Efficient Use of Exact Samples

by

Duncan J. Murdoch* and Jeffrey S. Rosenthal**

(June, 1998; revised April, 1999)

Abstract

Propp and Wilson (1996,1998) described a protocol called *coupling from the past* (CFTP) for exact sampling from the steady-state distribution of a Markov chain Monte Carlo (MCMC) process. In it a past time is identified from which the paths of coupled Markov chains starting at every possible state would have coalesced into a single value by the present time; this value is then a sample from the steady-state distribution.

Unfortunately, producing an exact sample typically requires a large computational effort. We consider the question of how to make efficient use of the sample values that are generated. In particular, we make use of *regeneration events* (cf. Mykland *et al.*, 1995) to aid in the analysis of MCMC runs. In a regeneration event, the chain is in a fixed reference distribution; this allows the chain to be broken up into a series of tours which are independent, or nearly so (though they do *not* represent draws from the true stationary distribution).

In this paper we consider using the CFTP and related algorithms to create tours. In some cases their elements are exactly in the stationary distribution; their length may be fixed or random. This allows us to combine the precision of exact sampling with the efficiency of using entire tours.

Several algorithms and estimators are proposed and analysed.

Keywords. *coupling from the past; exact sampling; perfect sampling; regeneration; tours.*

---

* Department of Statistical and Actuarial Sciences, University of Western Ontario, London, Ontario, Canada N6G 2E9. Internet: `dmurdoch@pair.com`.

** Department of Statistics, University of Toronto, Toronto, Ontario, Canada M5S 3G3. Internet: `jeff@utstat.toronto.edu`.

1. INTRODUCTION

Propp and Wilson (1996,1998) described a protocol called "coupling from the past" (CFTP) for exact sampling from a distribution using a coupled Markov chain Monte Carlo (MCMC) algorithm. Their idea is to consider running copies of the simulation from every possible starting state at a time in the indefinite past; if these are coupled properly, then they will all take on the same value by the present (i.e., by time 0). This value must be a sample from the steady-state distribution. A number of authors have used this idea to construct exact samplers for various point processes (Häggström *et al.*, 1996; Kendall, 1997, 1998) and for distributions arising as posterior distributions in Bayesian statistical models (Møller, 1997; Murdoch and Green, 1997; Green and Murdoch, 1998) and in other applications.

CFTP algorithms improve on traditional MCMC techniques (Gelfand and Smith, 1990; Gelfand *et al.*, 1990; Smith and Roberts, 1993; Tierney, 1994; Gilks, 1996) in that they guarantee an exact sample from the stationary distribution. In particular, they sidestep the need for provable bounds on convergence times of the algorithms (as in Meyn and Tweedie, 1994; Rosenthal, 1995). On the other hand, they require substantial computation to achieve just a single sample from the stationary distribution; to estimate functionals it may be necessary to obtain many such samples, each one requiring substantial additional computation.

We are interested in ways to obtain more efficient estimates using exact samples. In particular, we explore the extent to which previous and subsequent sample values from the chain (not just the single exact sample) may be used when estimating expected values under the stationary distribution.

To this end, we note that Mykland *et al.* (1995) considered MCMC processes in which *regeneration events* occur: the process returns at certain random times to a fixed distribution. The *tours* between regeneration events are then IID, and sample averages over the tours may be used to estimate corresponding means of the steady-state distribution. Such averages are very efficient (i.e., they lead to low-variance estimators) since they make use of a large number of sample values. Because they are IID, they may be used to form reliable variance estimates.

Section 2 of this paper describes CFTP in more detail. Section 3 suggests several ways in which we can modify CFTP to create tours. These tours will not necessarily be IID as in Mykland *et al.* (1995), but they will generally be identically distributed, and

if not independent, will have a structure like an MA(1) time series: non-adjacent tours will be independent. In Section 4 we show how to use these tours to construct estimators of means of functionals with respect to the target distribution. Section 5 considers the validity of these estimators. Section 6 attempts to quantify the computational cost of the different algorithms, and Section 7 illustrates the methods by computer simulation of a simple example. Finally, Section 8 discusses the relevance of these results.

