特集論文

Population Monte Carlo algorithms

Yukito Iba
The Institute of Statistical Mathematics
iba@ism.ac.jp, http://www.ism.ac.jp/~iba/

keywords: quantum Monte Carlo, transfer-matrix Monte Carlo, Monte Carlo filter, sequential Monte Carlo, pruned-enriched Rosenbluth method, annealed importance sampling, genetic algorithm

Summary

We give a *cross-disciplinary* survey on "population" Monte Carlo algorithms. In these algorithms, a set of "walkers" or "particles" is used as a representation of a high-dimensional vector. The computation is carried out by a random walk and split/deletion of these objects. The algorithms are developed in various fields in physics and statistical sciences and called by lots of different terms – "quantum Monte Carlo", "transfer-matrix Monte Carlo", "Monte Carlo filter (particle filter)","sequential Monte Carlo" and "PERM" *etc.* Here we discuss them in a coherent framework. We also touch on related algorithms – genetic algorithms and annealed importance sampling.

1. Introduction

In this paper, we give a *cross-disciplinary* survey on "population" Monte Carlo algorithms. These algorithms, which are developed in various fields, have a common structure: A set of "walkers" or "particles" is used for the representation of a high-dimensional vector and the computation is carried out by a random walk in the state space and split/deletion of these objects. These algorithms do *not* belong to the class of Markov chain Monte Carlo (MCMC, dynamical Monte Carlo), although they share common features and applications.

Monte Carlo filter [Kitagawa 96, Kitagawa 98] (or sequential Monte Carlo [Liu 98, Doucet 01, Crisan 00]) algorithm is a topic of recent interest in the field of statistical information processing and now being popular in related fields, *e.g.,* robot vision [Isard 98]. It is an example of population Monte Carlo algorithms defined here. It is, however, *not* the only example. Population Monte Carlo algorithms have proved to be useful tools in a number of fields — quantum physics, polymer science, statistical physics, and statistical sciences. They are powerful rivals of MCMC in these fields.

The aim of this paper is *neither* the design of a novel algorithm *nor* the presentation of new applications. Our goal is to give a minimal set of references and explanations for the cross-fertilization among the researchers in different fields. It will be useful because few people realize essentially the same algorithms are used in both in statistical information processing [Kitagawa 96, Kitagawa 98, Liu 98, Doucet 01, Crisan 00] and physics [Hetherington 84, Cerf 95, Kalos 62, Ceperley 79, Nightingale 88, Grassberger 00]. I also hope that this survey will be useful for the development of applications in machine learning and probabilistic artificial intelligence.

In Chapter 2, we give a common structure of the population Monte Carlo algorithms. We also discuss a relation to genetic algorithms and give remarks on the origins of the methods. In Chapter 3, we give examples of the algorithms both in physics and statistics.

2. An Overview

2·1 *Algorithm – General*

Essentially, the algorithms discussed in this paper are designed for the computation of the products of non-negative, sparse, $M \times M$ matrices $G^1, G^2, \ldots$ and a non-negative vector X^0. If we express the vector of nth step as X^n then

$$X^{n+1}(i) = \sum_j G^n(i,j)\, X^n(j) \qquad (1)$$

where $1 \le i, j \le M$ are indices of components of the vector X^n and the matrix G^n. Here, the dimension M of vector X^n is assumed to be very large and we cannot explicitly store the elements of the vector in

the memory. Saving the storage, we represent the non-negative vector as a weighted superposition of the "walker" or "particles" indexed by k $(k=1,\ldots,K)$, each of which is placed at $j^n(k)$ with a weight w_k^n:

$$\widetilde{X}^n(j)=\sum_k w_k^n\cdot\delta_{j^n(k),j} \tag{2}$$

The algorithm:

(1) STEP 1: Random walk in the state space.
Each walker k at the position $j^n(k)$ is moved independently to a new position i

$$j^{n+1}(k):=i \quad \sim \quad P^n(i,j^n(k))=\frac{G^n(i,j^n(k))}{\sum_{i'}G^n(i',j^n(k))}, \tag{5}$$

(2) STEP 2: Update of the weights.
For each walker, calculate the factor

$$W_k=\sum_{i'}G^n(i',j(k)) \tag{6}$$

and update the weight of the walker using

$$\log w_k^{n+1}:=\log W_k+\log w_k^n. \tag{7}$$

(3) STEP 3: Reconfiguration.
- Split walkers with a large weight. Each walker splits into multiple walkers, whose total weight is equal to that of the original walker.
- Remove walkers with a small weight.

— **reweighting**, **prune / enrichment**, **reconfiguration**, **rejuvenation**, **resampling**, **branching**, **selection**.

