THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCA

Article

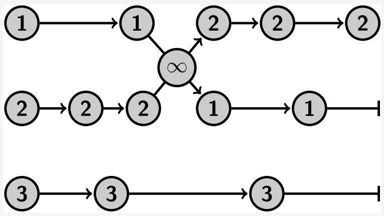
Exchanging Replicas with Unequal Cost, Infinitely and Permanently

Sander Roet, Daniel T. Zhang, and Titus S. van Erp*

Cite This: *J. Phys. Chem. A* 2022, 126, 8878−8886

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: We developed a replica exchange method that is effectively parallelizable even if the computational cost of the Monte Carlo moves in the parallel replicas are considerably different, for instance, because the replicas run on different types of processor units or because of the algorithmic complexity. To prove detailed-balance, we make a paradigm shift from the common conceptual viewpoint in which the set of parallel replicas represents a high-dimensional superstate, to an ensemble-based criterion in which the other ensembles represent an environment that might or might not participate in the Monte Carlo move. In addition, based on a recent algorithm for computing permanents, we effectively increase the exchange rate to infinite without the steep factorial scaling as a function of the number of replicas. We illustrate the effectiveness of this replica exchange methodology by combining it with a quantitative path sampling method, replica exchange transition interface sampling (RETIS), in which the costs for a Monte Carlo move can vary enormously as paths in a RETIS algorithm do not have the same length and the average path lengths tend to vary considerably for the different path ensembles that run in parallel. This combination, coined ∞RETIS, was tested on three model systems.

1. INTRODUCTION

The Markov chain Monte Carlo (MC) method is one of the most important numerical techniques for computing averages in high-dimensional spaces, like the configuration space of a many-particle system. The approach has applications in a wide variety of fields ranging from computational physics, theoretical chemistry, economics, and genetics. The MC algorithm effectively generates a selective random walk through state space in which the artificial steps are designed to ensure that the frequency of visiting any particular state is proportional to the equilibrium probability of that state. The Metropolis[1] or the more general Metropolis−Hastings[2] algorithms are the most common approaches for designing such random steps (MC moves) based on the detailed-balance principle. That is, the MC moves should be constructed such that the number of transitions from an old state s^o to a new state s^n is exactly balanced by the number of transitions from the new to the old state: $\rho(s^{(o)})\pi(s^{(o)} \to s^{(n)}) = \rho(s^{(n)})\ \pi(s^{(n)} \to s^{(o)})$, where $\rho(\cdot)$ is the state space equilibrium probability density and $\pi(\cdot)$ are the probabilities to make a transition between the two states given the set of possible MC moves. Further, the transition is split into a generation and an acceptance/rejection step such that $\pi(s \to s') = P_{\mathrm{gen}}(s \to s')\ P_{\mathrm{acc}}(s \to s')$. In the case that the sampled state space is the configuration space of a molecular system at constant temperature, P_{gen} might relate to moving a randomly picked particle in a random direction over a small random distance, and $\rho(s)$ is proportional to the Boltzmann weight $e^{-\beta E(s)}$, with $\beta = 1/k_B T$ the inverse temperature, k_B the Boltzmann constant, and $E(s)$ the state's energy.

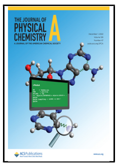
The Metropolis−Hastings algorithm takes a specific solution for the acceptance probability

$$P_{\mathrm{acc}}(s^{(o)} \to s^{(n)}) = \min\left[1, \frac{\rho(s^{(n)})P_{\mathrm{gen}}(s^{(n)} \to s^{(o)})}{\rho(s^{(o)})P_{\mathrm{gen}}(s^{(o)} \to s^{(n)})}\right] \tag{1}$$

The generation probabilities will cancel in the above expression if they are symmetric, $P_{\mathrm{gen}}(s \to s') = P_{\mathrm{gen}}(s' \to s)$ as in the less generic Metropolis scheme. At each MC step, the new state is either accepted or rejected based on the probability above. In case of a rejection, the old state is maintained and resampled. This scheme obeys detailed-balance. In addition, if the set of MC moves are ergodic, equilibrium sampling is guaranteed. When ergodic sampling, even if mathematically obeyed, is slowed down by a rough (free) energy landscape, replica exchange MC becomes useful.