2. The CFTP ALGORITHM

Suppose we wish to sample from a probability distribution $\pi(\cdot)$ on a state space $\mathcal{X}$, and have constructed an ergodic Markov chain $P(x, \cdot)$ having π as its stationary distribution. The idea of CFTP is to keep track simultaneously of Markov chain paths starting from all possible different starting points, from arbitrarily far backwards in time, and to get them all to coalesce to a single value at time $t = 0$.

Specifically, define random variables $N_{x,t}$ and G_t for each $t \in \mathbf{Z}$ and each $x \in \mathcal{X}$, so that $\mathbf{P}(N_{x,t} \in A) = P(x, A)$, and $\mathbf{P}(G_t \in A) = \pi(A)$, and $G_{t+1} = N_{G_t,t}$. Intuitively, $N_{x,t}$ is the "next" value a particular implementation of the Markov chain will take if it is at the point x at time t, and $\{G_t\}$ is a run of the Markov chain in stationarity. We require that N_{x,t_1} and N_{y,t_2} are independent if $t_1 \neq t_2$; however, the joint distribution of $N_{x_1,t}$ and $N_{x_2,t}$ is left unspecified and may be chosen as convenient.

For notation: for $s \geq t$, let $N^s_{x,t}$ be the value that the chain obtains at time s, if it follows the path specified by $\{N_{x,t}\}$ from time t onwards; thus, $N^t_{x,t} = x$, and $N^{s+1}_{x,t} = N_{N^s_{x,t},s}$ for $s \geq t$.

The idea of CFTP is as follows. Suppose we can find a time $-T$ such that $N^0_{x,-T}$ does not depend on x (that is, all paths from time $-T$ have coalesced to one particular value by time 0). Then this common value of $N^0_{x,-T}$ must be equal to G_0, and hence must be a draw from the stationary distribution.

Three points are worth noting. First, if we have coalescence from time $-T$, then we will also have coalescence from any earlier time. This allows the search for T to be quite rough; typically one starts with a guess T_g, and then doubles it until coalescence is found. The resulting multiple of T_g is used as T with no further refinement. The second point to note is that coalescence will typically occur at some time $-T + T_c < 0$, but it is still necessary to carry the value forward to time 0 to avoid a selection time bias. Finally, once G_0 is determined, we may choose to continue the Markov chain forwards in time, say for

another T_0 steps. These four stages of the CFTP algorithm are illustrated schematically in Figure 1.

Figure 1. A schematic representation of the four stages of the CFTP algorithm.

This paper will consider the question of how to use the structure of CFTP to estimate expected values of functionals with respect to π. At one extreme, we could repeatedly run CFTP, generating IID realisations of G_0, and form simple averages. However, this seems wasteful: most of the information in the $N_{x,t}$ values is lost. At the other extreme, we could run CFTP just once to generate the starting value for a long stationary realisation of the original chain, but this will almost always result in serially dependent values and complicate the evaluation of the precision of estimators.

3. TOURS USING CFTP

We propose several methods of obtaining tours related to the CFTP algorithm, starting with the default method that makes no use of the $N_{x,t}$ values:

Repeated CFTP (RCFTP): Run ordinary CFTP to obtain a sample $X \sim \pi(\cdot)$. For some fixed value T_0, set $X_0 = X$, and run any convenient implementation of the Markov chain forwards in time for an additional T_0 steps. The $T_0 + 1$ values observed represent one tour. Repeat independently.

In Figure 1, this would correspond to repeating steps 1 to 4 independently, and using the values generated from the end of step 3 and throughout step 4 as the tour.

Forward Coupling (FC): Run the chain *forwards* in time from time 0 until all paths have coalesced, to obtain a sample X at that time. Run it forwards again until all states

again coalesce, using the path from X as one tour. Repeat (using the coalescing value of one run as the starting value of the next run, but not counting it twice).

Because FC does not use CFTP, Figure 1 does not apply perfectly, but FC is much like applying step 2 repeatedly, and recording one particular path (started where the previous one ended) as the tour.

Concatenated CFTP (CCFTP): Run ordinary CFTP to obtain a sample $X \sim \pi(\cdot)$. Run it again to obtain a sample $Y \sim \pi(\cdot)$, a value of T, and a realization of $N_{x,t}$ for $t \in -T, \ldots, -1$. Our first tour of length T consists of the values in the path from X to Y using $N_{x,t}$ to define the transitions, counting Y but not X. Repeat using Y in place of X to generate subsequent tours.

