

# Stopping-time resampling for sequential Monte Carlo methods

Yuguo Chen and Junyi Xie

*Duke University, Durham, USA*

and Jun S. Liu

*Harvard University, Cambridge, USA*

[Received May 2003. Final revision July 2004]

**Summary.** Motivated by the statistical inference problem in population genetics, we present a new sequential importance sampling with resampling strategy. The idea of resampling is key to the recent surge of popularity of sequential Monte Carlo methods in the statistics and engineering communities, but existing resampling techniques do not work well for coalescent-based inference problems in population genetics. We develop a new method called ‘stopping-time resampling’, which allows us to compare partially simulated samples at different stages to terminate unpromising partial samples and to multiply promising samples early on. To illustrate the idea, we first apply the new method to approximate the solution of a Dirichlet problem and the likelihood function of a non-Markovian process. Then we focus on its application in population genetics. All our examples show that the new resampling method can significantly improve the computational efficiency of existing sequential importance sampling methods.

**Keywords:** Ancestral inference; Coalescent; Population genetics; Resampling; Sequential importance sampling

## 1. Introduction

Suppose that we are interested in estimating the mean of  $h(\mathbf{x})$  under distribution  $\pi(\mathbf{x})$ , which is known up to a normalizing constant. A standard Monte Carlo method is importance sampling, in which one draws random samples  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}$  from a trial or sampling distribution  $q(\mathbf{x})$  and estimates  $E_{\pi}\{h(\mathbf{x})\}$  by

$$\hat{\mu} = \frac{w^{(1)} h(\mathbf{x}^{(1)}) + \dots + w^{(m)} h(\mathbf{x}^{(m)})}{w^{(1)} + \dots + w^{(m)}}, \quad (1)$$

where  $w^{(i)} = \pi(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$ ,  $i = 1, \dots, m$ . The main objective of importance sampling is to design a sampling distribution that has a high probability mass in the region where the function  $h(\mathbf{x}) \pi(\mathbf{x})$  takes large values. Since it is difficult to prescribe a good high dimensional sampling distribution, we must often decompose a high dimensional vector into low dimensional components and then build up the sampling distribution by adding in components sequentially—this is the basic idea of sequential importance sampling (SIS).

SIS is a versatile and powerful tool for solving complex computational problems. The first SIS algorithm was developed in the 1950s for simulating long chain polymers (Hammersley and Morton, 1954; Rosenbluth and Rosenbluth, 1955). The ideas of pruning and enrichment

*Address for correspondence:* Jun S. Liu, Department of Statistics, Harvard University, 1 Oxford Street, Cambridge, MA 02138, USA.  
E-mail: [jliu@stat.harvard.edu](mailto:jliu@stat.harvard.edu)

(Wall and Erpenbeck, 1959; Grassberger, 1997) were later developed to improve the method greatly. These ideas have been widely used for molecular simulations and structural predictions (Kremer and Binder, 1988; Grassberger, 1997). More recently, the SIS methodology, together with a few key improvements such as resampling and Markov chain Monte Carlo iterations, has found a wide range of applications in computer science (Isard and Blake, 1996), financial data modelling (Pitt and Shephard, 1999), genetic linkage analysis (Irwing *et al.*, 1994), signal processing (Gordon *et al.*, 1993; Liu and Chen, 1995, 1998; Godsill *et al.*, 2000; Chen, 2001) and statistics (Kong *et al.*, 1994; Liu, 1996; Berzuini *et al.*, 1997; MacEachern *et al.*, 1999; Chen, 2001). Liu and Chen (1998) provided a general framework for this class of techniques under the name ‘sequential Monte Carlo’ sampling. Doucet *et al.* (2001) and Liu (2001) are good sources for obtaining detailed knowledge of sequential Monte Carlo sampling and its connection with Markov chain Monte Carlo sampling.

The resampling idea that was introduced in Gordon *et al.* (1993) and Liu and Chen (1995), which resembles the pruning and enrichment techniques (Wall and Erpenbeck, 1959; Grassberger, 1997), is crucial for improving the efficiency of an SIS algorithm. However, its standard implementations in the examples that are considered in this paper did not work well. We introduce a new approach called ‘stopping-time resampling (STR)’, which allows each partially built Monte Carlo sample to pause at a data-dependent stage for resampling considerations. In addition to illustrating the usefulness of the method by diverse examples ranging from the Dirichlet problem to panel data analysis, we focus specifically on statistical inference problems in population genetics. We show that SIS with STR outperformed its corresponding SIS method substantially with negligible computational overhead in all the cases that we considered.

The paper is organized as follows. Section 2 gives a brief summary of the general framework of SIS with resampling. Section 3 motivates our study by using a Dirichlet problem. Section 4 introduces the STR technique. Section 5 applies the SIS approach together with STR to analyse a set of panel data. Section 6 studies population genetics problems, explains how to incorporate STR into the SIS algorithms that have been developed in the literature and presents two examples to demonstrate the advantage of the new method. Section 7 concludes the paper with a discussion.

## 2. Sequential importance sampling with resampling

Suppose that  $\mathbf{x}$  can be decomposed as  $\mathbf{x} = (x_1, \dots, x_d)$ . We let  $\mathbf{x}_t = (x_1, \dots, x_t)$ ,  $1 \leq t \leq d$ , and call it a ‘partial sample’. The essence of SIS is to construct the sampling distribution  $q(\cdot)$  sequentially:

$$q(\mathbf{x}) = q_1(x_1) q_2(x_2|x_1) \dots q_d(x_d|\mathbf{x}_{d-1}).$$

Its importance weight is

$$w(\mathbf{x}) = \frac{\pi(\mathbf{x})}{q(\mathbf{x})} = \frac{\pi(x_1) \pi(x_2|x_1) \dots \pi(x_d|\mathbf{x}_{d-1})}{q_1(x_1) q_2(x_2|x_1) \dots q_d(x_d|\mathbf{x}_{d-1})}, \quad (2)$$

which can often be computed recursively as

$$w_t(\mathbf{x}_t) = w_{t-1}(\mathbf{x}_{t-1}) u_t(\mathbf{x}_t). \quad (3)$$

An obvious candidate for  $u_t$  is  $\pi(x_t|\mathbf{x}_{t-1})/q_t(x_t|\mathbf{x}_{t-1})$ . But this choice is infeasible since finding the marginal distributions  $\pi(x_1), \pi(x_1, x_2), \dots$  is perhaps more difficult than the original problem. In practice, we often can find a sequence of distributions  $\pi_1(x_1), \pi_2(x_1, x_2), \dots, \pi_d(x_1, \dots, x_d)$ , such that  $\pi_t(\mathbf{x}_t)$  is a reasonable approximation to the marginal distribution  $\pi(\mathbf{x}_t)$

for  $t = 1, \dots, d - 1$ , and  $\pi_d(x_1, \dots, x_d) = \pi(x_1, \dots, x_d)$ , and let

$$u_t(\mathbf{x}_t) = \frac{\pi_t(\mathbf{x}_t)}{\pi_{t-1}(\mathbf{x}_{t-1}) q_t(x_t|\mathbf{x}_{t-1})}. \tag{4}$$

It is easy to check that the final weight  $w_d(\mathbf{x}_d)$  is the same as that in equation (2). The  $\pi_t$ , which were called auxiliary distributions in Liu (2001), are used here both to help to design an efficient sampling distribution and to guide the resampling.

Consider as an example the filtering problem of the non-linear state space model:

$$y_t \sim f_t(y_t|x_t), \quad x_t \sim g_t(x_t|x_{t-1}), \quad t = 1, \dots, n,$$

in which we observe  $x_0$  and the  $y_t$  and are interested in estimating  $E(\mathbf{x}_n|\mathbf{y}_n)$ . The target distribution in this case is the posterior distribution

$$\pi(\mathbf{x}) = P(\mathbf{x}_n|\mathbf{y}_n) \propto \prod_{s=1}^n f_s(y_s|x_s) g_s(x_s|x_{s-1}).$$

A simple choice of the sampling distribution is  $q_t(x_t|\mathbf{x}_{t-1}) = g_t(x_t|x_{t-1})$ , and a more sophisticated choice can be  $q_t(\mathbf{x}_t|\mathbf{x}_{t-1}) \propto f_t(y_t|x_t) g_t(x_t|x_{t-1})$ , which incorporates the newly observed data point (Liu and Chen, 1998). A sequence of auxiliary distributions can be  $\pi_t(\mathbf{x}_t) = P(\mathbf{x}_t|\mathbf{y}_t)$ .

One way to implement SIS is to start  $m$  independent chains or paths and to proceed with them in parallel, i.e. we generate  $m$  independent samples  $\{x_1^{(1)}, \dots, x_1^{(m)}\}$  from  $q_1(\cdot)$  at step 1; then we generate  $x_2^{(j)}$  from  $q_2(\cdot|x_1^{(j)})$  for  $j = 1, \dots, m$  at step 2, and so on. The advantage of this parallel implementation is that we can compare the current weights  $w_t^{(1)}, \dots, w_t^{(m)}$  for the  $m$  partial samples  $\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(m)}$ . A very small weight of  $w_t^{(j)}$  suggests that we might want to stop the  $j$ th sample early because it will probably contribute very little to the final estimate. Formally, we can prune away those samples with small current weights and ‘amplify’ those with large current weights by resampling. Suppose that at step  $t$  we have  $m$  independent partial samples  $\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(m)}$ , with weights  $w_t^{(1)}, \dots, w_t^{(m)}$ . A resampling step can be carried out as follows.

