Available on-line at http://dse.usab-tm.ro
Banat's Journal of Biotechnology
2011, II(4),

THE SIS ALGORITHM AND ITS APPLICATIONS

Andrei Băncilă, Mihaela Păun, Ştefan Popescu, Laura Păun, George Roată, Iris Mateescu, Marian Buțu, Andrei Păun, Manuela Sidoroff

Bioinformatics Department, National Institute of Research and Development for Biological Sciences, Splaiul Independentei, Nr. 296, Sector 6, Bucharest, Romania, e–mail: *andreibancila@yahoo.com*

Abstract. A systematic use of the *Monte Carlo* method appeared since the early days of electronic computing and since then it is more present in different scientific research fields. Therefore, many techniques were developed based on this method and one of them is called *sequential importance sampling*. This technique is an adaptation of the Monte Carlo method that can be used to better extract samples form the domain using an importance weight function.

Key words: importance sampling, sequencial Monte Carlo, inference, importance weight

What is sequential importance sampling (SIS)

Importance sampling

For better understanding of this idea it is best to introduce an example at first.
Suppose that we want to evaluate a quantity

$$\theta = \int_{\chi} h(x)\pi(x)dx$$

where the support of the random variable *X* is denoted by χ and $h(x) \geq 0$. If we were to resolve this by a numeric method, we would discretize the domain $\mathcal{X}$ by regular grids and then we evaluate $h(x)\pi(x)$ on each of the grid points, and the use the Riemann sum as the approximation.

If we consider a target function

$$f(x,y) = 0.5e^{-90(x-0.5)^2-45(y+0.1)^4} + e^{-45(x+0.4)^2-60(y-0.5)^2}$$

where $(x,y) \in [-1,1] \times [-1,1]$, more than two–third of computing time are wasted on evaluating the points on which the function is almost zero. So this is wasted time and resources. If the space χ increases, then the situation will deteriorate more.

If we are to take random samples $\left(x^{(1)}, y^{(1)}, \ldots, x^{(m)}, y^{(m)}\right)$ uniformly in $[-1,1] \times [-1,1]$, we implemented a vanilla Monte Carlo algorithm to estimate the integral $\iint f(x,y)dxdy$.

Because the density for the sampling distribution is a constant, ¼ in the region, the estimate of the integral was produce as:

$$\hat{\mu} = \frac{4}{m}\left\{f^{(1)} + \ldots + f^{(m)}\right\}$$

where $f^{(i)} = f\left(x^{(i)}, y^{(i)}\right)$. With m=2,500, we obtained $\hat{\mu} = 0.1307$, with a standard deviation 0.009, which was estimated by

$$std(\hat{\mu}) = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{m}(f_i - \hat{\mu})^2}\,.$$

We can see that this method has the same problem as the one before, time is wasted for evaluation of samples located in unimportant regions (the function is almost zero).

In high–dimensional models from statistical physics, molecular simulation sand Bayesian statistics, because the region in which the function is not zero is meaningfully compared to the whole space, the vanilla Monte Carlo schemas are bound to fail.

Importance sampling suggests that the focus should be on the region(s) of importance so that we may save as many computational resources as we can [MARSHALL, 1956]. If the domain is small this seems rather unimportant but if the domain is large and the function takes values in only a small portion of that domain, the idea of biasing toward importance regions (the ones in where the function take values) becomes essential for Monte Carlo computation.

Suppose we want to evaluate

$$\mu = E_{\pi}\{h(x)\} = \int h(x)\pi(x)dx,$$

we can apply the following procedure that represents a simple form of the *importance sampling algorithm*:

a) we draw $x^{(1)},\ldots,x^{(m)}$ from a trial distribution ;

b) we calculate the *importance weight*

$$\omega^{(j)} = \pi\left(x^{(j)}\right) \Big/ g\left(x^{(j)}\right), \text{ for } j = 1,\ldots,m.$$

c) we approximate μ by

$$\hat{\mu} = \frac{\omega^{(1)}h\left(x^{(1)}\right)+\ldots+\omega^{(m)}h\left(x^{(m)}\right)}{\omega_1+\ldots+\omega_m} \quad 1.1$$

Practically if we want to make the estimation error small, we should choose *g(x)* as close as possible to $\pi(x)h(x)$. A major advantage in using (1.1) instead of the unbiased estimate

$$\hat{\mu} = \frac{1}{m}\left\{\omega^{(1)}h\left(x^{(1)}\right)+\ldots+\omega^{(m)}h\left(x^{(m)}\right)\right\} \quad 1.2$$

is that we only need to know the ratio $\pi(x) \Big/ g(x)$ and this estimate often has a smaller mean square error than the unbiased one.