In Figure 1, this repeats steps 1 to 3, recording the path from the end of one cycle through steps 2 and 3 of the next cycle.

We next consider the question of how to use these different tours to efficiently estimate linear functionals.

4. CONSTRUCTION of ESTIMATORS

If applied N times (with N fixed throughout), each of these methods produces tours having values which may be relabelled as $X_{i1}, X_{i2}, \ldots, X_{iT_i}$, $i = 1, \ldots, N$, where T_i is fixed or random. Thus, one might hope that for a functional $g : \mathcal{X} \to \mathbf{R}$, we could estimate the expected value of g with respect to $\pi(\cdot)$, i.e. $\pi(g) = \int g(x)\pi(dx)$, by averaging the tour averages:

$$\widetilde{\pi}(g) \;=\; (1/N) \sum_{i=1}^{N} \frac{\sum_{j=1}^{T_i} g(X_{ij})}{T_i}.$$

This estimator is clearly unbiased and consistent for RCFTP. However, the randomness of T_i in the other methods breaks this consistency, and we need to consider the ratio estimator used e.g. by Mykland *et al.* (1995):

$$\widehat{\pi}(g) \;=\; \frac{\sum_{i=1}^{N} \sum_{j=1}^{T_i} g(X_{ij})}{\sum_{i=1}^{N} T_i}.$$

Of course, $\widetilde{\pi}(g) = \widehat{\pi}(g)$ for RCFTP.

Two useful facts are:

1. If the sample values X_{ij} are marginally from $\pi(\cdot)$, and are independent of the tour lengths T_i, then $\mathbf{E}\left[\widetilde{\pi}(g)\right] = \pi(g)$, and in fact $\widetilde{\pi}(g)$ is a consistent (as $N \to \infty$) unbiased estimator of $\pi(g)$.
2. If the sample values X_{ij} may be strung together to form one long run which follows the transition law of the chain, then $\widehat{\pi}(g)$ is consistent (as $N \to \infty$). This is standard regeneration theory; see e.g. Asmussen (1987, Section V-1).

Generally speaking, $\widetilde{\pi}(g)$ is preferable to $\widehat{\pi}(g)$ in situations where it is unbiased. As Mykland *et al.* (1995) discuss, the ratio estimator $\widehat{\pi}(g)$ can be unstable unless N is large enough that the mean of T_i can be estimated with small relative error; they recommend 1% relative error as an upper limit to ensure reliability.

5. VALIDITY of the DIFFERENT METHODS

We now consider the question of which of the estimators $\widehat{\pi}(g)$ and $\widetilde{\pi}(g)$ are valid (i.e., unbiased, at least asymptotically) for each of our proposed methods. We also consider some other potential advantages and disadvantages of each.

RCFTP is simple, and is easily seen to be valid using $\widetilde{\pi}(g)$ (which is the same as $\widehat{\pi}(g)$ in this case). Moreover, it generates IID tours, so their analysis is very straightforward. However, it may be inefficient in a computing sense: it runs the CFTP algorithm N times, but only uses $T_0 + 1$ of the $T + T_0$ samples from each run. Unless T_0 is much larger than the expected time $E(T)$ to run CFTP, we may be throwing away a large fraction of our computing effort. We consider the "cost" of the algorithms in the next section.

FC is straightforward and is related to the Mykland *et al.* (1995) approach. Like that approach, it does not guarantee that we're sampling from the stationary distribution. Indeed, even if we *start* our chain in stationarity, the tour lengths will *not* be independent of the tour values, so that the estimator $\widetilde{\pi}(g)$ will *not* be a valid estimator in general. On the other hand, since the tour values are part of a single long run of the chain, we see that $\widehat{\pi}(g)$ will be a valid, consistent estimator of $\pi(g)$.

We note that one difference between FC and the method of Mykland *et al.* (1995) is that the regeneration that occurs when a Mykland *et al.* (1995) tour returns to its starting point takes FC the whole next tour to be realized. As a result of this the tours are not IID; each one is dependent on the value at the end of the previous tour. (The tours still

have the structure of an MA(1) time series: non-adjacent tours will be independent. This is still considered a valid regeneration structure; see e.g. Asmussen, 1987, p. 125.) This may increase the variance of $\widehat{\pi}(g)$ somewhat, though not by more than a factor of 2. It will also complicate the estimation of the variance.

