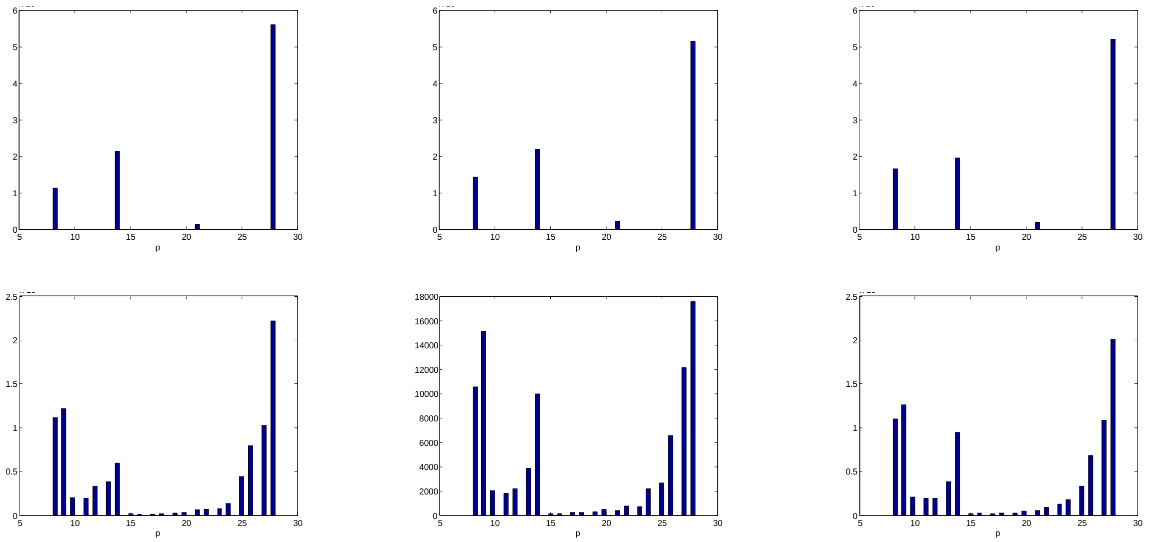

Figure 5: PMMH for the coalescent model. Histograms of number of levels p in the resulting posterior for one repetition under the two adaptive algorithms. We present the output for $N = 50, 100, 200$ from left to right. Top: allowing $p \in \{8, 14, 21, 28\}$. Bottom: allowing $p \in \{8, 9, ..., 28\}$.

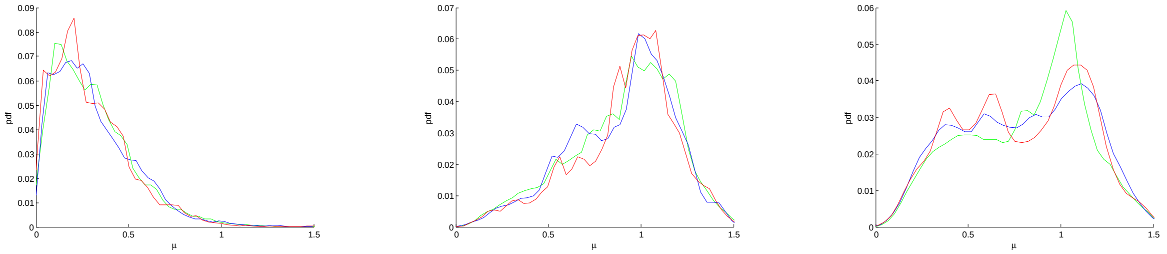

Figure 6: PMMH for the coalescent model for $N = 50$ (red) , 100 (blue) , 200 (green) . We select one run from each algorithm for each value of N and compare the estimated pdfs of μ. Left: fixing $p = 8$. Middle: allowing $p \in \{8, 14, 21, 28\}$. Right: allowing $p \in \{8, 9, ..., 28\}$.

Acknowledgement

We thank Arnaud Doucet and Maria De Iorio for many conversations on this work. The first author was supported by a Ministry of Education grant. The second author was kindly supported by the EPSRC programme grant on Control For Energy and Sustainability EP/G066477/1. We thank two referees and an associate editor for comments that vastly improved the paper.