Another scenario for resorting to importance sampling is when we want to generate i.e. random samples from π but doing so directly is infeasible. In this case, we can generate random samples from a different, but similar, trivial distribution $g(\cdot)$, and then correct the bias using the importance sampling procedure. Similar to the rejection method the importance sampling requires that the sampling distribution *g* is reasonable close to π (i.e. that *g* has a longer tail then π).

Sequential Importance Sampling

It is nontrivial to design a good trial distribution for doing importance sampling in high–dimensional problems. One of the most useful strategies in these problems is to build up the trial density sequentially. Suppose we can decompose x as $x = (x_1,\ldots,x_d)$ where each of the x, may be multidimensional. Then, our trial density can be constructed as

$$g(x) = g_1(x_1)g_1(x_2|x_1)\ldots g_1(x_d|x_1,\ldots,x_{d-1}) \quad 1.3$$

by which we hope to obtain some guidance from the target density while building up the trial density. Corresponding to the decomposition of x, we can rewrite the target density as

$$\pi(x) = \pi_1(x_1)\pi_1(x_2|x_1)\ldots\pi_1(x_d|x_1,\ldots,x_{d-1}) \quad 1.4$$

and the importance weight as

$$\omega(x) = \frac{\pi_1(x_1)\pi_1(x_2|x_1)\ldots\pi_1(x_d|x_1,\ldots,x_{d-1})}{g_1(x_1)g_1(x_2|x_1)\ldots g_1(x_d|x_1,\ldots,x_{d-1})} \quad 1.5$$

If the importance weight is calculated this way, we suggest a recursive way of computing and monitoring the importance weight by denoting $x = (x_1,\ldots,x_d)$ we have

$$\omega_t(x_t) = \omega_{t-1}(x_{t-1})\frac{\pi\left(x_t|x_{t-1}\right)}{g\left(x_t|x_{t-1}\right)}$$

At the end, ω_d is equal to $\omega(x)$ in (1.5). Potential advantages can be:

a. we can stop generating further components of x if the *partial weight* derived from the sequentially generated *partial sample* is too small;
b. we can take advantage of $\pi\left(x_t|x_{t-1}\right)$ in designing $g\left(x_t|x_{t-1}\right)$.

In other words, the marginal distribution $\pi(x_t)$ can be used to guide the generation of *x*.

This method is interesting but the problems appear when we must calculate the decomposition of π in (1.4) and of ω as in (1.5). The problem is that we need to have the marginal distribution

$$\pi(x_t) = \int \pi(x_1,\ldots,x_d)dx_{t+1}\ldots dx_d$$

whose computation involves integrating out components $x_{t+1},\ldots,x_d$ in $\pi(x)$ and is as difficult as the original problem.

Suppose we can find a sequence of auxiliary distributions $\pi_1(x_1)\pi_2(x_2),\ldots,\pi_d(x_d)$, so that $\pi_t(x_t)$ is a reasonable approximation to the marginal distribution $\pi(x_t)$ for $t = 1,\ldots,d-1$ and $\pi_d = \pi$. We want to reach the conclusion that it is necessary to know x_t up to a normalizing constant and they represent guides to our construction of the whole sample $x = (x_1,\ldots,x_d)$.

A recursive procedure that defines sequential importance sampling (SIS) is:

a. draw $X_t = x_t$ from

$g_t(x_t|x_{t-1})$ and let $x_t=(x_{t-1},x_t)$,

b. compute

$$u_t=\frac{\pi_t(x_t)}{\pi_{t-1}(x_{t-1})g_t(x_t|x_{t-1})},$$

and let $\omega_t=\omega_{t-1}u_t$.