- (a) Draw  $m$  samples (with replacement) from  $\{\mathbf{x}_t^{(1)}, \dots, \mathbf{x}_t^{(m)}\}$  with probabilities proportional to  $\{w_t^{(1)}, \dots, w_t^{(m)}\}$ .
- (b) Assign equal weights  $(w_t^{(1)} + \dots + w_t^{(m)})/m$  to each of the  $m$  new samples.

Other more efficient resampling schemes, such as residual resampling (Liu and Chen, 1998) and stratified resampling (Kitagawa, 1996), can reduce the Monte Carlo variation and are usually preferable to the multinomial resampling strategy that was described above. Recently, Fearnhead and Clifford (2003) proposed a new optimal resampling method for the discrete state space case.

Although resampling is essential for an SIS algorithm, doing resampling at every step may not be desirable (Liu and Chen, 1995, 1998). Two resampling schedules have been proposed: deterministic and dynamic. In a deterministic schedule, we do resampling at fixed times  $t_1, t_2, \dots$ , where  $t_i$  is often chosen to be  $i \times t_1$ . In a dynamic schedule, we check the square of the coefficient of variation (denoted as  $cv^2$  henceforth) of the current weights  $w_t$  at each time  $t$  and do resampling if  $cv^2$  is greater than a certain bound. Our experience shows that the dynamic schedule tends to be more efficient than the deterministic schedule. In practice,  $cv^2$  for  $w_t$  is estimated by the ratio between the sample variance and the square of the sample mean of the weights  $w_t^{(1)}, \dots, w_t^{(m)}$ .

In both schedules, we carry all the partial samples in parallel to the same sampling stage (i.e. all samples have their first  $t$  components generated) and then conduct resampling. For some problems such as the Dirichlet problem in the next section and the likelihood inference problem

in population genetics, this ‘same stage’ resampling schedule does not improve the efficiency (see Sections 4 and 6.4). The main reason for its ineffectiveness is that the partial samples at the same sampling stage may have very distinct features so that a large current weight often does not lead to a large future weight. In other words, the sampling stage is not a natural ‘timescale’ for the progress of each partial sample. The example in the next section sheds some light on how to do resampling appropriately.

### 3. A motivating example: the Dirichlet problem

Solving the Dirichlet problem is one of the earliest applications of Monte Carlo methods (Hammersley and Handscomb, 1964). The following example from Farlow (1993) is to find a function  $u(x, y)$ , defined over the unit square  $[0, 1] \times [0, 1]$ , satisfying

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad \text{on } (x, y) \in (0, 1) \times (0, 1) \tag{5}$$

with boundary condition

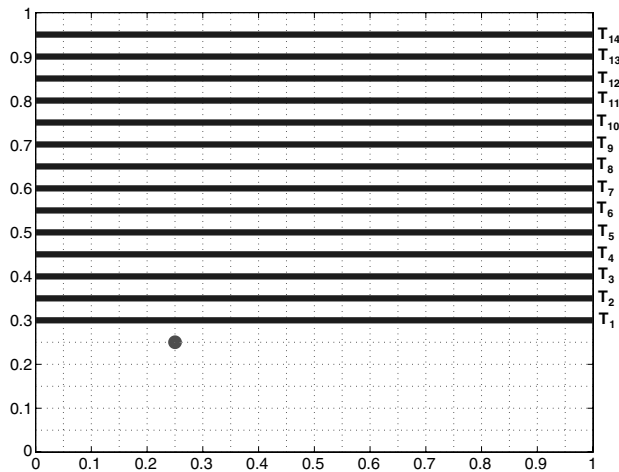
$$u(x, y) = \begin{cases} 1, & \text{on the top of the square,} \\ 0, & \text{on the sides and bottom of the square.} \end{cases} \tag{6}$$

The Monte Carlo approach to solve the equation numerically starts by discretizing the unit square into  $n^2$  equal-sized small squares (Fig. 1) and then uses the finite difference approximation to replace condition (5) by

$$u\left(\frac{i}{n}, \frac{j}{n}\right) = \frac{1}{4} \left\{ u\left(\frac{i-1}{n}, \frac{j}{n}\right) + u\left(\frac{i+1}{n}, \frac{j}{n}\right) + u\left(\frac{i}{n}, \frac{j-1}{n}\right) + u\left(\frac{i}{n}, \frac{j+1}{n}\right) \right\}, \tag{7}$$

where  $(i/n, j/n)$  is an interior point. Suppose that a simple random walk starts from the interior point  $u(i_0/n, j_0/n)$  and hits the boundary at  $(i'/n, j'/n)$ . It is well known that

$$E \left\{ u\left(\frac{i'}{n}, \frac{j'}{n}\right) \right\} = u\left(\frac{i_0}{n}, \frac{j_0}{n}\right).$$



**Fig. 1.** Discretization of the unit square into equal-sized small squares: the value of  $u$  at the point  $(0.25, 0.25)$  can be estimated by running many random walks starting from  $(0.25, 0.25)$  and resampling if  $cv^2$  for the partial weights at  $T_i$  is greater than  $B$ , where  $T_i$  is the stopping time when all random walks hit the thick line with height  $(25 + 5i)/100$ , for  $i = 1, \dots, 14$

Hence, we can run  $m$  independent simple random walks from  $(i_0/n, j_0/n)$ , record the ending points of these random walks as  $(i^{(1)}/n, j^{(1)}/n), \dots, (i^{(m)}/n, j^{(m)}/n)$  and use

$$\frac{\sum_{k=1}^m u(i^{(k)}/n, j^{(k)}/n)}{m}$$

to estimate  $u(i_0/n, j_0/n)$ . We call this strategy the naïve Monte Carlo method.

For the boundary condition (6), the naïve Monte Carlo estimate is equal to the proportion of random walks that hit the top edge of the unit square. A potentially better strategy is to encourage the random walks to move up towards the top edge. This can be realized by assigning a higher probability to move up at each step of the walk, and then using the importance weight to correct the bias. For example, we can implement a random walk that moves to its four neighbours  $((i - 1)/n, j/n), ((i + 1)/n, j/n), (i/n, (j + 1)/n)$  and  $(i/n, (j - 1)/n)$  with probabilities  $0.25, 0.25, 0.25 + \delta$  and  $0.25 - \delta$  respectively, where  $0 < \delta < 0.25$ . If we use  $\mathbf{x}_t$  to denote the path of a random walk up to time  $t$ , the importance weight  $w_t(\mathbf{x}_t)$  that is associated with  $\mathbf{x}_t$  can be computed by using equation (3), where  $\pi(x_t|\mathbf{x}_{t-1}) = 0.25$  and  $q_t(x_t|\mathbf{x}_{t-1})$  is the probability of moving from the current position to  $x_t$  according to the proposal distribution. The final estimate of  $u(i_0/n, j_0/n)$  is the average of the importance weights.

Because of the sequential nature of the random walk, resampling might be incorporated to increase the efficiency. The standard resampling strategy runs all the walks the same number of steps and decides whether to do resampling by checking  $cv^2$  for the partial weights  $w_t$ . However, this strategy does not improve the accuracy of the estimate (see the numerical results in Section 4) because different random walks take different numbers of steps to reach the top edge and only those that hit the top edge contribute to the estimate. At time  $t$  (i.e. the length of each random walk is  $t$ ), some walks with large weights might be far from the top edge and their final weights when hitting the top edge tend to be very small because moving up will decrease the importance weights. In contrast, some samples with small weights may be very close to the top edge. Although their ‘current’ weights are small, they are already very close to the end so their final weights tend to be large comparing with other samples. Since large current weights often imply small future weights and vice versa, resampling according to the natural time ‘ $t$ ’ actually prunes away many ‘good’ samples.

A more natural timescale for this problem is the time that random walks hit the thick horizontal lines (see Fig. 1), because it roughly determines how far the sample is from the top edge. Thus, an alternative strategy is to run the  $m$  walks until the first time that they all reach the line indexed by  $T_1$  in Fig. 1. Then we check  $cv^2$  for the importance weights of all the walks: if it is greater than a certain bound, we do resampling; if not, we continue to run all the random walks until they all reach line  $T_2$ , and so on. The next section gives a formal description of this idea and proves that this resampling approach is proper.

#### 4. Sequential importance sampling with stopping-time resampling

Now we describe the general method of sequential importance sampling with stopping-time resampling (SISSTR), whose preliminary version was first used in Chen and Liu (2000). In SIS, we obtain each sample  $\mathbf{x}$  by generating its components  $x_1, x_2, \dots$  sequentially. A sequence of stopping times  $1 < T_1 < \dots < T_L < d$  is defined on the sample path  $\{x_1, x_2, \dots\}$  so that the event  $\{T_l = t\}$  is measurable with respect to the  $\sigma$ -field that is generated by  $\{x_1, \dots, x_t, T_1, \dots, T_{l-1}\}$ . In other words, after we observe  $\{x_1, \dots, x_t, T_1, \dots, T_{l-1}\}$ , we know whether we have reached the stopping time  $T_l$ . We check  $cv^2$  for the weights at these stopping times and do resampling if  $cv^2$  is greater than a certain bound  $B$ . More precisely, we start by running in parallel  $m$  sample

paths until they reach their respective first stopping time  $T_1$ . If  $cv^2$  is greater than a certain bound at this stage, we do resampling; otherwise we continue to run these partial samples until they reach their second stopping time  $T_2$  and check  $cv^2$  again and so on.

