Lifted directed-worm algorithm

Hidemaro Suwa
Department of Physics, The University of Tokyo, Tokyo 113-0033, Japan

(Received 28 June 2022; accepted 7 October 2022; published 9 November 2022)

Nonreversible Markov chains can outperform reversible chains in the Markov chain Monte Carlo method. Lifting is a versatile approach to introducing net stochastic flow in state space and constructing a nonreversible Markov chain. We present here an application of the lifting technique to the directed-worm algorithm. The transition probability of the worm update is optimized using the geometric allocation approach; the worm backscattering probability is minimized, and the stochastic flow breaking the detailed balance is maximized. We demonstrate the performance improvement over the previous worm and cluster algorithms for the four-dimensional hypercubic lattice Ising model. The sampling efficiency of the present algorithm is approximately 80, 5, and 1.7 times as high as those of the standard worm algorithm, the Wolff cluster algorithm, and the previous lifted worm algorithm, respectively. We estimate the dynamic critical exponent of the hypercubic lattice Ising model to be $z \approx 0$ in the worm and the Wolff cluster updates. The lifted version of the directed-worm algorithm can be applied to a variety of quantum systems as well as classical systems.

DOI: 10.1103/PhysRevE.106.055306

I. INTRODUCTION

Many physical models, such as the Ising model, can be mapped to configurations of bond variables, also called dimers [1]. The physical bond configurations are restricted by constraints on the state space, and it is nontrivial to sample bond configurations efficiently under the constraint. The worm algorithm has an advantage in sampling configurations under constraints on state space, and it is one of the most efficient algorithms for various classical and quantum systems [1–3]. Perhaps surprisingly, the worm algorithm significantly alleviates critical slowing down and reduces the dynamic critical exponent. For example, the dynamic critical exponent of the cubic lattice Ising model is $z \approx 0.27$ in the worm update [4], while it is $z \approx 0.46$ [5] in the Swendsen-Wang cluster update [6]. The worm algorithm has been applied to a wide variety of physical models, such as the $|\phi|^4$ model [1], the Potts model [7], the O(n) loop model [8,9], frustrated Ising models [10], spin glasses [11], lattice QCD [12], quantum spins [13–15], bosons [3,16,17], and fermions [18,19]. Thus, it is crucial to improve the worm algorithm and push the limit of the Monte Carlo method for physical models.

The heart of the worm algorithm is to enlarge the state space to those with kinks that break the constraint. A nonlocal update in the original state space, otherwise hard to perform, can be achieved as a series of local updates in the enlarged state space. In the worm update, the Monte Carlo dynamics can be viewed as a diffusion process of the kinks. Because the configuration is updated by diffusion of the kink, a higher diffusivity is expected to yield a higher sampling efficiency. Here, a bottleneck of the algorithm is the worm backscattering process, in which the kink traces back the path. Because the backscattering, a rejection process in the Markov chain Monte Carlo (MCMC) method, naturally lowers the diffusivity, it is desirable to minimize the worm backscattering probability for efficient sampling [20].

As improved versions of the worm algorithm, the directed loop (or the directed worm) algorithm for quantum systems [13] and an extended version for classical systems were proposed [4]. In classical cases, the kinks in the directed-worm algorithm are located on bonds of a lattice instead of sites, in contrast to the conventional worm algorithm [1]. The probability of worm scattering can be easily optimized using the geometric allocation approach [20–22], and the backscattering probability is reduced to zero in many cases. The directed-worm algorithm enhances the diffusivity of the kink and significantly improves sampling efficiency.

In the meantime, it has been actively discussed that nonreversible Markov chains can outperform reversible chains [23,24]. It was mathematically proved that adding perturbative stochastic flows breaking detailed balance, or reversibility, always leads to faster distribution convergence [25,26]. Lifting is a versatile approach to constructing a nonreversible Markov chain, and it has been applied to various physical systems [27]. The idea of lifting is to enlarge the state space and introduce stochastic flows in the enlarged state space. The lifted nonreversible Markov chain can achieve up to a square root improvement in the convergence time to the steady state [28–30]. Although it is challenging to reach square root improvement in general cases, lifted nonreversible chains can achieve a significant variance reduction in many cases. A variety of efficient Monte Carlo methods have been developed based on the concept of lifting, such as the event-chain Monte Carlo method [31,32]. In particular, lifted versions of the worm algorithm [33] and self-avoiding walks [34,35] were recently proposed and shown to be much more efficient than the original reversible algorithm. The improvement factor is especially large in high dimensions.

In the present paper, we show a lifted version of the directed-worm algorithm to improve the worm algorithm further. The transition probability is optimized using the geometric allocation approach to minimize backscattering probability and maximize net stochastic flow. Reviewing the bond representation of the Ising model and the directed-worm algorithm in Sec. II, we provide an optimized probability set for the Ising model on the d-dimensional torus in Sec. III. Investigating the statistical errors of the energy and the magnetic susceptibility at the critical temperature of the four-dimensional hypercubic lattice Ising model, we demonstrate a significant variance reduction in Sec. IV: the sampling efficiency of the present worm algorithm is approximately 80, 5, and 1.7 times those of the standard worm algorithm, the Wolff algorithm, and the previous lifted worm algorithm, respectively. The dynamic critical exponent is estimated to be $z \approx 0$ in the worm and the Wolff algorithms. The present paper is summarized with discussions in Sec. V.

II. MODEL

The worm algorithm for classical models was originally proposed by Prokof'ev and Svistunov [1], which we call the PS worm hereafter. We adopt the same representation of the partition function of the ferromagnetic Ising model on a bipartite lattice. The model is represented by $-\beta H = K \sum_{\langle ij \rangle} \sigma_i \sigma_j$, where $\beta = 1/T$ is the inverse temperature, H is the Hamiltonian, $\sigma_i = \pm 1$ is the Ising spin at site i of a lattice, and $\langle i, j \rangle$ runs over all the pairs of nearest-neighbor sites. Using the identity $e^{K\sigma_i\sigma_j} = \cosh(K) \sum_{n_b=0,1} [\sigma_i \sigma_j \tanh K]^{n_b}$, the partition function is represented as follows:

$$\begin{aligned} Z &= \sum_{\sigma_i=\pm 1} e^{K \sum_{\langle ij \rangle} \sigma_i \sigma_j} \\ &= \sum_{\sigma_i=\pm 1} \prod_{b=\langle ij \rangle} e^{K\sigma_i\sigma_j} \\ &= \sum_{\sigma_i=\pm 1} \prod_{b=\langle ij \rangle} \cosh(K) \sum_{n_b=0,1} [\sigma_i \sigma_j \tanh K]^{n_b} \\ &= 2^N [\cosh K]^{N_b} \sum_{\{n_b\} \in \mathrm{loop}} [\tanh K]^{\ell}, \end{aligned} \tag{1}$$

where n_b denotes the bond variable on bond b, $\ell \equiv \sum_b n_b$, and N and N_b are the total numbers of sites and bonds of a lattice, respectively. The sum in the last line is taken over all the bond configurations that form loops of the activated bonds ($n_b = 1$). Integrating out the spin variables first removes all the bond configurations containing an open string of activated bonds.