There are several different ways to resample the population of the walkers. A simple way is to set

$$Q_k=\frac{w_k}{\sum_{k'}w_{k'}} \tag{8}$$

and resample the walkers with probability Q_k (Note that walkers with large Q_k can be sampled several times.). Variants are found in the references, for example, [Nightingale 88]. In some variants, the number of walkers K is not strictly constant, but fluctuates within a range.

To complete the algorithm, the initial positions $\{j^0(k)\}$ and weights $\{w_k^0\}$ of the walkers should be given. In most cases, w_k^0 is a constant independent of k and $\{j^0(k)\}$ are independent samples from the initial density defined by X^0. The choice of X^0 depends on a specific problem and usually the same as the choice of X_0 in the corresponding deterministic algorithm based on the direct iteration of (1). For example, we can use an arbitrary density as an initial vector X_0 in quantum Monte Carlo (Section 3･1), although systematic choice will improve the speed of the convergence. In Monte Carlo filtering (Section 3･3), X_0 represents a prior distribution for initial states of series.

Let us define $[\cdots]$ as an average over random numbers used in the past steps of the algorithm. Then, if we assume

$$X^n(j)=[\widetilde{X}^n(j)], \tag{9}$$

it is not difficult to show that the relation

$$X^{n+1}(j)=[\widetilde{X}^{n+1}(j)] \tag{10}$$

holds with STEP 1 $\sim$ STEP3. This property is essential for the adequacy of population Monte Carlo algorithm. When we modify the STEP 3, we should be careful to conserve this property. With the condition (10), we expect the convergence of the algorithm in the limit of the number of the walkers $K\to\infty$ with a fixed number n of the iteration [*1]. We refer to [Crisan 00] for a recent result on the convergence of population Monte Carlo algorithms [*2]. Practically, the rate of the convergence severely depends on the problem treated by the method.

Finally, we note that it is easy to generalize the algorithm to the cases with continuous state space, where the matrix G is replaced by an operator and the summations become integrals. Such continuous versions of the algorithm are successfully used in some of the examples discussed below, *e.g.,* applications to quantum many-body problems.

2･2 The Role of STEP 3

The STEP 3 of the algorithm is not necessary for the validity of the algorithm, i.e., the condition that we discussed in the previous section is satisfied only with STEP 1 and STEP 2. The role of STEP 3 is to suppress the variation of weights $\{w_k\}$ and improve the efficiency of the algorithm.

*1 Note that the convergence is not assured in the limit $n\to\infty$ with a fixed number K of walkers. It is an important remark for the problems where the limit $n\to\infty$ is required (for example, see Section 3･1).

*2 Some remarks by physicists on the property of the algorithm are found in [Hetherington 84, Cerf 95].

For some problems, the algorithm without STEP 3 is sufficient. Hereafter, we call the algorithm without STEP 3 as "simple" population Monte Carlo. Note that without STEP 3, there is no interaction between walkers. Then, parallel evolution of the walkers can be replaced by sequential runs each of which corresponds to the simulation of a walker.

In some references, a part of STEP 3 is separately discussed as a "population control" procedure (*e.g.,* [Kalos 62, Grassberger 98]) and/or a part of STEP 3 is merged into STEP 1 (*e.g.*, [Kalos 62]). In such cases, we should be careful to judge whether the algorithm effectively contains STEP 3 or not.

2·3 Relation to Genetic Algorithm

There is an obvious analogy between population Monte Carlo and genetic algorithms (GA) [Higuchi 97]. However, there is an essential difference in the goal of the algorithm. Population Monte Carlo is a tool for the computation of the product of matrices, multiple summations (integrals) and calculation of marginals. On the other hand, GA is a tool specialized in optimization. Usually, we can modify population Monte Carlo for the optimization. However, the converse is often not true. Specifically, we should be careful to the introduction of *crossover operators* and other tricks popular in GA to population Monte Carlo, because they can easily spoil the convergence to the exact result in the limit of infinite number of walkers, $K \to \infty$.

There are some theoretical studies on GA that assume the absence of the crossover operation. It might be interesting to apply them to the study of population Monte Carlo.

2·4 Origins

It is difficult to fix the origin of the population Monte Carlo. An origin seems the algorithms for the calculation of the elements of an inverse Matrix developed by von Neumann and others [Hammersley 64, Dreitlein 85], which lacks the STEP 3. Another reference is Metropolis and Ulam [Metropolis 49] *3, which discuss a solver of the Schrödinger equation by simulation. The algorithm is attributed to Fermi (It is an origin of quantum Monte Carlo algorithms with random walkers discussed in Section 3·1). The algorithm for eigenvalue problems are also discussed in the classic book [Hammersley 64] on Monte Carlo. Simple algorithm for self-avoiding walks is introduced by Rosenbluth and Rosenbluth [Rosenbluth 55]. According to [Crisan 00], one of the earliest studies of simple algorithm in statistical science is found in [Handschin 69].

