

## Chapter 2

# Geometric Allocation Approach in Markov Chain Monte Carlo

### 2.1 Markov Chain Monte Carlo

The Monte Carlo method was first devised by Ulam et al. [61] in 1947, and it has been enjoyed for a wide variety of applications in mathematics, physics, statistics, etc. According to the literature, the method is defined as a simulation that solves a deterministic problem by means of a stochastic way. Ulam and the coworkers used the Monte Carlo method for neutron diffusion problem in fissile material of atomic bombs and for the eigenvalue problem of the Schrödinger equation at Los Alamos during the World War II. In the both applications, the average of random sampling converges to a target integral.

Let  $s \in N$  be a dimension and  $D \subset R^s$  be an integral range on the Lebesgue measure (now we assume  $D$  is compact as topological space for simplicity),  $x \in D$  be a continuous state variable,  $f : R^s \rightarrow R$  be an integrand function, and  $I$  be a target integral defined as

$$I = \int_D f(x)dx, \quad (2.1)$$

which takes a finite value. If  $f$  has a finite mean and we generate  $M$  independent samples,  $x_1, x_2, \dots$ , in  $D$ , then the average of the function values at the samples

$$\hat{I}_M = \frac{1}{M} \sum_{i=1}^M f(x_i) \quad (2.2)$$

converges to the target value  $I$  almost surely, which is proved by *the strong law of large numbers*:

$$\hat{I}_M \xrightarrow{a.s.} I. \quad (2.3)$$

Moreover, if  $f$  has a finite variance  $v$ , *the central limit theorem* states the weak convergence (convergence in distribution):

$$\sqrt{M}(\hat{I}_M - I) \xrightarrow{d} N(0, v). \quad (2.4)$$

Here  $N(\mu', v')$  is the normal (Gaussian) distribution with the mean  $\mu'$  and the variance  $v'$ .

The key point of the method is how the samples are chosen. The simple uniform sampling on the range  $D$  is feasible only for very low-dimensional problems. The relative error of the Monte Carlo average gets exponentially large as the dimension  $s$  becomes high, which is called the *the curse of dimensionality* [6]. For some low-dimensional problems (typically  $s < 10$  or 100), the quasi Monte Carlo techniques have been developed [14, 33]. These are very useful in finance, for example; the price of derivative needs to be calculated very quickly. The idea of the quasi Monte Carlo is that the samples should be distributed in integral range with low “discrepancy” as shown in the widely known Koksma-Hlawka inequality [27]. Some algorithms for generation of low-discrepancy (artificial) sequence have been proposed, such as the digital  $(t, m, s)$ -nets (including the Sobol’ sequence [53], the Faure sequence [17], and the Niederreiter sequence [47]), the Halton sequence, the lattice rule, the Kronecker sequence and so on. By using these artful sequences, the order of convergence is drastically improved for some cases. For such a successful improvement, one of the key concepts is the *effective dimension* that is corresponding to the number of effective variables contributing to the variance of Monte Carlo average. Although these recent techniques have many interesting applications, it is still difficult to work well in high-dimensional problems including many physical systems.

An effective way for conquering the dimensional problem is *the importance sampling* where samples that have larger contribution to the integral are selected more often. The Markov chain Monte Carlo (MCMC) method [24, 32, 52] is a computational method that *approximately* generates samples stochastically from any target distribution. As a result of the importance sampling, an expectation value of a function (an observable)  $O : X \rightarrow R$  ( $X$  is a state space) on the target distribution can be approximated from the simple average of the  $M$  samples:

$$\langle O \rangle = \frac{\int w(x) O(x) dx}{\int w(x) dx} \simeq \frac{1}{M} \sum_i O(x_i), \quad (2.5)$$

where  $w : X \rightarrow R$  is a weight function (measure). The MCMC method is a powerful tool for systems with multiple degrees of freedom (correlated high-dimensional problem) that are very important in the condensed matter physics. In this thesis, we use and extend the MCMC method for investigation of strongly correlated systems.

In the MCMC method, we start from an initial state (configuration), and the next state is stochastically chosen depending on the present configuration. Then the sequence of states (samples) constructs a Markov chain; the Markov property comes from the fact of dependency only on the one-step previous state. In other words, we set an initial distribution and continue to operate a kernel to the distribution at each Monte Carlo step. When an ergodicity condition holds, the distribution converges to an

equilibrium distribution. In the end, we can get samples from the target (equilibrium) distribution.

For a finite state space, the condition becomes easy if we take the natural discrete topology for the state space. It is because every open set is closed in this topology space and thus the Borel algebra (field) is trivial, which is the smallest closed set including all open sets. For a general space, however, the convergence condition is complicated. Usually, if a Markov chain is Harris positive recurrent and aperiodic, the chain is called ergodic [41, 59].<sup>1</sup> It is far from trivial to prove the ergodicity of a Markov chain in general. Necessarily in many practical simulations, the condition is simply assumed. Then, of course, we have to check the MC averages converge to the correct value. About our simulations in this thesis, we first confirmed the correctness by comparing with a more precise calculation, e.g., the numerical exact diagonalization, in simple or small systems (the diagonalization can be done only when the dimension of the Hilbert space is very small). Although we have not succeeded in mathematically proving the ergodicity in some cases, such a check is convincing enough to assure us the validity of the simulations. We, however, should keep in mind the mathematical conditions especially when we try to develop and improve the method.

Instead of the curse of dimensionality, the MCMC method suffers from the sample correlation. Since the next configuration is generated (updated) from the previous one, the samples are not independent of each other. Then the correlation gives rise to two problems: we have to wait for the distribution convergence (equilibration) before sampling, and the number of effective samples is decreased.

The former convergence problem is quantified by a distribution distance to the target. In many cases, the total variation distance

$$\| P^m(x, \cdot) - \pi(\cdot) \| = \sup_{A \in \mathcal{B}(X)} |P^m(x, A) - \pi(A)|, \quad (2.6)$$

is used, where  $\mathcal{B}(X)$  is the Borel algebra,  $A$  is a Borel set, and  $P^m(x, A) = P(x_m \in A | x_0 = x)$  is the  $m$ -step transition probability. For a finite state space, this is corresponding to the second largest eigenvalue in absolute value.

As an assessment for the latter problem, the decrease of the number of effective samples, the integrated autocorrelation time is defined as

$$\tau_{\text{int}} = \sum_{t=1}^{\infty} C(t) \quad (2.7)$$

$$C(t) = \frac{\langle O_{i+t} O_i \rangle - \langle O \rangle^2}{\langle O^2 \rangle - \langle O \rangle^2}, \quad (2.8)$$

---

<sup>1</sup> In the literature of physics, the ergodicity is confused with the irreducibility in many cases. If a state space is finite and a Markov chain is aperiodic, the ergodicity and the irreducibility are equivalent but they differ in a general state space.

where  $O_i$  is an observable at the  $i$ -th Monte Carlo step, and  $C(t)$  is almost independent of  $i$  after the distribution convergence. This autocorrelation decreases the number of effective samples as

$$M_{\text{eff}} \simeq \frac{M}{1 + 2\tau_{\text{int}}}, \quad (2.9)$$

where  $M$  is the total number of samples in simulations. Although an MCMC method satisfying appropriate conditions guarantees correct results asymptotically in principle<sup>2</sup> [41], variance reduction of relevant estimators is crucial for the method to work in practice. If the central limit theorem holds, as we mentioned, the variance of expectations decreases as  $v/M \simeq \text{var}(f)/M_{\text{eff}}$ , where  $v$  is called the asymptotic variance that depends on the integrand function and the update method through the autocorrelation time.

What we have to concern is, thus, to shorten the distribution convergence (burn-in) time and to reduce the asymptotic variance. Optimal strategies against these two criteria must differ as we will explain in the next section. For most lattice systems (Markov random fields), however, it is presumably possible to improve conventional sampling methods in the both quantities. It is because the usual method is far from the global optimal strategy in the both respect.

There are three key points for the MCMC method to be effective. One is the choice of the ensemble. From the view of this respect, the extended ensemble methods, such as the multicanonical method [7] and the replica exchange method [28], have been proposed and applied successfully to protein folding problems, spin glasses, etc. The second is the selection of candidate configurations. The cluster algorithms, e.g., the Swendsen-Wang algorithm [57] and the loop algorithm [16], can overcome the critical slowing down by taking advantage of mapping to graph configurations in many physical models.<sup>3</sup> The hybrid (Hamiltonian) Monte Carlo method performs a simultaneous move, where the candidate state is chosen by taking the advantage of the Newtonian dynamics. The third is the determination of the transition probability, given candidate configurations. We focus our interest on this optimization problem of the probabilities in the following sections of this chapter.

## 2.2 Optimization of Transition Kernel

We will explain previous optimization approaches for transition kernel in the MCMC method in this section. Let us consider a finite state space now for simplicity. In the method, for the equilibrium distribution to be a target distribution, the (total) balance, the invariance of target distribution,

---

<sup>2</sup> All estimators measured in the MCMC method are biased. The ergodicity, however, ensures they are (strong) consistent. That is, the sequence of Monte Carlo average converges to the correct result in probability (almost surely).

<sup>3</sup> The improved estimator on the graph configuration reduces also the variance of the integrand function  $\text{var}(f)$ .

$$w(c_i) = \sum_j w(c_i) p(c_i \rightarrow c_j) = \sum_j w(c_j) p(c_j \rightarrow c_i) \quad \forall i, \quad (2.10)$$

is imposed to the transition kernel.<sup>4</sup> In this equation,  $p(c_i \rightarrow c_j)$  is a transition probability from configuration  $c_i$  to  $c_j$ , and  $w(c_i)$  is an unnormalized weight (measure) of configuration  $c_i$  that is proportional to the normalized weight  $\pi_i$ . In most practical implementations, the Metropolis-Hastings algorithm [26, 40] (we call it simply the Metropolis algorithm below) or the heat bath algorithm [5, 11], namely, the Gibbs sampler [23] have been used for the determination of the transition probabilities. The next state is chosen by probability

$$p(c_i \rightarrow c_j) = \frac{1}{n-1} \min\left(1, \frac{w(c_j)}{w(c_i)}\right) \quad i \neq j \quad (2.11)$$

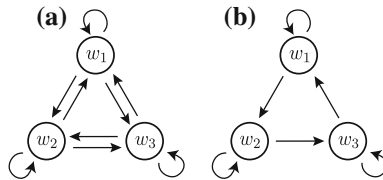
in the Metropolis algorithm, and

$$p(c_i \rightarrow c_j) = \frac{w(c_j)}{\sum_k w(c_k)} \quad \forall i, j \quad (2.12)$$

in the heat bath algorithm among  $n$  candidate states. These canonical algorithms satisfy the detailed balance, the reversibility,

$$w(c_i) p(c_i \rightarrow c_j) = w(c_j) p(c_j \rightarrow c_i) \quad \forall i, j, \quad (2.13)$$

which is a sufficient condition for the total balance (2.47). Under this condition, thanks to the simple property that every elementary transition balances with a corresponding inverse process (Fig. 2.1), it becomes easy to find a qualified transition probability by solving the equation for each pair of configurations as the Metropolis and the heat bath algorithm do. Attempts to optimizing the transition matrix have concentrated within this sufficient condition so far. We will review the previous optimization approaches below.