Replica exchange MC (or replica exchange molecular dynamics) is based on the idea to simulate several copies of the system with different ensemble definitions,[3−5] most commonly ensembles with increasing temperature (parallel tempering). By performing "swaps" between adjacent replicas, the low-temperature replicas gain access to the broader space region that are explored by the high-temperature replicas. The detailed-balance and corresponding acceptance−rejection step can be derived by viewing the set of states in the different ensembles (replicas) as a single high-dimensional superstate $S = (s_1, s_2, \cdots, s_N)$ representing the system in a set of N

Received: September 2, 2022
Revised: October 18, 2022
Published: November 17, 2022



https://doi.org/10.1021/acs.jpca.2c06004
J. Phys. Chem. A 2022, 126, 8878−8886

independent "parallel universes". The Metropolis scheme applied to the superstate yields

$$P_{\text{acc}}(S^{(o)} \rightarrow S^{(n)}) = \min\left[1, \frac{\rho(S^{(n)})}{\rho(S^{(o)})}\right] \tag{2}$$

in which the probability of the superstate equals

$$\rho(S) = \rho(s_1, s_2, \cdots, s_N) = \prod_{i=1}^{N} \rho_i(s_i) \tag{3}$$

where $\rho_i(\cdot)$ is the specific probability density of ensemble i. For example, the move that attempts to swap the first two states, implying $S^o = (s_1, s_2, \cdots, s_N)$ and $S^n = (s_2, s_1, \cdots, s_N)$, will be accepted with a probability

$$P_{\text{acc}} = \min\left[1, \frac{\rho_1(s_2)\rho_2(s_1)}{\rho_1(s_1)\rho_2(s_2)}\right] \tag{4}$$

In a replica exchange simulation, swapping moves and standard MC or MD steps are applied alternately. Parallel computing will typically distribute the same number of processing units per ensemble to carry out the computationally intensive standard moves. The swapping move is cheap but requires that the ensembles involved in the swap have completed their previous move. If the standard moves in each ensemble require different computing times, then several processing units have to wait for the slow ones to finish.

If the disbalance per move is relatively constant, then the replicas could effectively be made to progress in cohort by trying to differentiate the number of processing units per ensemble or the relative frequency of doing replica exchange versus standard moves per ensemble. However, this disbalance is not constant in several MC methods, such as with configurational bias MC[6−8] or path sampling.[9] The number of elementary steps to grow a polymer in configurational bias MC obviously depends on the polymer's length that is being grown, but also early rejections lead to a broad distribution of the time it takes to complete a single MC move even in uniform polymer systems. Analogously, the time required to complete an MC move in path sampling simulations will depend on the length of the path being created. Other examples of complex Monte Carlo methods with a fluctuating CPU cost per move are cluster Monte Carlo algorithms[10] and event-chain Monte Carlo.[11,12]

We will show that the standard acceptance eqs 1 and 4 can be applied in a parallel scheme in which ensembles are updated irregularly in time and the average frequency of MC moves is different for the ensembles. In addition, we show that we can apply an infinite swapping[13] scheme between the available ensembles. For this, we develop a new protocol based on the evaluation of permanents that circumvents the steep factorial scaling. This last development is also useful for standard replica exchange.

2. METHODS

2.1. Finite Swapping. In the following, we will assume that we have two types of MC moves. One move that is CPU-intensive and can be carried out within a single ensemble, and replica exchange moves between ensembles which are relatively cheap to execute. The CPU-intensive move will be carried out by a single worker (one processor unit, one node, or a group of nodes) and these workers perform their task in parallel on the different ensembles. One essential part of our algorithm is that we have less workers than ensembles such that whenever the worker is finished and produced a new state for one ensemble, this state can directly be swapped with the states of any of the available ensembles (the ones not occupied by a worker). After that, the worker will randomly switch to another unoccupied ensemble for performing a CPU-intensive move.

In its most basic form, the algorithm consists of the following steps:

1. Define N ensembles and let $\rho_i(\cdot)$ be the probability distribution of ensemble i. We also define P_{RE} which is the probability of doing a replica exchange move.
2. Assign $K < N$ workers (processing units) to K of the N ensembles for performing a CPU-intensive MC move. Each ensemble is at all times occupied by either 1 or 0 workers. The following steps are identical for all of the workers.
3. If the worker is finished with its MC move in ensemble i, the new state is accepted or rejected according to eq 1 (with ρ_i for ρ). Ensemble i is updated with the new state (or by resampling the old state in case of rejection) and is then considered to be free.
4. Take a uniform random number ν between 0 and 1. If $\nu > P_{\text{RE}}$, go to step 7.
5. Among the free ensembles, pick a random pair (i, j).
6. Try to swap the states of ensembles i and j using eq 4 (with labels i, j instead of 1, 2). Update ensembles i, j with the swapped state or the old state in case of a rejection. Return to step 4.
7. Select one of the free ensembles at random and assign the worker to that ensemble for performing a new standard move. Go to step 3.