The following argument shows that SISSTR produces a consistent estimate of  $E_\pi\{h(\mathbf{x})\}$  as the sample size  $m$  goes to  $\infty$ . Without loss of generality, we consider only a fixed and finite number  $d$  of components and one stopping time  $T$  satisfying the condition that  $P(T \leq d) = 1$ . We would need that  $P(T < \infty) = 1$  if  $d$  is infinite. Under these settings, our SISSTR procedure produces a population of weighted samples  $(w_T, \mathbf{x}_T)$  at the stopping time and then resamples from this population. As  $m \rightarrow \infty$ , resampling is equivalent to letting each  $\mathbf{x}_T$  survive with a probability  $c w_T(\mathbf{x}_T)$ , where  $c$  is a constant. The survived  $\mathbf{x}_T$  is given a constant weight  $c^{-1}$ . If  $\mathbf{x}_T$  with  $T = t$  survives the resampling, it is continued with SIS to produce  $\mathbf{x} = (\mathbf{x}_t, x_{t+1}, \dots, x_d)$ . Its weight at the end of the process is

$$w'(\mathbf{x}) = \frac{1}{c} \times u_{t+1} \times \dots \times u_d = \frac{1}{c} \frac{\pi(\mathbf{x})}{\pi_t(\mathbf{x}_t) q_{t+1}(x_{t+1}|\mathbf{x}_t) \dots q_d(x_d|\mathbf{x}_{d-1})}.$$

Thus, for any integrable function  $h(\mathbf{x})$ , we have

$$\begin{aligned} E\{h(\mathbf{x}) w'(\mathbf{x}) | \mathbf{x} \text{ survived}\} &= \frac{\sum_t E\{h(\mathbf{x}) w'(\mathbf{x}), T = t, \mathbf{x} \text{ survived}\}}{P(\mathbf{x} \text{ survived})} \\ &= c' \sum_t E\{h(\mathbf{x}) w'(\mathbf{x}) c w_t(\mathbf{x}_t)\} P(T = t) \\ &= c' \sum_t E_\pi\{h(\mathbf{x})\} P(T = t) = c' E_\pi\{h(\mathbf{x})\}, \end{aligned}$$

where  $c' = 1/P(\mathbf{x} \text{ survived})$ . Therefore, it is valid to estimate  $E_\pi\{h(\mathbf{x})\}$  by equation (1).

If we choose  $T_i \equiv l$ , SISSTR is identical to the standard dynamic resampling schedule that we discussed in Section 2. The new SISSTR method enables us to choose appropriate stopping times, so that at these stopping times all the samples tend to have ‘similar’ futures. Therefore the current weights can better reflect the future weights. The value of this method for applications of SIS in molecular population genetics and other fields is demonstrated in Sections 5 and 6.4. The examples differ slightly from the current setting by having a non-fixed number ( $d$ ) of components, but the same theory applies.

To examine the effectiveness of the SISSTR method, we revisited the Dirichlet problem of Section 3 and compared different algorithms on estimating the value of  $u$  at the point  $(0.25, 0.25)$  (the dot in Fig. 1) for the Dirichlet problem (5). The unit square was discretized into  $100 \times 100$  equal-sized small squares. We chose  $\delta = 0.01$  in the SIS procedure so that the probabilities of moving to the four directions left, right, up and down are 0.25, 0.25, 0.26 and 0.24 respectively. For each method, we generate 100 estimates with each based on 5000 samples. In Table 1, we report the mean of the 100 estimates and the standard error of the mean.

**Table 1.** Comparison of different methods on estimating  $u(0.25, 0.25)$

Method	Estimate	Standard error
Naïve Monte Carlo	0.0677	0.0035
SIS without resampling	0.0686	0.0029
Traditional SIS with resampling	0.0673	0.0037
SISSTR	0.0681	0.0021

In traditional SIS with resampling, we checked  $cv^2$  at every 100 steps and do resampling if  $cv^2 > B$ . In SISSTR we choose  $T_i$  as the first time that all samples hit the horizontal line with height  $(25 + 5i)/100$ , for  $i = 1, \dots, 14$  (see the thick lines in Fig. 1) and do resampling at the stopping time  $T_i$  if  $cv^2$  for the partial weights at  $T_i$  is greater than  $B$ . We chose  $B = 0.3$ , which incurred about 33 resamplings for traditional SIS and two resamplings for SISSTR. All the methods took about 5 min on a 3 GHz Dell workstation and gave estimates that were close to the true value. (A long simulation showed that  $u(0.25, 0.25) \approx 0.0683$ .) From Table 1, we can see that SISSTR gives the smallest standard error among the four methods. SIS without resampling is better than both the naïve method and traditional SIS with resampling, indicating that doing resampling in a wrong way may decrease the efficiency of SIS. We observed similar patterns when estimating the values of other points in the unit square.

To see how sensitive the performance of SISSTR is to the choice of  $B$ , we ran the simulation for  $B$  ranging from 0.1 to 1. The number of resamplings incurred ranged from 1 to 5 and the corresponding standard error ranged from 0.0021 to 0.0024. If we choose  $B > 1$ , then the average number of resamplings is less than 1 and the standard error is close to that of SIS without resampling.

**5. Another example: inference in a non-Markov binary process**

A common scientific objective in studies of episodic phenomena is to characterize the duration in and between episodes. Many examples can be found in the study of chronic diseases, chronic infections and life history events (de Stavola, 1988). It is often not possible or not practical to observe the episodic process continuously. Instead, panel studies are often used in which the current state of the episodic process is observed at a number of predetermined points. Although there is an extensive literature on the analysis of panel studies under the assumption that the episodic phenomena follow a continuous time Markov process (Kalbfleisch and Lawless, 1985), many episodic phenomena exhibit duration dependence and are hence non-Markovian, e.g. spells of employment and unemployment (Jovanovic, 1979; Lippman and McCall, 1976). The likelihood for panel studies of non-Markov processes generally does not have a computationally tractable closed form expression. Chen *et al.* (2004) developed Monte Carlo methods that are suitable for analysing such panel studies.

As a concrete example, we consider here a two-state process  $\mathbf{Y} = (Y_t : 0 \leq t \leq \tau)$  where the duration  $X$  in state  $j$  ( $j = 0, 1$ ) has probability density function

$$f_j(x) = \frac{c_j}{b_j} \left(\frac{x}{b_j}\right)^{c_j-1} \exp\left\{-\left(\frac{x}{b_j}\right)^{c_j}\right\}, \quad x > 0, \quad b_j, c_j > 0. \tag{8}$$

This is the probability density function for the Weibull family of distributions, which is widely used to model duration. The parameters of interest are  $\theta = (b_0, c_0, b_1, c_1)$ . Let  $t^*(\mathbf{Y}) = \{t_1^*(\mathbf{Y}), t_2^*(\mathbf{Y}), \dots, t_K^*(\mathbf{Y})\}$  be the times at which  $\mathbf{Y}$  moves between states, where  $K$  is a random variable denoting the total number of transitions. Assume that the process  $\mathbf{Y}$  is in equilibrium. Let  $P(j), j = 0, 1$ , denote the probability that  $Y_0 = j$ . Then the density function of  $\mathbf{Y}$  for  $Y_0 = j, j = 0, 1$ , is

$$p(\mathbf{y}) = P(j) \frac{\mathcal{G}_j\{t_1^*(\mathbf{Y})\}}{\mu_{j,X}} f_{1-j}\{t_2^*(\mathbf{Y}) - t_1^*(\mathbf{Y})\} f_j\{t_3^*(\mathbf{Y}) - t_2^*(\mathbf{Y})\} f_{1-j}\{t_4^*(\mathbf{Y}) - t_3^*(\mathbf{Y})\} \dots$$

where  $\mathcal{G}_j(X) = P(X > x), j = 0, 1$ , denotes the survivor function of the duration  $X$  for state  $j$  and  $\mu_{j,X}, j = 0, 1$ , denotes the mean of the duration  $X$  in state  $j$ , which can be found from equation (8) (Cox and Isham, 1981). In panel studies, instead of observing the exact transition times into

and out of states, we observe the states at predetermined times  $t_1, t_2, \dots, t_k$ ,  $(Y_{t_1}, Y_{t_2}, \dots, Y_{t_k}) := (Z_1, \dots, Z_k) := \mathbf{Z}$ . The likelihood function of  $Z_1, \dots, Z_k | \theta$  is

$$\pi(z_1, \dots, z_k) = \int p(\mathbf{y}) \mathbf{1}_{\{y_{t_1}=z_1, \dots, y_{t_k}=z_k\}} d\mathbf{y}. \tag{9}$$

Conceptually, we can draw  $m$  samples  $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(m)}$  from  $p(\mathbf{y})$  and estimate  $\pi(z_1, \dots, z_k)$  by

$$\hat{\pi}(z_1, \dots, z_k) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{\{y_{t_1}^{(i)}=z_1, \dots, y_{t_k}^{(i)}=z_k\}}.$$

However, this naïve Monte Carlo method will be very inefficient because  $\mathbf{1}_{\{y_{t_1}=z_1, \dots, y_{t_k}=z_k\}}$  will be 0 for most  $\mathbf{y}$  drawn from  $p(\mathbf{y})$ . Instead, as indicated in algorithm 1 below, we may sample  $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(m)}$  from a trial distribution  $q(\mathbf{y})$  whose support is  $\{\mathbf{y} : y_{t_1} = z_1, \dots, y_{t_k} = z_k\}$  and then correct the bias by importance weights.