**Fig. 2.1** Stochastic flow with the detailed balance (a) and without it (b). In the former, the flow balances with a corresponding inverse process. On the other hand, in the latter, a net stochastic flow exists as the result of breaking the detailed balance

<sup>4</sup> Note that there has been some interesting progress for generating samples from the target distribution asymptotically, the transition kernel being modified in an adaptive procedure [3].

### 2.2.1 Criteria of Markov Chain

Let  $S$  be a finite state space and  $n$  be the number of elements of  $S$ . Assume  $n$  is small enough to calculate the transition probabilities from all states to all. The discussion here can be easily extended to a conditional case in a huge state space; that is, we choose a state variable and update it on the condition that other variables are fixed. Let us also define an average of function  $f$  and inner product of  $f$  and  $g$  as

$$\langle f \rangle = \sum_{s \in S} f(s) \pi(s), \quad (2.14)$$

and

$$\langle f, g \rangle_\pi = \sum_{s \in S} f(s) g(s) \pi(s), \quad (2.15)$$

respectively. The following theorem connects the asymptotic variance and the eigenvalues of the reversible transition matrix, which are always real.

**Theorem 2.2.1** *Let  $P$  be an irreducible and a reversible (w.r.t.  $\pi$ ) stochastic matrix. Let  $x(0), x(1), \dots$  be a Markov chain on state space  $S$  with transition matrix  $P$  and  $f: S \rightarrow \mathbb{R}$ . For any initial distribution,*

$$v(f, P, \pi) = \lim_{M \rightarrow \infty} M \operatorname{var} \left( \frac{1}{M} \sum_{k=0}^{M-1} f[x(k)] \right) \quad (2.16)$$

$$= \langle (I - P)^{-1} (I + P) (f - \langle f \rangle 1), f - \langle f \rangle 1 \rangle_\pi. \quad (2.17)$$

This fact was proved first by Peskun [48]. Easy computations show that the above limit is equal to

$$\sum_{k=2}^n \frac{1 + \lambda_k}{1 - \lambda_k} \langle f, \bar{e}_k \rangle_\pi^2, \quad (2.18)$$

where  $P \bar{e}_k = \lambda_k \bar{e}_k$ ,  $\|\bar{e}_k\|_\pi = 1$ . This is also described as

$$\sum_{k=2}^n \frac{1 + \lambda_k}{1 - \lambda_k} a_k \operatorname{var}(f), \quad (2.19)$$

where  $a_k$  are some nonnegative constants such that  $\sum_k a_k = 1$ . This formula links the asymptotic variance  $v(f, P, \pi)$  to the eigenvalues and eigenvectors of  $P$ . In order to make the asymptotic variance small, it seems useful to use dynamics  $P$  with possibly all negative and small eigenvalues (except the largest one which is 1). In this sense, one can say that negative eigenvalues help the asymptotic variance get small.

The negative eigenvalues, however, do not always help the weak convergence (convergence in distribution) gets rapid. The  $m$ -step transition matrix  $P^m(x, y) = P(x_m = y | x_0 = x)$  is represented as

$$P^m(x, y) = \sum_{k=1}^n e_k(x) e_k(y) \lambda_k^m. \quad (2.20)$$

The dominant term becomes  $e_1(\cdot) = \pi(\cdot)$  for large  $m$  since  $\lambda_1 = 1$ . It is clear that the speed of the convergence depends on the second largest eigenvalue in absolute value.

These two goals seem to be in different point. We also mention the noticeable relation between them:

$$v(f, P, \pi) \leq \frac{1 + |\lambda_2|}{1 - |\lambda_2|} \text{var}(f) \leq \frac{2}{1 - |\lambda_2|} \text{var}(f). \quad (2.21)$$

Therefore, the spectral gap  $1 - |\lambda_2|$  that determines the convergence rate gives also the upper bound of the asymptotic variance.

If the convergence is so slow that it is not possible to ignore the intrinsic bias of the MCMC sampler, we have to reduce the absolute value of the second eigenvalue. Once the equilibrium state becomes accessible in feasible time, the statistical efficiency, the asymptotic variance is more important to calculation.

### 2.2.2 Peskun's Theorem

There is a simple theorem as a guideline for the optimization. Let  $X$  be a discrete random variable following distribution  $\pi$ , and let  $P$  be the transition matrix of a Markov chain with  $\pi$  as its invariant distribution. Following Peskun [48], we define  $P_2 \geq P_1$  for any two transition matrices if each of the off-diagonal elements of  $P_2$  is greater than or equal to the corresponding off-diagonal elements of  $P_1$ . The following lemma is Theorem 2.1.1 of Peskun [48].

**Lemma 2.2.1 (Peskun)** *Suppose each of the irreducible transition matrices  $P_1$  and  $P_2$  is reversible for the same invariant probability distribution  $\pi$ . If  $P_2 \geq P_1$ , then, for any  $f$ ,*

$$v(f, P_1, \pi) \geq v(f, P_2, \pi), \quad (2.22)$$

where

$$v(f, P, \pi) = \lim_{M \rightarrow \infty} M \text{var}(\hat{I}_M), \quad (2.23)$$

and  $\hat{I}_M = \sum_{i=1}^M f(x_i)/M$  is an estimator of  $I = E_\pi(f)$  using  $M$  consecutive samples from the Markov chains.

Based on this theorem, a modified Gibbs sampler called the “Metropolized Gibbs sampler” was proposed by Liu [34, 35]. Assume ordering of the states as  $\pi_1 \leq \pi_2 \leq \dots \leq \pi_n$ , where  $\pi_i = w_i / \sum_i w_i$ . In the heat bath algorithm, we choose the next state with forgetting the current state. Meantime, it is obvious that the rejection that the current state is chosen as the next state should be avoided as much as possible from the Peskun’s theorem. Then, some will think we can exclude the current state at first proposal and use the Metropolis algorithm for going to the chosen state actually. Applying the heat bath algorithm at the first proposal seems plausible. This modified Gibbs sampler is reduced to the Metropolis algorithm for  $n = 2$ , not to the usual Gibbs sampler, which is also called the Barker’s algorithm [5]. That is why it is called the Metropolized Gibbs sampler. By this approach, the transition matrix  $P$  is described as

$$P_{ij}^{MG} = \min(\pi_i / (1 - \pi_j), \pi_j / (1 - \pi_i)). \quad (2.24)$$

The matrix forms as

$$P^{MG} = \begin{bmatrix} 0 & \frac{\pi_1}{1-\pi_1} & \frac{\pi_1}{1-\pi_1} & \dots & \frac{\pi_1}{1-\pi_1} \\ \frac{\pi_2}{1-\pi_1} & 1 - \dots & \frac{\pi_2}{1-\pi_2} & \dots & \frac{\pi_2}{1-\pi_2} \\ \frac{\pi_3}{1-\pi_1} & \frac{\pi_3}{1-\pi_2} & 1 - \dots & \dots & \frac{\pi_3}{1-\pi_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\pi_n}{1-\pi_1} & \frac{\pi_n}{1-\pi_2} & \frac{\pi_n}{1-\pi_3} & \dots & 1 - \dots \end{bmatrix}, \quad (2.25)$$

where  $P_{ij}$  is the transition probability from  $j$  to  $i$ . Let  $P^{HB}$  be the transition matrix by using the normal heat bath algorithm. It follows that  $P^{MG} \geq P^{HB}$  and hence the Peskun’s theorem says the following theorem.

**Theorem 2.2.2** *The Metropolized Gibbs sampler for discrete random variables as defined above is statistically more efficient than the usual heat bath algorithm (Gibbs sampler).*

### 2.2.3 Worst Case Solution

The above “Metropolization” always set to zero the diagonal element for the smallest-weight states. Interestingly, Frigessi et al. [21] showed that this procedure gives a kind of optimal property. Let us follow their theorems below.

At first, we have a general fact about the second largest eigenvalue as follows.

**Theorem 2.2.3 (Frigessi, Hwang, Younes 1)** *(a) The second largest eigenvalue of any stochastic matrix  $P$ , reversible w.r.t.  $\pi$ , is greater than or equal to*



$$-\frac{\pi_1}{1-\pi_1}. \quad (2.26)$$

For all matrices whose second largest eigenvalue attains this lower bound, the corresponding eigenvector is

$$e_2 = (1 - \pi_1, -\pi_1, \dots, -\pi_1)^T. \quad (2.27)$$

Furthermore, their first column has a zero as first entry and all other elements are equal to

$$\frac{\pi_1}{1-\pi_1}. \quad (2.28)$$

That is, the matrix forms as

$$P = \begin{bmatrix} 0 & \frac{\pi_1}{1-\pi_1} & \cdots & \frac{\pi_1}{1-\pi_1} \\ \frac{\pi_2}{1-\pi_1} & & & \\ \vdots & & P_2 & \\ \frac{\pi_n}{1-\pi_1} & & & \end{bmatrix}, \quad (2.29)$$

where the submatrix  $P_2$  is in detailed balance again. This is nothing but the Metropolization for the smallest-weight state.