The worm algorithm aims to sample bond configurations efficiently under the loop constraint. In the worm algorithm, kinks that break the constraint are inserted into the system, and the kink random walk updates the configuration. A serious bottleneck of the algorithm is worm backscattering, in which the kink traced the path backwards. As the backscattering lowers the diffusivity of the kink and the sampling efficiency, it is desirable to minimize the worm backscattering probability for efficient simulations [20,36].

A directed version of the worm algorithm [4] was recently proposed to avoid backscattering. In contrast to the standard worm algorithm, the kinks are located on bonds and move from bond to bond stochastically, which we call worm scattering hereafter. The kink is assumed to be located at the center of a bond; the bond variable takes 0, $\frac{1}{2}$, or 1 accordingly. The kink possesses the moving direction and keeps it forward after each scattering.

III. LIFTED DIRECTED-WORM ALGORITHM

A. Lifting on the energy axis

We apply lifting to the directed-worm algorithm to further improve the worm algorithm. In the lifting technique, the state space is enlarged by introducing lifted variables, and net stochastic flows are induced in the enlarged space [23,24,27]. As relevant applications, lifted versions of the Berretti-Sokal (BS) algorithm [37] were recently proposed for the Ising model [33] and self-avoiding walks [34,35]. In the original BS algorithm, one randomly chooses to increase n_b or decrease n_b in each local worm update. When increasing (decreasing) n_b, a state with increased (decreased) n_b is randomly chosen and accepted using the standard Metropolis algorithm. In the lifted BS algorithm, one keeps the label + corresponding to the n_b increasing mode and − corresponding to the n_b decreasing mode, and flips it ($+ \leftrightarrow -$) only when the proposed state is rejected by the Metropolis filter. Thus, the state space is enlarged by introducing the lifted variable $\sigma = \pm$. Because the number of activated bonds $\ell = \sum_b n_b$ corresponds to the energy of the original spin system, as represented in Eq. (A1), this approach introduces net stochastic flows on the energy axis. The lifted BS worm algorithms significantly improve sampling efficiency compared to the original BS worm algorithm.

We use the same type of lifting in the directed-worm framework, adding the lifted variable to the configuration space. The worm configuration has the mode ($\sigma = +$ or $-$) in addition to the position of the kinks and the moving direction. The lifted worm algorithm does not modify the other update processes and is described by Algorithm 1. Clearly, probability optimization is crucial to the effectiveness of the lifting.

Algorithm 1 Lifted directed-worm algorithm.

```
Require: A bond configuration under the loop constraint
  $N_w$ ← the total number of worms
  for $n$ ← 1 to $N_w$ do
      Choose a bond $b_0$, a moving direction, and a mode (+ or −)
       randomly.
      $b$ ← $b_0$
      do
          Choose the next bond $c$ and the next mode with a certain
          probability.
          if $b \neq c$ then
              Update the bond variables $n_b$ and $n_c$ and keep the
               moving direction forward.
              $b$ ← $c$
          else
              Flip the moving direction.
          end if
      while $c \neq b_0$
      Measure observables.
  end for
  Calculate the averages of observables.
```

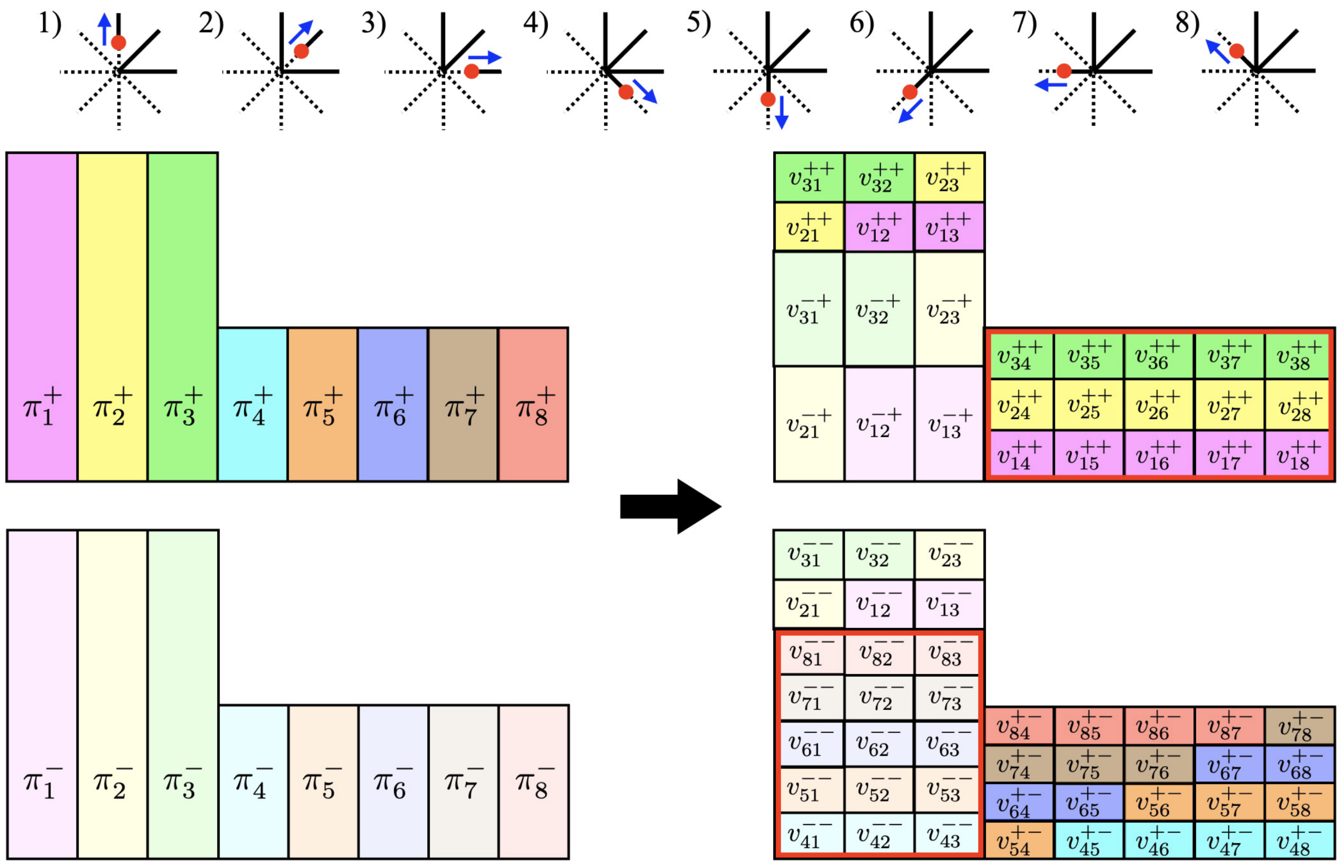
1)
2)
3)
4)
5)
6)
7)
8)
π_1^+
π_2^+
π_3^+
π_4^+
π_5^+
π_6^+
π_7^+
π_8^+
π_1^-
π_2^-
π_3^-
π_4^-
π_5^-
π_6^-
π_7^-
π_8^-
v_{31}^{++} v_{32}^{++} v_{23}^{++}
v_{21}^{++} v_{12}^{++} v_{13}^{++}
v_{31}^{-+} v_{32}^{-+} v_{23}^{-+}
v_{21}^{-+} v_{12}^{-+} v_{13}^{-+}
v_{34}^{++} v_{35}^{++} v_{36}^{++} v_{37}^{++} v_{38}^{++}
v_{24}^{++} v_{25}^{++} v_{26}^{++} v_{27}^{++} v_{28}^{++}
v_{14}^{++} v_{15}^{++} v_{16}^{++} v_{17}^{++} v_{18}^{++}
v_{31}^{--} v_{32}^{--} v_{23}^{--}
v_{21}^{--} v_{12}^{--} v_{13}^{--}
v_{81}^{--} v_{82}^{--} v_{83}^{--}
v_{71}^{--} v_{72}^{--} v_{73}^{--}
v_{61}^{--} v_{62}^{--} v_{63}^{--}
v_{51}^{--} v_{52}^{--} v_{53}^{--}
v_{41}^{--} v_{42}^{--} v_{43}^{--}
v_{84}^{+-} v_{85}^{+-} v_{86}^{+-} v_{87}^{+-} v_{78}^{+-}
v_{74}^{+-} v_{75}^{+-} v_{76}^{+-} v_{67}^{+-} v_{68}^{+-}
v_{64}^{+-} v_{65}^{+-} v_{56}^{+-} v_{57}^{+-} v_{58}^{+-}
v_{54}^{+-} v_{45}^{+-} v_{46}^{+-} v_{47}^{+-} v_{48}^{+-}