### 5.1. Algorithm 1

*Step 1:* set initial time  $x_0 = t_1$ , initial state  $s_0 = z_1$  and  $Y_{x_0} = s_0$ .

*Step 2:* suppose that the current time is  $x_i$  and  $Y_{x_i} = s_i$ . Let  $l = \min\{j : x_i < t_j \text{ and } z_j \neq s_i, 1 \leq j \leq k\}$  and assume that  $\min(\emptyset) = \infty$ .

- (a) In case I,  $l$  is a finite number. If  $\{(t_l - x_i)/b_{s_i}\}^{c_{s_i}} < c$ , where  $c$  is a chosen constant, draw a random time  $t$  from a scaled beta distribution  $g(t) \propto \{t/(t_l - x_i)\}^{c_{s_i}-1} (0 < t < t_l - x_i)$ . Otherwise, draw a random time  $t$  from a truncated Weibull distribution which is proportional to  $f_{s_i}(t) \mathbf{1}_{\{t < t_l - x_i\}}$  (see equation (8)). Set  $x_{i+1} = x_i + t$  and  $s_{i+1} = 1 - s_i$ .
- (b) In case II,  $l = \infty$ . Draw a random time  $t$  from the Weibull distribution  $f_{s_i}(t)$  (see equation (8)). Set  $x_{i+1} = x_i + t$  and  $s_{i+1} = 1 - s_i$ .

*Step 3:* the process stops when the current time  $x_i > t_k$ .

At step 2, when  $\{(t_l - x_i)/b_{s_i}\}^{c_{s_i}} < c$  is small, the truncated Weibull distribution tends to give a low acceptance rate. Thus, instead we draw samples from a scaled beta distribution since the truncated Weibull distribution with large truncation is roughly proportional to the scaled beta distribution. Simulation results ( $c = 0.1$  in the example below) show that adaptively choosing between truncated Weibull and scaled beta distributions as the proposal distribution greatly improves the efficiency. Drawing samples from the scaled beta distribution  $g(t)$  can be realized by first drawing a sample from  $\text{beta}(c_{s_i}, 1)$  and multiplying it by  $t_l - x_i$ . Drawing from truncated Weibull distributions can be realized by drawing samples from the Weibull distribution and rejecting those that are not within the range  $(0, t_l - x_i)$ .

To incorporate resampling to improve algorithm 1, we need to define an appropriate time-scale to reflect the trend of the importance weights. If two neighbouring observations  $z_i$  and  $z_{i+1}$  are different, then for every sample there is at least one transition between  $t_i$  and  $t_{i+1}$ . Thus, we define a stopping time as the first transition between  $t_i$  and  $t_{i+1}$ . At this stopping time, all samples are roughly the same distance from the end. The total number of stopping times is equal to the total number of neighbouring pairs with different values.

To show that STR can indeed help algorithm 1, we simulated a two-state process  $\mathbf{Y}$  with  $b_0 = 1.1, c_0 = 0.8, b_1 = 0.9$  and  $c_1 = 1.2$ . The observations taken at times  $0, 1, \dots, 34$  were

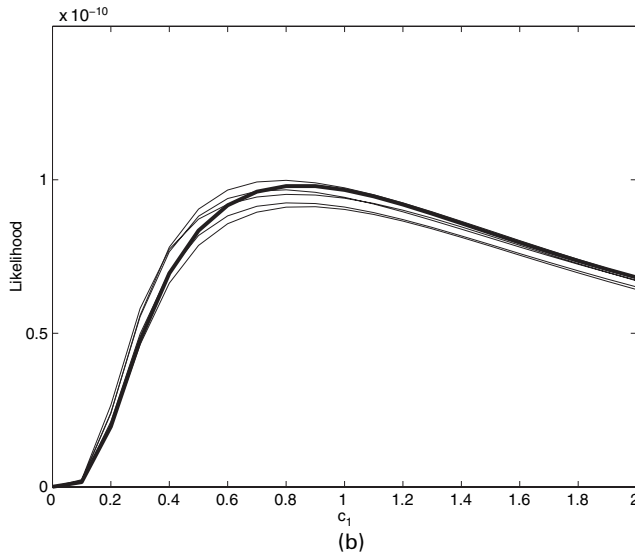
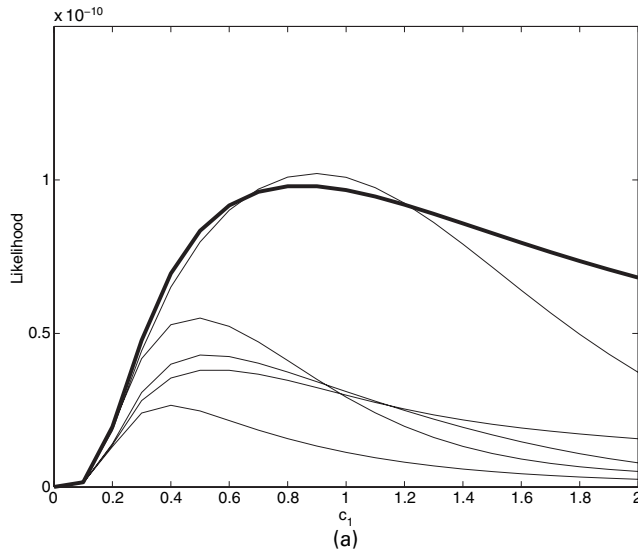
01110011000000010000100101100100010.

For ease of comparison, we solely consider inference for  $c_1$  conditional on the true values of  $b_0, c_0$  and  $b_1$ . We simulated the two-state process by using algorithm 1 with a guessed value of  $c_1$ ,

denoted by  $c_1^* = 1.5$ . A by-product of this approach is that we can use one sampling distribution  $q_{c_1^*}$  to estimate  $\pi_{c_1}(z_1, \dots, z_k)$  for a range of  $c_1$  close to  $c_1^*$ , and thus the likelihood curve, by

$$\hat{\pi}_{c_1}(z_1, \dots, z_k) = \frac{1}{m} \sum_{i=1}^m \frac{p_{c_1}(\mathbf{y}^{(i)})}{q_{c_1^*}(\mathbf{y}^{(i)})}. \tag{10}$$

Fig. 2(a) displays the estimated likelihood curves for  $c_1$  by algorithm 1. We ran it five times, with each run based on 50000 samples, which took about 15 min on a 3 GHz Dell workstation.



**Fig. 2.** Comparison of estimated likelihood curves for the panel data example (—, accurate estimate of the likelihood curve based on 500 000 samples): (a) five independent likelihood curve estimates based on 50 000 samples by using algorithm 1 without resampling; (b) five independent likelihood curve estimates based on 50 000 samples by using algorithm 1 with STR

The large variability of the curves indicates that algorithm 1 without resampling is not very efficient for this problem. We also implemented algorithm 1 with STR, which has 16 stopping times, and observed a significant improvement. With sample size  $m = 50000$ , and  $cv^2$  bound  $B = 1$ , about 10 resamplings were incurred in each of the five independent runs. The additional computational cost of STR was negligible. Fig. 2(b) displays the likelihood curves that were estimated from five runs of the SISSTR method. As a comparison, an ‘accurate’ estimate of the likelihood curve (the bold curves in Figs 2(a) and 2(b)) was obtained on the basis of 500000 samples by using algorithm 1 with STR which took about  $2\frac{1}{2}$  h.

We also ran simulations for various values of  $B$  ranging from 0 to 100. The number of resamplings ranged from 2 to 15 and the performance of SISSTR did not change much (with slightly large variation for  $B = 100$ ). This shows that the performance of SISSTR is quite robust to the choice of  $B$ .

## 6. Applications to population genetics

Population genetics studies genetic variations within and between species. It seeks to understand the evolutionary process that produced these variations and provides the genetic foundation for evolutionary biology (Hartl and Clark, 1997). Statistical methods have played an important role in studying population genetics throughout its development (Ewens, 1979; Donnelly and Tavaré, 1995; Nordborg, 2001). Recent advances in biotechnology have provided an abundance of data on genetic variations of deoxyribonucleic acid within a population. These data facilitate population genetics studies and help to address a broad array of biological questions concerning topics such as the rate of mutation, the time to the most recent common ancestor (MRCA) and the demographic history of a population.

A typical data set in a population genetics study consists of chromosomes that are randomly sampled from a population. Since these chromosomes share ancestry, they are not independent. Consequently the amount of information that is contained in the data grows very slowly as the sample size increases, typically proportional to the logarithm of the sample size (Donnelly and Tavaré, 1995; Ewens, 1972). It is thus desirable to employ a likelihood-based inference method to make the most efficient use of the limited information. However, the distribution of a random sample of chromosomes depends on the model parameters in a complicated way, and calculating the exact likelihood of these parameters is usually infeasible. Griffiths and Tavaré (1994a, b, c) pioneered the use of importance sampling in population genetics. Kuhner *et al.* (1995) introduced a Markov chain Monte Carlo approach, which was further studied in Wilson and Balding (1998) and Markovtsova *et al.* (2000). It has been shown that importance sampling and Markov chain Monte Carlo methods often complement each other (Felsenstein *et al.*, 1999; Stephens and Donnelly, 2000; Liu, 2001), suggesting that both will continue to play an important role in population genetics. We focus our discussion here on importance sampling.