It is also not easy to determine who introduced STEP 3. For example, STEP 3 is included in the algorithm in Metropolis and Ulam [Metropolis 49] and Kalos [Kalos 62] in a somewhat implicit manner. The "enrichment" algorithm [Wall 59] for self-avoiding walks contains a very special case of STEP 3. Anyway, many authors have used the algorithms with STEP 3 by the end of 1980s as a version of "quantum Monte Carlo" or "transfer-matrix Monte Carlo", while the algorithms without STEP 3 have also been used up to now.

We left a comprehensive treatment on the history of the population Monte Carlo methods to future surveys.

3. Examples

3·1 Quantum Many-Body Problems

First, we discuss "quantum Monte Carlo" algorithms *4 [Ceperley 79, Schmidt 84, Schmidt 92, De Raedt 92, Hetherington 84, Cerf 95, Negele 86, Anderson 75, Blankenbecler 83, John 87]. In spite of their name, they are not algorithms specialized to quantum mechanics. They are most naturally understood as methods for the approximation of the smallest (or largest) eigenvalue and the corresponding eigenvector of a large sparse matrix. Essentially, they are stochastic versions of the power method for eigenvalue problems.

Consider a symmetric, non-negative matrix A. To formulate the eigenvalue problem into population Monte Carlo, we introduce the exponential $\exp(-\beta A)$ of A. For sufficiently large β, the expression

$$\exp(-\beta A)\, X^0 \tag{11}$$

approximately gives an eigenvector with the smallest eigenvalue for an arbitrary vector X^0 that has non-zero projection to the eigenvector. In general, the calculation of $\exp(-\beta A)$ is difficult. In most of interesting problems, however, A is decomposed into

*3 It is interesting to point out that this paper by famous physicists appeared in the Journal of American Statistical Association, which is one of the well-known journals of statistics.

*4 There is a different type of "quantum Monte Carlo" algorithms which are based on MCMC (dynamical Monte Carlo). They simulate a "world-line" instead of a collection of random walkers.

several simple components, *e.g.,* $A = B + C$ where B and C is symmetric, non-negative matrices whose exponential is easily calculated. Thus, the exponential of A is expressed as

$$\exp(-\beta A) = \lim_{N\to\infty} \left(\exp(-\frac{\beta}{N}B)\exp(-\frac{\beta}{N}C) \right)^N \quad (12)$$

It is straightforward to fit the problem into the form of the equation (1) with

$$G^n = \exp(-\frac{\beta}{N}B)\exp(-\frac{\beta}{N}C) \quad (13)$$

where N is a sufficiently large number and $n = 0, \ldots, N-1$.

Another approach to the eigenvalue problem is a method based on an iterative solver of the eigenvalue equation. Consider the product of matrices

$$G^n = B^{-1}(\lambda I - D) \quad (14)$$

where $A = B + D$ and D are a diagonal matrix (I denotes an identity matrix.). Under the suitable condition on the "Green's function" B^{-1} and a properly chosen value of λ [*5], the vector X^n converges to an eigenvector of A.

In the treatment on quantum mechanical problems, the matrix A is usually *Hamiltonian* of the given system. It is easy to construct continuous version of the algorithm, where the matrices and summation is replaced by operators and integrals, respectively. In such cases, the operator B is often represented by the summation of Laplace operators and a constant.

These algorithms are successfully applied to various problems in quantum physics, *e.g.,* the computation of the properties of nuclei [Kalos 62, Negele 86], superfluids [Ceperley 79, Schmidt 92], quantum spin systems [Blankenbecler 83, De Raedt 92], and quantum dynamics [John 87]. They are called by the terms "projector Monte Carlo", "Green's function Monte Carlo", and "diffusion Monte Carlo" [*6].

In some cases, we should deal with matrices or eigenvectors that are not non-negative. Formally, these cases can be dealt with the introduction of negative weight w_k^n of walkers. It, however, seriously spoils the efficiency and convergence of the algorithm. It is an example of "negative sign crisis" in stochastic computation of quantum mechanical problems. Although there are some interesting ideas to improve the algorithm [Kalos 84, Carlson 85, Negele 89, Anderson 91, Liu 94], it seems difficult to remove the difficulty entirely with a smart trick, because it originates from radical difference between classical stochastic systems and quantum systems.