In this algorithm, ensembles are not updated in cohort like in standard replica exchange, but updates occur at irregular intervals. In addition, the different ensemble conditions can result in systematic differences in the number of states that are being created over time. To prove that the above scheme actually samples the correct distributions requires a fundamentally new conceptual view as the superstate picture is no longer applicable. Despite that the algorithm uses the same type of eqs 1 and 4, as one would use in standard replica exchange, it does not rely on eqs 2 and 3 that are no longer valid. In the SI, we provide a proof from the individual ensemble's perspective in which the other ensembles provide an "environment" $\mathcal{E}$ that might, or might not, participate in the move of the ensemble considered. By doing so, we no longer require that the number of transitions from old to new, $S^{(o)} \rightarrow S^{(n)}$, is the same as from new to old, $S^{(n)} \rightarrow S^{(o)}$. Instead, by writing $S = (s_1, \mathcal{E})$, from ensemble 1's perspective, we have that the number of $(s_1^{(o)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(n)}, {}^{a}\mathcal{E}^{(n)})$ transitions should be equal to the number of $(s_1^{(n)}, \mathcal{E}^{(o)}) \rightarrow (s_1^{(o)}, {}^{a}\mathcal{E}^{(n)})$ transitions when the standard move is applied where ${}^{a}\mathcal{E}^{(n)}$ refers to *any* new environment. In the SI, we show a similar detailed-balance condition for the replica exchange moves. At step 6 we sample only ensemble i and j or, alternatively, all free ensembles get a sample update. This would mean resampling the existing state of those not involved in a swap ("null move"). This makes the approach more similar to the superstate sampling albeit using only free ensembles, as described in the SI. The null move does not reduce the statistical uncertainty, but we mention it here as it makes it easier to explain the infinite swapping approach. But for the detailed-balance conditions to be valid it is imperative that occupied ensembles are not sampled.

An essential aspect of the efficiency of our algorithm is that the number of workers K is less than the number of ensembles N. The case $K = N$ is valid but would reduce the number of replica exchange moves to zero as only one ensemble is free at the maximum. Reducing the K/N ratio will generally imply a higher acceptance in the replica exchange moves as we can expect a higher number of free ensembles whose distributions have significant overlap. What gives the optimum number of workers is therefore a nontrivial question that we will further explore in Section 4. However, for case K < N we can maximize the effect of the replica exchange moves by taking the P_{RE} parameter as high as possible. In fact, we can simulate the effect of the limit $P_{\mathrm{RE}} \to 1$ without having to do an infinite number of replica exchange moves explicitly. This leads to an infinite swapping[13] version of our algorithm.

2.2. Infinite Swapping.

If in the previously described algorithm we take $P_{\mathrm{RE}} = 1 - \delta$, we will loop through steps 4−6 for many iterations ($n_{\mathrm{it}} = \sum_{n=0}^{\infty} n(1-\delta)^n\delta = 1/\delta$ in the limit $\delta \to 0$) before getting to step 7. When δ vanishes and n_{it} becomes infinitely large, we expect that all possible swaps will be executed an infinite number of times. Since the swaps obey detailed-balance between unoccupied ensembles, these will essentially sample the distribution of eq 3 (for the subset S^* of unoccupied ensembles). Hence, when the loop is exited, each possible permutation $\sigma \in S^*$ has been sampled $n_{\mathrm{it}} \times \rho(\sigma)/\sum_\sigma \rho(\sigma)$ times. By lumping all the times that the same permutation was sampled and normalizing by division with n_{it}, we simply sample all of the possible permutations in one go using fractional weights that sum up to 1. This is then the only sampling step, as the single update in step 3 can be skipped due to its negligible $1/n_{\mathrm{it}}$ weight.