FIG. 1. Example of the lifted worm scattering and the geometric allocation in the four-dimensional hypercubic lattice Ising model. In each state 1, 2, ..., 8, projected onto the two-dimensional space for illustrative purposes, the solid black lines, the red circles, and the blue arrows represent activated bonds, the moving kink, and the moving direction, respectively. The weight of each state determined by the partition function is denoted by π_i $(i = 1, 2, \dots, 8)$, and the weight of the lifted configuration is $\pi_i^\sigma = \frac{1}{2}\pi_i$ $(\sigma = \pm)$. L and S denote the sets of the states with the larger and smaller weights, respectively: $L = \{i : \pi_i > \pi_{j \in L^C}\}$ and $S = L^C$. The numbers of elements are $n_L = |L|$ and $n_S = |S|$, and $n_L = 3$ and $n_S = 5$ in this case. Defining $\pi_L = \pi_{i\in L}$, $\pi_S = \pi_{i \in S}$, $v_a = \frac{\pi_S}{n_L}$, $v_b = \frac{\pi_L - n_S v_a}{n_L - 1}$, we set $v_{ij}^{++} = v_b$ $(i, j \in L, i \neq j)$, v_a $(i \in L, j \in S)$, $v_{ij}^{--} = v_b$ $(i, j \in L, i \neq j)$, v_a $(i \in S, j \in L)$, $v_{ij}^{+-} = \frac{\pi_S}{n_S - 1}$ $(i, j \in S, i \neq j)$, $v_{ij}^{-+} = \frac{n_S v_a}{n_L - 1}$ $(i, j \in L, i \neq j)$, and $v_{ij}^{\sigma\sigma'} = 0$ for the other cases. The areas enclosed by the red lines represent the total net stochastic flow in each mode, which is maximized in this case and amounts to $n_S \pi_S$. The transition probability is given by $p_{(i,\sigma)\to(j,\sigma')} = v_{ij}^{\sigma\sigma'}/\pi_i^\sigma$. The order of the states within L and S does not matter to the transition probability.

B. Probability optimization

In the standard lifting technique, the lifted variable determines the next proposed state, and the Metropolis filter gives the acceptance probability [33]. This scheme is simple and effective, but can we optimize the whole transition probability without decomposing the transition probability into the proposal and the acceptance probabilities? We show here that we can easily optimize the whole probability using the geometric allocation approach [20–22].

To see our idea of probability optimization, let us consider a directed-worm scattering process for the d-dimensional hypercubic lattice model. Let π_i $(i = 1, \dots, n)$ denote the weight, or the measure, of each state, with $n = 2d$. An example of the worm scattering for $d = 4$ is illustrated in Fig. 1. There are eight possible states $(n = 8)$ after the worm scattering. Let L and S denote the sets of the states with the larger and smaller weights, respectively, that is, $L = \{i : \pi_i > \pi_{j \in L^C}\}$ and $S = L^C$. Accordingly, let us define $\pi_L = \pi_{i \in L}$, $\pi_S = \pi_{i \in S}$, $n_L = |L|$, and $n_S = |S|$. In the Ising model (1), the smaller weight states have a larger number of activated bonds $\ell = \sum_b n_b$, and the weight ratio is given by $\pi_S/\pi_L = \tanh K$. Note that n_L and n_S take odd numbers in the worm scattering.

We duplicate here the configurations and introduce the lifted variables. The weight of each lifted configuration is set to half the original weight: $\pi_i^\sigma = \frac{1}{2}\pi_i$ $(\sigma = \pm)$. Our purpose is to optimize the transition probabilities from (i, σ) to (j, σ'), $p_{(i,\sigma)\to(j,\sigma')}$ $(i, j = 1, \dots, n$ and $\sigma, \sigma' = +, -)$, under the global balance condition:

$$\pi_i^\sigma = \sum_{j,\sigma'} \pi_i^\sigma p_{(i,\sigma)\to(j,\sigma')} = \sum_{j,\sigma'} \pi_j^{\sigma'} p_{(j,\sigma')\to(i,\sigma)} \quad \forall\, i, \sigma. \tag{2}$$

It is convenient to define the stochastic flow from (i, σ) to (j, σ'): $v_{ij}^{\sigma\sigma'} \equiv \pi_i^\sigma p_{(i,\sigma)\to(j,\sigma')}$. The balance condition (2) is expressed by

$$\pi_i^\sigma = \sum_{j,\sigma'} v_{ij}^{\sigma\sigma'} = \sum_{j,\sigma'} v_{ji}^{\sigma'\sigma} \quad \forall\, i, \sigma. \tag{3}$$

The guideline of the probability optimization in our approach is to minimize the worm backscattering flow

$v_{\rm back} = \sum_{i,\sigma,\sigma'} v_{ii}^{\sigma\sigma'}$ and maximize the net stochastic flow in each mode $v_{\rm net}^{+} = \sum_{i\in L, j\in S} v_{ij}^{++}$ and $v_{\rm net}^{-} = \sum_{i\in S, j\in L} v_{ij}^{--}$.