3·2 Lattice Spin Systems

Using the transfer matrix (transfer integral) formalism, we can translate a calculation for classical statistical mechanics (and combinatorics) to the computation of a product of matrices. Then, it is a natural idea to apply population Monte Carlo algorithms to these problems. For example, consider a classical spin model (Markov field model) defined on a $L_1 \times L_2 (L_2 >> L_1)$ strip [*7]. We define a set of subsystems each of which consists of the $1, \ldots, n$ rows of the original strip ($n = 1, 2, \ldots, L_2$). Hereafter we assume nearest neighbor interaction on the lattice, although extensions to cases with next-nearest interaction *etc.* are formally easy. The partition function $Z_n(S_n)$ of the subsystem n conditioned with the values of the variables in the nth row $S_n = \{S_{ln}\}_{l=1\ldots L_1}$ is

$$\sum_{S_1}\sum_{S_2}\cdots\sum_{S_{n-1}} \exp(-\beta E_n(S_1, S_2, \ldots, S_n)) \quad (15)$$

where β is inverse temperature and $E_n(S_1, S_2, \ldots, S_n)$ is the energy of the nth subsystem. When each variable S_{ln} is a binary variable, the number of the possible values of S_n is 2^{L_1}. Then, a recursion relation

$$Z_{n+1}(S_{n+1}) = \sum_{S_n} Z_n(S_n)\exp(-\beta \triangle E_n(S_n, S_{n+1})) \quad (16)$$

where

$$\triangle E_n(S_n, S_{n+1}) = E_{n+1}(S_1, \ldots, S_n, S_{n+1}) - E_n(S_1, \ldots, S_n) \quad (17)$$

holds. Obviously, it is essentially the same as the filtering formula for general state space models and the recursive formula of the Baum-Welch algorithm for hidden Markov models. We define $Z_0 = 1$ and $\triangle E_0(S_1) = E_1(S_1)$ for the free boundary condition at the top of the strip (Here and hereafter, the symbol S_0 in the formulae should be neglected.).

A population Monte Carlo algorithm is constructed by setting G and X as

$$G^n(S_{n+1}, S_n) = \exp(-\beta \triangle E_n(S_n, S_{n+1})) \quad (18)$$

*5 When we implement it in a form of population Monte Carlo, the choice of the constant λ is absorbed in STEP 2 and STEP 3. Then we can obtain the eigenvalue λ and the corresponding eigenvector simultaneously.

*6 Here we avoid the details of the terminology, because the classification and naming of these algorithms may depend on the authors.

*7 For a three-dimensional problem, a $L_1 \times L_1 \times L_2$ column substitutes for the strip.

$$X^n(S_n) = Z_n(S_n). \tag{19}$$

The algorithm is called "transfer-matrix Monte Carlo" and applied to classical Heisenberg and XY models [Nightingale 88, Thijssen 90] and enumeration of the number of Penrose tiling [Shaw 91] on a lattice.

3·3 Bayesian Computation: MCF and Sequential Monte Carlo

Algorithms that are similar to the "transfer-matrix Monte Carlo" are developed by statisticians for the models of time-series analysis and models with sequential structures (*e.g.,* models for gene-propagation analysis). In these algorithms, we consider marginal likelihood conditioned with the state x_n of the latest step (time-step) n

$$Z_n(x_n) = \sum_{x_0}\sum_{x_1}\cdots\sum_{x_{n-1}} \hat{P}(x_0)\tilde{P}(x_0|x_1) P(y_1|x_1)\cdots\tilde{P}(x_{n-1}|x_n)P(y_n|x_n) \tag{20}$$

instead of partition function *8. Here $P(y|x)$ is likelihood associated with data y (representation of observational noise). $\tilde{P}(x|x')$ and $\hat{P}(x)$ are prior probability for the state transition $x \to x'$ (representation of system noise) and prior probability of initial state x, respectively. The application of population Monte Carlo algorithms is straightforward.

Kitagawa [Kitagawa 96, Kitagawa 98] introduced "Monte Carlo filter" (MCF, "particle filter") for non-Gaussian/non-linear state space models. Gordon and coworkers independently developed an idea similar to the one by Kitagawa, which they call "bootstrap filter" [Gordon 93, Gordon 95]. Another contribution is "sequential imputation" that first used in gene-propagation analysis [Irwin 94] and later used for blind deconvolution [Liu 95] and other problems. The original form of sequential imputation is a simple algorithm that lacks STEP 3, but STEP 3 is introduced later ("rejuvenated sequential imputation"). The term "sequential Monte Carlo" is also used to indicate a family of algorithms that include the above-mentioned methods as special cases [Liu 98, Crisan 00]. In [Liu 98, Crisan 00], more references in statistical sciences, some of which are earlier than above-mentioned studies, are discussed. Applications in robot vision are described in [Isard 98]. A forthcoming book [Doucet 01] on sequential Monte Carlo by statisticians will be a good guide to the applications in statistical sciences.