CCFTP has the advantage that each sample in the tour is marginally drawn from $\pi(\cdot)$. Unfortunately, the values in each tour will typically not be independent of T_i, so $\widetilde{\pi}(g)$ will still be biased. (This is a manifestation of the "user impatience bias" of the CFTP algorithm that was discussed by Fill, 1998, Sections 5.2 and 6.1.) For example, if the state space is the non-negative integers, with $P(x, \{x+1\}) = P(x, \{0\}) = \frac{1}{2}$, and with the simple coupling that all states move up or move to 0 simultaneously, then $\pi\{x\} = 1/2^{x+1}$. If g is the identity function, $\pi(g) = 1$; on the other hand, we may compute that $E[\widetilde{\pi}(g)] = \frac{1}{2}$. Thus, CCFTP requires the ratio estimator $\widehat{\pi}(g)$.

The question is, in view of this bias, can we make use of the extra structure that CCFTP gives us? One possibility is to modify this algorithm as follows, to make use of a guaranteed minimum value for T:

Guarantee Time CFTP (GTCFTP): Fix a value $T_g > 0$ to use as an initial guess for T in the CFTP algorithm. Run ordinary CFTP to obtain a sample $X \sim \pi(\cdot)$. Run it again to obtain a sample $Y \sim \pi(\cdot)$, a value of $T \geq T_g$, and a realization of $N_{x,t}$ for $t \in -T, \ldots, -1$. Our first tour of length T consists of the final T_g values in the path from X to Y using $N_{x,t}$ to define the transitions. Repeat using Y in place of X to generate subsequent tours.

Like CCFTP, this algorithm repeats steps 1 to 3 of Figure 1, but only T_g of the values in steps 2 and 3 will be included in the tour.

Because this method fixes T_g, the tour length is fixed and so trivially independent of the sample values. Since the sample values are marginally distributed as $\pi(\cdot)$, we conclude (as in note 1 above) that $\widetilde{\pi}(g)$ is an unbiased consistent estimator in this case.

Any of these methods may be modified to include an additional T_0 points generated using a simple implementation of the chain starting from Y, as was done in RCFTP. (This adds step 4 of Figure 1 after the other steps.) In our cost analysis below and in the simulations we used this modification.

We have also considered other methods for generating tours, which appeared promising but turned out not to be valid. For example, CCFTP may be modified to use the value

Y as the starting value X, yielding independent cyclic tours. Unfortunately, because of the dependence between Y and T, the marginal $\pi(\cdot)$ distribution may be lost. Similarly, Fill (1998) describes an algorithm (see also Fill *et al.*, 1999) based on rejection sampling that removes the dependence between Y and T, but it also loses the marginal distributions within the tour. We haven't found a way to make use of the $N_{x,t}$ values with either of these algorithms.

6. COMPUTATIONAL COSTS

The CFTP algorithm requires several tasks. One of these is the evaluation of the update function $N_{x,t}$ given x and t. Sometimes this is more difficult than a simple Markov chain update, because a coalescing update rule is required. On a discrete state space this might not impose any cost, since many different update rules will coalesce. However, we will usually choose one with particularly convenient properties for CFTP such as monotonicity (Propp and Wilson, 1996), and this may cost something. On a continuous state space, most simple update rules will not coalesce, and more complicated and expensive schemes are required (Murdoch and Green, 1997; Green and Murdoch, 1998). In either case there is also the overhead of storing or regenerating random values for subsequent evaluations of $N_{x,t}$ at the same time t.

Denote the mean cost (i.e., mean computation time) of one evaluation of $N_{x,t}$ as U.

If $T_0 > 0$, we also need to compute single updates of the Markov chain forwards in time. Denote the mean cost of these by M; as mentioned above, $M \leq U$.