(b) The above construction can be iterated to finally obtain a matrix with the following properties: (i) all the elements along the diagonal are zero, except possibly the last one; (ii) its eigenvalues are  $1 = \lambda_1 > 0 > \lambda_2 \geq \cdots \geq \lambda_n$  and satisfy the property that  $\lambda_{i+1}$  attains the smallest possible value among all matrices (reversible w.r.t.  $\pi$ ) that already possess the eigenvalues  $1, \lambda_2, \dots, \lambda_i$ ; (iii) its columns have constant entries under the diagonal, which are, respectively,  $-\lambda_2, \dots, -\lambda_n$ . (iv) its eigenvectors are

$$e_1 = 1 \quad (2.30)$$

and

$$e_{k+1} = \delta_k - \langle \delta_k | 1, 2, \dots, k-1 \rangle_{\pi}, \quad k = 1, \dots, n-1, \quad (2.31)$$

where  $1 = (1, \dots, 1)^T$ ,  $\delta_k = (0, \dots, 1, \dots, 0)^T$ , and  $\langle f | 1, 2, \dots, k-1 \rangle_{\pi}$  is the conditional expectation of  $f$  given the  $\sigma$ -algebra generated by the sets  $1, \dots, k-1$  under the probability  $\pi$ ; in vector notation,

$$e_{k+1} = \left( \underbrace{0, \dots, 0}_{k-1 \text{ terms}}, 1 - \frac{\pi_k}{\pi_k + \cdots + \pi_n}, -\frac{\pi_k}{\pi_k + \cdots + \pi_n}, \dots, -\frac{\pi_k}{\pi_k + \cdots + \pi_n} \right)^T. \quad (2.32)$$

(c) Moreover, this matrix is the unique one which satisfies the previous condition (ii).

The resulting matrix, which we call the iterative Metropolized Gibbs sampler, is described as

$$P^{IMG} = \begin{bmatrix} 0 & y_1 & y_1 & \cdots & y_1 \\ \frac{w_2}{w_1} y_1 & 0 & y_2 & \cdots & y_2 \\ \frac{w_3}{w_1} y_1 & \frac{w_3}{w_2} y_2 & 0 & \cdots & y_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} y_1 & \frac{w_n}{w_2} y_2 & \frac{w_n}{w_3} y_3 & \cdots & 1 - y_1 - y_2 - \cdots \end{bmatrix}, \quad (2.33)$$

where  $y_1 = \pi_1/(1 - \pi_1)$ ,  $y_2 = (1 - y_1)\pi_2/(1 - \pi_1 - \pi_2)$ ,  $\dots$ . We do not prove this theorem here, but refer also some remarks.

**Remark 2.2.1** A final nonzero element on the diagonal may remain in the matrix of part (b). In fact this happens if and only if  $\pi_{n-1} \neq \pi_n$ .

**Remark 2.2.2** It is not difficult to write down the values of  $\lambda_2, \dots, \lambda_n$ . They are

$$\lambda_{k+1} = -y_k = -\frac{\pi_k}{\pi_{k+1} + \cdots + \pi_n} \prod_{\ell=1}^{k-1} \left(1 - \frac{\pi_\ell}{\pi_{\ell+1} + \cdots + \pi_n}\right). \quad (2.34)$$

If  $\pi_{k-1} = \pi_k$ , then  $\lambda_k = \lambda_{k+1}$ . It is not surprising that  $\delta_1 - \langle \delta_1 \rangle 1$  realizes the above bound. In other words, this means that the most difficult quantity to estimate by time averages is the probability of the least likely state. This seems consistent with intuition.

In addition, we have a relation between the eigenvalues and the asymptotic variance of integrand.

**Corollary 2.2.1** Let  $P$  be a stochastic matrix, reversible w.r.t.  $\pi$ . Let  $v(P, \pi)$  be the maximum asymptotic variance of  $(1/M) \sum_{i=1}^M f(x_i)$  for norm 1 functions  $f$ . Then

$$v(P, \pi) \geq \frac{1 + \lambda_2}{1 - \lambda_2} \quad (2.35)$$

$$\geq 1 - 2\pi_1, \quad (2.36)$$

and any matrix that realizes this equality must have the properties given in Theorem 2.2.3(a) and hence be of the form (2.29).

This corollary says that the second largest eigenvalue is the lower bound of the asymptotic variance in the *worst* case, and the function is proportional to the eigenvector (2.32) that belongs to the second largest eigenvalue.

### 2.2.4 Four Optimization Problems

For an optimization problem, there are two interesting cases theoretically: the worst case and the average case. Our problem of the asymptotic variance is the minimization of the following two quantities; for the worst case

$$v(P, \pi) = \sup_{\langle f \rangle=0, \langle f^2 \rangle=1} v(f, P, \pi), \quad (2.37)$$

and for the average case

$$\bar{v}(P, \pi) = \int_{\langle f \rangle=0, \langle f^2 \rangle=1} v(f, P, \pi) dS(f), \quad (2.38)$$

where  $dS(f)$  stands for the normalized surface area. We can consider these problem with/without the detailed balance. Set  $\mathcal{P} = \{P : \pi P = \pi\}$  and  $\mathcal{R} = \{P : P \text{ is reversible w.r.t. } \pi\}$ . There are following 4 optimization problems:

- (i) Minimize  $v(P, \pi)$  over all  $P \in \mathcal{R}$ .
- (ii) Minimize  $v(P, \pi)$  over all  $P \in \mathcal{P}$ .
- (iii) Minimize  $\bar{v}(P, \pi)$  over all  $P \in \mathcal{R}$ .
- (iv) Minimize  $\bar{v}(P, \pi)$  over all  $P \in \mathcal{P}$ .

The above theorem we mentioned is the solution of (i), the worst case of reversible kernels.

### 2.2.5 Average Case Solution

Interestingly, Hwang these days has given the solution of (iv), the average case of irreversible kernel [30]. He proved the following important lemma and showed optimal solutions.

**Theorem 2.2.4 (Hwang)** *The average asymptotic variance is expressed by the trace of an inverse matrix:*

$$\bar{v}(P, \pi) = \int_{\langle f \rangle=0, \langle f^2 \rangle=1} v(f, P, \pi) dS(f) \quad (2.39)$$

$$= \frac{2}{n-1} \text{tr}(1 - P)^{-1} - 1. \quad (2.40)$$

Then the trace has a lower bound:

$$\text{tr}(1 - P)^{-1} = \sum_j \pi_j E_\pi(T_j) \geq \sum_{i=1}^n (i-1) \pi_i, \quad (2.41)$$

where

$$T_i = \inf\{t \geq 0 : x_t = i\}, \quad (2.42)$$

is a stopping time and  $x_t$  is a generated state at  $t$ -th Monte Carlo step.

Then, there are at most  $2^{n-2}$  transition matrices reaching the minimum. For the each transition matrix,

$$p_{i,i} = \begin{cases} 0 & i < n \\ \frac{\pi_n - \pi_{n-1}}{\pi_n} & i = n \end{cases} \quad (2.43)$$

and one of the following holds:

- $p_{2,1} = 1$  and  $p_{1,i} = \pi_1/\pi_i$  for some  $i$ .
- $p_{1,2} = \pi_1/\pi_2$  and  $p_{j,1} = 1$  for some  $j$ .

According to this theorem, the specific transition matrix,

$$P^{OPA} = \begin{bmatrix} 0 & 0 & \dots & 0 & \frac{\pi_1}{\pi_n} \\ 1 & 0 & \dots & 0 & \frac{\pi_2 - \pi_1}{\pi_n} \\ 0 & 1 & \dots & 0 & \frac{\pi_3 - \pi_2}{\pi_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \frac{\pi_n - \pi_{n-1}}{\pi_n} \end{bmatrix}, \quad (2.44)$$

is one of the optimal matrices that gives the minimized average asymptotic variance. Note that the problems (ii) and (iii) are still open.

### 2.2.6 For Lattice Systems

When we can control the all matrix elements, the above theorems will be powerful. For lattice models (Markov random fields), however, we have to deal with a exponentially large number of states such that we need to update variables *locally* one by one. In fact, this is the main philosophy of the MCMC. Then, the optimization problem of the asymptotic variance becomes very difficult, and the above optimization schemes for local variables may not be efficient for the whole transition matrix.

On lattice systems, Frigessi et al. proved an interesting theorem. Let  $D$  be a finite lattice (say,  $D \subset \mathbb{Z}^2$ ),  $S_0$  be a finite set and  $S = (S_0)^D$ . An element  $x$  of  $S$  will thus be a  $|D|$ -tuple,  $x = (x_\sigma)_{\sigma \in D}$ ,  $x_\sigma \in S_0$ . We consider the measure  $\pi$  on  $S$  given by  $\pi(x) = \exp(-\beta U(x))/Z_\beta$ , where  $\beta$  is an inverse temperature and  $Z_\beta$  is a partition function (normalizing constant). We, now, consider the random update where the site to be updated is chosen uniformly at random. Their theorem is as follows.

**Theorem 2.2.5 (Frigessi, Hwang, Younes 2)** (a) For any nonconstant  $f$ ,

$$v(f, P^{MG_1}, \pi) < v(f, P^{HB}, \pi), \quad (2.45)$$

where  $P^{MG_1}$  is the transition matrix which is constructed by Metropolizing for only the minimum weight states in the local updates as the form (2.29), and  $P^{HB}$  is that by the heat bath algorithm. As a consequence, the second eigenvalue of  $P^{MG_1}$  is strictly smaller than the that of  $P^{HB}$ .

(b) If  $\beta$  is large enough and the potential  $U$  has at least two local minimums, then the second eigenvalue in absolute value of  $P^{MG_1}$  is strictly smaller than that of  $P^{HB}$ .

This theorem states that the iterative Metropolized Gibbs sampler is always better than the heat bath algorithm in terms of asymptotic variance, and also weak convergence at least low temperature.

### 2.2.7 Other Approaches

As applications of the Metropolized Gibbs sampler, Loison et al. [38] researched the efficiency of the (single) Metropolized Gibbs sampler (they called it the restricted direct heat bath algorithm) in some  $O(N)$  models. Pollet et al. [49] applied the iterative version to the Potts model and quantum  $XY$  model by using the worm (directed-loop) algorithm, which will be explained in Sect. 3.4.3.

As other approaches, Mira tried to modify the transition matrix by directly moving the probabilities from the diagonal elements to off diagonal elements [42, 43]. Green et al. thought a multi trial after rejections [25]. Chiang et al. investigated the asymptotic convergence rate in low-temperature limit [10]. Baldi et al. researched the convergence issue of lattice systems (Markov random fields) [4], and Frigessi et al. investigated the computational complexity of finite Markov random fields [22]. As a reinterpretation, Billera et al. showed a simple diagram of the Metropolis algorithm [8]. Moreover, for enhancement of negative correlation, Frigessi et al. considered an antithetic coupling of two Gibbs sampler chains [20]. Although some optimization approaches for the transition kernel have been proposed, they are based on the Metropolis algorithm or the Gibbs sampler and they do not seem to improve the efficiency drastically.

For the optimization of local transition matrix, it seems intuitively efficient to minimize the diagonal elements (we call them the rejection rates). Actually, even for a simple update case of finite number of candidates, the conventional methods fail to minimize the diagonal elements. We will present a new method that constructs a rejection-minimized transition matrix in the next section.