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:

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

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 normalising constant as $\hat{Z}_p(\cdot)$ with $\cdot$ denoting which random variables generate the estimate.

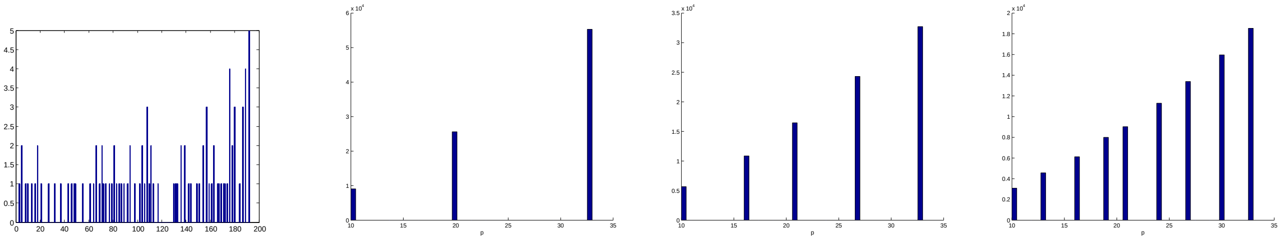

Figure 7: Coalescent with migration. Far left: Dataset for the coalescent with migration. The types $(g \times d)$ run along the horizontal axis, and the height of each bar represents the count of each genetic type at time τ. Right: Histograms of number of levels p in the resulting posteriors for $N = 100$ under three adaptive PMMH algorithms. Only one of the ten repetitions of each simulation is shown.

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:

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

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

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

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 normalising constant Z_p:

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

and we can easily conclude. □

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 $q^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) = \psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1}) f(k|W_p)\overline{q}(\theta(i-1)|\theta') = \psi_\theta(\bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})\bar{W}_p^{(k)}\overline{q}(\theta|\theta')$. Then

$$\begin{aligned}
\frac{\overline{\pi}_p^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})}{q^N(\theta, k, \bar{\mathcal{X}}_{1:p}, \bar{\mathbf{a}}_{1:p-1})} &= \frac{N^{-p}\overline{\pi}(\theta, \mathcal{X}_{1:p}^{(k)})}{\bar{W}_p^{(k)}\mathcal{M}_1(\mathcal{X}_1^{(b_1^k)})\left(\prod_{n=2}^{p}\bar{W}_{n-1}^{(b_{n-1}^k)}\mathcal{M}_n(\mathcal{X}_n^{(b_n^k)}|\mathcal{X}_{n-1}^{(b_{n-1}^k)})\right)\overline{q}(\theta(i-1)|\theta')} \\
&= \frac{N^{-p}\overline{\gamma}_p(\mathcal{X}_{1:p}^{(k)})\overline{p}(\theta)/Z_p}{\mathcal{M}_1(\mathcal{X}_1^{(b_1^k)})\prod_{n=2}^{p}\mathcal{M}_n(\mathcal{X}_n^{(b_n^k)}|\mathcal{X}_{n-1}^{(b_{n-1}^k)})\left(\prod_{n=1}^{p}\bar{W}_n^{(b_n^k)}\right)\overline{q}(\theta(i-1)|\theta')} \\
&= \frac{\overline{\gamma}_p(\mathcal{X}_{1:p}^{(k)})\left(\prod_{n=1}^{p}N^{-1}\left(\sum_{j=1}^{N}w_n(\mathcal{X}_n^{(j)})\right)\right)\overline{p}(\theta)}{Z\mathcal{M}_1(\mathcal{X}_1^{(b_1^k)})\left(\prod_{n=2}^{p}\mathcal{M}_n(\mathcal{X}_n^{(b_n^k)}|\mathcal{X}_{n-1}^{(b_{n-1}^k)})\right)\left(\prod_{n=1}^{p}w(\mathcal{X}_n^{(b_n^k)})\right)\overline{q}(\theta|\theta')} \\
&= \frac{\hat{Z}_p}{Z} \times \frac{\overline{p}(\theta)}{\overline{q}(\theta|\theta')},
\end{aligned}$$