The first stage of CFTP searches for a time $-T$ from which all paths coalesce. The cost of this stage depends on the search strategy. We assume an initial guess T_g will be tried, and if it fails, it will be doubled until it succeeds. Efficiency is helped by a good guess: if it is too small, more doublings will be required, and if it is too large, the paths will coalesce early but still need to be carried forward to time 0. We'll approximate the total cost as $S(T_g) + C + (T - T_c)U$, where $S(T_g) > 0$ is a function of T_g representing the time spent computing the *failures* before the successful T is found, $C > 0$ is the cost to determine the number of steps T_c for all states to coalesce to a single state and to simulate one path through this stage, and $(T - T_c)U$ approximates the cost to carry a single path forward from the coalescence time to time 0. Typically $S(T_g)$ decreases as T_g increases, because failures are less likely with larger T_g. However, the overall cost of a large T_g may be higher, because of the constraint $T \geq T_g$.

We can now calculate the running times of each of our algorithms. These calculations are summarized below.

Method	Computational Cost	# Samples	Estimator
RCFTP	$S(T_g) + C + (T - T_c)U + T_0 M$	$T_0 + 1$	$\widetilde{\pi}(g)$
FC	$C + T_0 M$	$T + T_0$	$\widehat{\pi}(g)$
CCFTP	$S(T_g) + C + (T - T_c)U + T_0 M$	$T + T_0$	$\widehat{\pi}(g)$
GTCFTP	$S(T_g) + C + (T - T_c)U + T_0 M$	$T_g + T_0$	$\widetilde{\pi}(g)$

Figure 2. A summary of the methods.

RCFTP costs $S(T_g) + C + (T - T_c)U + T_0 M$ to achieve one tour of fixed length $T_0 + 1$. Separate tours are IID, and the estimator $\widetilde{\pi}(g)$ may be used.

FC costs $C + T_0 M$ to achieve one tour of random length $T + T_0$. Since the tour values and T are not independent, $\widetilde{\pi}(g)$ will not produce consistent estimates and $\widehat{\pi}(g)$ must be used. Note that C has a different distribution under forward and backward coupling. The coalescence time T_c depends on the realization of $N_{x,t}$ and the time t from which we start all paths. In forward coupling, the starting time is fixed at $t = 0$. In backward coupling, CFTP searches backwards until it finds a time $t = -T$ for which $T > T_c$; this search is biased towards smaller values of T_c (and thus smaller cost C, as well). (To see this, suppose that in CFTP we increase our guess by one each time it fails, instead of doubling it. Then we will always find the minimal T, and we know that paths from time $-(T-1)$ will *not* coalesce. A simple calculation then shows that $P(T > x)$ in CFTP is the same as $P(T_c > x)$ in FC; since $T_c \leq T$ in CFTP, we see the bias mentioned above.) However, when doubling is used in the search, very little selection bias occurs and the distributions of T_c and C are quite similar under both coupling methods.

CCFTP has an initial one-time-only cost of $S(T_g) + C + (T - T_c)U$ to generate the first perfect sample X. From then on, it costs $S(T_g) + C + (T - T_c)U + T_0 M$ each time to achieve a tour of random length $T + T_0 \geq T_g + T_0$. As with FC, $\widehat{\pi}(g)$ must be used.

GTCFTP costs $S(T_g) + C + (T - T_c)U + T_0 M$ to achieve a tour of fixed length $T_g + T_0$. Separate tours are MA(1), but the estimate $\widetilde{\pi}(g)$ may be used. This method may be preferable to RCFTP if the correlation of successive tour means is sufficiently low,

because the tours are longer for the same approximate cost. It is more expensive than FC; this extra expense buys the $\widetilde{\pi}(g)$ estimator, but is it worth the cost? We will approach this question by examining an example.

7. EXAMPLE

The Markov chain in our example is a random walk on the integers from 0 to 20, with equal probability of stepping up or down by one unit, and steps outside the range being rejected. Here π is the uniform distribution with mass 1/21 on each state. It is particularly easy to couple this chain, because the simple coupling obtained by attempting the same jump from every state (i.e. $N_{x,t} = \max[0, \min(20, x \pm 1)]$) is a monotone coupling: paths from different starting points can never cross, but are certain to eventually coalesce. Only the two paths starting from states 0 and 20 need to be simulated to determine coalescence, and the coalescing update rule is as efficient to simulate as any other.

Figure 3. CFTP applied to the monotone coupling of the random walk example. The four attempts at coalescence are shown in thin lines; the solid black path is the value of G_t. The underlined range of X_t values would be averaged by the RCFTP algorithm.