### 2.3 Geometric Allocation

In this section, we introduce a novel method that constructs a transition kernel by a geometric approach. This method can find solutions by applying a graphical procedure, *weight allocation*, instead of solving the detailed balance equation algebraically as before. Surprisingly, it is *always* possible to find a solution that minimizes the average rejection rate.

In the MCMC method, we update configuration (or state) variables locally and run over the whole system. Now, let us consider updating one discrete variable as an elementary process, e.g., flipping a single spin in the Ising or Potts models [63] as Fig. 2.2. Given an environmental configuration, we would have  $n$  candidates (including the current one) for the next configuration. The weight of each candidate configuration (or state) is given by  $w_i$  ( $i = 1, \dots, n$ ), to which the equilibrium probability measure is proportional. Although the total and detailed balance are usually expressed in terms of the weights  $\{w_i\}$  and the transition probabilities  $\{p_{i \rightarrow j}\}$  from state  $i$  to  $j$ , it is more convenient to introduce a quantity  $v_{ij} := w_i p_{i \rightarrow j}$ , which corresponds to the amount of (raw) stochastic flow from state  $i$  to  $j$ . The law of probability conservation and the total balance are then expressed as

$$w_i = \sum_{j=1}^n v_{ij} \quad \forall i \quad (2.46)$$

$$w_j = \sum_{i=1}^n v_{ij} \quad \forall j, \quad (2.47)$$

respectively. The average rejection rate is written as

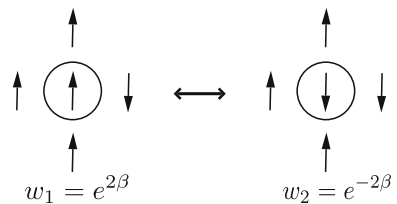
$$\sum_i v_{ii} / \sum_i w_i. \quad (2.48)$$

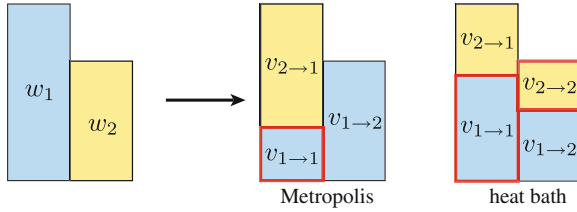
Also, it is straightforward to confirm that  $\{v_{ij}\}$  satisfy

$$v_{ij} = \frac{1}{n-1} \min[w_i, w_j] \quad i \neq j \quad (2.49)$$

for the Metropolis algorithm with the flat proposal distribution, and

**Fig. 2.2** Single spin (Elementary) update in the ferromagnetic Ising model and the weights of each configuration. The parameter  $\beta$  is an inverse temperature





**Fig. 2.3** Example of the weight allocation by the Metropolis and heat bath algorithms for  $n = 2$ . The regions with thick frame denote the rejection rates

$$v_{ij} = \frac{w_i w_j}{\sum_{k=1}^n w_k} \quad \forall i, j \quad (2.50)$$

for the heat bath algorithm (Gibbs sampler), where the detailed balance, i.e., the absence of net stochastic flow, is manifested by the symmetry under the interchange of the indices:

$$v_{ij} = v_{ji} \quad \forall i, j. \quad (2.51)$$

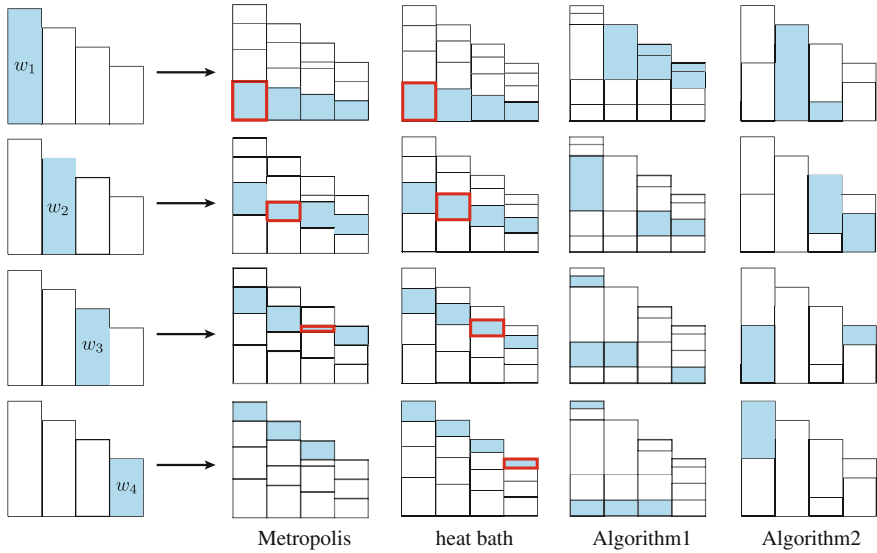
Our aim is to find a set  $\{v_{ij}\}$  that minimizes the average rejection rate while satisfying Eqs. (2.46) and (2.47). The procedure for the task can be understood visually as *weight allocation*, where we move (or allocate) some amount of weight ( $v_{ij}$ ) from state  $i$  to  $j$  keeping the entire shape of the weight boxes intact. For catching on this allocation picture, let us think at first the case with  $n = 2$  as in the single spin update of the Ising model. Figure 2.3 shows the allocation when the Metropolis and heat bath algorithms are applied, where the average rejection rate ( $\propto v_{11} + v_{22}$ ) clearly remains finite. Indeed, for  $n = 2$  the Metropolis algorithm gives the best solution, i.e., the minimum average rejection rate even within the total balance (see Eq. (2.52)).

For  $n \geq 3$ , these two methods fail to minimize the rejection rate as we will mention. Besides, a generic method that accomplishes the minimization has not been known before. We will show that we can easily make it possible by this geometric picture. Although many optimal solutions are found actually, here we will introduce two specific algorithms. One makes a reversible kernel, and the other makes an *irreversible* kernel without the detailed balance.

### 2.3.1 Reversible Kernel

For describing our algorithm, let us introduce an operation named Swap:

```
Swap( i, j, w ) {
    vii ← vii - w
    vij ← vij + w
    vji ← vji + w
    vjj ← vjj - w
}
```



**Fig. 2.4** Example of weight allocation by the Metropolis, the heat bath, and the proposed two algorithms for  $n = 4$ . Algorithm 1 constructs a reversible kernel, and Algorithm 2 does an irreversible kernel. Both proposed algorithms minimize the average rejection rate in general, and they are rejection free in this case while the conventional methods remain finite rejection rates as indicated by the thick frames

We note that if  $\{v_{ij}\}$  satisfy the conditions (2.46), (2.47) and (2.51), this Swap operation does not break them. A certain algorithm for the construction of reversible kernel that minimizes the average rejection rate is described in Algorithm 1 [56]. This algorithm starts with the diagonal matrix  $[v_{ij}]$  and uses only Swap operation for construction. Therefore the three conditions (2.46), (2.47) and (2.51) are automatically satisfied in the whole procedure. This algorithm can be depicted visually as Algorithm 1 in Fig. 2.4. As a result, the self-allocated weight that produces rejection is expressed as

$$v_{ii} = \begin{cases} \max(0, w_1 - \sum_{i=2}^n w_i) & i = 1 \\ 0 & i \geq 2. \end{cases} \quad (2.52)$$

That is, a rejection-free solution is obtained if the condition

$$w_1 \leq \frac{S_n}{2} \equiv \frac{1}{2} \sum_{k=1}^n w_k \quad (2.53)$$



is satisfied. When it is not satisfied, one has to necessarily assign the maximum weight to itself since it is larger than the sum of the rest. Thus, the present solution is optimal in the sense that it minimizes the average rejection rate.

### 2.3.2 Irreversible Kernel

Next, we show another algorithm that constructs an irreversible kernel [54]. Since the invention by Metropolis and the coworkers, the reversibility, the detailed balance, has been imposed to the Markov chain in most MCMC simulations. The reversibility is sufficient for the invariance of target distribution (sampling from the target distribution asymptotically). It is, however, *not necessary*. If it is possible to find a solution beyond the sufficient condition, further optimization can be achieved. In the meanwhile, it has long been considered difficult to satisfy the total balance without imposing the detailed balance. Thus attempts to optimizing transition probabilities have concentrated within the reversibility as we mentioned in the previous section [21, 34].

In fact, the reversibility is often broken secretly, even though the detailed balance is used apparently to define the transition probabilities. The single spin update in a classical system is such an example. The random update, where a spin to be flipped is chosen uniformly randomly among all spins, satisfies the detailed balance strictly.

---

#### Algorithm 1 Construction of Reversible Kernel with Minimized Rejection

---

Sort  $n$  candidate configurations as  $w_1 \geq w_2 \geq w_3 \geq \dots \geq w_n$  ( $n \geq 3$ ).

$v_{ij} \leftarrow w_i \delta_{ij}$

$w_{\text{diff}} \leftarrow w_1 - w_2$

$S_3 \leftarrow \sum_{i=3}^n w_i$

**if**  $w_{\text{diff}} \geq S_3$  **then**

**for**  $i = 2, \dots, n$  **do**

        Swap(  $1, i, w_i$  )

        //  $v_{ii}$  becomes 0

**end for**

**else**

**for**  $i = 3, \dots, n$  **do**

$v \leftarrow w_{\text{diff}} * w_i / S_3$

        Swap(  $1, i, v$  )

        //  $v_{11} = v_{22} \geq v_{33} \geq \dots \geq v_{nn}$

**end for**

**for**  $j = n, \dots, 2$  **do**

$v' \leftarrow v_{jj} / (j - 1)$

**for**  $k = j - 1, \dots, 1$  **do**

            Swap(  $j, k, v'$  )

        //  $v_{11} = v_{22} \geq \dots \geq v_{j-1, j-1}$  and  $v_{jj} = 0$

**end for**

**end for**

**end if**

---

On the other hand, the reversibility is broken in the sequential update, where spins are swept in a fixed order. The detailed balance is satisfied only locally, that is, only in each spin flip, and the total balance is eventually fulfilled in one sweep [39].

In this subsection, we present another geometric algorithm that fulfills the total balance but breaks the detailed balance even locally. Furthermore, breaking the detailed balance introduces a net stochastic flow in the configuration space. It will boost up the convergence further by suppressing random walk behavior [1, 13, 15].<sup>5</sup>

Our approach is the first method that can generally satisfy the total balance without the detailed balance. Although a solution is not unique obviously, we propose the following procedure as a concrete algorithm to find a solution for general  $n$ .