Here, we can view the optimization problem as an allocation problem of the stochastic flow $v_{ij}^{\sigma\sigma'}$. As illustrated in Fig. 1, an allocation of $v_{ij}^{\sigma\sigma'}$ satisfies Eq. (3) if each *color* area and the entire *box* shape are unchanged. In our approach, we maximize the net stochastic flow under the condition that the backscattering probability is minimized. We geometrically found the optimal allocation and obtained the analytical form. For $1 < n_L < n-1$, defining $v_a = \frac{\pi_S}{n_L}$ and $v_b = \frac{\pi_L - n_S v_a}{n_L - 1}$, we set

$$v_{ij}^{\sigma\sigma'} = \begin{cases} v_b & (\sigma = \sigma' = +, i, j \in L, i \neq j), \\ v_a & (\sigma = \sigma' = +, i \in L, j \in S), \\ v_b & (\sigma = \sigma' = -, i, j \in L, i \neq j), \\ v_a & (\sigma = \sigma' = -, i \in S, j \in L), \\ \frac{\pi_S}{n_S - 1} & (\sigma = +, \sigma' = -, i, j \in S, i \neq j), \\ \frac{n_S v_a}{n_L - 1} & (\sigma = -, \sigma' = +, i, j \in L, i \neq j), \\ 0 & (\text{otherwise}). \end{cases} \tag{4}$$

This solution for $n_L = 3$ is shown graphically in Fig. 1 and is possible if $v_b \geqslant 0$, that is,

$$n_L \pi_L \geqslant n_S \pi_S. \tag{5}$$

We successfully achieve $v_{\rm back} = 0$ and $v_{\rm net}^{+} = v_{\rm net}^{-} = n_S \pi_S$, which are maximized in this case. For $n_L = 1$ and $n-1$, we avoid backscattering without net flow. Specifically, for $n_L = 1$, defining $v_c = \frac{\pi_L}{n_S}$, we set

$$v_{ij}^{\sigma\sigma'} = \begin{cases} v_c & (\sigma = \sigma', i \in L, j \in S), \\ v_c & (\sigma = \sigma', i \in S, j \in L), \\ \frac{\pi_S - v_c}{n_S - 1} & (\sigma = \sigma', i, j \in S, i \neq j), \\ 0 & (\text{otherwise}). \end{cases} \tag{6}$$

This solution exists if

$$n_S \pi_S \geqslant \pi_L. \tag{7}$$

For $n_L = n - 1$, we set

$$v_{ij}^{\sigma\sigma'} = \begin{cases} v_a & (\sigma = \sigma', i \in L, j \in S), \\ v_a & (\sigma = \sigma', i \in S, j \in L), \\ \frac{\pi_L - v_a}{n_L - 1} & (\sigma = \sigma', i, j \in L, i \neq j), \\ 0 & (\text{otherwise}). \end{cases} \tag{8}$$

This solution always exists. Note that for $n_L = 1$ and $n-1$, introducing a net stochastic flow between L and S necessarily causes worm backscattering. We put backscattering minimization before stochastic flow maximization.

As in the other applications of lifting, our solution partially satisfies the skewed detailed balance [28]: $\pi_i^{+} p_{(i,+)\to(j,+)} = \pi_j^{-} p_{(j,-)\to(i,-)}$, that is, $v_{ij}^{++} = v_{ji}^{--}$. However, in contrast to the standard lifting method, we allow the simultaneous update of both the original state variables and the lifted variable to avoid backscattering: $v_{ij}^{\sigma\sigma'} = \frac{\pi_S}{n_S - 1}$ $(\sigma = +, \sigma' = -, i, j \in S, i \neq j)$ and $\frac{n_S v_a}{n_L - 1}$ $(\sigma = -, \sigma' = +, i, j \in L, i \neq j)$ in Eq. (4). Although it is nontrivial to prove the ergodicity of the present algorithm mathematically because of this diagonal update, we have not observed any sign of the ergodicity breaking in our simulations.

In the Ising model (1), $\pi_S/\pi_L = \tanh K$, and the condition (5) is satisfied if $n_L/n_S \geqslant \tanh K$. For $n_L \geqslant n_S$, this is always satisfied, and for $n_L < n_S$, it is satisfied if $K \leqslant \tanh^{-1} \frac{n_L}{n_S} = \frac{1}{2} \ln \frac{n}{n_S - n_L}$. The case of $n_L = 3$ requires the tightest condition. Hence, the solution (4) exists for all the cases if $K \leqslant \frac{1}{2} \ln \frac{n}{n-6} = \frac{1}{2} \ln \frac{d}{d-3} \Longleftrightarrow \frac{2}{\ln \frac{d}{d-3}} \leqslant T$. On the other hand, the condition (7) is satisfied if $1/n_S \leqslant \tanh K$. Thus, the solution (6) exists if $K \geqslant \frac{1}{2} \ln \frac{n}{n-2} = \frac{1}{2} \ln \frac{d}{d-1} \Longleftrightarrow T \leqslant \frac{2}{\ln \frac{d}{d-1}} = T_{\rm Bethe}$, which is the transition temperature in the Bethe approximation and always higher than the true transition temperature in finite dimensions.

In simulation, we calculate all the transition probabilities before Monte Carlo sampling and store them in memory. We choose the next bond and the mode for each worm scattering process using the stored probabilities and Walker's alias method [38,39]. The computational cost of generating a random event is $O(1)$ and almost independent of the number of choices n in contrast to the cost of the binary search $O(\log_2 n)$. The present worm algorithm has practically no extra computational cost compared to the previous worm algorithms. We confirmed that in our implementation, the wall clock time per worm scattering of the present worm algorithm was almost the same with the PS worm algorithm, as discussed in Appendix C.

IV. RESULTS

We investigate the performance of the lifted version of the directed-worm algorithm, comparing it with the PS worm algorithm [1], the lifted BS worm algorithm [33], and the Wolff cluster algorithm [40]. We focus on the critical slowing down of the four-dimensional ($d = 4$) hypercubic Ising model, and we calculate the energy and the magnetic susceptibility up to $L = 56$, where L is the system length, and the number of sites is $N = L^4$. See Appendix A for the measurements of these quantities. The temperature was set to the critical temperature $1/T_{\rm c} = 0.149\,694\,7(5)$ [41], and periodic boundaries were used in all spatial directions. The solutions presented in Sec. III B exist because $1.443 \simeq \frac{2}{\ln \frac{d}{d-3}} \leqslant T_{\rm c} \leqslant \frac{2}{\ln \frac{d}{d-1}} \simeq 6.952$, as mentioned above. In the present simulations, we set the initial state with $n_b = 0 \;\forall\; b$. We ran 2^{28}, 2^{24}, 2^{24}, and 2^{22} Monte Carlo steps for each Markov chain of the PS worm, the lifted BS worm, the present worm, and the Wolff cluster algorithms, respectively. The first half of the samples was discarded for thermalization (burn-in). The mean squared errors and the asymptotic variances of the observables were calculated using the binning analysis such that the size of each bin is much larger than the autocorrelation time [42]. We confirmed that the mean squared errors converged within the error bars with respect to the bin size. We also averaged the quantities over 128 independent Markov chains for each L, allowing us to estimate the error bars of the present results reliably. In the Wolff algorithm, we test two susceptibility estimators: $\hat{\chi} = \beta M_z^2/N$ (dubbed Wolff spin), where M_z is the total magnetization, and $\hat{\chi} = \beta \ell_{\rm cl}$ (Wolff cluster), where $\ell_{\rm cl}$ is the cluster size. To compare efficiency fairly, we normalize the autocorrelation time in units of the number of sites for each L, using the average worm length and cluster size. See Appendix B for the normalization of the autocorrelation time