Figure 3 illustrates CFTP on this example. Here T_g was taken to be 25, but the paths started at $t = -25$ did not coalesce, nor did those started at the doublings to $t = -50$ or -100. The paths started from $t = -200$ did all coalesce, at $t = -81$; the coalesced values were carried forward (using the same jumps as for the previous attempts) to time 0, and then further forward to time $T_0 = 100$. In RCFTP, g evaluated at the last 101 values would be averaged into the estimate $\widetilde{\pi}(g)$.

Figure 4. Illustration of the CFTP-based algorithms through four cycles. They differ only in the sample values used in the estimates, as indicated by the underlines: solid for RCFTP, dotted for CCFTP, and dashed for GTCFTP. In this figure, the times t have been shifted to positive values in all cycles after the first.

Figure 4 compares RCFTP with CCFTP and GTCFTP, with $T_g = T_0 = 100$. All three algorithms are based on the same application of CFTP; the differences lie in which values of G_t are employed in estimating $\pi(g)$. RCFTP used the IID cycles of length 101 starting at times 0, 300, 600 and 1100. CCFTP used all values from time 100 onwards, divided into 2 cycles of length 300 and one of length 500. These cycles are not independent, nor are the cycles of fixed length 200 used by the GTCFTP algorithm. Figure 5 illustrates the FC algorithm, again with $T_0 = 100$.

The same simulation allowed us to form estimates of some of the computational timings. The times depend to a large degree on the computer (a 233 Mhz Pentium PC), the compiler (Delphi Pascal version 3) and the programmer (the first author), but should give a rough idea of the relative magnitudes. The counts of steps depend only on the algorithms. We observed that the distribution of steps to coalescence was skewed to the right, with a median of $T_c = 120$ steps (with the 1st and 3rd quartiles at 86 and 182 steps respectively, hereafter written as $T_c = 120[86, 182]$). Simulating to coalescence took a median time of $C = 39[29, 56]$ μs. Simulating G_t from coalescence to the sample at time 0 took a median time of $39[23, 67]$ steps. The time taken to do a single step update was $U = M = 0.30[0.23, 0.54]$ μs. In this simulation T_g was set to 1, making the search times $S(T_g)$ quite large: $703[441, 1207]$ μs. (In a separate simulation with $T_g = 100$, this was reduced to $406[402, 804]$ μs.)

Figure 5. Illustration of FC through four cycles. The line below the plot indicates the sample values used in the estimator; the ticks on it mark the beginning of each cycle.

In order to compare the cost efficiency of the different algorithms, we simulated calculation of $\widehat{\pi}(g)$ or $\widetilde{\pi}(g)$ for $g(x) = I(x = 0)$, the indicator of residency in state 0. Our simulation consisted of 1000 runs of the chain under 128 different conditions: all com-

binations of forward vs. backward coupling (all 3 CFTP estimators can be calculated from the same backward coupled run), values of $N = 2, 25, 100, 400$ (note that all algorithms except RCFTP make no use of the first cycle's values in the estimator), values of $T_g = 1, 25, 100, 400$, and values of $T_0 = 0, 25, 100, 400$.

For each of the simulations we calculated the computational efficiency, defined as the ratio of the precision (inverse sample variance) to the total computational cost averaged over the 1000 simulations. As can be seen in Figure 6, CCFTP and FC gave the most computationally efficient estimators; both make use of all simulated values, so we would expect this. Somewhat surprisingly, large values of N were always more efficient than small values. This is partly explained by the fact that FC, CCFTP and GTCFTP all discard the first cycle; it is likely also partly caused by the instability of the ratio estimator for small N. (Only $N = 400$ attained 1% relative error in estimating the mean cycle length.) The value T_g predictably played a strong part in determining the efficiency of GTCFTP and had a small influence on RCFTP (and none at all on FC, where it isn't used). Likewise T_0 affected the efficiency of RCFTP most strongly.

Figure 6. Mean $\log_2$ efficiency in estimating $\pi(X = 0)$. The numbers shown are the values of the variables N, T_g and T_0, plotted at the position corresponding to the mean of the base 2 log efficiency. The horizontal lines show the overall mean for each algorithm.

This figure shows a fairly small range. Most of the simulations lie within a factor of 8 of each other in efficiency. The only method that clearly stands out as inefficient is

RCFTP with $T_0 = 0$, i.e. the naive generation of an IID sample by CFTP one observation at a time.