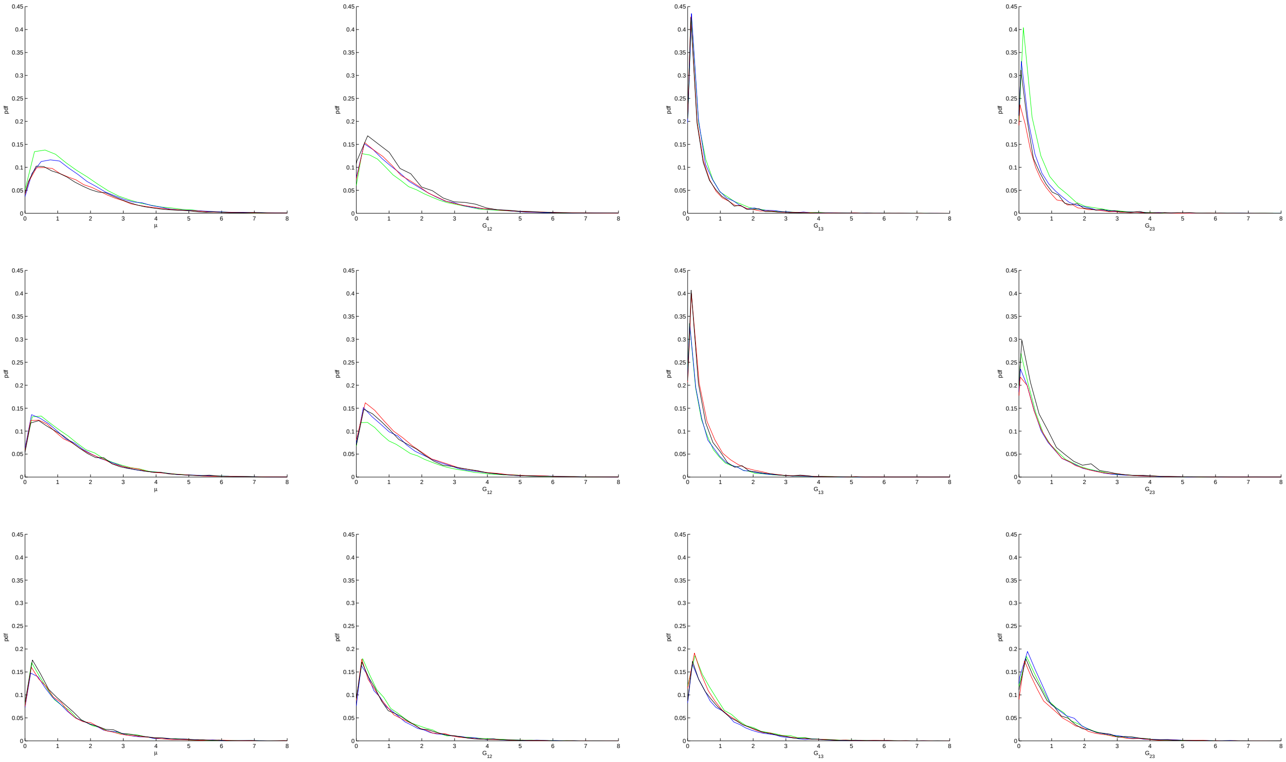

Figure 8: PMMH for the coalescent with migration for $N = 10$ (black) , 50 (red) , 100 (blue) , 200 (green) . Estimated pdfs for μ, G_{12}, G_{13}, G_{23} (from left to right). Only one of the ten repetitions of each simulation is shown. Top row: choosing $p \in \{10, 20, 33\}$. Middle row: choosing $p \in \{10, 16, 21, 27, 33\}$. Bottom row: choosing $p \in \{10, 13, 16, 19, 21, 24, 27, 30, 33\}$.