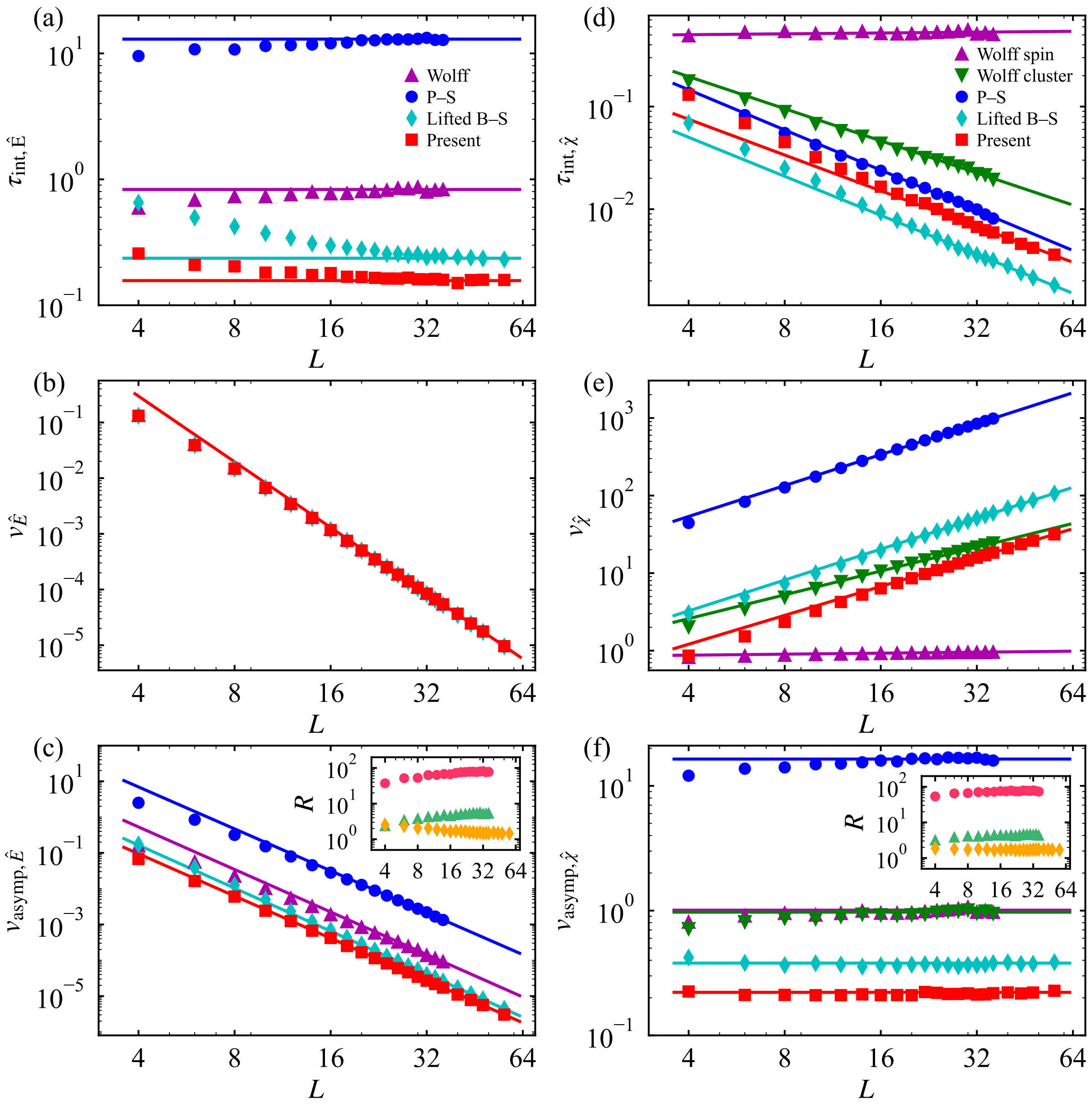

FIG. 2. (a), (d) Integrated autocorrelation time; (b), (e) variance; and (c), (f) asymptotic variance of the energy estimator [(a)–(c)] and the susceptibility estimators [(d)–(f)] as a function of the system length L of the four-dimensional hypercubic lattice Ising model at the critical temperature. The Wolff algorithm (triangles) [40], the Prokof'ev-Svistunov worm (circles) [1], the lifted Berretti-Sokal worm (diamonds) [33], and the present worm (squares) algorithms are compared. In the Wolff algorithm, we test two susceptibility estimators using the spins dubbed the Wolff spin (upper triangles) and the cluster size dubbed the Wolff cluster (lower triangles). The data are fitted to a power-law function of L. The exponents of $\tau_{\mathrm{int},\hat{E}}$, $v_{\hat{E}}$, and $v_{\mathrm{asymp},\hat{E}}$ are estimated to be approximately 0, −3.9, and −3.9, respectively, and likely the same for all four algorithms. The exponents of $\tau_{\mathrm{int},\hat{\chi}}$ are estimated to be approximately 0, −1.0, −1.3, −1.3, and −1.2 for the Wolff spin, the Wolff cluster, the Prokof'ev-Svistunov worm, the lifted Berretti-Sokal worm, and the present worm algorithms, respectively; those of $v_{\hat{\chi}}$ are 0, 1.0, 1.3, 1.3, and 1.2, respectively; that of $v_{\mathrm{asymp},\hat{\chi}}$ is 0 and presumably the same for the four algorithms. The insets of panels (c) and (f) show the ratios of the asymptotic variances in the Wolff (triangles), the Prokof'ev-Svistunov worm (circles), and the lifted Berretti-Sokal (diamonds) algorithms to the one in the present worm algorithm, which are, for the largest system sizes, approximately 5.4, 80, and 1.5 for (c) the energy; and 4.5, 77, and 1.7 for (f) the susceptibility. The error bars are smaller than the symbol sizes.

and the definitions of the integrated autocorrelation time, the variance, and the asymptotic variance. Note that the inverse of the asymptotic variance is proportional to the sampling efficiency of an MCMC sampler.

Figure 2 shows the integrated autocorrelation time $\tau_{\mathrm{int},\hat{O}}$, the variance $v_{\hat{O}}$, and the asymptotic variance $v_{\mathrm{asymp},\hat{O}}$ of the energy ($\hat{O} = \hat{E}$) and the susceptibility estimators ($\hat{O} = \hat{\chi}$). Fitting a power law to the data, we estimated the exponents, as shown in the caption. The exponents of the energy estimator are likely the same for the four algorithms. Nevertheless, the sampling efficiency of the present worm algorithm is approximately 5.4, 80, and 1.5 times as high as those of the Wolff cluster, the PS worm, and the lifted BS worm algorithms, respectively, as shown in the inset of Fig. 2(c). Interestingly, for small system sizes, $\tau_{\mathrm{int},\hat{E}}$ decreases with L for the lifted BS worm and the present algorithms while increasing for the PS worm and the Wolff cluster algorithms. Also, the exponent of the asymptotic variances of the susceptibility estimators is almost zero for the four algorithms. Again, the sampling efficiency of the present worm algorithm is approximately 4.5, 77, and 1.7 times those of the Wolff cluster, the PS worm, and the lifted BS worm algorithms, respectively, as shown in the inset of Fig. 2(f). Although the size scalings of the $\tau_{\mathrm{int},\hat{\chi}}$ and $v_{\hat{\chi}}$ are quite different for the two estimators in the Wolff algorithm, the asymptotic variance, that is, the product of the two quantities, as shown in Eq. (B5), is almost the same. Noticeably, the exponents for the PS worm and the lifted BS worm algorithms are almost identical.