The idea of doing an “infinite number” of swapping moves has been proposed before,[13−16] but here we give a different flavor to this approach by a convenient reformulation of the problem into permanents that allows us to beat the steep factorial scaling reported in earlier works.[13] The permanent formulation goes as follows. Suppose that after step 3, there are four free ensembles (we name them e_1, e_2, e_3, e_4) containing four states (s_1, s_2, s_3, s_4). Which state is in which ensemble after this step is irrelevant. We can now define a weight-matrix W

$$W = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline s_1 & W_{11} & W_{12} & W_{13} & W_{14} \\ s_2 & W_{21} & W_{22} & W_{23} & W_{24} \\ s_3 & W_{31} & W_{32} & W_{33} & W_{34} \\ s_4 & W_{41} & W_{42} & W_{43} & W_{44} \end{array}$$

where $W_{ij} \propto \rho_j(s_i)$. Essential to our approach is the computation of the permanent of the W-matrix, perm(W), and that of the $W\{ij\}$-matrices in which the row i and column j are removed.

The permanent of a matrix is similar to the determinant but without alternating signs. We can, henceforth, write perm(W) = $\sum_{j=1}^{4} W_{1j}\mathrm{perm}(W\{1j\})$. As the permanent of the 1 × 1 matrix is obviously equal to the single matrix value, the permanent of arbitrary dimension could in principle be solved recursively using this relation. Based on the permanents of W, we will construct a probability matrix P

$$P = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline s_1 & P_{11} & P_{12} & P_{13} & P_{14} \\ s_2 & P_{21} & P_{22} & P_{23} & P_{24} \\ s_3 & P_{31} & P_{32} & P_{33} & P_{34} \\ s_4 & P_{41} & P_{42} & P_{43} & P_{44} \end{array}$$

where P_{ij} is the chance to find state s_i in ensemble e_j. As for each permutation each state is in one ensemble and each ensemble contains one state, the P-matrix is bistochastic: both the columns and the rows sum up to 1. If we consider S_{ij}^* the set of permutations in which state s_i is in e_j, we can write $P_{ij} = \sum_{\sigma \in S_{ij}^*} \rho(\sigma)/\sum_{\sigma' \in S^*}\rho(\sigma')$. We can, however, also use the permanent representation in which

$$P_{ij} = \frac{W_{ij}\mathrm{perm}(W\{ij\})}{\mathrm{perm}(W)} \tag{5}$$

So far we have not won anything as computing the permanent via the recursive relation mentioned above has still the factorial scaling. The Gaussian elimination approach, which allows an order $O(n^3)$ computation for determinants of $n \times n$ matrices, will not work for permanents as only some but not all row and column operations have the same effect to a permanent as to a determinant. One can for instance swap rows and columns without changing the permanent. Multiplying a row by a nonzero scalar multiplies the permanent by the same scalar. Hence, this will not affect the P-matrix based on eq 5. Unlike the determinant, adding or subtracting to a row a scalar multiple of another row, an essential part of the Gaussian elimination method, does change the permanent. This makes the permanent computation of a large matrix excessively more expensive than the computation of a determinant. Yet, recent algorithms based on the Balasubramanian-Bax-Franklin-Glynn (BBFG) formula[17−20] scale as $O(2^n)$. This means that the computation of the full P-matrix scales as $O(2^n \times n^2)$, which seems still steep but is nevertheless a dramatic improvement compared to factorial scaling.

For our target time of 1 second, for instance, we could only run the algorithm up to $N = 7$ in the factorial approach, while we reach $N = 12$ in the BBFG method using a mid-to-high-end laptop (DELL XPS 15 with an Intel Core i7-8750H). If matrix size of $N = 20$ is the target, the BBFG method can perform a full P-matrix determination in ∼711 s, while it would take ∼15.3 × 10^6 years in the factorial approach. The BBFG method is the fastest completely general solution for the problem of computing a P-matrix from any W-matrix. For several algorithms, the W-matrix has special characteristics that can be exploited to further increase efficiency. For instance, if by shuffling the rows and columns the W-matrix can be made into a block form, where squared blocks at the diagonal have only zeros at their right and upper side, the permanent is equal to the product of the block's permanents. For instance, if $W_{14} = W_{24} = W_{34} = 0$, we have two blocks, 3 × 3 and 1 × 1. If $W_{13} = W_{14} = W_{23} = W_{24} = 0$, we can identify two blocks of 2 × 2, etc. Identification of blocks can hugely decrease the computation of a large permanent. Another speed-up can be made if all rows in the W-matrix are a sequence of ones followed by all zeros, or can be made into that form after the previously mentioned column and row operations. This makes an order $O(n^2)$ approach possible. We will further discuss this in Section 3.1.