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 may write

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

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). □

Proof. [Proof of Proposition 4.3] The proof is similar to 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 the 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 $\bar{\pi}^N$, one re-writes the target as

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

Let $\bar{\pi}_p^N(\theta)$ denote the marginal of $\bar{\pi}_p^N(\cdot)$ 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 $\bar{\pi}^N(\cdot)$ w.r.t. v, $\bar{\mathcal{X}}_{1:p(v)}, \bar{\mathbf{a}}_{1:p(v)-1}$, k we have that

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

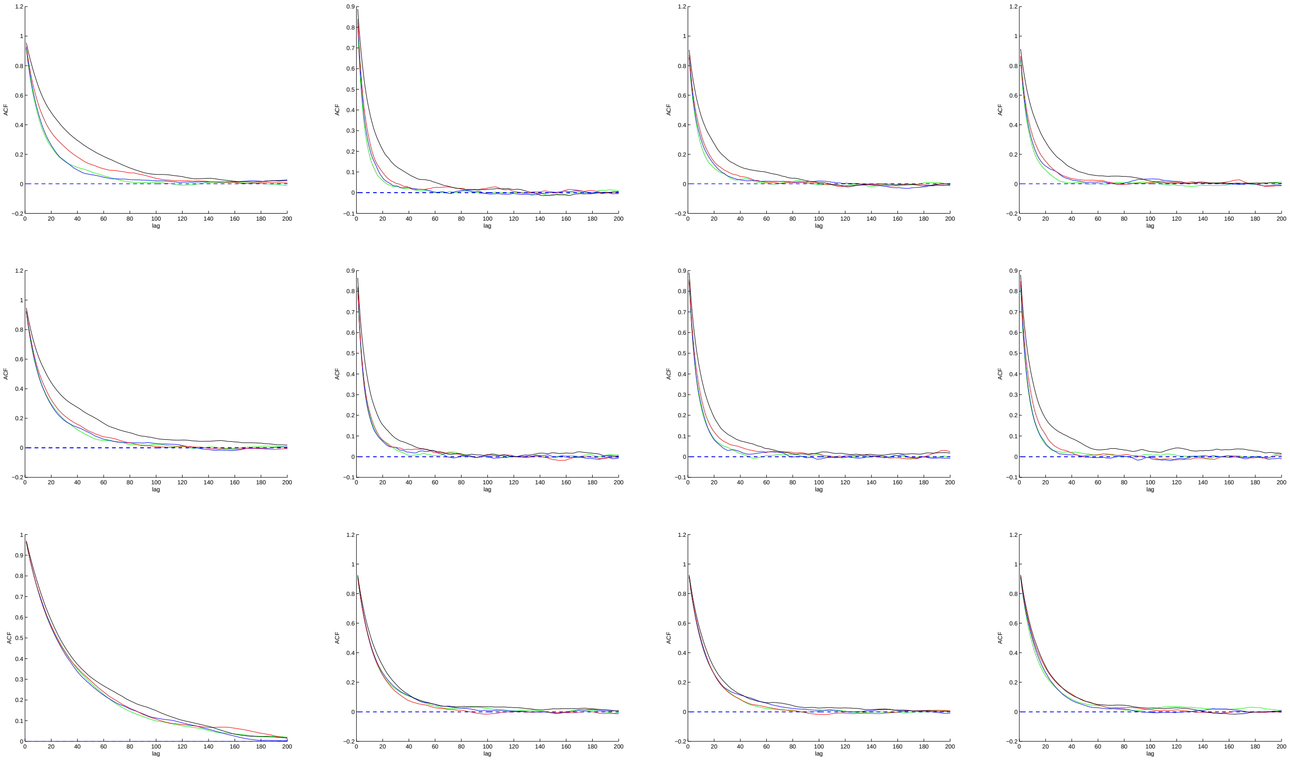

Figure 9: PMMH for the coalescent with migration for $N = 10$ (black) , 50 (red) , 100 (blue) , 200 (green) . ACF plots for μ, G_{12}, G_{13}, G_{23} (from left to right). Only one of the ten repetitions of each simulation is shown. Top row: choosing $p \in \{10, 20, 33\}$. Middle row: choosing $p \in \{10, 16, 21, 27, 33\}$. Bottom row: choosing $p \in \{10, 13, 16, 19, 21, 24, 27, 30, 33\}$.

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 $\mathcal{X}^{(k)}_{1:p(v)}$ is

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

Hence the sequence $\left(\theta(i), \mathcal{X}^{(k)}_{1:p(v)}(i)\right)_{i\geq 0}$ satisfies the required property as a direct consequence of Theorem 1 in [3] and Assumption (A2). □

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.