In the SIS step, we can call u_t an incremental weight. It is easy to show that x_t is properly weighted by ω_t with respect to π_t provided that x_{t-1} is properly weighted by ω_{t-1} with respect to π_{t-1}. We can say that the sample x is properly weighted by the final importance weight ω_d with respect to the target density $\pi(x)$. The sequential buildup of the trial density breaks a difficult task into manageable pieces. We can use the sequence of auxiliary distributions $\pi_1,\pi_2,\ldots,\pi_d$ to help construct more efficient trial distribution:

- we can build g_t in light of π_t

.

Example: we can choose: $g_t(x_t|x_{t-1})=\pi_t(x_t|x_{t-1})$

Than the incremental weight becomes:

$$u_t=\pi_t(x_t)/\pi_{t-1}(x_{t-1})$$

- when ω_t is getting too small, we can chose to reject the sample when we want and restart again, this way we can save time. However by rejecting a sample a bias can incur so we can use a rejection control technique to correct the bias.

Rejection control in sequential importance sampling

Suppose a sequence of check points $0<t_1<t_2<\ldots<t_k\le d$ and a sequence of threshold values $c_1,\ldots,c_k$ are given in advance. The following procedure can be implemented:

1) at each check point t_j start $RC(t_1)$ with the threshold value $c=c_j$. If the partial sample $(x_1,\ldots,x_t)$ has a weight ω_t, then we accept this partial sample with probability $\min\{1,\omega_{t_j}/c_j\}$ and replace its weight by $\omega_{t_j}^*=\max\{\omega_{t_j},c_j\}$ if accepted.
2) For each partial sample that is rejected, restart from the beginning let it pass through all check points at $t_1,\ldots,t_j$, with threshold values $c_1,\ldots,c_j$ and if rejected in any middle check point start again.

After the first rejection control at stage t_1, the sampling distribution $g_t^*(x_t)$ for X_t is no longer the same as the one described in (1.3). It is sown by Liu, Chen and Wong (1998) that for time t, partial sample x_t that result from the procedure is properly weighted with respect to π_t by their modified ω_j^*.

Since this method requires that each rejected sample be restarted from stage *0*, it tends to be impractical when the number of components *d* is large.

Sis application

Application of SIS in Population Genetics

The evolutionary theory states that stochastic mutational events may alter the genome and these changes may pass on to the progeny. If we compare homologous DNA regions from a random sample of individuals taken from certain population can shed light on the evolutionary process of this population.

This comparison can also yield important information for locating genes that are responsible for genetic diseases.

Recent advances in the biotechnology revolution have provided a wealth of DNA sequence data for which meaningful studies on the evolution process can be made and biologically verified.

We present a simple demographic model, focusing on populations of constant size *N* which evolve in non–overlapping generations.

Each individual in a population is a sufficiently small chromosomal region in which no recombination is allowed (in reality, recombination can happen with a very small probability).

Thus, each chromosomal segment seen in the dataset can be treated as a

descendent of a single parental segment in the previous generation (it is sufficient to consider that each individual has one parent).

Each segment has a genetic type and the set of all possible types is denoted as *E*.

If a parental segment is of type $\alpha \in E$, then its progeny is of type $\alpha \in E$ with probability $1-\mu$ and of type $\beta \in E$ with probability $\mu P_{\alpha\beta}$.

Thus, μ can be seen as the mutation rate per chromosome per generation.

The mutation transition matrix $P=(P_{\alpha\beta})$ is assumed to have a unique stationary distribution.

After observing a random sample from the current population, assumed to be at stationary, we can represent the ancestral relationships by a tree.

We can represent a tree for a segment that has only two genetic types, C and T and consider five observations.

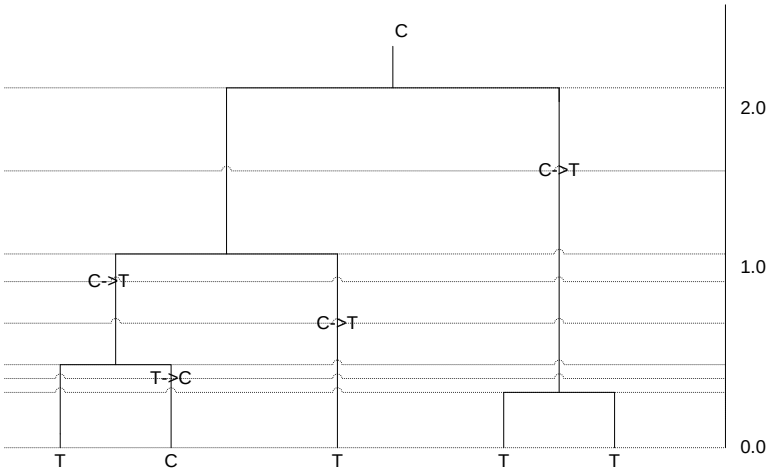

Figure 1. Illustration of a genealogical tree

In *Figure 1*, the set contains five observations at the current time {T, C, T, T, T}. The tree represents a possible route from these five individuals to descend from a common ancestor of type C.