The infinite swapping approach changes the aforementioned algorithm from step 3:

3 If the worker is finished with its MC move in a specific ensemble, the new state is accepted or rejected (but not yet sampled) according to eq 1. The ensemble is free.

4 Determine the W-matrix based on all unoccupied ensembles, calculate the P-matrix based on eq 5, and update all of the unoccupied ensembles by sampling all free states with the fractional probabilities corresponding to the columns in the P-matrix.

5 Pick randomly one of the free ensembles e_j.

6 Pick one of the available states $(s_1, s_2, \cdots)$ based on a weighted random selection in which state s_i has a probability of P_{ij} to be selected.

7 The worker is assigned to do a new standard move in ensemble e_j based on previous state s_i. Go to step 3.

3. APPLICATION: ∞RETIS

Replica Exchange Transition Interface Sampling (RETIS)[21,22] is a quantitative path sampling algorithm in which the sampled states are short molecular trajectories (paths) with certain start and end conditions, and a minimal progress condition. New paths are being generated by a Monte Carlo move in path space, such as the shooting move[23] in which a randomly selected phase point of the previous path is randomly modified and then integrated backward and forward in time by means of molecular dynamics (MD). The required minimal progress increases with the rank of the ensemble such that the final ensemble contains a reasonable fraction of transition trajectories. The start and end conditions, as well as the minimal progress, are administered by the crossings of interfaces $(\lambda_0, \lambda_1, \cdots, \lambda_M)$ with $\lambda_{k+1} > \lambda_k$, that can be viewed as nonintersecting hypersurfaces in phase space having a fixed value of the reaction coordinate. A MC move that generates a trial path not fulfilling the path ensemble's criteria is always rejected. RETIS defines different path ensembles based on the direction of the paths and the interface that has to be crossed, but all paths start by crossing λ_0 (near the reactant state/state A) and they end by either crossing λ_0 again or reaching the last interface λ_M (near the product state/state B). There is one special path ensemble, called $[0^-]$, that explores the left side of λ_0, the reactant well, while all other path ensembles, called $[k^+]$ with $k = 0, 1, \cdots M - 1$, start by moving to the right from λ_0 reaching at least λ_k.

A central concept in RETIS is the so-called overall crossing probability, the chance that a path that crosses λ_0 in the positive direction reaches λ_M without recrossing λ_0. It provides the rate of the process when multiplied with the flux through λ_0 (obtained from the path lengths in $[0^-]$ and $[0^+]$[22]) and is usually an extremely small number. The chance that any of the sampled paths in the $[0^+]$ path ensemble crosses λ_M is generally negligible, but a decent fraction of those (∼0.1−0.5) will cross λ_1 and some even λ_2. Likewise, paths in the $[k^+]$, $k > 0$, path ensembles have a much higher chance to cross λ_{k+1} than a $[0^+]$-path as they already cross λ_k. This leads to the calculation of M local conditional crossing probabilities, the chance to cross λ_{k+1} given λ_k was crossed for $k = 0,1, \cdots, M - 1$, whose product gives an exact expression for the overall crossing probability with an exponentially reduced CPU cost compared to MD.

The efficiency is further hugely improved by executing replica exchange moves between the path ensembles. These swaps are essentially cost-free since there is no need to simulate additional ensembles that are not already required. An accepted swapping move in RETIS provides new paths in two ensembles without the expense of having to do MD steps. The enhancement in efficiency is generally even larger than one would expect based on these arguments alone as path ensembles higher-up the barrier provide a similar effect as the high-temperature ensembles in parallel tempering. In addition, point exchange moves between the $[0^-]$ and $[0^+]$ ensembles are performed by exchanging the end and start points of these paths that are then continued by MD at the opposite site of the λ_0 interface.

While TIS[24] (without replica exchange) can run all path ensembles embarrassingly parallel, the RETIS algorithm increases the CPU-time efficiency but is difficult to parallelize and open source path sampling codes, like OpenPathSampling[25] and PyRETIS,[26] implement RETIS as a fully sequential algorithm. The path length distributions are generally broad with an increasing average path length as a function of the ensemble's rank. This becomes increasingly problematic the more ensembles you have as they all have to wait for the slowest ensemble. This means that while RETIS will give the best statistics per CPU-hour, it might not give the best statistics in wall-time. Our parallel scheme can effectively deal with the unequal CPU cost of the replicas, which allows us to increase the wall-time efficiency with no or minimal reduction in CPU-time efficiency. On the contrary, our method does not give an equal distribution of the CPU-time over the different ensembles nor an equal number of samples per ensemble, leading to a smarter distribution of CPU hours. The CPU efficiency therefore even seems to improve slightly.