- (i) Choose a configuration with maximum weight. If two or more configurations have the same maximum weight, choose one of them. In the following, we assume  $w_1$  is the maximum without loss of generality. The order of the remaining weights does not matter.
- (ii) Allocate the maximum weight  $w_1$  to the next box ( $i = 2$ ). If the weight still remains after saturating the box, reallocate the remainder to the next ( $i = 3$ ). Continue until the weight is all allocated.
- (iii) Allocate the weight of the first landfilled box ( $w_2$ ) to the last partially filled box in step (ii). Continue the allocation likewise.
- (iv) Repeat step (iii) for  $w_3, w_4, \dots, w_n$ . Once all the boxes with  $i \geq 2$  are saturated, landfill the first box ( $i = 1$ ) afterward.

The whole algorithm is described in Algorithm 2. In the algorithm, if two or more configurations have the same maximum weight, choose one of them at first. Any order of configurations accomplishes the same minimized rejection rate. In the above procedure, all the boxes are filled without any space as well as the reversible case, as in Fig. 2.4; it satisfies the two conditions (2.46) and (2.47). However, the reversibility (2.51) is broken. (For example,  $v_{12} > 0$ , but  $v_{21} = 0$  as depicted in the figure.) Since  $w_1$  is the maximum, it is also clear that the second and subsequent boxes must be already saturated when the allocation of its own weight is initiated.

By this procedure,  $\{v_{i \rightarrow j}\}$  are determined as

$$v_{i \rightarrow j} = \max(0, \min(\Delta_{ij}, w_i + w_j - \Delta_{ij}, w_i, w_j)), \quad (2.54)$$

where

$$\Delta_{ij} := S_i - S_{j-1} + w_1 \quad 1 \leq i, j \leq n \quad (2.55)$$

$$S_i := \sum_{k=1}^i w_k \quad 1 \leq i \leq n \quad (2.56)$$

$$S_0 := S_n. \quad (2.57)$$

---

<sup>5</sup> The overrelaxation method [1] and the hybrid Monte Carlo [15] both satisfy the detailed balance, mistakenly believed to break it.

**Algorithm 2** Construction of Irreversible Kernel with Minimized Rejection

---

Choose a configuration that has the maximum weight and number it 1.  
Sort other configurations in an arbitrary order.

```

 $i \leftarrow 1$ 
 $j \leftarrow 2$ 
while  $i \leq n$  do
   $w_r \leftarrow w_i$ 
  while  $w_r > 0$  do
    if  $w_r \geq w_j$  then
       $v_{ij} \leftarrow w_j$ 
       $w_r \leftarrow w_r - w_j$ 
      if  $j = n$  then
         $j \leftarrow 1$ 
      else
         $j \leftarrow j + 1$ 
      end if
    else
       $v_{ij} \leftarrow w_r$ 
       $w_j \leftarrow w_j - w_r$ 
       $w_r \leftarrow 0$ 
    end if
  end while
   $i \leftarrow i + 1$ 
end while

```

---

It is easy to understand from Eq. (2.54) that the rejection flow is expressed as the same Eq. (2.52) with Algorithm 1. In contrast to the reversible case, a net stochastic flow is introduced as the result of breaking the detailed balance, and it is expected to further boost up the sampling efficiency [13].

We close the introduction of our algorithm with a note about the ergodicity. Our new algorithms both minimize the average rejection rate; they are methods of choice. One hand, it is difficult to prove that the present irreversible kernel satisfies the ergodicity in the sequential update since many of the transition probabilities become zero. On the other hand, our reversible kernel can be ergodic almost as likely as the heat bath algorithm because the transition probabilities to other states except the current one are all positive. Since it is quite easy to prove the ergodicity of Markov chain by the heat bath algorithm, this reversible version must be ergodic in almost every cases. As we mentioned, the net stochastic flow in state space will boost the convergence; the irreversible version will be better than the reversible one. Thus, it is a good strategy to first check the ergodicity of the irreversible chain by comparing to the reversible chain and then use the irreversible one basically. Another way to ensure the ergodicity in the irreversible version is to randomly choose one of the probability sets obtained by different allocation order. Although we have not observed any glimpse of ergodicity breaking in the following simulations, such a prescription will assure users of the ergodicity.

## 2.4 Benchmarks in Potts Model

In order to assess the effectiveness of the present algorithms, we investigate the convergence and the autocorrelations in the ferromagnetic  $q$ -state Potts models on the square lattice [63]; the local (spin) state is expressed as  $\sigma_k$  that takes an integer ( $1 \leq \sigma_k \leq q$ ). These systems exhibit a continuous ( $q \leq 4$ ) or first-order ( $q > 4$ ) phase transition at  $T_4 = 1/\ln(1 + \sqrt{q})$ . We calculate the square of order parameter (structure factor) for  $q = 4$  and 8 by several algorithms. The order parameter [64], which is one of the most important quantity in statistical mechanical models, is defined as

$$O = \frac{q-1}{q} \|m\|_2^2, \quad (2.58)$$

where

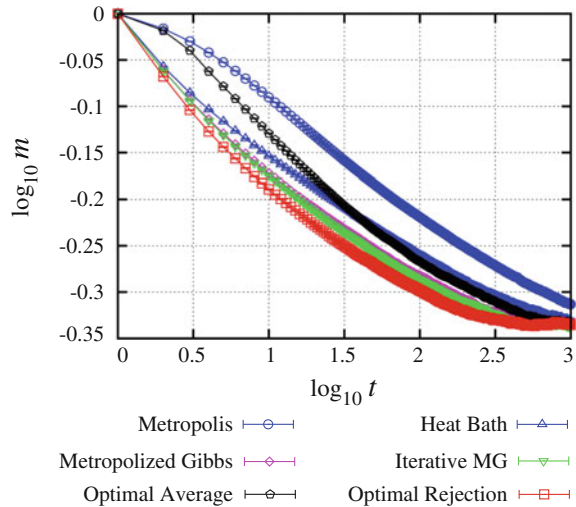
$$m = (m_1, m_2, \dots, m_q) \quad (2.59)$$

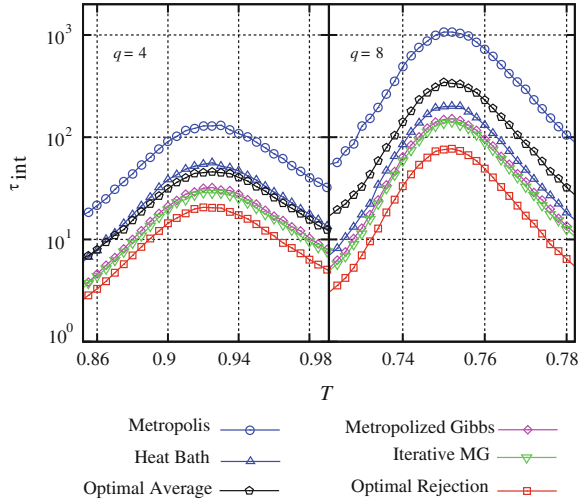
$$m_i = \frac{1}{L^d(q-1)} \left\langle \sum_k (q \delta_{i,\sigma_k} - 1) \right\rangle \quad 1 \leq i \leq q,$$

in the Potts model.

The order parameter equilibration (convergence) is shown in Fig. 2.5, where the simulation starts with the fully ordered state and the local variables are sequentially updated by the several algorithms. The square lattice with  $L = 32$  and the critical temperature  $T = 0.9102392266$  are used. The Metropolis algorithm (2.11) [40], the heat bath algorithm (2.12) [5], the Metropolized Gibbs sampler (2.24) [34, 35], the itera-

**Fig. 2.5** Convergence of the order parameter (square root of the structure factor) in the ferromagnetic 4-state Potts model on the *square* lattice with  $L = 32$  at the critical temperature. The *horizontal axis* is the Monte Carlo step. The simulation starts with the ordered (all “up”) state. Our rejection-minimized samplers achieve the fastest convergence. There is no difference, in this scale, of two data by the reversible and irreversible kernel (That is why only one data is plotted)





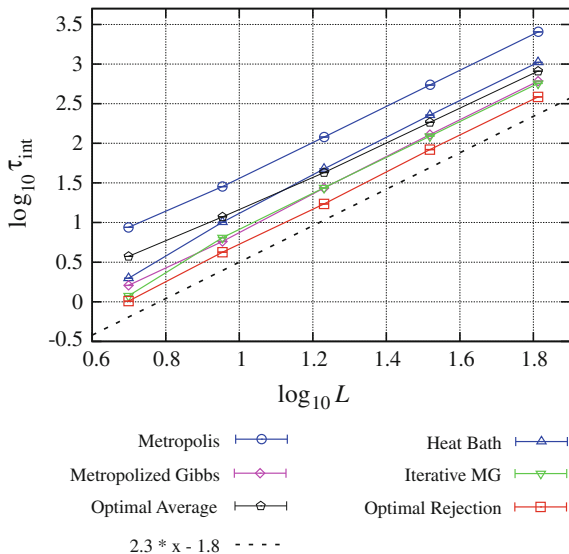
**Fig. 2.6** Autocorrelation time of the square of order parameter near the transition temperature ( $T \simeq 0.910$  and  $0.745$ , respectively) in the 4-state (*left*) and 8-state (*right*) Potts models by several methods. The system size is  $16 \times 16$ . In the both cases, the present methods realize the shortest autocorrelation time. The error bars are the same order with the point sizes

tive Metropolized Gibbs sampler (2.33) [21], the optimal average sampler (2.44) [30], our rejection-minimized algorithms, Algorithm 1 [56] and Algorithm 2 (2.54) [54], are compared. Our samplers accomplish the fastest convergence of the quantity (square root of the structure factor). The two reversible and irreversible kernel perform equally. This acceleration implies that locally rejection-minimized algorithms reduce the second largest eigenvalue of the whole transition matrix in absolute value, which real part is presumably positive expected from the overdamping form.

In Fig. 2.6, on the other hand, it is clearly seen that our algorithms significantly reduce the autocorrelation time for  $q = 4, 8$  in comparison with the conventional methods. The autocorrelation time  $\tau_{\text{int}}$  is estimated through the relation:  $\sigma^2 \simeq (1 + 2\tau_{\text{int}})\sigma_0^2$ , where  $\sigma_0^2$  and  $\sigma^2$  are the variances of the estimator without considering autocorrelation and with calculating correlation from the binned data using a bin size much larger than the  $\tau_{\text{int}}$  [32]. In the 4 (8)-state Potts model, the autocorrelation time becomes nearly 6.4 (14) times as short as that by the Metropolis algorithm, 2.7 (2.6) times as short as the heat bath algorithm, and even 1.4 (1.8) times as short as the iterative Metropolized Gibbs sampler [21, 49], which was considered as one of the best solutions before our approach. The autocorrelations of our two algorithms are much the same both for  $q = 4, 8$ .