Ancestral lineages are joined by horizontal lines when they share a common ancestor.

The dots represent mutations and the horizontal dotted lines indicate the times at which the events occur. The history $H=(H_{-k},H_{-(k-1)},\ldots,H_{-1},H_0)$ in this case is

$$\{C\},\{C,C\},\{C,T\},\{C,C,T\},\{C,T,T\},\{T,T,T\},\{T,T,T,T\},$$
$$\{C,T,T,T\},\{C,T,T,T,T\}.$$

Stephens and Donnelly (2000) used $H=(H_{-k},H_{-(k-1)},\ldots,H_{-1},H_0)$ to denote the whole ancestral history (unobserved) of the observed individuals at the present time, where k is the first time when all the individuals in the sample coalesce (i.e., the first time we find a common ancestor). H_{-i} is an unordered list of genetic types of the ancestors i generations ago.

Thus, the history H has a one–to–one correspondence with the tree topology (evolution time is not reflected in H). Note that only H_0 is observable. For any given H that is compatible with H_0, however, we can compute the likelihood function $p_\theta(H)$ as

$$p_\theta(H)\propto p_\theta(H_{-k})p_\theta(H_{-k+1}|H_{-k})\ldots p_\theta(H_0|H_{-1})p_\theta(stop|H_0)$$

.

The stationary distribution of P is π_0 because $p_\theta(H_{-k})=\pi_0(H_{-k})$.

The coalesce theory [GRIFFITHS *et al.*, 1994, STEPHENS *et al.*, 2000] tells us that

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

for $i=-(k-1),\ldots,0$ and the process is stopped before a new genetic type is produced

Available on-line at http://dse.usab-tm.ro
Banat's Journal of Biotechnology
2011, II(4),

$$p_\theta(stop|H_0)=\sum_\alpha \frac{n_\alpha}{n}\frac{n-1}{n-1+\theta}$$

Here, n is the sample size at generation H_{i-1} and n_α is the number of chromosomes of type α in the sample. The notation $H_i = H_{i-1}+\alpha$ indicates that the new generation H_i is obtained from H_{i-1} by a split of a line of type α, and the notation $H_i = H_{i-1}-\alpha+\beta$ means that H_i is obtained from H_{i-1} by a mutation from a type α to a type β.

The parameter $\theta = 2N\mu/\nu$, with N being the population size (assumed to be constant throughout the history) and ν^2 being the variance of the number of progeny of a random chromosome. To find the value of θ, one can use the *maximum likelihood estimate* (MLE) method, which requires us to compute for each given B the likelihood value

$$p_\theta(H_0)=\sum_{H:compatible\,with\,H_0} p_\theta(H)$$

This computation cannot be solved analytically and we use some approximation methods like Monte Carlo. In a native Monte Carlo, one may randomly choose the generation number k and then simulate forward from H_{-k}, which only has a single individual, too H_0.

However, except for trivial dataset, such simulated history H has little chance to be compatible with the observed H_0. An alternative strategy is to simulate H backward starting from H_0 and then use weight to correct bias. In a sequential importance sampling strategy we can simulate $H_{-1}, H_{-2}, \ldots$ from a trial distribution built up sequentially by reversing the forward sampling probability at a fixed θ_0; that is, for $t=1,\ldots,k$, we have

$$g_t(H_t|H_{-t-1})=\frac{p_{\theta_0}(H_{-t+1}|H_{-t})}{\sum_{H'_{-t}} p_{\theta_0}(H_{-t+1}|H'_{-t})}$$

and the final trial distribution

$$g(H)=g_1(H_{-1}|H_0)\ldots g_t(H_k|H_{-k+1})$$

In other words, each g_t is the local posterior distribution of H_{-t} under a uniform prior, conditional on H_{-t+1}. By simulating from $g(\cdot)$ multiple copies of the history, $H^{(i)}$, $j=1,\ldots,m$ we can approximate the likelihood function as