*8 We can also use formulation based on posterior distributions of subsystems, which is similar to the one used in Section 3·4. It is useful in some problems, *e.g.,* smoothing of time-series.

3·4 Polymer Science

"Polymer" means a flexible chain consists of small molecules ("monomers") connected in a fixed order. It is called a heteropolymer ($\leftrightarrow$ homopolymer) when it is made by monomers of more than two species. Well-known examples of heteropolymer is protein, RNA and DNA. A good model of polymer is a flexible chain where each unit is mutually repulsed by short range force (van der Waals core) and repulsed/attracted by longer range force. In lattice models, the former interaction is modeled by "self-avoiding condition", i.e., the condition that a lattice point should not be occupied by more than two monomers.

Population Monte Carlo algorithms for the study of polymer models are designed in a manner similar to those for lattice spin models. Subchains of length $n = 1, 2, \ldots, N$ that consists of the first n monomers of the original polymer of length N is used as subsystems for the definition of the algorithm. An important difference is that the index of a vector X^n is *not* the coordinates r_n of the monomer located at the end of a subchain, but coordinates $\{r'_n\}, (n' = 1, \ldots, n)$ of all monomers in the subchains. This is because a pair of monomers with a large distance along the chain can interact each other, even with short range interaction. Thus, the recursion is better described by the Gibbs distribution P_n of subchains, rather than partition functions Z_n.

If we express the Gibbs distribution of a subchain n as $P_n(r_1, r_2, \ldots, r_n)$, the recursion relation

$$P_{n+1}(r_1, r_2, \ldots, r_{n+1}) \propto P_n(r_1, r_2, \ldots, r_n) \cdot \exp(-\beta \triangle E(\{r_1, \ldots, r_n\}, r_{n+1})) \tag{21}$$

holds, where $\triangle E(\{r_1, \ldots, r_n\}, r_{n+1})$ is the interaction energy between a new monomer and the monomers in the subchain n. For lattice models, we set $\triangle E = +\infty$ if the conformation of the subchain specified by $\{r_1, \ldots, r_{n+1}\}$ is non-physical, i.e., self-avoiding condition or connectivity condition of the chain is violated.

To define a population Monte Carlo algorithm, we must specify the index i of $X^n(i)$ in the equation (1). The index i should determine the conformation $\{r_1^i, \ldots, r_n^i\}$ of the nth subchain and should have the dimension N independent of n. Here we define i as a

vector with components $i_s = r_s^i (s \leq n)$ and $i_s = \mathbf{1}(s > n)$, i.e.,

$$i = (r_1^i, r_2^i, \ldots, r_n^i, \mathbf{1}, \mathbf{1}, \ldots, \mathbf{1}) \tag{22}$$

where $\mathbf{1}$ is the coordinates of an arbitrary point, for example, the origin. Next we define $X^n(i)$ as

$$X^n(i) \propto P_n(r_1^i, r_2^i, \ldots, r_n^i) \tag{23}$$

for i of the form (22) and set $X^n(i) = 0$ for other indices i. Finally, we define the matrix G^n as

$$\begin{aligned} G_{ij}^n &= \exp(-\beta \triangle E(\{r_1^i, \ldots, r_n^i\}, r_{n+1}^i)) \\ &\times \textstyle\prod_{s=1}^n \delta(i_s - j_s) \prod_{s=n+2}^N \delta(i_s - \mathbf{1}). \end{aligned} \tag{24}$$

With these expressions, we can write the recursion (21) in the form of the equation (1), which gives population Monte Carlo algorithms based on a simulation of growth/selection of polymer subchains (here we start from X^1 instead of X^0, with which the index n equals to the length of the subchain).

Population Monte Carlo algorithms for polymer models are developed by two or more groups ([Garel 90, Orland 98] and [Grassberger 97, Grassberger 98, Bastolla 98, Grassberger 00]) [*9]. Grassberger pointed out that his algorithm, PERM (pruned-enriched Rosenbluth method) is understood as an unification of "enrichment algorithm" [Wall 59] (STEP 1 and 3) and "Rosenbluth algorithm" [Rosenbluth 55, Seno 96] (STEP 1 and 2). PERM has proved to be an useful tool to study statistical mechanics of long homopolymers [Grassberger 97] and other lattice statistical models [Grassberger 98, Grassberger 00]. It is also the most efficient algorithm to compute finite temperature properties of lattice protein models [Bastolla 98, Grassberger 00], although it is recently challenged by a MCMC algorithm developed by the author and coworkers [Chikenji 99].