Next, we investigated the dynamical exponent of the autocorrelation time at the critical temperature. The system-size dependence of the autocorrelation for  $q = 4$  is shown in Fig. 2.7. All the update methods suffer from the critical slowing down with the exponent  $z \sim 2.3$ . Unfortunately, the locally optimized method does not reduce

**Fig. 2.7** System-size dependence of the autocorrelation time for  $q = 4$  at the critical temperature. The dynamics of all update methods experiences the critical slowing down with the dynamical exponent  $z \sim 2.3$ . Although the local optimization does not improve the exponent, the rejection-minimized samplers get always factor over 6, compared with the Metropolis algorithm



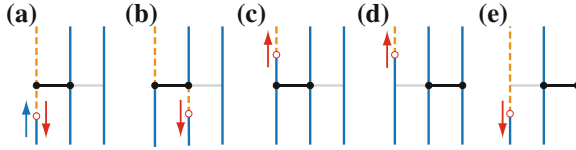
the exponent. The factor, however, over 6 is always gained against the Metropolis algorithm.

As we have seen, our rejection-minimized samplers boost the convergence and reduce the autocorrelation time of the relevant estimator (order parameter). The present methods will be effective in not only the Potts model but also the many kinds of systems. A further eigenvalue analysis is important. Particularly, modifications of the above theorems for irreversible kernels are greatly of interest.

## 2.5 Bounce-Free Worm (Directed-Loop) Algorithm

Next, we will move onto the quantum Monte Carlo (QMC) method. Although we will explain the detailed formalism of the QMC method with new modifications in the next chapter, here let us see an example of the kernel optimization in the QMC method.

The worm algorithm for quantum spin and lattice boson models is formulated based on either the Euclidean path integral or the high-temperature series [50, 58]. One Monte Carlo sweep of the worm algorithm consists of the diagonal update, where operators are inserted or removed without changing the shape of worldlines, and the off-diagonal update, where the worldlines (and the type of operators) are updated with keeping the position of operators unchanged. In the latter process, a pair of creation and annihilation operators, which is called a worm, is inserted on a worldline (pair creation), and one of them (called the head) is moved stochastically until the head and the tail destroy each other (pair annihilation). As a thorny problem,



**Fig. 2.8** Extension of the worm-going pathway in the  $S = 1/2$  model. Here, *dashed (solid) vertical lines* denote spin up (down). The head of worm (*open circle*) moves on the worldline (a), and scatters at the operator (*horizontal thick line*). As candidate configurations, we introduce operator-flip updates (d), (e) in addition to the conventional ones (a)–(c). Note that in (e) the position of the operator is shifted simultaneously in contrast to the simple bounce process (a)

a bounce process, where the head just backtracks and cancels the last update, has been generally inevitable within the detailed balance. Here, as an example, we consider the  $S = 1/2$  antiferromagnetic  $XXZ$  model:

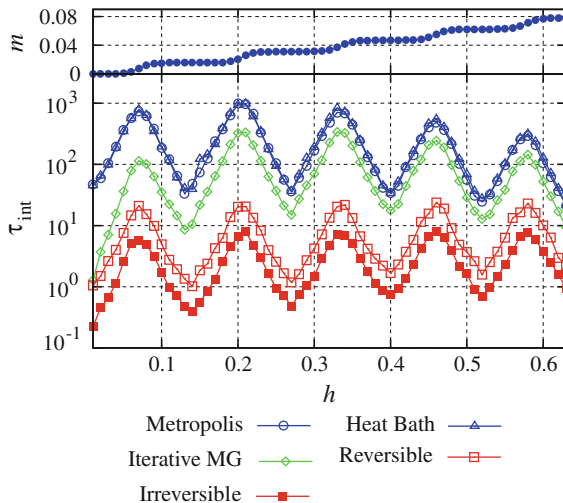
$$\mathcal{H} = \sum_{\langle i, j \rangle} \left( S_i^x S_j^x + S_i^y S_j^y + \Delta S_i^z S_j^z - C \right) - h \sum_i S_i^z, \quad (2.60)$$

where we introduce a parameter  $C$  controlling the ratio between the diagonal and off-diagonal weights. In the head scattering process at an operator, only three among four exits have a nonzero weight due to the conservation of the total  $S^z$  (Fig. 2.8a–c). At the Heisenberg point ( $\Delta = 1$ ), there remain finite bounce probabilities except at  $h = 0$  within the detailed balance [2, 49, 58]. Unfortunately, the situation does not improve much even in the total balance because the number of candidates is too small. However, the condition (2.53) provides us a clear prospect; by increasing the number of candidates, a bounce-free algorithm will be realized. According to this strategy, we introduce an *operator-flip* update, where sites on which an operator acts are shifted simultaneously (Fig. 2.8). By the operator flip together with the constant  $C$  chosen as

$$C = \max \left( \frac{1}{4}(2\Delta + 3h - 1), \frac{1}{8}(\Delta + 3h + 1) \right), \quad (2.61)$$

we can indeed eliminate the bounce process.

The autocorrelation data of the magnetization in the Heisenberg chain ( $\Delta = 1$ ) are shown in Fig. 2.9. Amazingly, the bounce-free worm algorithm with the operator flip is faster by about 2 orders of magnitude than the Metropolis and the heat bath algorithms. In this model, our irreversible kernel is nearly 3 times as fast as our reversible kernel; the net stochastic flow effectively works in this case. Also in high- $S$  spin systems, the bounce-free worms can be constructed by having the head hold the matrix element of the ladder operator or by representing the partition function of general  $S$  system as that of decomposed  $S = 1/2$  spin system [60]. Our idea of breaking the detailed balance and operator-flip updates are also applied to general bosonic models and the efficiency is improved because bosonic worms get



**Fig. 2.9** Magnetic field dependence of magnetization (*upper*) and autocorrelation time (*lower*) of the  $S = 1/2$  antiferromagnetic Heisenberg chain ( $L = 64$ ,  $T = 1/2L$ ). The maximum autocorrelation time is  $1.0 \times 10^3$  by the worm update with Metropolis (*circles*),  $9.8 \times 10^2$  by the worm with heat bath (*triangles*), and  $3.3 \times 10^2$  by the iterative Metropolized Gibbs sampler (*diamonds*). By the bounce-free worm with the operator flip,  $\tau_{\text{int}}$  is further reduced down to 23.8 (*open squares*) for the reversible version and 8.1 (*solid squares*) for the irreversible version

bounce-minimized with more candidates. This application is very important also for our calculation of spin-Peierls models in the following chapters.

## 2.6 General Construction of Irreversible Kernel

We will mention also new methods constructing improved kernel for general state space that is not finite. Let us consider updating a continuous variable here. When the inversion method is applicable, the variable is updated by the heat bath algorithm (Gibbs sampler) usually. That is, the next state is determined from a uniformly random variable  $r \in [0, 1]$  on conditional cumulative distribution. The calculation of the inverse function is needed in this procedure. On the other hand, when the inversion method cannot be applied, a candidate state is chosen from a proposal distribution and accepted/rejected by the Metropolis algorithm usually. In this situation, where we are in most cases, the inevitable rejection will be a bottleneck for sampling. For continuous variables, it is not possible to apply directly the previous allocation algorithm because the measure of each state is zero. We can, nevertheless, improve the efficiency for both cases by extending the idea of breaking the detailed balance.



### 2.6.1 Beyond Heat Bath Algorithm

First, we introduce an improved sampling that is an alternative method to the Gibbs sampler. Let us review our allocation algorithm for the irreversible kernel for finite-size problems. We start at the configuration with the maximum weight and allocate the weight to the next. This procedure can be also represented by shifting each position in the maximum weight on the cumulative distribution. We compare the shifted distribution to the original (non-shifted) one as Fig. 2.10 and assign the next position (state). It is possible to set the amount of shift any value. If there is a self-allocation as a result of the shift, the amount is nothing but the rejection rate. It is obvious that the amount of shift that can avoid the self-allocation is not unique; the rejection-free kernel can be achieved as long as the amount of shift is such that the maximum weight has no overlap with its original position as in the figure. For continuous variables, we can set the start point of allocation (the amount of shift) at our disposal. Let us consider the bivariate Gaussian distribution as a simple example:

$$P(x_1, x_2) \propto e^{-\frac{(x_1-x_2)^2}{2\sigma_1^2} - \frac{(x_1+x_2)^2}{2\sigma_2^2}}. \quad (2.62)$$

Given  $x_2$ ,  $x_1$  is updated by using the conditional cumulative distribution

$$F(x_1|x_2) = \int_{-\infty}^{x_1} P(x, x_2) dx. \quad (2.63)$$

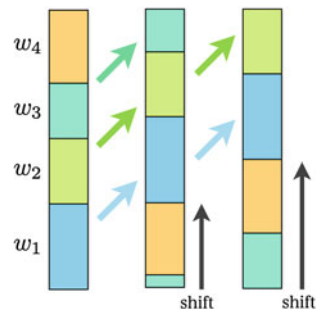
The heat bath algorithm determines the next state as

$$x'_1 = F^{-1}(r), \quad (2.64)$$

where  $r \in [0, 1]$  is an uniformly (pseudo) random variable. This process satisfies the detailed balance.