3.1. *W*-Matrix in RETIS.

If there are $M + 1$ interfaces, λ_0, $\lambda_1, \cdots, \lambda_M$, there are also $N = M + 1$ ensembles, $[0^-]$, $[0^+]$, $[1^+]$, $\cdots$, $[(M - 1)^+]$. For K workers, the size of the W-matrix is, hence, either $(N - K + 1) \times (N - K + 1)$ or $(N - K) \times (N - K)$ as swappings are executed when 1 of the K workers is free, while the remaining $K - 1$ workers occupy path ensembles that are locked and do not participate in the swap. The smallest matrix occurs when one worker is occupying both $[0^-]$ and $[0^+]$ during the point exchange move, as described in the SI.

Paths can be represented by a sequence of time slices, the phase points visited by the MD trajectory. For a path of length $L + 1$, $X = (x_0, x_1, \cdots, x_L)$, the plain path probability density $\rho(X)$ is given by the probability of the initial phase point times the dynamical transition probabilities to go from one phase point to the next: $\rho(X) = \rho(x_0)\phi(x_0 \to x_1)\phi(x_1 \to x_2) \cdots \phi(x_{L-1} \to x_L)$. Here, the transition probabilities depend on the type of dynamics (deterministic, Langevin, Nosé-Hoover dynamics, etc). The weight of a path within a specific path ensemble $\rho_j(X)$ can be expressed as the plain path density times the indicator function $\mathbf{1}_{e_j}$ and possibly an additional weight function $w_j(X)$: $\rho_j(X) = \rho(X) \times \mathbf{1}_{e_j}(X) \times w_j(X)$. The indicator function equals 1 if path X belongs to ensemble e_j. Otherwise, it is 0. The additional weight function $w_j(X)$ is part of the high-acceptance protocol that is used in combination with the more recent path generation MC moves such as stone skipping[27] and wire fencing.[28] Using these "high-acceptance weights", nearly all of the CPU-intensive moves can be accepted as they are tuned to cancel the P_{gen}-terms in the Metropolis−Hastings scheme, eq 1, and the effect of the

Table 1. Results of the Three Model Systems Showing Crossing Probabilities (P_{cross}), Permeabilities (perm.), and Rates for Different Number of Workers (K)[a]

MSVS		two-channel system			double well with wire fencing		
K	$P_{\text{cross}}/10^{-50}$	K	$P_{\text{cross}}/\ 10^{-5}$	perm./10^{-6}	K	$P_{\text{cross}}/10^{-7}$	rate/10^{-7}
1	0.61 ± 0.33	1	1.52 ± 0.17	1.28 ± 0.14	1	5.91 ± 0.18	2.59 ± 0.07
5	1.47 ± 1.04	2	1.63 ± 0.24	1.37 ± 0.20	2	5.70 ± 0.13	2.51 ± 0.06
10	0.86 ± 0.51	3	1.52 ± 0.07	1.28 ± 0.06	3	5.57 ± 0.19	2.45 ± 0.08
15	0.68 ± 0.08	4	1.42 ± 0.10	1.19 ± 0.08	4	5.20 ± 0.30	2.34 ± 0.12
20	1.02 ± 0.13	5	1.40 ± 0.12	1.18 ± 0.10	5	5.05 ± 0.41	2.23 ± 0.18
25	1.02 ± 0.17	6	1.54 ± 0.06	1.30 ± 0.05	6	5.49 ± 0.29	2.42 ± 0.13
30	1.26 ± 0.24	7	1.48 ± 0.08	1.24 ± 0.07	7	4.99 ± 0.39	2.21 ± 0.17
35	1.05 ± 0.15	8	1.46 ± 0.08	1.23 ± 0.06	8	4.88 ± 0.43	2.15 ± 0.19
40	1.05 ± 0.14	9	1.42 ± 0.10	1.20 ± 0.08			
45	0.93 ± 0.09	10	1.44 ± 0.08	1.21 ± 0.07			
50	1.00 ± 0.07	11	1.41 ± 0.09	1.19 ± 0.08			
		12	1.30 ± 0.15	1.09 ± 0.12			
			Literature/Theoretical Result				
			1.23 ± 0.16[b]	1.06 ± 0.14[b]			2.79 ± 0.70[d]
						5.84 ± 0.13[e]	2.58 ± 0.06[e]
	1.00[a]		1.61[c]	1.37[c]		5.83[c]	2.58[c]

[a] All results are shown in dimensionless units. Errors are based on single standard deviations. Values shown in the lower part are a: exact result, b: ref 31, c: approximated value based on Kramers' theory (see the SI), d: ref 32, and e: ref 28.

nonphysical weights is undone in the analysis by weighting each sampled path with the inverse of $w_j(X)$.