3·5 *Bayesian Computation: Annealed Importance Sampling*

In the example of Section 3·1, an intermediate state indexed by n is a fictitious one and we are only interested in the limit of large n. In Section 3·2–Section 3·4, we dealt with examples where the index n corresponds to an index in the real world, *e.g.,* the index of row of a lattice, time-step of series, and the length of the subchain. Here we consider cases where the index n indicates a system with a parameter γ_n, *e.g.,* temperature in statistical physics and hyperparameters in Bayesian models. Such an example is discussed by Neal and called by "annealed importance sampling" [Neal 98].

Consider a parametric family of distributions $\{P_{\gamma_n}(x), n = 1, \ldots, N\}$. An example is

$$P_{\gamma_n}(x) = \{P(y|x)\}^{\gamma_n} \tilde{P}(x) \tag{25}$$

where $P(y|x)$ is likelihood with data y and $\tilde{P}(x)$ is a prior. We can formally define a "recursion relation" by

$$P_{\gamma_{n+1}}(x) = \frac{P_{\gamma_{n+1}}(x)}{P_{\gamma_n}(x)} \cdot P_{\gamma_n}(x). \tag{26}$$

Unfortunately, population Monte Carlo algorithm derived from (26) is trivial and inefficient, because the matrix $G^n(x, x')$ defined by the recursion is diagonal.

An idea by Neal is to introduce MCMC steps into the algorithm. The insertion of probabilistic state-changes that makes $P_{\gamma_n}(x)$ invariant corresponds to the insertion of a non-diagonal matrix $\widehat{G}^n$ to the right-hand-side of the equation (1), which satisfies

$$X^n(i) = \sum_j \widehat{G}^n(i, j) X^n(j). \tag{27}$$

Obviously, the introduction of such $\widehat{G}^n$ does not have an effect on the validity of the results. The adequacy of the procedure is also ensured by the relation

$$X^n(j) = [\widetilde{X}^n(j)] \tag{28}$$

is not changed by the addition of MCMC steps. The resultant algorithm consists of *finite* MCMC sweeps at each n and STEP 2 with

$$W_k = \frac{P_{\gamma_{n+1}}(x)}{P_{\gamma_n}(x)} \tag{29}$$

where $x = j(k)$ is the position of the kth walker.

As far as I know, Neal does not discuss algorithms with STEP 3, but there seems no reason against the use of STEP 3 in the algorithm. On the other hand, as Neal himself pointed out, the algorithm without STEP 3 can be regarded as a version of non-adiabatic thermodynamic integration discussed by Jarzynski [Jarzynski 97, Hendrix 01].

It is interesting to remark that the introduction of MCMC steps [Crisan 00] is possible for some of the other population Monte Carlo algorithms.

Acknowledgments

This paper is a revised version of the paper presented at 2000 Workshop on Information-Based Induction Sciences (IBIS2000, Syuzenji, Japan, 2000, July). We would like to thank to Prof. T. Higuchi

*9 A related problem (counting of "meanders") is treated in [Golinelli 00] by population Monte Carlo.

who kindly taught us about the book [Doucet 01] and some of other references in Section 3·3. We also acknowledge Dr. A. Doucet for giving us kind advices and significant information on references.

♢ References ♢

[Anderson 75] Anderson, J. B.: "A random-walk simulation of the Schrödinger equation H_3^+," *The Journal of Chemical Physics,* Vol. 63, pp. 1499-1503, 1975.

[Anderson 91] Anderson, J. B., Traynor, C. A., and Boghosian, B. M.: "Quantum chemistry by random walk: Exact treatment of many-electron systems", *The Journal of Chemical Physics,* Vol. 95, pp. 7418-7425, 1991.

[Bastolla 98] Bastolla, U., Frauenkron, H., Gerstner, E. Grassberger P., and Nadler W.,: "Testing a new Monte Carlo algorithm for protein folding," *PROTEINS: Structure, Function, and Genetics,* Vol. 32, pp. 52-66, 1998.

[Blankenbecler 83] Blankenbecler, R., and Sugar, R. L.: "Projector Monte Carlo method," *Physical Review D*, Vol. 27, pp. 1304-1311, 1983.

[Carlson 85] Carlson, J., and Kalos, M. H.: "Mirror potentials and the fermion problem," *Physical Review C*, Vol. 32, pp. 1735-1741, 1985.

[Ceperley 79] Ceperley D. M., and Kalos, M. H.: "Quantum many-body problems," *Monte Carlo Methods in Statistical Physics*, Topics in Current Physics 7, ed K. Binder, Springer, (1979, 1986),pp. 145-194.