$$p_\theta(H_0)=\frac{1}{m}\sum_{j=1}^{m}\frac{p_\theta(H^{(j)})}{g(H^{(j)})}$$

In this approach, the choice of θ_0 can influence the final result. We tested this importance sampling method on a small test dataset in Stephens and Donnelly [STEPHENS *et al.*, 2000], *{8, 11, 11, 11, 11, 12, 12, 12, 12, 13}*, with $E=\{0,1,\ldots,19\}$ and a simple random walk mutation transition on E. Stephens and Donnelly (2000) recently proposed a new SIS construction of the trial distribution and are significantly better than the simple construction described in this section.

Inference in Population Genetics

The presented model can be improved by the general method of resampling that improves the computation [CHEN *et al.*, 200b].

As many SIS applications, the trial distribution for simulating the evolutionary history H has the form:

$$g(H)=\prod_{t=1}^{k} g_t(H_{-t}|H_{-t+1})$$

We define the weight (for $t\le k$) for the trial density:

$$\omega_{-t}=\frac{p_\theta(H_{-t+1}|H_{-t})\ldots p_\theta(H_0|H_{-1})}{g_t(H_{-t}|H_{-t+1})\ldots g_1(H_{-1}|H_0)}\equiv\omega_{-t+1}\frac{p_\theta(H_{-t+1}|H_{-t})}{g_t(H_{-t}|H_{-t+1})}$$

The final weight is then $\omega=\omega_{-k}p_\theta(H_{-k})p_\theta(stop|H_0)$ where:

$$p_\theta(stop|H_0)=\sum_\alpha \frac{n_\alpha}{n}\frac{n-1}{n-1+\theta}$$

If we want to implement SIS then we first generate the m samples from $q_\theta(H_{-1}|H_0)$, we generate $\{H_{-t}^{(1)},\ldots,H_{-t}^{(m)}\}$, called the current sample, for $t=2,3,\ldots$ until unifies in all m process. We now can monitor the weight and, at any time t, when the coefficient of variation in $\{\omega_{-t}^{(1)},\ldots,\omega_{-t}^{(m)}\}$ exceeds a threshold apply additional steps of resampling. In resampling we produce a new current sample by drawing with replacement from $\{H_{-t}^{(1)},\ldots,H_{-t}^{(m)}\}$ according to the

probability $\propto \left\{\omega_{-t}^{(1)},\ldots,\omega_{-t}^{(m)}\right\}$. The weight for each new sample is set as the *sample average* of $\omega_{-t}^{(j)}$ this way assuring that in the end we obtain a proper estimate of the likelihood function.

It is inefficient to resample among $\left\{H_{-t}^{(1)},\ldots,H_{-t}^{(m)}\right\}$ because these samples differ greatly in their convergence speeds. In the small population sizes we start with small current weights and finish with large final weights. If we resample among $H_{-t}^{(m)}$ we throw out many good samples.

A solution can be resampling at the same coalescence time instead of the same sequential sampling time. This means we wait until all the processes reach the same population size, and then resample from $\left\{H_{-i_1}^{(1)},\ldots,H_{-i_m}^{(m)}\right\}$, where

$$i_j = \min\left\{t : \left|H_{-t}^{(j)}\right| = i\right\}$$

$\left|H_{-t}^{(j)}\right|$ represents the population size of that generation. We need to keep the early histories of the processes that survive the resampling.

The resampling procedure can be implemented as follows. Suppose that we start m parallel processes, than for $i = n-1,\ldots,1$:

- for $j = 1,\ldots,m$, we run the j process until the population size first reaches i; mark this time as i_j;
- compute the coefficient of variation for $\omega_{-i_1}^{(1)},\ldots,\omega_{-i_m}^{(m)}$ as CV_i;
- we resample among $H_{-i_1}^{(1)},\ldots,H_{-i_m}^{(m)}$ when CV_i is greater than the threshold, otherwise continue the usual sampling sequence;
- the weight for each sample after resampling is set as the sample average of the $\omega_{-i_j}^{(j)}$.