The exponent of $v_{\hat{E}}$ should be asymptotically the same for the specific heat, $\alpha/\nu - d = -4$, where $\alpha = 0$ and $\nu = 1/2$ are the mean-field critical exponents of the specific heat and the correlation length, respectively. Note that the variance is relative to the mean squared, defined by Eq. (B7). The estimate obtained by a simple power law, -3.9, is slightly larger than the theoretical value and still decreases with L. The slow convergence of the powers of $\tau_{\mathrm{int},\hat{E}}$ and $v_{\hat{E}}$ is presumably due to the logarithmic correction at the upper critical dimension [43].

V. CONCLUSIONS

We propose a lifted version of the directed-worm algorithm, adding the lifted variable ($+$ and $-$) to the original configuration. The probability optimization in the worm scattering process is essential to efficient computation. Using the geometric allocation approach, we minimize the worm backscattering probability and maximize net stochastic flow, as illustrated in Fig. 1. We provide the optimal probability set for the d-dimensional hypercubic lattice Ising model. The backscattering probability reduces to zero, and the net stochastic flow is maximized in a wide range of temperatures, including the critical temperature. The entire procedure of the lifted directed-worm algorithm is described by Algorithm 1.

We calculate the integrated autocorrelation time, the variance, and the asymptotic variance of the energy and the magnetic susceptibility estimators of the four-dimensional hypercubic lattice Ising model at the critical temperature. Several relevant exponents are the same for the Wolff cluster [40], the PS worm [1], the lifted BS worm [33], and the present worm algorithms. Nevertheless, the sampling efficiency of the present algorithm is approximately 5, 80, and 1.7 times those of the Wolff cluster, the PS worm, and the lifted BS worm algorithms, respectively, shown in Fig. 2. In our implementation, the present algorithm achieves the shortest real time needed for a certain relative error, as discussed in Appendix C. The improvement factor of the lifted BS worm algorithm over the PS worm algorithm is slightly different from the value reported in Ref. [33]. This is likely because, in the previous report, the number of activated bonds was measured not in the original physical space but in the enlarged state space, including the kinks.

We estimate the dynamic critical exponent associated with the integrated autocorrelation time of the energy to be $z_{\mathrm{int},\hat{\mathrm{E}}} \approx 0$ for all the compared algorithms. For reversible cases, the integrated autocorrelation time satisfies a Li-Sokal-type bound ([44], Corollary 9.2.3): $\tau_{\mathrm{int},\hat{E}} \times L^d \geqslant \mathrm{const} \times \mathrm{var}[\hat{E}]$. This leads to $z_{\mathrm{int},\hat{\mathrm{E}}} \geqslant \alpha/\nu = 0$ for $d = 4$. Therefore, interestingly, the reversible algorithms, that is, the PS worm and the Wolff algorithms, achieve this bound.

The autocorrelation function of the number of activated bonds has several exponential terms in $d \geqslant 4$, shown in Ref. [33]. Nevertheless, the scaling of the exponential autocorrelation time with respect to the system length is the same as that of the integrated autocorrelation time in $d = 5$. Assuming that this is the case for $d \geqslant 4$, equal to or greater than the upper critical dimension, we also estimate the dynamic critical exponent associated with the exponential autocorrelation time to be $z_{\mathrm{exp}} \approx 0$ from the scaling of the integrated autocorrelation time of the energy estimator, which is consistent with a previous estimate for the Wolff algorithm [45]. In contrast, the dynamic critical exponent of the three-dimensional Ising model in the worm and Wolff cluster updates is greater than the Li-Sokal-type bound: $z \approx 0.27 > \alpha/\nu \simeq 0.174$ [4]. Since the bound is achieved in $d = 4$, we expect $z = 0$ in $d \geqslant 4$.

The present approach of the lifted directed-worm algorithm can be applied to a variety of quantum models as well as classical models. As long as the lookup table of the transition probability is prepared, the lookup cost in the simulation is almost independent of the number of states, thanks to Walker's alias method, as demonstrated in Appendix C. Although it is nontrivial to obtain an analytical solution of the transition probability in general cases, linear programming can numerically find a reasonable solution. It is noteworthy that a numerically obtained solution may break ergodicity, so we need to check the validity of the solution by comparing it to other samplers. As demonstrated here, the lifted directed-worm algorithm is expected to improve the sampling efficiency of the worm update for various systems. Applications to other physical models are promising and need to be investigated in the future.

ACKNOWLEDGMENTS

The author is grateful to Synge Todo for the fruitful discussions. Simulations were performed using computational resources of the Supercomputer Center at the Institute for Solid State Physics, the University of Tokyo. This work was supported by the JSPS KAKENHI Grants-in-Aid for Scientific Research (No. 22K03508).

APPENDIX A: MEASUREMENTS

From the bond representation of the Ising model (1), the total energy of the original spin system is given by

$$\begin{aligned} E &= -\frac{\partial \ln Z}{\partial \beta} \\ &= -N_b \tanh K - \left(\frac{1}{\tanh K} - \tanh K\right)\langle \ell \rangle, \end{aligned} \qquad \text{(A1)}$$

where the brackets $\langle O \rangle$ denote the Monte Carlo average of an observable O. The number of Monte Carlo steps in our algorithm is identical to the number of worms from insertion to removal. In the directed-worm algorithm, a susceptibility estimator is given by

$$\hat{\chi} = \frac{\beta}{4zw} \sum_{\text{scattering}} f_{\text{rew}}, \tag{A2}$$

where z is the coordination number ($z = 2d$ for the d-dimensional hypercubic lattice), $w = \frac{1}{2}$ is the extra weight the worm carries, and $f_{\text{rew}} = (u + \frac{1}{u}) f_h$ is the reweighting factor with $u = \sqrt{\tanh K}$ after the worm scattering. f_h takes $2/u$ if the head is on an activated bond ($n_b = 1$), $2u$ on a deactivated bond ($n_b = 0$), and $u + \frac{1}{u}$ on a half-activated bond ($n_b = \frac{1}{2}$). The reweighting factor f_{rew} is calculated after each worm scattering process and summed over the scattering processes from worm insertion to removal, as denoted by the summation in Eq. (A2). We refer the reader to Ref. [4] for detailed discussions of measurement in the enlarged state space.

APPENDIX B: EFFICIENCY QUANTIFICATION

To assess the efficiency of an MCMC sampler, we calculate the integrated autocorrelation time, the variance, and the asymptotic variance of estimators. For a fair comparison, we normalize the number of Monte Carlo steps and the autocorrelation time in units of the number of sites N:

$$M = M' \frac{\langle \ell_{\text{worm}} \rangle}{N}, \tag{B1}$$

$$\tau_{\text{int},\hat{O}} = \tau'_{\text{int},\hat{O}} \frac{\langle \ell_{\text{worm}} \rangle}{N}, \tag{B2}$$

where M' and τ'_{int} are the original number of Monte Carlo steps and the original integrated autocorrelation time obtained from the simulations, respectively. The value of M' is equal to the total number of worms during a simulation, and $\langle \ell_{\text{worm}} \rangle$ is the average number of worm scattering processes from worm insertion to removal.