[Cerf 95] Cerf, N., and Martin, O. C.: "Finite population-size effects in projection Monte Carlo methods", *Physical Review E*, Vol. 51, pp. 3679-3693, 1995.

[Chikenji 99] Chikenji, G., Kikuchi, M., and Iba, Y.: "Multi-self-overlap ensemble for protein folding: Ground state search and thermodynamics", *Physical Review Letters*, Vol. 83, pp. 1886-1889, 1999.

[Crisan 00] Crisan, D. and Doucet, A.:"Convergence of sequential Monte Carlo methods", CUED/F-INFENG/TR381, available at `http://www.statslab.cam.ac.uk/~mcmc/`.

[Doucet 01] Doucet, A., de Freitas, N., and Gordon N. (eds.),: "Sequential Monte Carlo Methods in Practice" Springer-Verlag, New York, 2001.

[Dreitlein 85] Dreitlein, J.: "Stochastic simulation of classical lattice systems," *Physical Review A*, Vol. 31, pp. 1658-1666, 1985.

[Garel 90] Garel, T., and Orland, H.: "Guided replication of random chain: a new Monte Carlo method," *Journal of Physics A: Mathematical and General*, Vol. 23, pp. L621-L626, 1990.

[Golinelli 00] Golinelli, O.: "A Monte Carlo study of meanders", *The European Physical Journal B*, Vol. 14, pp. 145-155, 2000.

[Gordon 95] Gordon, N., Salmond, D., and Ewing, C.: "Bayesian state estimation for tracking and guidance using the bootstrap filter," *Journal of Guidance, Control, and Dynamics,* Vol. 18, pp. 1434-1443, 1995.

[Gordon 93] Gordon, N. J., Salmond D. J., and Smith, A. F. M.:"Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE PROCEEDINGS-F,* Vol. 140, pp. 107-113, 1993.

[Grassberger 97] Grassberger, P.: "Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1000000," *Physical Review E*, Vol. 56, pp. 3682-3693, 1997.

[Grassberger 98] Grassberger, P., Fraunenkron, H., and Nadler, W.: PERM: A Monte Carlo strategy for simulating polymers and other things, *Monte Carlo Approach to Biopolymers and Protein Folding*, ed. P. Grassberger, G. T. Barkema and W. Nadler (World Scientific, Singapore, 1998) pp. 301-315.

[Grassberger 00] Grassberger,P. and Nadler, W.: "Go with the winners"-simulations, a talk presented at Heraeus Summer School, Chemnitz, Oct. 2000, e-print, available at `http://xxx.lanl.gov/abs/cond-mat/0010265`.

[Hammersley 64] Hammersley J. M., and Handscomb, D. C.: *Monte Carlo Methods,* Methuen, London and John & Sons New York, 1964.

[Handschin 69] Handschin J. E. and Mayne, D. Q.: Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering, *International Journal of Control*, Vol. 9, pp. 547-559, 1969.

[Hendrix 01] Hendrix, D.A. and Jarzynski C.: "A "fast growth" method of computing free energy differences," e-print, available at `http://xxx.lanl.gov/abs/cond-mat/0101307`.

[Hetherington 84] Hetherington, J. H.: "Observations on the statistical iteration of matrices," *Physical Review A*, Vol. 30, pp. 2713-2719, 1984.

[Higuchi 97] Higuchi, T.: "Monte Carlo filter using the genetic algorithm operators," *The Journal of Statistical Computation and Simulation,* Vol. 59, pp. 1-23, 1997.

[Irwin 94] Irwin, M., Cox, N., and Kong, A.: "Sequential imputation for multilocus linkage analysis," *Proc. Natl. Acad. Sci. USA,* Vol. 91, pp. 11684-11688, 1994.

[Isard 98] Isard, M., and Blake, A.: "Condensation - Conditional density propagation for visual tracking," *International Journal of Computer Vision,* Vol. 29, pp. 5-28, 1998.

[Jarzynski 97] Jarzynski, C.: "Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach," *Physical Review E,* Vol. 56, pp. 5018-5035, 1997.

[John 87] John S., and Remler, E. A.: "Solution of the quantum Liouville equation as a stochastic process," *Annals of Physics*, Vol. 180, pp. 152-165, 1987.

[Kalos 62] Kalos, M. H.: "Monte Carlo calculation of the ground state of three- and four-body nuclei," *Physical Review,* Vol. 128, pp. 1791-1795, 1962.

[Kalos 84] Kalos, M. H.: "Optimization and the many-fermion problem," *Monte Carlo Methods in Quantum Problems,* ed. M. H. Kalos, NATO ASI Series C, Vol. 125, Reidel, 1984, pp. 19-31.