At the end of this procedure, we produce a sample of histories $H^{(*j)}$, $j = 1,\ldots,m$, and their associated weights $\omega^{(*j)}$, $j = 1,\ldots,m$. We use the sample average of these weights

$$\frac{1}{m}\left(\omega^{(*1)} + \ldots + \omega^{(*m)}\right)$$

to estimate the likelihood function $p_\theta(H_0)$. If we wanted to estimate $p_{\theta'}(H_0)$ for $\theta' = \theta$, we can use a modified weighted average:

$$p_{\theta'}(H_0) = \frac{1}{m}\left[\omega^{(*1)}\frac{p_{\theta'}\left(H^{(*1)}\right)}{p_\theta\left(H^{(*1)}\right)} + \ldots + \omega^{(*m)}\frac{p_{\theta'}\left(H^{(*m)}\right)}{p_\theta\left(H^{(*m)}\right)}\right]$$

Cheng and Liu applied this modified resampling step to the trial distribution

Molecular Simulation

Chemists and structural biologists have developed numerous lattice–based models, and others more complicated, to predict native structures of important macromolecules, such as protein molecules.

One of the most important problems is the *protein folding problem* in which it is required to predict the three–dimensional fold shape of a protein molecule based only on the sequence of amino acids. One simple model is a 2–D or 3–D lattice–bead model.

These models have identical structures as some SAW models but they usually use a more complicated function for interactive energy between the beds on the lattice. These beads correspond to amino acid residue, which are of 20 different types (letters of the alphabet).

A simple model used by Unger and Moult [UNGER *et al.*, 1993] has two different kinds of beads, white and black.

They correspond to hydrophilic and hydrophobic residue. This is an example:

WWWBBWWBBWWWWWBBBBBBBWWBBWWWWBBWWBWW

The scope is to find out the most "favorable fold" in a 2–D lattice space.

This means that we have to define an energy function for each configuration of this bead sequence:

$$U_n = -\sum_{|i-j|>1} c(x_i, x_j)$$

where $c(x_i, x_j) = 1$ if x_i and x_j are non–bonding neighbors and the identities of beads i and j are both black (hydrophobic), and $c(x_i, x_j) = 0$ otherwise.

Clearly these simple models favor the close packing of hydrophobic residue and therefore mimic some aspects of a real protein fold.

We apply SIS with various

modifications (resampling, rejection control and one–step–look–ahead) with the target distribution

$$\pi_n(x_n) \propto \exp\{-U_n(x_n)/2\}$$

on all examples from Unger and Moult (1993). Comparing the result of Bastolla, Frauenkron, Gerstner, Grassberger and Nadler (1998) with the results of Unger and Moult (1993) we find an equal or better minimum energy.

Actually [BASTOLLA *et. al.*, 1998] applied the SIS method with pruning and enrichment modifications to the examples of Unger and Moult and many others and achieved some excellent results compared to these.

We represent the configurations that resulted and noticed that for 36 atoms, 14 were found, for 60 atoms, 36 were found meanwhile the Unger and Moult found only 34 out of 60 atoms.

An additional experiment shows that when repeated 125 times, the SIS algorithm provides the normalizing constant $\log(Z) = 89.65$ and the mean squared extension $E_\pi(R^2) \approx 157$.

Resampling and partial rejection techniques can be used to obtain better and accurate results.

Also the two–step–look–ahead strategy can be used to greatly increase simulation efficiency.

Used on a chain with $N = 100$ particles we obtain $\log Z_n \approx 96.39 \pm 0.037$.

The article was funded by projects BIODIV 105

References

1. Liu J. S., **2001**, *Monte Carlo Strategies in Scientific Computing*, Springer Series in Statistics, New York
2. Stephens M., Donnelly P., **2000**, *Inference in molecular population genetics*, Journal of the Royal Statistical Society, Series B
3. Griffiths R. C., Tavere S., **1994**, *Simulating probability distributions in the coalescent*, Theoretical Population Biology
4. Liu J. S., Cheng R., Wond W. H., **1998**, Rejection control and sequential importance sampling and resampling, *in A. Doucet*, DE FREITAS J. F. G., N. J. Gordon (eds), *Sequential Monte Carlo in Practice*, Springer–Verlag, New York;
5. Chen Y., Liu J. S., **2000**, *Discussion of the paper by Stephens and Donnelly*, Journey of the Royal Statistical Society, Series B

Received: May 17, 2011
Accepted: September 10, 2010