### 6.1. The coalescent

Consider a sample of chromosomes,  $A_n = (a_1, a_2, \dots, a_n)$ , drawn from the current population completely at random, where  $a_i$  denotes the genetic type of the  $i$ th chromosome in the sample. The simplest model to describe the data is the celebrated Wright–Fisher model (Hartl and Clark, 1997), which assumes that the number of offspring of each individual is independent and identically Poisson distributed, and the population has a fixed number  $N$  of chromosomes throughout its history and evolves in non-overlapping generations. To avoid the complication of recombinations, we assume that the samples are from a very small chromosomal segment so

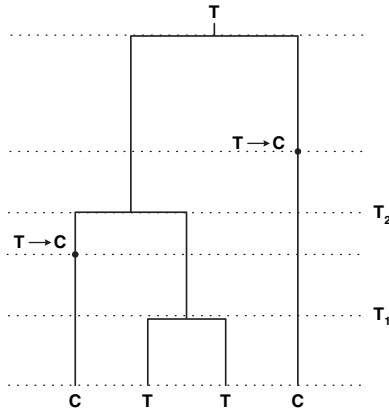


Fig. 3. Illustration of a genealogical tree

that we can view the population as being haploid, i.e. each individual has a single parent. Let  $E$  denote the set of possible genetic types, which is assumed to be countable. To accommodate mutations, we postulate that, for a chromosome of type  $\alpha$ , the genetic type of its child is  $\alpha$  with probability  $1 - \mu$  and is  $\beta \in E$  with probability  $\mu p_{\alpha\beta}$ , where  $\mu$  is the mutation rate per chromosome per generation and  $P = (p_{\alpha\beta})$  is the probability transition matrix, which is known and has a unique stationary distribution. In this paper we sometimes call the chromosomal segment in consideration a ‘gene’.

Since the Wright–Fisher model is too cumbersome to make inference from, the coalescent process was introduced by Kingman (1982) as a continuous time approximation. It is shown that, if time is measured in units of  $N$  generations, the ancestral process of the Wright–Fisher model converges in distribution to the coalescent as  $N \rightarrow \infty$ . As opposed to the ‘prospective’ viewpoint of classical population genetics that considers the future of a population given its current state (Ewens, 1979), the coalescent is ‘retrospective’, looking backwards in time and describing how chromosomes merge at times of common ancestry. A recent review of the subject can be found in Nordborg (2001).

The coalescent describes the ancestral process of the sampled chromosomes by a binary tree, with the MRCA as the root and the sampled chromosomes as the leaves. The tree can be constructed backwards. Each ancestor of the sample is represented by a branch of the tree. Starting from the chromosomes sampled, when there are  $j$  branches in the tree, we wait for a random time  $T_j$ , which has an exponential distribution with mean  $j(j - 1)/2$ , until two branches that are chosen at random coalesce. When coalescences occur, the number of branches decreases by 1. Mutations occur as a Poisson process with rate  $\theta/2 = N\mu$  along the branches of the tree, where  $\mu$  is the mutation rate per chromosome per generation. The Poisson rate is  $N\mu$  since time is measured in units of  $N$  generations.

Fig. 3 is a simple genealogical tree illustrating the coalescent process. There are only two genetic types  $E = \{C, T\}$ . The current sample,  $\{C, T, T, C\}$ , is given at the bottom of the tree and  $\{T\}$  is the root of the tree. Each vertical line represents an ancestor. The horizontal full lines denote the occurrences of coalescences. The dots represent mutations.

### 6.2. Sequential importance sampling for the coalescent process

Our observed data consist of a random sample  $A_n$  of  $n$  chromosomes from the population in equilibrium. Of interest is the estimation of the mutation rate  $\theta/2$ . We denote the likelihood

function, i.e. the probability of obtaining the sample under the coalescent model, as  $\pi_\theta(A_n)$ . Although we cannot calculate  $\pi_\theta(A_n)$  analytically, algorithm 2 below can be used to generate  $A_n$  from  $\pi_\theta$  (Griffiths and Tavaré, 1994a).

6.2.1. Algorithm 2

*Step 1:* generate one genetic type according to the stationary distribution of the transition matrix  $P$ , and then immediately split it into two copies.

*Step 2:* if there are currently  $k$  genes (chromosomal segments),  $2 \leq k \leq n$ , choose one at random from them. Then, with probability  $(k - 1)/(k + \theta - 1)$  split this gene into two of the same type, or with probability  $\theta/(k + \theta - 1)$  mutate this gene according to  $P$ .

*Step 3:* if there are fewer than  $n + 1$  genes, go back to step 2. When there are  $n + 1$  genes, delete the last duplicated gene to form a sample of size  $n$  and stop.

Stephens and Donnelly (2000) used  $\mathcal{H} = (H_{-M}, H_{-(M-1)}, \dots, H_{-1}, H_0)$  to denote the split and mutation events in the ancestry, where  $H_0$  is the current sample (i.e.  $H_0 = A_n$ ),  $H_{-M}$  is the MRCA of the sample and  $H_{-i}$  is the set of unordered genetic types after the  $i$ th coalescent or mutational event. In Fig. 3, the dotted lines indicate the time when coalescences or mutations occur. The history  $\mathcal{H}$  in this case can be recorded as  $(\{T\}, \{T, T\}, \{T, C\}, \{T, T, C\}, \{C, T, C\}, \{C, T, T, C\})$ . Thus, the probability of a particular genealogical tree  $\mathcal{H} = (H_{-M}, H_{-(M-1)}, \dots, H_{-1}, H_0)$  can be written as

$$p_\theta(\mathcal{H}) = p_\theta(H_{-M}) p_\theta(H_{-(M-1)}|H_{-M}) \dots p_\theta(H_0|H_{-1}) (n - 1)/(n - \theta + 1), \tag{11}$$

where  $p_\theta(H_{-M})$  is the probability of genetic type  $H_{-M}$  under the stationary distribution of the transition matrix  $P$ . In equation (11), the transition probability is given by

$$p_\theta(H_i|H_{i-1}) = \begin{cases} \frac{|H_{i-1}|_\alpha}{|H_{i-1}|} \frac{\theta}{|H_{i-1}| - 1 + \theta} P_{\alpha\beta}, & \text{if } H_i = H_{i-1} - \alpha + \beta, \\ \frac{|H_{i-1}|_\alpha}{|H_{i-1}|} \frac{|H_{i-1}| - 1}{|H_{i-1}| - 1 + \theta}, & \text{if } H_i = H_{i-1} + \alpha, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

where  $|H_i|$  denotes the number of individuals (genes) in  $H_i$  and  $|H_i|_\alpha$  is the number of genes of type  $\alpha$  in  $H_i$ . The probability of obtaining the observed sample  $A_n$  is the sum of the probabilities of all trees that are consistent with  $A_n$ , i.e.

$$\pi_\theta(A_n) = \sum_{\{\mathcal{H}: H_0=A_n\}} P_\theta(\mathcal{H}) = \sum_{\mathcal{H}} \pi_\theta(A_n|\mathcal{H}) P_\theta(\mathcal{H}), \tag{13}$$

where

$$\pi_\theta(A_n|\mathcal{H}) = \begin{cases} 1, & \text{if } H_0 = A_n, \\ 0, & \text{if } H_0 \neq A_n. \end{cases}$$

Therefore, we can estimate  $\pi_\theta(A_n)$  by

$$\hat{\pi}_\theta(A_n) = \frac{1}{m} \sum_{i=1}^m \pi_\theta(A_n|\mathcal{H}^{(i)}), \tag{14}$$

where  $\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(m)}$  are independent and identically distributed samples from  $P_\theta(\mathcal{H})$ .

Because algorithm 2 simulates forwards from the MRCA, it is extremely unlikely that its simulated tree will have  $A_n$  as its leaves. Consequently, the naïve estimator (14) is very inefficient and it is preferable to sample the ancestry backwards starting from the current sample  $A_n$ . More

precisely, we need first to choose a sequence of backward transition functions  $q_{\theta_0}(H_{i-1}|H_i)$ , whose support includes the set  $\{H_{i-1} : p_{\theta}(H_i|H_{i-1}) > 0\}$ . Then we generate  $m$  independent and identically distributed genealogical trees  $\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(m)}$  from the distribution

$$Q_{\theta_0}(\mathcal{H}) = \prod_{j=0}^{-M+1} q_{\theta_0}(H_{j-1}|H_j)$$

and estimate  $\pi_{\theta_0}(A_n)$  by

$$\tilde{\pi}_{\theta_0}(A_n) = \frac{1}{m} \sum_{i=1}^m \pi_{\theta_0}(A_n|\mathcal{H}^{(i)}) \frac{P_{\theta_0}(\mathcal{H}^{(i)})}{Q_{\theta_0}(\mathcal{H}^{(i)})}.$$

We can estimate  $\pi_{\theta}(A_n)$  for a range of  $\theta$  by using one sampling distribution  $Q_{\theta_0}$  because

$$\tilde{\pi}_{\theta}(A_n) = \frac{1}{m} \sum_{i=1}^m \pi_{\theta}(A_n|\mathcal{H}^{(i)}) \frac{P_{\theta}(\mathcal{H}^{(i)})}{Q_{\theta_0}(\mathcal{H}^{(i)})}. \tag{15}$$

It is noted that the Griffiths–Tavaré algorithm corresponds to using the trial distribution

$$q_{\theta}(H_{j-1}|H_j) \propto p_{\theta}(H_j|H_{j-1}), \tag{16}$$

where the  $p_{\theta}$  are defined in equation (12). From a Bayesian point of view, the scheme that is based on expression (16) is equivalent to putting a uniform prior distribution on  $H_{i-1}$ , so that  $q_{\theta}$  is the posterior of  $H_{i-1}$  conditional on  $H_i$ .