[Kitagawa 96] Kitagawa, G.: "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics,* Vol. 5, pp. 1-25, 1996.

[Kitagawa 98] Kitagawa, G.: "A self-organizing state-space model," *Journal of American Statistical Association,* Vol. 93, pp. 1203-1215, 1998.

[Liu 94] Liu, Z., Zhang, S., and Kalos, M. H.: "Model fermion Monte Carlo method with antithetical pairs," *Physical Review E*, Vol. 50, pp. 3220-3229, 1994

[Liu 95] Liu J. S., and Chen, R.: "Blind deconvolution via sequential imputation," *Journal of the American Statistical Association,* Vol. 90, pp. 567-574, 1995.

[Liu 98] Liu J. S., and Chen, R.: "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association,* Vol. 93, pp. 1032-1044, 1998.

[Metropolis 49] Metropolis, N., and Ulam, S.: "The Monte Carlo method, " *Journal of the American Statotical Aassociation,* Vol. 44, pp. 335-341, 1949.

[Neal 98] Neal, R. M.: "Annealed importance sampling," Technical Report No. 9805, Dept. of Statistics, Univ. of Tronto, 1998, available at `http://www.cs.utoronto.ca/~radford/` .

[Negele 86] Negele, J. W.: "Monte Carlo studies of nuclear many-particle systems," *Journal of Statistical Physics*, Vol. 43, pp. 991-1015, 1986.

[Negele 89] Negele J. W., and Brommer, K. D.: "Calculation of ground states for many-fermion systems", *Annals of Physics*, Vol. 192, pp. 119-132, 1989.

[Nightingale 88] Nightingale, M. P., and Blöte, H. W. J.: "Monte Carlo calculation of free Energy, critical point, and surface critical behavior of three-dimensional Heisenberg ferromagnets," *Physical Review Letters,*, Vol. 60, pp. 1562-1565, 1988.

[Orland 98] Orland, H.: "The Monte Carlo growth method," *Monte Carlo Approach to Biopolymers and Protein Folding*, ed. P. Grassberger, G. T. Barkema and W. Nadler (World Scientific, Singapore, 1998) pp. 90-97.

[De Raedt 92] De Raedt, H., and von der Linden, W.: "Quantum lattice problems," *The Monte Carlo Method in Condensed Matter Physics*, Topics in Applied Physics 71, ed K. Binder, Springer, (1992, 1995), pp. 249-284.

[Rosenbluth 55] Rosenbluth, M. N., and Rosenbluth, A. W.: "Monte Carlo calculation of the average extension of molecular chains," *The Journal of Chemical Physics*, Vol. 23, pp. 356-359, 1955.

[Schmidt 84] Schmidt, K. E., and Kalos, M. H.: "Few- and many- fermion problems," *Applications of the Monte Carlo Method in Statistical Physics*, Topics in Current Physics 36, ed. K. Binder, Springer, (1984, 1987), pp. 125-143.

[Schmidt 92] Schmidt, K. E., and Ceperley, D. M.: "Monte Carlo techniques for quantum fluids, solids and droplets," *The Monte Carlo Method in Condensed Matter Physics*, Topics in Applied Physics 71, ed. K. Binder, Springer, (1992, 1995), pp. 205-248.

[Seno 96] Seno, F., Vendruscolo, M., Maritan, A., and Banavar, J. R.: "Optimal protein design procedure," *Physical Review Letters*, Vol. 77, pp. 1901-1904, 1996.

[Shaw 91] Shaw, L. J., and Henley, C. L.: "A transfer-matrix Monte Carlo study of random Penrose tilings," *Journal of Physics A*, Vol. 24, pp. 4129-4153, 1991.

[Thijssen 90] Thijssen, J. M., and Knops, H. J. F.: "Monte Carlo transfer-matrix study of the frustrated XY model," *Physical Review B*, Vol. 42, pp. 2438-2444, 1990.

[Wall 59] Wall, F. T., and Erpenbeck, J. J.: "New method for the statistical computation of polymer dimensions," *The Journal of Chemical Physics,* Vol. 30, pp. 634-637, 1959.

〔担当委員：新田克己〕

Received October 31, 2000.

Author's Profile

Iba, Yukito

Yukito Iba is an Associate Professor of the Institute of Statistical Mathematics (ISM), Minato-ku, Tokyo. He was Born in Tokyo in 1959 and majored in statistical physics at the University of Tokyo. Since 1988, he is a staff member of ISM. He obtained Ph. D degree in physics from Osaka University in 2000. His thesis title is "Bayesian Statistics, Statistical Physics, and Monte Carlo Methods: An interdisciplinary study". He is interested in cross-fertilization of different fields of sciences, specifically, statistics and statistical physics.