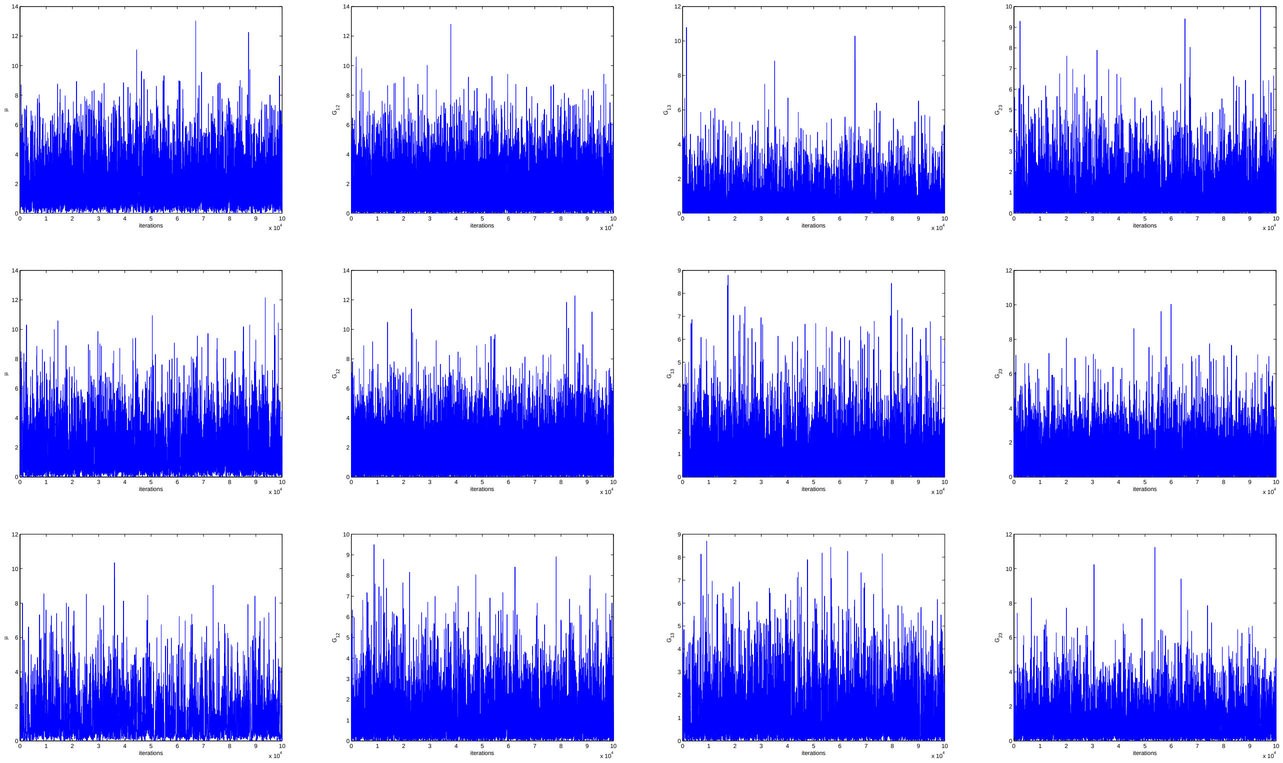

Figure 10: PMMH for the coalescent with migration for $N = 100$. Trace plots for μ, G_{12}, G_{13}, G_{23} (from left to right). Only one of the ten repetitions of each simulation is shown. Top row: choosing $p \in \{10, 20, 33\}$. Middle row: choosing $p \in \{10, 16, 21, 27, 33\}$. Bottom row: choosing $p \in \{10, 13, 16, 19, 21, 24, 27, 30, 33\}$.

[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. (2012). Sequential Monte Carlo for rare event estimation. *Statist. Comp.*, **22**, 795–808.

[10] Cerou, F., Del Moral, P. & Guyader, A. (2011). A non asymptotic variance theorem for unnormalized Feynman-Kac particle models. *Ann. Inst. Henri Poincare*, **47**, 629–649.

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

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

[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). Multi-level 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, Technical Report, Imperial College London.

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