The ‘ideal’ trial distribution, however, is  $Q_{\theta}^*(\mathcal{H}) = P_{\theta}(\mathcal{H}|A_n)$ , the posterior distribution of genealogical trees given the random sample  $A_n$ , because this will result in equal importance weights and, consequently, the exact value of  $\pi_{\theta}(A_n)$ . Using the ‘particle representation’ of Donnelly and Kurtz (1996) for algorithm 2, Stephens and Donnelly (2000) developed a clever characterization of  $Q_{\theta}^*$  in terms of its backward transition probabilities and constructed a better trial distribution.

Both Griffiths and Tavaré’s and Stephens and Donnelly’s sampling distributions can be put in the framework of SIS since both have the form

$$q_{\theta}(\mathcal{H}) = \prod_{i=0}^{-(M-1)} q_{\theta}(H_{i-1}|H_i).$$

The weight function can be written in a sequential form as

$$w = \frac{P_{\theta}(\mathcal{H})}{Q_{\theta}(\mathcal{H})} = \frac{p_{\theta}(H_0|H_{-1}) \dots p_{\theta}(H_{-(M-1)}|H_{-M})}{q_{\theta}(H_{-1}|H_0) \dots q_{\theta}(H_{-M}|H_{-(M-1)})} c_1,$$

where  $c_1 = p_{\theta}(H_{-M})(n-1)/(n-\theta+1)$ . The numerator (including  $c_1$ ) is the probability of a certain genealogical tree under the actual distribution. The denominator is the probability of constructing the tree backwards by using the trial distribution. If we define the current weight  $w_{-t}$  as the weight that we have at time  $-t$  (for  $t \leq M$ ),

$$w_{-t} = \frac{p_{\theta}(H_0|H_{-1}) \dots p_{\theta}(H_{-(t-1)}|H_{-t})}{q_{\theta}(H_{-1}|H_0) \dots q_{\theta}(H_{-t}|H_{-(t-1)})} = w_{-(t-1)} \frac{p_{\theta}(H_{-(t-1)}|H_{-t})}{q_{\theta}(H_{-t}|H_{-(t-1)})},$$

we can update the current weight recursively; the final weight is

$$w = w_{-M} p_{\theta}(H_{-M})(n-1)/(n-\theta+1).$$

### 6.3. Resampling genealogical trees

Suppose that we simulate in parallel  $m$  genealogical trees according to  $Q_\theta(\mathcal{H})$ , i.e. we first generate  $m$  samples  $\{H_{-1}^{(1)}, \dots, H_{-1}^{(m)}\}$  from  $q_\theta(H_{-1}|H_0)$ . Then we recursively generate  $\{H_{-t}^{(1)}, \dots, H_{-t}^{(m)}\}$  from  $q_\theta(H_{-t}|H_{-t+1})$ , for  $t = 2, 3, \dots$ . The simulation stops when all the samples reach their MRCA. Standard resampling procedures (Liu and Chen, 1998), which stop all the  $m$  partially generated trees at the same sampling step  $t$  for resampling, do not work well because genealogical trees differ in their speeds of coalescence. At time  $-t$ , trees with large current weights tend to be those that are still very far from their MRCA. Their final weights when reaching the root tend to be very small. In contrast, trees with small current weights are often those that have advanced much more in their coalescence process and will result in large final weights. Resampling according to the natural time  $t$  actually prunes away many ‘good’ samples.

A more natural timescale for this problem is the number of coalescences because it determines how far the sample is from its MRCA. By defining the stopping time  $T_i$  as the time that a total of  $i$  coalescences have occurred in each sample (Fig. 3), we implemented the SISSTR method for this problem as follows. We wait until all the  $m$  parallel samples reach their first coalescence. The size of each sample at this point becomes  $n - 1$ . Then, we check  $cv^2$  for the weights and conduct resampling if  $cv^2$  is greater than a certain bound  $B$ . Otherwise we continue the usual sequential sampling for all the samples until they reach their second coalescence and check  $cv^2$  again. The process continues till all the samples reach their MRCA.

### 6.4. Data analysis under the coalescent model

We applied the SISSTR method to two data sets, under the coalescent model with fixed population size and compared its performance with existing importance sampling algorithms.

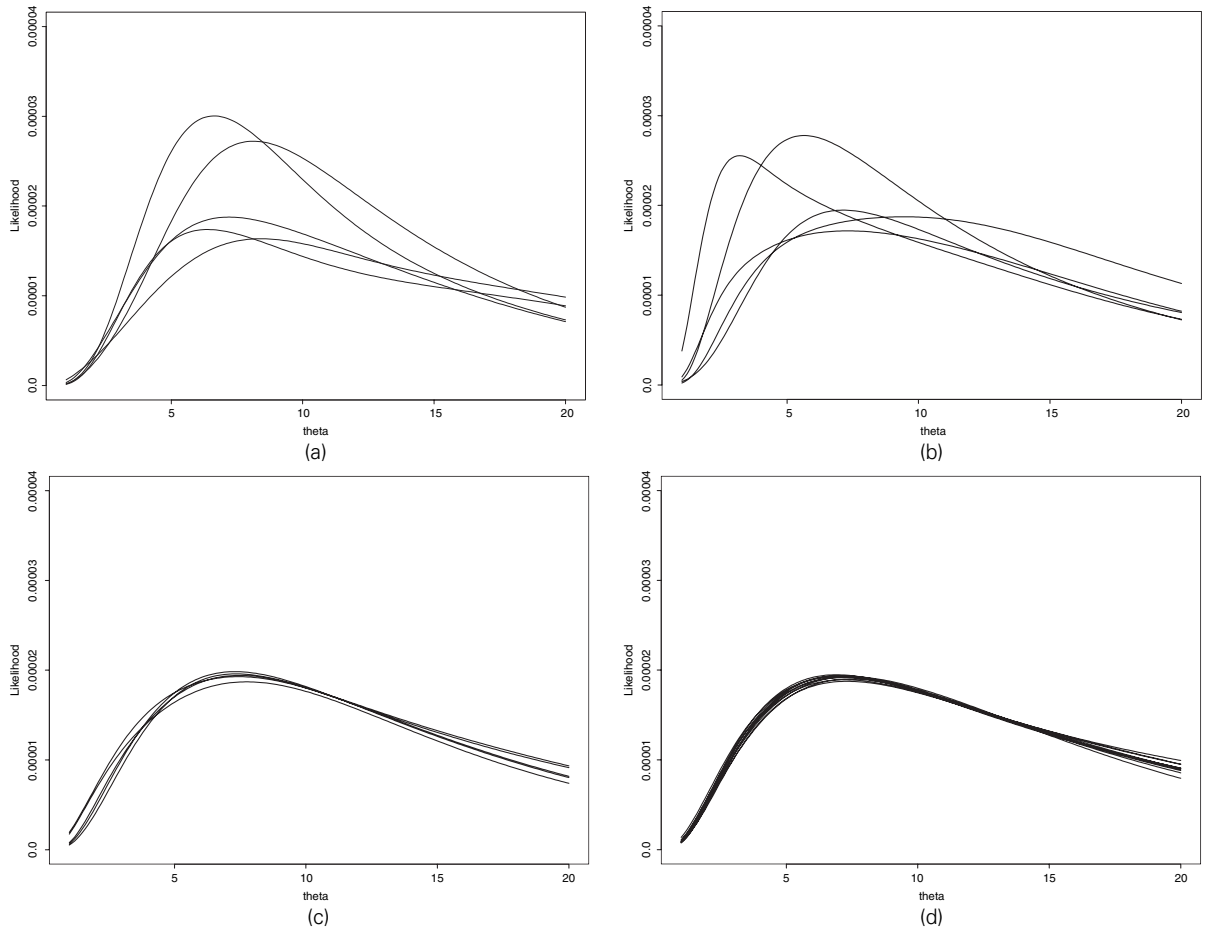
#### 6.4.1. An example from Stephens and Donnelly (2000)

The space of genetic types in this example is  $E = \{0, 1, \dots, 19\}$ , where  $i$  denotes an allele with  $i$  repetitions of a short deoxyribonucleic acid motif at microsatellite loci. The mutation model is a simple random walk, in which the number of repetitions increases or decreases by 1 with probability  $\frac{1}{2}$  each. For alleles with 0 or 19 repetitions, the mutation will only increase or decrease one repeat respectively. The data consist of alleles with repetitions  $A_n = \{8, 11, 11, 11, 11, 12, 12, 12, 12, 13\}$ .