The overrelaxation method has been known as one of the best ways to update the Gaussian variables. The name “overrelaxation” comes from an idea to make the

**Fig. 2.10** Picture of the cumulative distribution shifts. The algorithm for the irreversible kernel is corresponding to the shift in the maximum weight as shown in the midst. If we shift in a larger value, we could get a rejection free algorithm as in the *right* bar



Markov chain to have negative correlation. We have already seen that the negative eigenvalues, which results in the negative correlation, make the asymptotic variance smaller and the sampling more efficient. This method was first proposed by Adler [1] for updating variables with a Gaussian conditional distribution. Whitmer [62] discussed slightly an extended application range of the method. In the overrelaxation method, for generation of a variable from a conditional distribution

$$P(z_i | \cdot) \sim N(\mu_i, \sigma_i^2), \quad (2.65)$$

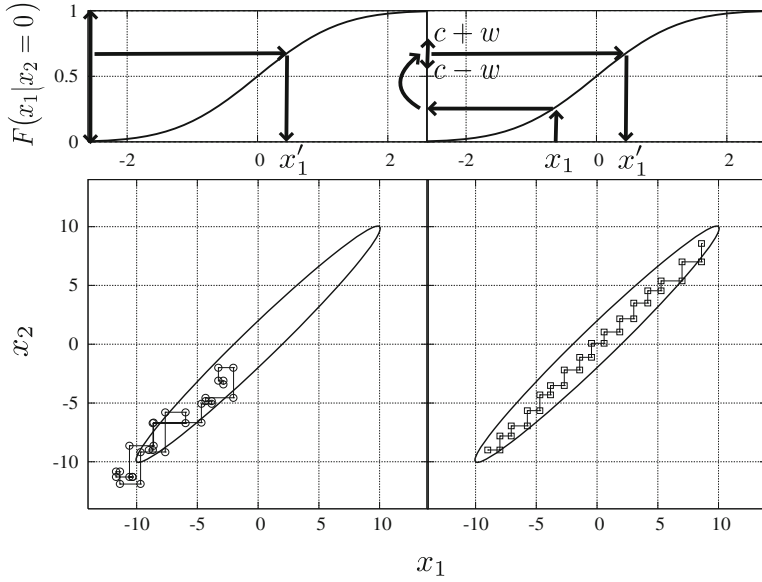
the next state is chosen as

$$z'_i = \mu_i + \alpha(z_i - \mu_i) + \sigma_i \sqrt{1 - \alpha^2} v, \quad (2.66)$$

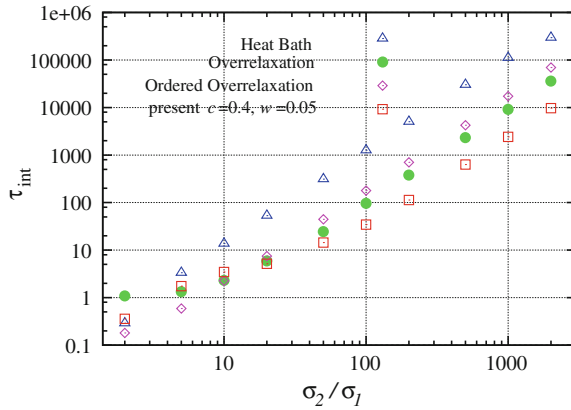
where  $v$  is a random variable generated from  $N(0, 1)$  and  $\alpha$  is a parameter with  $-1 < \alpha < 1$ . Later, Brown et al. [9] and Creutz [12] applied the idea of generating negative correlation to the SU(2) and SU(3) gauge theory, mapping the state space into an Euclidean space where the distribution can be described by the Gaussian distribution. Mira et al. rediscovered the overrelaxation method [44] in statistics. Moreover, this overrelaxation technique can be used for not only the Gaussian but some simple distributions, e.g., the conditional distribution of the classical XY model or the Heisenberg model. The main idea is to try to go to the opposite side over the center of the conditional distribution under the detailed balance. Thus, we need to know where the center is. This condition hinders us from applying the method to general cases. As an approach for general models, Neal [46] proposed the ordered overrelaxation where some candidates are generated from the conditional distribution and ordered, then we go to approximately opposite side through the order of states. This method, however, still needs to generate configurations from the conditional distribution. Thereby, the application range is not so extended practically. Now, as another update method [55], let us choose the next state as

$$x'_1 = F^{-1}(\{F(x_1|x_2) + c + wu\}), \quad (2.67)$$

where  $x_1$  is the current state,  $c$  and  $w$  is a positive real parameter with  $c \geq w$ , and  $u$  is a uniformly random variable in  $[-1, 1]$ , respectively. The symbol  $\{a\}$  takes the fractional portion of a real number  $a$ . If we use  $c = w = 1/2$ , this process is nothing but the heat bath algorithm. On the other hand, when  $c \neq 0, 1/2$ , it does not satisfy the detailed balance and there is a net stochastic flow. This flow can push the configuration globally as in Fig. 2.11. As the result, the autocorrelation time of  $(x_1 + x_2)^2$  is significantly reduced as shown in Fig. 2.12. In this figure, the overrelaxation methods and the ordered overrelaxation method are also tested. Compared with the conventional methods, the present update method can be better on the whole parameter region; at any ratio  $\sigma_2/\sigma_1$ , there is a better parameter set of present scheme than the best parameter of the overrelaxation or the ordered overrelaxation methods. Figure 2.12 shows the results by using the heat bath algorithm, the overrelaxation method with  $\alpha = -0.86$ , the ordered overrelaxation with the number of candidates



**Fig. 2.11** Trajectories of configurations updated by the heat bath algorithm (*left*) and by the present irreversible algorithm with  $c = 0.4$  and  $w = 0.1$  (*right*) in the bivariate Gaussian distribution with  $\sigma_1 = 1$  and  $\sigma_2 = 10$ . The ellipsoidal line is the three-sigma line of the Gaussian distribution. The *upper* figures show the update procedures of each algorithm



**Fig. 2.12** Autocorrelation times of  $(x_1 + x_2)^2$  in the bivariate Gaussian distribution by using the heat bath algorithm (*triangles*), the overrelaxation (*circles*) with  $\alpha = -0.86$ , the ordered overrelaxation (*diamonds*) with the number of candidates 10, and the present method with  $c = 0.4$  and  $w = 0.05$  (*squares*). The horizontal axis  $\sigma_2 / \sigma_1$  is corresponding to the sampling difficulty

10, and the present algorithm with  $c = 0.4$  and  $w = 0.05$ . The present method produces the shortest correlation time over  $\sigma_2/\sigma_1 \geq 50$  and achieves about 50 times as short correlation time as the heat bath algorithm in the region. As we mentioned, we can surely find a better parameter set of the present algorithm than the best parameter of the conventional overrelaxation methods in the whole region.

### 2.6.2 Beyond Metropolis Algorithm

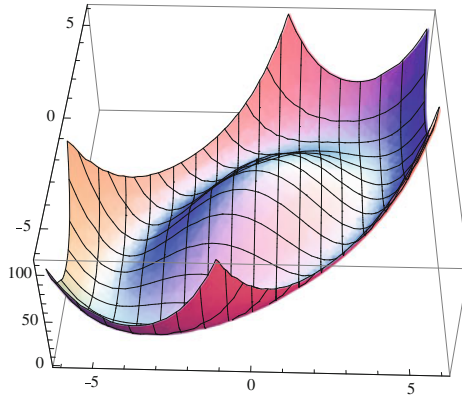
We will explain here it is possible to significantly reduce the rejection rate for general cases. When the direct inversion method as in the previous subsection cannot be applied, we resort to the Metropolis algorithm usually, where a candidate is generated and the update trial is accept/reject according to the weight ratio. The Metropolis algorithm is very simple and easy to implement. It has been a canonical algorithm for the MCMC method since the invention in 1953 [40]. However, the inevitable rejection often obstructs the efficient sampling. Obviously it is important to reduce the rejection rate. When the number of candidates is two, the Metropolis algorithm achieves the minimized rejection rate that is easily proved by the geometric picture we introduced in Sect. 2.3. On the other hand, the Barker algorithm, which is nothing but the heat bath algorithm for two candidates, has more rejection. Therefore, in order to reduce the rejection rate, we have to prepare more candidates than two. As an alternative to the simple Metropolis algorithm, some methods have been proposed so far. An example is the multipoint Metropolis methods, where after generating some candidate states the next configuration is stochastically chosen with the detailed balance kept. See references Frenkel et al. [19], Liu et al. [37], Qin et al. [51], and Liu [36]. Another example is the window algorithm proposed by Neal [45]. This is a variant of the multipoint Metropolis methods for the hybrid Monte Carlo method [15].

We can apply our rejection-minimized algorithms after creating some candidate states. Let us consider sampling from the wine-bottle (Mexican-hat) potential:

$$P(x_1, x_2) \propto \exp\left(-\left(\frac{(x_1 - x_2)^2}{2\sigma_1^2} + \frac{(x_1 + x_2)^2}{2\sigma_2^2}\right)\left(\frac{(x_1 - x_2)^2}{2\sigma_1^2} + \frac{(x_1 + x_2)^2}{2\sigma_2^2} - h\right) + \frac{h^2}{4}\right). \quad (2.68)$$

The example of the potential is shown in Fig. 2.13. We propose a candidate configuration by the isotropic bivariate Gaussian distribution  $\propto \exp(-(\Delta x_1)^2 - (\Delta x_2)^2)$ . Here, we try to make some proposals. If we propose candidates from the current position and naively make a transition matrix (probability) taking into account only the weights, the total balance is broken. It is because we have to consider also the different proposal probability. Avoiding this difficult task, we use the following strategy:

1. A configuration is chosen as a hub from the current configuration by a proposal distribution.



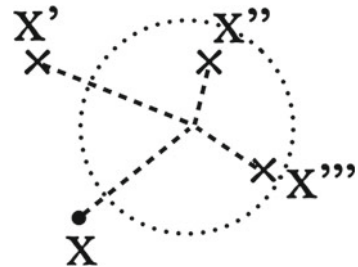
**Fig. 2.13** Wine-bottle (Mexican-hat) potential for  $\sigma_1 = 1$ ,  $\sigma_2 = 2$ ,  $h = 16$

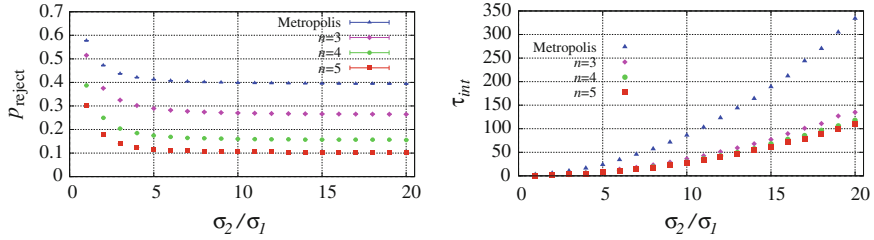
2. Candidates are generated by using the same proposal distribution with process 1 from the hub.
3. The next state is chosen among the candidates and the current configuration by using the transition probabilities taking into account only the weights of the states.