The integrated autocorrelation time of an estimator $\hat{O}$ can be estimated by

$$\tau'_{\text{int},\hat{O}} = \frac{\sigma_{\hat{O}}^2}{2\bar{\sigma}_{\hat{O}}^2}, \tag{B3}$$

where $\sigma_{\hat{O}}^2$ is the mean squared error, that is, the square of the statistical error, calculated by binning analysis using a much larger bin size than the autocorrelation time, and $\bar{\sigma}_{\hat{O}}^2$ is calculated without binning [42].

According to the central limit theorem, the relative error squared is asymptotically given by

$$\left(\frac{\sigma_{\hat{O}}}{\mu_{\hat{O}}} \right)^2 \approx \frac{v_{\text{asymp},\hat{O}}}{M}, \tag{B4}$$

where $\mu_{\hat{O}} = \langle \hat{O} \rangle$ is the mean of an unbiased estimator, and $v_{\text{asymp},\hat{O}}$ is the asymptotic variance of the estimator. In other words, the number of Monte Carlo steps needed to achieve a certain precision ϵ of an observable O is approximately given by $v_{\text{asymp},\hat{O}}/\epsilon^2$. Thus, the sampling efficiency of an MCMC sampler can be quantified by the asymptotic variance and proportional to $v_{\text{asymp},\hat{O}}^{-1}$.

The asymptotic variance is represented by

$$v_{\text{asymp},\hat{O}} = 2\tau_{\text{int},\hat{O}} v_{\hat{O}}, \tag{B5}$$

where $v_{\hat{O}}$ is the variance of an estimator $\hat{O}$. While $\tau_{\text{int},\hat{O}}$ depends on MCMC updates, $v_{\hat{O}}$ does not. Therefore, we need to consider both an efficient MCMC update and a beneficial estimator to reduce asymptotic variance. From Eqs. (B1)–(B5), the asymptotic variance and the variance of an estimator can be estimated by

$$v_{\text{asymp},\hat{O}} = M' \left(\frac{\sigma_{\hat{O}}}{\mu_{\hat{O}}} \right)^2 \frac{\langle \ell_{\text{worm}} \rangle}{N}, \tag{B6}$$

$$v_{\hat{O}} = M' \left(\frac{\bar{\sigma}_{\hat{O}}}{\mu_{\hat{O}}} \right)^2, \tag{B7}$$

respectively. In the Wolff cluster algorithm, ℓ_{worm} is replaced with the cluster size ℓ_{cl}.

APPENDIX C: WALL CLOCK TIME

We discuss here the elapsed real time, or the wall clock time, of the present algorithm. To assess the performance of the algorithms, we focus on the autocorrelation time and the asymptotic variance in the main text, which are independent of the implementation of the simulation code. However, the wall clock time is also important to the actual calculation.

We measured the elapsed real time of the worm scattering in our implementation of the PS, the lifted BS, and the present worm algorithms on an Intel Xeon Platinum 9242 Processor using the GNU C++ compiler. The wall clock times of the PS and the present worm algorithms were almost the same, ∼100 ns per scattering. In our code, the elapsed real time of the lifted BS algorithm was longer than that of the PS worm algorithm, but we expect that this difference can be removed by further tuning. We do not expect that the wall clock time of the lifted BS algorithm can be shorter than that of the PS worm algorithm because the PS worm update is simple and concise. The three algorithms thus consume almost the same real time after tuning, which is reasonable from the perspective of the shared fundamental nature of the local worm update. From Eq. (B4), the real time needed to achieve a certain relative error squared is given by the product of the asymptotic variance and the wall clock time of the worm scattering process. Therefore, the asymptotic variances shown in Figs. 2(c) and 2(f) are practically proportional to the needed real time of the compared worm algorithms.

We also measured the wall clock time of the Wolff cluster update per site consisting of the Wolff cluster in our implementation; it was approximately three times that of the PS and the present worm algorithms. This longer time is due to the fact that the number of bonds checked during the Wolff cluster update is much larger than the resulting cluster size. Thus, the needed real time of the Wolff cluster algorithm becomes larger since we normalize the number of Monte Carlo steps such that N sites are updated in one Monte Carlo step, as discussed in Appendix B. Consequently, the present algorithm also provides the most efficient sampler in real time.

We also discuss the lookup cost of the transition probability. The number of possible states in worm scattering is $n = 2d$ in the d-dimensional hypercubic lattice and may be

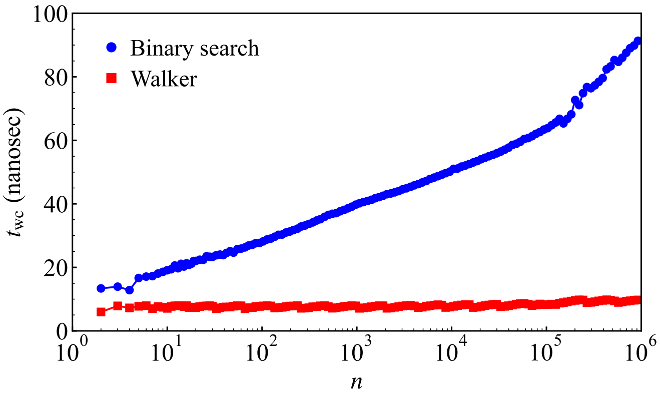

FIG. 3. Wall clock time per sample of random event generation from a discrete probability distribution using the binary search and Walker's alias method. While $O(\log_2 n)$ in the binary search, the computation time is $O(1)$ and almost independent of n in Walker's alias method. It increases more for n larger than corresponding to the cache size, $n \sim 10^5$ in this benchmark.

quite large for other systems. Once a lookup table of the probability is prepared, we can easily generate a random event from any discrete probability distribution. Figure 3 shows the wall clock time per sample generated using the binary search and Walker's alias method [38,39]. We implemented the binary search using std::upper_bound of the C++ Standard Library and Walker's method using the Balance Condition Library (BCL) [46]. The computation time of the binary search is proportional to $\log_2 n$ and increases more for n larger than corresponding to the L2 cache, $n \sim 10^5$ in this benchmark. On the other hand, the computation time of Walker's method is $O(1)$ and almost independent of n. Therefore, thanks to Walker's method, the lookup cost does not depend on the model as long as the lookup table is prepared. We also note that the computational cost of building a lookup table is $O(n)$ and is negligible compared to the Monte Carlo simulation cost [38]. In many cases, the present algorithm needs almost no additional computation cost compared to the PS worm algorithm.

[1] N. Prokof'ev and B. Svistunov, Worm Algorithms for Classical Statistical Models, Phys. Rev. Lett. **87**, 160601 (2001).

[2] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, Exact, complete, and universal continuous-time worldline Monte Carlo approach to the statistics of discrete quantum systems, J. Exp. Theor. Phys. **87**, 310 (1998).

[3] M. Boninsegni, N. V. Prokof'ev, and B. V. Svistunov, Worm algorithm and diagrammatic Monte Carlo: A new approach to continuous-space path integral Monte Carlo simulations, Phys. Rev. E **74**, 036701 (2006).