Fig. 4(a) shows the estimated likelihood curves for  $\theta$  based on the Griffiths–Tavaré method without resampling, with  $\theta_0 = 10$ . We ran SIS five times, with each run based on  $m = 10000$  importance samples, which took about 2 min on a 1.2 GHz Athlon workstation. The large variability of the curves indicates that Griffiths and Tavaré’s trial distribution is not very efficient for this problem. Fig. 4(b) shows the estimated likelihood curves based on their method with standard resampling methods. We tested the dynamic schedule resampling method with different  $cv^2$  bounds, but none of these schedules helped much in this problem.

Fig. 4(c) displays the likelihood curves estimated from five runs of the SISSTR method based on Griffiths and Tavaré’s sampling distribution. With  $m = 10000$  and bound  $B = 4$ , two resampling steps were incurred in each of the five independent runs. The additional computational cost due to resampling was negligible. Fig. 4(c) is almost indistinguishable from Fig. 4(d), which resulted from the use of Stephens and Donnelly’s (2000) sampling distribution without resampling (with  $m = 10000$ ), demonstrating the significant effect of the new resampling method.

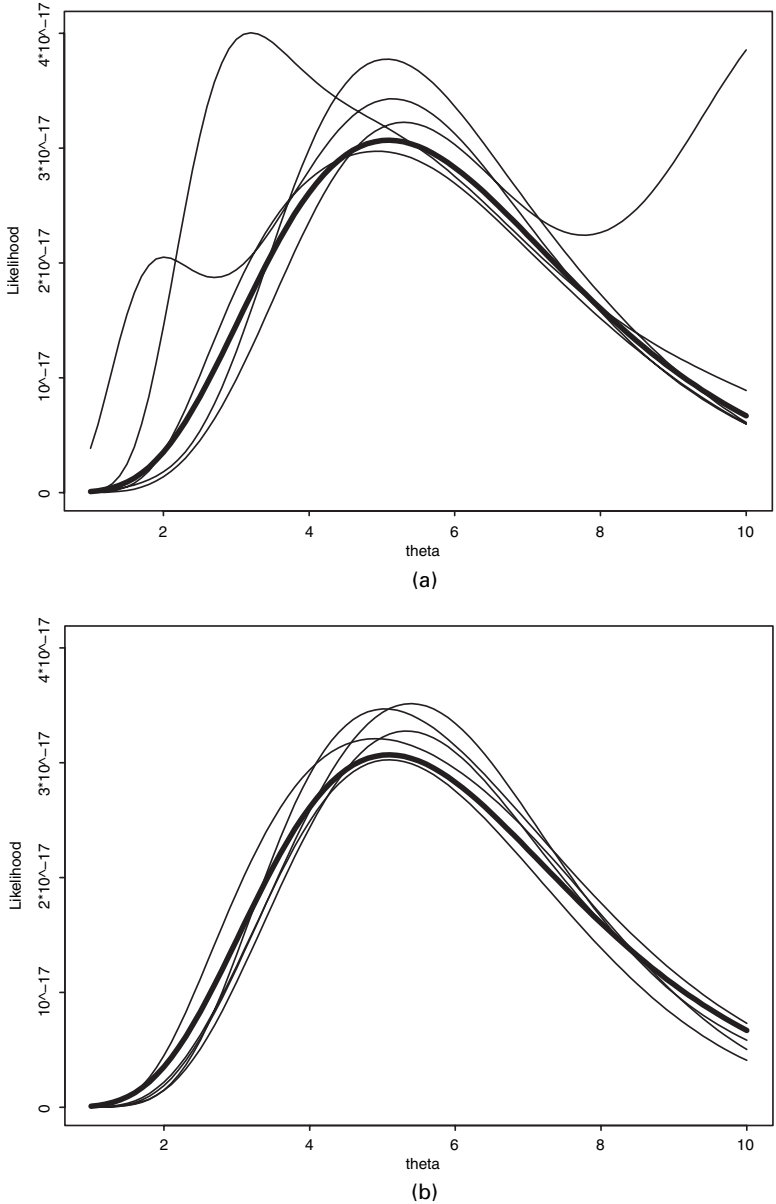
The performance of the Stephens–Donnelly algorithm is nearly perfect for this example, which makes it difficult to assess whether SISSTR can further improve it. We consider next a more complicated microsatellite example in which the Stephens–Donnelly algorithm does not perform as well.



**Fig. 4.** Comparison of estimated likelihood curves for the microsatellite data in Section 6.4.1: (a) five independent likelihood curve estimates based on 10 000 samples by using the Griffiths–Tavaré method without resampling; (b) five independent likelihood curve estimates based on 10 000 samples by using the Griffiths–Tavaré method with traditional resampling according to  $cv^2$ ; (c) five independent likelihood curve estimates based on 10 000 samples by using the Griffiths–Tavaré method with STR according to  $T_i$ ; (d) five independent likelihood curve estimates based on 10 000 samples by using the Stephens–Donnelly method without resampling

6.4.2. A larger data set

For a sample of 296 brown bears from the Western Brooks Range of Alaska, the allele counts at locus G10M are  $\{0, 0, 0, 0, 0, 24, 134, 16, 32, 81, 0, 8, 0, 1, 0, 0, 0, 0, 0\}$  (Paetkau *et al.*, 1997), in which number  $n$  in the  $i$ th ( $0 \leq i \leq 19$ ) position indicates that  $n$  bears have  $i + 98$  microsatellite repeats at that locus. Under the same setting as in Section 6.4.1, we are interested in estimating the likelihood function of  $\theta$ . We chose  $\theta_0 = 6$  in the following computation.



**Fig. 5.** Comparison of estimated likelihood curves for the microsatellite data in Section 6.4.2 (—, accurate estimate of the likelihood curve based on 1 million samples): (a) five independent likelihood curve estimates based on 10000 samples by using Stephens and Donnelly's (2000) method without resampling; (b) five independent likelihood curve estimates based on 10000 samples by using SISSTR

Fig. 5(a) shows five estimated likelihood curves by using the Stephens–Donnelly method without resampling with each curve based on  $m = 10000$  samples, which took about 3 min on a 1.2 GHz Athlon workstation.  $cv^2$  for these runs were between 100 and 9000. Fig. 5(b) gives five estimated likelihood curves after using STR, which took about the same amount of computing time. On average four resamplings were incurred for each run with  $B = 9$ . We also obtained an ‘accurate’ estimate of the likelihood curve (the bold curves in Figs 5(a) and 5(b)) based on 1 million samples by using the Stephens–Donnelly method without resampling, which took about 4 h. These figures indicate clearly the advantage of STR for this example.

We also tested a few different values of  $B$ . For  $B$  ranging from 1 to 99, the number of resamplings ranged from 1 to 15 and the performance of SISSTR did not change much, showing that the algorithm is robust to the choice of  $B$ . See Section 7 for more discussion.

## 7. Discussion

In this paper, we described a general sequential Monte Carlo method, SISSTR, and applied it to problems in population genetics and other fields. The examples in Sections 3, 5 and 6.4 demonstrate that substantial gains in efficiency can be obtained by the new method without incurring additional computational cost. The method can be applied not only to standard Monte Carlo estimation problems but also to solving linear equations and integral equations (Rubinstein (1981), chapter 5). In fact,  $\pi_\theta(A_n)$  in the coalescent model can be viewed as the solution to a system of linear equations (Sawyer *et al.*, 1987; Lundstrom *et al.*, 1992; Griffiths and Tavaré, 1994a). Griffiths and Tavaré’s algorithm was originally designed to solve this linear system numerically. When a resampling step is incurred at a stopping time, there are several ways to resample from the current partial sequences. We used the simple multinomial resampling scheme in the examples. The new STR strategy can also be used in conjunction with other resampling strategies, such as residual resampling or stratified resampling, to improve the efficiency.

In SISSTR as well as other resampling algorithms, there is a tuning parameter  $B$ , which is the threshold value for determining whether to do resampling at a certain stage.  $B = 0$  corresponds to the case that we do resampling at almost every check point, and  $B = \infty$  corresponds to SIS without resampling. We see from the examples that both SIS without resampling and that with too many resamplings can be inefficient. For all the examples in this paper, we found that the performance of SISSTR is insensitive to the choice of  $B$  within a reasonable range. In practice, we can run a few samples and see how  $cv^2$  for the importance weights behave, and then choose an appropriate bound so that a reasonable number of resamplings will be incurred. In the four examples that we studied, we incurred two, 10, two and four resamplings. For other problems in which the history is more quickly forgotten, such as certain non-linear space models (Doucet *et al.*, 2001), the number of resamplings incurred can be even larger.

Several ideas that are similar to resampling have been proposed in the population genetics literature. Griffiths and Tavaré (1994a) and Nielsen (1997) noticed that, in their coalescent simulations, discarding trees with too many mutations ‘seems particularly promising’. Stephens and Donnelly (2000) also pointed out that their importance sampling scheme may be viewed as sequential imputation of the ancestral states, and they suggested applying the rejection control idea (Liu *et al.*, 1999), which has a similar effect to that of resampling. The motivation behind resampling is to avoid wasting computational efforts on samples that will contribute very little to the final estimate. By resampling according to the weights of partial samples at certain stages, we implicitly assume that there is a ‘trend’ in the weight sequence: samples with small current weights are likely to have small final weights. If this is not true, then resampling can only have the adverse effect of increasing variability. The new resampling scheme that is developed in this

paper, which allows us to use a more appropriate timescale to pause the samples for resampling, is a way to capture the future trend more accurately.