8. DISCUSSION and CONCLUSIONS

Exact sampling methods represent an important new development in the use of Markov chains to estimate the means of functionals with respect to complicated probability distributions. However, such methods are computationally intensive, and it is important to use the resulting values efficiently.

In this paper, we have proposed a number of modified algorithms and estimators designed to make use of values available from the exact sampling methods. We have proved the unbiasedness of certain estimators associated with them, and we have considered the computational costs involved (both theoretically and through simulation).

These algorithms are all roughly comparable in efficiency, provided some effort is made to obtain more than a single observation from each coupling run. If IID unbiased tours are needed, use RCFTP with T_g just large enough to avoid long search times, and large T_0 for efficiency. N should be large enough to allow a stable variance estimate from an IID sample (say 30 or more). If unbiased MA(1) tours are sufficient, GTCFTP will generally be more efficient. Here larger values of either T_g or T_0 will result in greater efficiency. However, when simple MCMC steps are easier to simulate than coalescing ones (i.e. $U > M$), large T_0 should be the choice. If exact unbiasedness is not needed, FC is probably the simplest algorithm to implement and we would recommend its use when N is large enough to make the estimator stable. For large samples, CCFTP will produce essentially the same estimates as FC for given length of chain (since all simulated values in the chain are used except for a few at the beginning and the end), but it is more difficult to implement, so we do not recommend its use. Finally, in cases where regeneration events are available, the Mykland *et al.* (1995) algorithm should do at least as well as FC.

Overall we conclude that these methods are useful, efficient ways to make good use of exact sampling. They are all better than naive use of a single perfect value per CFTP run. They may even alleviate the need for clever coupling algorithms that coalesce rapidly: if a slower than necessary coupler is chosen, all algorithms but RCFTP will still gain precision from the values observed during the search for T.

ACKNOWLEDGEMENTS

We are grateful to the referees for their helpful comments. This research was supported in part by NSERC Research Grants to both authors.

REFERENCES

Asmussen, S. (1987). *Applied Probability and Queues.* John Wiley & Sons, New York.

Fill, J. A. (1998). An interruptible algorithm for perfect sampling via Markov chains. *Annals of Applied Probability* 8:131-162.

Fill, J. A., Machida, M., Murdoch, D. J. and Rosenthal, J. S. (1998). An extension of Fill's exact sampling algorithm to non-monotone chains. Submitted.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association*, 85:972–985.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Gilks, W. R. (1996). Full conditional distributions. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in Practice*, pages 75–88. Chapman and Hall.

Green, P. J. and Murdoch, D. J. (1998). Exact sampling for Bayesian inference: towards general purpose algorithms. In *Bayesian Statistics 6.* To appear.

Häggström, O., Lieshout, M. N. M., and Møller, J. (1996). Characterisation results and Markov chain Monte Carlo algorithms including exact simulation for some spatial point processes. Technical Report R-96-2040, Department of Mathematics, Aalborg University. To appear in *Bernoulli.*

Kendall, W. S. (1997). On some weighted Boolean models. In D. Jeulin, editor, *Advances in Theory and Applications of Random Sets*, pp. 105–120. World Scientific Publishing Company, Singapore.

Kendall, W. (1998). Perfect simulation for the area-interaction point process. In Accardi, L. and Heyde, C. C., editors, *Probability Towards 2000.* Springer, New York.

Meyn, S. P. and Tweedie, R. L. (1994). Computable bounds for convergence rates of Markov chains. *Annals of Applied Probability*, 4:981–1011.

Møller, J. (1997). Perfect simulation of conditionally specified models. Technical Report R-97-2006, Department of Mathematics, Aalborg University. To appear in *J. Roy. Stat. Soc., ser. B*.

Murdoch, D. J. and Green, P. J. (1997). Exact sampling from a continuous state space. *Scandinavian Journal of Statistics*, 25:483–502.

Mykland, P., Tierney, L., and Yu, B. (1995). Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, 90:233–241.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252.

Propp, J. G. and Wilson, D. B. (1998). How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27:170–217.

Rosenthal, J. S. (1995). Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 90:558–566.

Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion). *Journal of the Royal Statistical Association, Series B*, 55:3–24.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22:1701–1762.