[4] H. Suwa, Geometric allocation approach to accelerating directed worm algorithm, Phys. Rev. E **103**, 013308 (2021).

[5] G. Ossola and A. D. Sokal, Dynamic critical behavior of the Swendsen-Wang algorithm for the three-dimensional Ising model, Nucl. Phys. B **691**, 259 (2004).

[6] R. H. Swendsen and J.-S. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, Phys. Rev. Lett. **58**, 86 (1987).

[7] Y. D. Mercado, H. G. Evertz, and C. Gattringer, Worm algorithms for the 3-state potts model with magnetic field and chemical potential, Comput. Phys. Commun. **183**, 1920 (2012).

[8] W. Janke, T. Neuhaus, and A. M. J. Schakel, Critical loop gases and the worm algorithm, Nucl. Phys. B **829**, 573 (2010).

[9] Q. Liu, Y. Deng, and T. M. Garoni, Worm Monte Carlo study of the honeycomb-lattice loop model, Nucl. Phys. B **846**, 283 (2011).

[10] G. Rakala and K. Damle, Cluster algorithms for frustrated two-dimensional Ising antiferromagnets via dual worm constructions, Phys. Rev. E **96**, 023304 (2017).

[11] J.-S. Wang, Worm algorithm for two-dimensional spin glasses, Phys. Rev. E **72**, 036706 (2005).

[12] D. H. Adams and S. Chandrasekharan, Chiral limit of strongly coupled lattice gauge theories, Nucl. Phys. B **662**, 220 (2003).

[13] O. F. Syljuåsen and A. W. Sandvik, Quantum Monte Carlo with directed loops, Phys. Rev. E **66**, 046701 (2002).

[14] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods* (Cambridge University Press, Cambridge, 2016).

[15] S. Yasuda, H. Suwa, and S. Todo, Stochastic approximation of dynamical exponent at quantum critical point, Phys. Rev. B **92**, 104411 (2015).

[16] S. Wessel, F. Alet, M. Troyer, and G. G. Batrouni, Quantum Monte Carlo simulations of confined bosonic atoms in optical lattices, Phys. Rev. A **70**, 053615 (2004).

[17] H. Suwa and S. Todo, Generalized Moment Method for Gap Estimation and Quantum Monte Carlo Level Spectroscopy, Phys. Rev. Lett. **115**, 080601 (2015).

[18] E. Burovski, N. Prokof'ev, B. Svistunov, and M. Troyer, The fermi–hubbard model at unitarity, New J. Phys. **8**, 153 (2006).

[19] P. Gunacker, M. Wallerberger, T. Ribic, A. Hausoel, G. Sangiovanni, and K. Held, Worm-improved estimators in continuous-time quantum Monte Carlo, Phys. Rev. B **94**, 125153 (2016).

[20] H. Suwa and S. Todo, Markov Chain Monte Carlo Method without Detailed Balance, Phys. Rev. Lett. **105**, 120603 (2010).

[21] S. Todo and H. Suwa, Geometric allocation approaches in Markov Chain Monte Carlo, J. Phys.: Conf. Ser. **473**, 012013 (2013).

[22] H. Suwa, *Geometrically Constructed Markov Chain Monte Carlo Study of Quantum Spin-phonon Complex Systems*, Springer Theses (Springer, Tokyo, 2014).

[23] P. Diaconis, S. Holmes, and R. M. Neal, Analysis of a nonreversible Markov Chain sampler, Ann. Appl. Probab. **10**, 726 (2000).

[24] F. Chen, L. Lovász, and I. Pak, Lifting markov chains to speed up mixing, in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 1999), p. 275.

[25] C.-R. Hwang, S.-Y. Hwang-Ma, and S.-J. Sheu, Accelerating diffusions, Ann. Appl. Probab. **15**, 1433 (2005).

[26] A. Ichiki and M. Ohzeki, Violation of detailed balance accelerates relaxation, Phys. Rev. E **88**, 020101(R) (2013).

[27] M. Vucelja, Lifting-a nonreversible Markov Chain Monte carlo algorithm, Am. J. Phys. **84**, 958 (2016).

[28] K. S. Turitsyn, M. Chertkov, and M. Vucelja, Irreversible Monte Carlo algorithms for efficient sampling, Physica D **240**, 410 (2011).

[29] H. C. M. Fernandes and M. Weigel, Non-reversible Monte Carlo simulations of spin models, Comput. Phys. Commun. **182**, 1856 (2011).

[30] Y. Sakai and K. Hukushima, Dynamics of one-dimensional ising model without detailed balance condition, J. Phys. Soc. Jpn. **82**, 064003 (2013).

[31] E. P. Bernard, W. Krauth, and D. B. Wilson, Event-chain Monte Carlo algorithms for hard-sphere systems, Phys. Rev. E **80**, 056704 (2009).

[32] M. Michel, S. C. Kapfer, and W. Krauth, Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps, J. Chem. Phys. **140**, 054116 (2014).

[33] E. M. Elçi, J. Grimm, L. Ding, A. Nasrawi, T. M. Garoni, and Y. Deng, Lifted worm algorithm for the Ising model, Phys. Rev. E **97**, 042126 (2018).

[34] H. Hu, X. Chen, and Y. Deng, Irreversible Markov Chain Monte Carlo algorithm for self-avoiding walk, Front. Phys. **12**, 120503 (2017).

[35] H. Zhao and M. Vucelja, Nonreversible Markov Chain Monte Carlo algorithm for efficient generation of self-avoiding walks, Front. Phys. **9**, 782156 (2022).

[36] H. Suwa and S. Todo, Geometric allocation approach for transition Kernel of Markov Chain, Monte Carlo Meth. Appl. **8**, 213 (2012).

[37] A. Berretti and A. D. Sokal, New Monte Carlo method for the self-avoiding walk, J. Stat. Phys. **40**, 483 (1985).

[38] K. Fukui and S. Todo, Order-N cluster Monte Carlo method for spin systems with long-range interactions, J. Comput. Phys. **228**, 2629 (2009).

[39] T. Horita, H. Suwa, and S. Todo, Upper and lower critical decay exponents of Ising ferromagnets with long-range interaction, Phys. Rev. E **95**, 012143 (2017).

[40] U. Wolff, Collective Monte Carlo Updating for Spin Systems, Phys. Rev. Lett. **62**, 361 (1989).

[41] P. H. Lundow and K. Markström, Critical behavior of the Ising model on the four-dimensional cubic lattice, Phys. Rev. E **80**, 031104 (2009).

[42] B. A. Berg, *Markov Chain Monte Carlo Simulations and their Statistical Analysis: With Web-based Fortran Code* (World Scientific, Singapore, 2004).

[43] R. Kenna, Finite size scaling for $O(N)\phi^4$-theory at the upper critical dimension, Nucl. Phys. B **691**, 292 (2004).

[44] N. Madras and G. Slade, *The Self-avoiding Walk* (Springer Science & Business Media, Boston, 2013).

[45] P. Tamayo, R. C. Brower, and W. Klein, Single-cluster Monte Carlo dynamics for the Ising model, J. Stat. Phys. **58**, 1083 (1990).

[46] BCL, Balance Condition Library, https://github.com/cmsi/bcl, accessed: 24 September 2022.