The population genetics model that we have considered is the simplest evolutionary model. More realistic assumptions would include population structure, selection and recombination, which make it much more difficult to estimate the likelihood function. We expect that the insight that we have gained from the simple model will be helpful for studying these more complex problems. In fact, we have successfully applied the new SISSTR scheme to estimate the population growth rate and mutation rate in the infinite sites model for varying population size. The results are not presented here for brevity. Recently Fearnhead and Donnelly (2001) extended the Stephens–Donnelly method to estimating likelihood surfaces for the mutation and recombination rates. It is of interest to investigate whether an appropriate resampling scheme can be designed to improve estimation efficiency in this case.

## Acknowledgements

We thank Dylan Small and Simon Tavaré for their careful reading of our earlier drafts, and Robert Griffiths for sharing his codes. We are very grateful to the Associate Editor and two referees for helpful suggestions. This work was supported in part by National Science Foundation grants DMS-0203762 and DMS-0204674, National Institutes of Health grant R01-HG02518-01 and an Arts and Sciences Research Council grant from Duke University.

## References

- Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997) Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Am. Statist. Ass.*, **92**, 1403–1412.
- Chen, Y. (2001) Sequential importance sampling with resampling: theory and applications. *PhD Dissertation*. Department of Statistics, Stanford University, Stanford.
- Chen, Y. and Liu, J. S. (2000) Discussion on ‘Inference in molecular population genetics’ (by M. Stephens and P. Donnelly). *J. R. Statist. Soc. B*, **62**, 644–645.
- Chen, Y., Small, D. and Xie, J. (2004) Analysis of non-Markovian panel studies via Monte Carlo methods. To be published.
- Cox, D. R. and Isham, V. (1981) *Point Processes*. New York: Chapman and Hall.
- De Stavola, B. L. (1988) Testing departures from time homogeneity in multistate Markov processes. *Appl. Statist.*, **37**, 242–250.
- Donnelly, P. and Kurtz, T. G. (1996) A countable representation of the Fleming-Voit measure-valued diffusion. *Ann. Probab.*, **24**, 698–742.
- Donnelly, P. and Tavaré, S. (1995) Coalescents and genealogical structure under neutrality. *A. Rev. Genet.*, **29**, 401–421.
- Doucet, A., de Freitas, N. and Gordon, N. (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.
- Ewens, W. J. (1972) The sampling theory of selectively neutral alleles. *Theoret. Popln Biol.*, **3**, 87–112.
- Ewens, W. J. (1979) *Mathematical Population Genetics*. Berlin: Springer.
- Farlow, S. J. (1993) *Partial Differential Equations for Scientists and Engineers*. New York: Dover Publications.
- Fearnhead, P. and Clifford, P. (2003) On-line inference for hidden Markov models via particle filters. *J. R. Statist. Soc. B*, **65**, 887–899.
- Fearnhead, P. and Donnelly, P. (2001) Estimating recombination rates from population genetic data. *Genetics*, **159**, 1299–1318.
- Felsenstein, J., Kuhner, M. K., Yamato, J. and Beerli, P. (1999) Likelihoods on coalescents: a Monte Carlo sampling approach to inferring parameters from population samples of molecular data. In *Statistics in Molecular Biology and Genetics* (ed. F. Seillier-Moiseiwitsch). Hayward: Institute of Mathematical Statistics and American Mathematical Society.
- Godsill, S., Doucet, A. and West, M. (2000) Monte Carlo smoothing for non-linear time series. *Discussion Paper 00-01*. Institute of Statistics and Decision Sciences, Duke University, Durham.
- Gordon, N. J., Salmon, D. J. and Smith, A. F. M. (1993) A novel approach to non-linear and non-Gaussian Bayesian state estimation. *IEEE Proc. Radar Signal Process.*, **140**, 107–113.

- Grassberger, P. (1997) Pruned-enriched Rosenbluth method: simulations of  $\theta$  polymers of chain length up to 100000. *Phys. Rev. E*, **56**, 3682–3693.
- Griffiths, R. C. and Tavaré, S. (1994a) Simulating probability distributions in the coalescent. *Theoret. Popul. Biol.*, **46**, 131–159.
- Griffiths, R. C. and Tavaré, S. (1994b) Ancestral inference in population genetics. *Statist. Sci.*, **9**, 307–319.
- Griffiths, R. C. and Tavaré, S. (1994c) Sampling theory for neutral alleles in a varying environment. *Phil. Trans. R. Soc. Lond. B*, **344**, 403–410.
- Hammersley, J. M. and Handscomb, D. C. (1964) *Monte Carlo Methods*. New York: Wiley.
- Hammersley, J. M. and Morton, K. W. (1954) Poor man's Monte Carlo. *J. R. Statist. Soc. B*, **16**, 23–38.
- Hartl, D. L. and Clark, A. G. (1997) *Principles of Population Genetics*. Sunderland: Sinauer.
- Irwing, M., Cox, N. and Kong, A. (1994) Sequential imputation for multilocus linkage analysis. *Proc. Natn. Acad. Sci. USA*, **91**, 11684–11688.
- Isard, M. and Blake, A. (1996) Contour tracking by stochastic propagation of conditional density. In *Proc. Eur. Conf. Computer Vision, Cambridge*, pp. 343–356. London: Springer.
- Jovanovic, B. (1979) Job matching and the theory of turnover. *J. Polit. Econ.*, **87**, 972–990.
- Kalbfleisch, J. D. and Lawless, J. F. (1985) The analysis of panel data under a Markov assumption. *J. Am. Statist. Ass.*, **80**, 863–871.
- Kingman, J. F. C. (1982) On the genealogy of large populations. *J. Appl. Probab. A*, **19**, 27–43.
- Kitagawa, G. (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Statist.*, **5**, 1–25.
- Kong, A., Liu, J. S. and Wong, W. H. (1994) Sequential imputations and Bayesian missing data problems. *J. Am. Statist. Ass.*, **89**, 278–288.
- Kremer, K. and Binder, K. (1988) Monte Carlo simulation of lattice models for macromolecules. *Comput. Phys. Rep.*, **7**, 259–310.
- Kuhner, M. K., Yamato, J. and Felsenstein, J. (1995) Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics*, **140**, 1421–1430.
- Lippman, S. and McCall, J. (1976) The economics of job search: a survey. *Econ. Inq.*, **14**, 113–126.
- Liu, J. S. (1996) Nonparametric hierarchical Bayes via sequential imputations. *Ann. Statist.*, **24**, 911–930.
- Liu, J. S. (2001) *Monte Carlo Strategies for Scientific Computing*. New York: Springer.
- Liu, J. S. and Chen, R. (1995) Blind deconvolution via sequential imputations. *J. Am. Statist. Ass.*, **90**, 567–576.
- Liu, J. S. and Chen, R. (1998) Sequential Monte-Carlo methods for dynamic systems. *J. Am. Statist. Ass.*, **93**, 1032–1044.
- Liu, J. S., Chen, R. and Wong, W. H. (1999) Rejection control and sequential importance sampling. *J. Am. Statist. Ass.*, **93**, 1022–1031.
- Lundstrom, R., Tavaré, S. and Ward, R. H. (1992) Modeling the evolution of the human mitochondrial genome. *Math. Biosci.*, **112**, 319–335.
- MacEachern, S. N., Clyde, M. and Liu, J. S. (1999) Sequential importance sampling for nonparametric Bayes models: the next generation. *Can. J. Statist.*, **27**, 251–267.
- Markovtsova, L., Marjoram, P. and Tavaré, S. (2000) The age of a unique event polymorphism. *Genetics*, **156**, 401–409.
- Nielsen, R. (1997) A likelihood approach to population samples of microsatellite alleles. *Genetics*, **146**, 711–716.
- Nordborg, M. (2001) Coalescent theory. In *Handbook of Statistical Genetics* (eds D. J. Balding, M. Bishop and C. Cannings). Chichester: Wiley.
- Paetkau, D., Waits, L. P., Clarkson, P. L., Craighead, L. and Strobeck, C. (1997) An empirical evaluation of genetic distance statistics using microsatellite data from bear (Ursidae) populations. *Genetics*, **147**, 1943–1957.
- Pitt, M. and Shephard, N. (1999) Filtering via simulation: auxiliary particle filters. *J. Am. Statist. Ass.*, **94**, 590–599.
- Rosenbluth, M. and Rosenbluth, A. (1955) Monte Carlo calculation of the average extension of molecular chains. *J. Chem. Phys.*, **23**, 356–359.
- Rubinstein, R. (1981) *Simulation and the Monte Carlo Method*. New York: Wiley.
- Sawyer, S., Dykhuizen, D. and Hartl, D. (1987) Confidence interval for the number of selectively neutral amino acid polymorphisms. *Proc. Natn. Acad. Sci. USA*, **84**, 6225–6228.
- Stephens, M. and Donnelly, P. (2000) Inference in molecular population genetics. *J. R. Statist. Soc. B*, **62**, 605–635.
- Wall, F. T. and Erpenbeck, J. (1959) New method for the statistical computation of polymer dimensions. *J. Chem. Phys.*, **30**, 634–637.
- Wilson, I. J. and Balding, D. J. (1998) Genealogical inference from microsatellite data. *Genetics*, **150**, 499–510.