This procedure is depicted as in Fig. 2.14. In the process 3, we can make the rejection rate minimized by applying our algorithms we introduced. Figure 2.15 shows that the rejection rate is indeed reduced by using this multi-proposal algorithm and the irreversible kernel. The correlation time of  $(x_1 + x_2)^2$  also gets shorter as the number of candidates is increased. Although this example is a simple case, this procedure is applicable to any MCMC sampling. The further investigation of the validity is a future problem. These days, the role of the net stochastic flow (irreversible drift) has caught much attention [18, 29, 31]. The performance of our algorithm and the efficient flow structure need to be investigated further in the future.

Finally, we close this chapter with noting of the CPU time cost. As we mentioned in the introduction of the geometric allocation algorithms, there is no extra CPU time cost when we can prepare all transition probabilities before sampling. As for the continuous variables, it is not the case; we have to calculate probabilities at each

**Fig. 2.14** Multi-proposal strategy for  $n = 4$ . At first, a center position is chosen from the current position  $x$ . Then, candidates  $x'$ ,  $x''$ ,  $x'''$  are generated from the center. The dot line shows the 1 sigma line of the Gaussian distribution





**Fig. 2.15** Rejection rates (*left*) and the correlation times of  $(x_1 + x_2)^2$  (*right*) by the simple Metropolis algorithm and the rejection-minimized method for  $n = 3$  and  $n = 4$ . The rejection rate is definitely reduced as the number of candidates is increased. Accompanying the rejection rate, the correlation time gets shorter

transition. Although the rejection rate is certainly reduced by using the multi proposal strategy, meanwhile, the CPU time cost is increased. Thus, the question arises that the needed calculation time is truly decreased. Here, we mention a combination with parallelization method. The process of preparing candidates can be done independently. Thus, we can control the CPU time by applying the parallelization for the multi proposal. This multi-core update scheme is effective especially for the cases where the long thermalization (convergence) time is needed by a simple single core update. The effectiveness of this combination method needs to be researched.

## References

1. Adler, S. L. (1981). Over-relaxation method for the Monte Carlo evaluation of the partition function for multi-quadratic actions. *Physical Review D*, 23, 2901.
2. Alet, F., Wessel, S., & Troyer, M. (2005). Generalized directed loop method for quantum Monte Carlo simulations. *Physical Review E*, 71, 036706. doi:[10.1103/PhysRevE.71.036706](https://doi.org/10.1103/PhysRevE.71.036706).
3. Andrieu, C., & Toms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18, 343–373.
4. Baldi, P., Frigessi, A., & Piccioni, M. (1993). Importance sampling for Gibbs random fields. *Annals of Applied Probability*, 3, 914–933.
5. Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18, 119.
6. Bellman, R. E. (1957). *Dynamic programming*. Princeton: Princeton University Press.
7. Berg, B. A., & Neuhaus, T. (1992). Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters*, 68, 9.
8. Billera Louis, J., & Diaconis, P. (2001). A geometric interpretation of the Metropolis-Hastings algorithm. *Statistical Science*, 16, 335–339.
9. Brown, F. R., & Woch, T. J. (1987). Overrelaxed heat-bath and Metropolis algorithms for accelerating pure gauge Monte Carlo calculations. *Physical Review Letters*, 58, 2394–2395.
10. Chiang, T. S., & Chow, Y. (1993). Asymptotic behavior of eigenvalues and random updating schemes. *Applied Mathematics and Optimization*, 28, 259–275.
11. Creutz, M. (1980). Monte Carlo study of quantized SU(2) gauge theory. *Physical Review D*, 21, 2308.

12. Creutz, M. (1987). Overrelaxation and Monte Carlo simulation. *Physical Review D*, 36, 515–519.
13. Diaconis, P., Holmes, S., & Neal, R. M. (2000). Analysis of a nonreversible Markov chain sampler. *Annals of Applied Probability*, 10, 726.
14. Dick, J., & Pillichshammer, F. (2010). *Digital nets and sequences: Discrepancy theory and Quasi Monte Carlo integration*. Cambridge: Cambridge University Press.
15. Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195, 216.
16. Evertz, H. G., Lana, G., & Marcu, M. (1993). Cluster algorithm for vertex models. *Physical Review Letters*, 70, 875.
17. Faure, H. (1982). Discrepance de suites associées à un système de numération (en dimension  $s$ ). *Acta Arithmetica*, 41, 337–351.
18. Franke, B., Hwang, C. R., Pai, H. M., & Sheu, S. J. (2010). The behavior of the spectral gap under growing drift. *American Mathematical Society*, 362, 1325–1350.
19. Frenkel, D., & Smit, B. (2001). *Understanding molecular simulation: From algorithms to applications* (2nd ed.). San Diego: Academic Press.
20. Frigessi, A., Gâsemayr, J., & Rue, H. (2000). Antithetic coupling of two Gibbs sampler chains. *Annals of Statistics*, 28, 1128–1149.
21. Frigessi, A., Hwang, C. R., & Younes, L. (1992). Optimal spectral structure of reversible stochastic matrices, Monte Carlo methods and the simulation of Markov random fields. *Annals of Applied Probability*, 2, 610–628.
22. Frigessi, A., Martinelli, F., & Stander, J. (1997). Computational complexity of Markov chain Monte Carlo methods for finite Markov random fields. *Biometrika*, 84, 1–18.
23. Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721.
24. Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. London/New York: Chapman & Hall.
25. Green, P. J., & Mira, A. (2001). Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika*, 88, 1035–1053.
26. Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97.
27. Hlawka, E. (1961). Funktionen von beschränkter variation in der theorie der gleichverteilung. *Annali di Matematica Pura ed Applicata*, 54, 325–333.
28. Hukushima, K., & Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, 65, 1604.
29. Hwang, C. R. (2006). Nonreversible perturbations accelerate convergence. *RIMS Kokyuroku*, 1462, 26–34.
30. Hwang, C. R., Chen, T. L., Chen, W. K., & Pai, H. M. (2012). On the optimal transition matrix for MCMC sampling. *SIAM Journal on Control and Optimization*, 50, 2743–2762.
31. Hwang, C. R., Hwang-Ma, S. Y., & Sheu, S. J. (2005). Accelerating diffusions. *Annals of Applied Probability*, 15, 1433–1444.
32. Landau, D. P., & Binder, K. (2005). *A guide to Monte Carlo simulations in statistical physics* (2nd ed.). Cambridge: Cambridge University Press.
33. Lemieux, C. (2009). *Monte Carlo and Quasi-Monte Carlo sampling*. New York: Springer.
34. Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6, 113.
35. Liu, J. S. (1996). Peskun's theorem and a modified discrete-state Gibbs sampler. *Biometrika*, 83, 681–682.
36. Liu, J. S. (2001). *Monte Carlo strategies in scientific computing* (1st ed.). Springer: New York.
37. Liu, J. S., Liang, F., & Wong Wing, H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95, 121.
38. Loison, D., Qin, C. L., Schotte, K. D., & Jin, X. F. (2004). Canonical local algorithms for spin systems: Heat bath and Hasting's methods. *European Physical Journal B*, 41, 395–412.

39. Manousiouthakis, V. I., & Deem, M. W. (1999). Strict detailed balance is unnecessary in Monte Carlo simulation. *Journal of Chemical Physics*, 110, 2753.
40. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21, 1087.
41. Meyn, S. P., & Tweedie, R. L. (1993). *Markov chains and stochastic stability*. New York: Springer.
42. Mira, A. (2001). Ordering and improving the performance of Monte Carlo Markov chains. *Statistical Science*, 16, 340–350.
43. Mira, A. (2006). Stationarity preserving and efficiency increasing probability mass transfers made possible. *Computational Statistics*, 21, 509–522.
44. Mira, A., & Sargent Daniel, J. (2003). A new strategy for speeding Markov chain Monte Carlo algorithms. *Statistical Methods and Applications*, 12, 49–60.
45. Neal, R. M. (1994). An improved acceptance procedure for the hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 111, 194–203.
46. Neal, R. M. (1999). Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 205–228). Cambridge: MIT Press.
47. Niederreiter, H. (1988). Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30, 51–70.
48. Peskun, P. H. (1973). Optimum Monte Carlo sampling using Markov chains. *Biometrika*, 60, 607.
49. Pollet, L., Rombouts, S. M. A., Van Houcke, K., & Heyde, K. (2004). Optimal Monte Carlo updating. *Physical Review E*, 70, 056705.
50. Prokof'ev, N. V., Svistunov, B. V., & Tupitsyn, I. S. (1998). Exact, complete, and universal continuous-time world-line Monte Carlo approach to the statistics of discrete quantum systems. *Soviet Physics JETP*, 87, 310.
51. Qin, Z. S., & Liu, J. S. (2001). Multipoint Metropolis method with application to hybrid Monte Carlo. *Journal of Computational Physics*, 172, 827.
52. Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods* (2nd ed.). New York: Springer.
53. Sobol, I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *Žrnal Vyčislitel'noi Matematiki i Matematicheskoi Fiziki*, 7, 784–802.
54. Suwa, H., & Todo, S. (2010). Markov chain Monte Carlo method without detailed balance. *Physical Review Letters*, 105, 120603.
55. Suwa, H., & Todo, S. (2011). *Butsuri*, 66, 370 (in Japanese).
56. Suwa, H., & Todo, S. (2012). Geometric allocation approach for transition kernel of Markov chain. In *Monte Carlo methods and applications*, (pp. 213–221). Berlin: De Gruyter.
57. Swendsen, R. H., & Wang, J. S. (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58, 86.
58. Syljuasen, O. F., & Sandvik, A. W. (2002). Quantum Monte Carlo with directed loops. *Physical Review E*, 66, 046701.
59. Tierney, L. (1994). Markov chain for exploring posterior distributions. *Annals of Statistics*, 22, 1701.
60. Todo, S., & Kato, K. (2001). Cluster algorithms for general- $S$  quantum spin systems. *Physical Review Letters*, 87, 047203.
61. Ulam, S., Richtmyer, R. D., & von Neumann, J. (1947). *Statistical methods in neutron diffusion*. Los Alamos, NM: Los Alamos Scientific Laboratory. (Report LAMS-551).
62. Whitmer, C. (1984). Over-relaxation methods for Monte Carlo simulations of quadratic and multiquadratic actions. *Physical Review D*, 29, 306–311.
63. Wu, F. Y. (1982). The Potts model. *Reviews of Modern Physics*, 54, 235.
64. Zheng, B. (1998). Monte Carlo simulations of short-time critical dynamics. *International Journal of Modern Physics B*, 12, 1419.



Geometrically Constructed Markov Chain Monte Carlo  
Study of Quantum Spin-phonon Complex Systems

Suwa, H.

2014, XII, 126 p. 61 illus., 46 illus. in color., Hardcover

ISBN: 978-4-431-54516-3