While the path probability $\rho(s_i = X)$ is difficult to compute, determining $\mathbf{1}_j(s_i)$ and $w_j(s_i)$ is trivial. It is therefore a fortunate coincidence that we can replace $W_{ij} = \rho_j(s_i)$ with

$$W_{ij} = \mathbf{1}_{e_j}(s_i)w_j(s_i) \tag{6}$$

because the P-matrix does not change if we divide or multiply a row by the same number, as mentioned in Section 2. Except for $[0^-]$, all path ensembles have the same start and end conditions and only differ with respect to the interface crossing condition. A path that crosses interface λ_k automatically crosses all lower interfaces $\lambda_{l<k}$. Reversely, if the path does not cross λ_k, it will not cross any of the higher interfaces $\lambda_{l>k}$. This implies that if the columns of W_{ij} are ordered such that the first column (e_1) is the first available ensemble from the sequence ($[0^-]$, $[0^+]$, $[1^+]$, $\cdots$, $[(M-1)^+]$), the second column (e_2) is the second available ensemble, and so on, most rows will end with a series of zeros.

Reordering the rows with respect to the number of trailing zeros, almost always ensures that the W-matrix can be brought into a block form such that the permanent can be computed faster based on smaller matrices. In particular, if $[0^-]$ is part of the free ensembles, it will always form a 1 × 1 block as there is always one and no more than one available path that fits in this ensemble.

If high acceptance is not applied, we have $w_j(X) = 1$ and each row in the W-matrix (after separating the $[0^-]$ ensemble if it is part of the free ensembles) is a sequence of ones followed by all zeros. The W-matrix can hence be represented by an array (n_1, n_2, n_3, $\cdots$ n_n), where each integer n_i indicates the number of ones in row i. As we show in the SI, the permanent of such a W-matrix is simply the product of $(n_i + 1 - i)$: $\text{perm}(W) = \prod_i (n_i + 1 - i)$. Further, the P-matrix can be constructed from the following order $O(n^2)$ method.

The first step is to order the rows of the W-matrix such that $n_1 \le n_2 \le \cdots \le n_n$. We then fill in the P-matrix from top to bottom for each row using

$$P_{ij} = \begin{cases} 0, \text{ if } W_{ij} = 0 \\ \dfrac{1}{n_i + 1 - i}, \text{ if } W_{ij} = 1 \text{ and } [W_{(i-1)j} = 0 \text{ or } i = 1] \\ \left(\dfrac{n_{i-1} + 1 - i}{n_i + 1 - i}\right) P_{(i-1)j}, \text{ otherwise} \end{cases} \tag{7}$$

The approach is extremely fast and allows the computation of P-matrices from a large W-matrix, up to several thousands, within a second of CPU-time. The above method applies whenever the rows of the W-matrix can be transformed into sequence of ones followed by all zeros. Besides RETIS without high acceptance, this would apply to other MC methods like subset sampling[29] or umbrella sampling[30] with semi-infinite rectangular windows.

4. RESULTS AND DISCUSSION

To test our algorithms we ran three types of ∞RETIS simulations. First, a memoryless single variable stochastic (MSVS) process was simulated to mimic a RETIS simulation in which the average path length increases linearly with the rank of the ensemble. A "path" is created by drawing 2 random numbers where the first determines how much progress a path makes and the second determines the path length. These two outcomes are variable and depend on the rank of the ensemble such that the fictitious path in ensemble $[k^+]$ has a 0.1 probability to cross λ_{k+1} and has an average path length of approximately $k/10$ seconds (see Section 2). The worker is paused for a number of seconds equal to the path length before it can participate in replica exchange moves to mimic the time it would take to perform all of the necessary MD steps. While this artificial simulation allows us to investigate the potential strength of the method to tackle extremely rare events, it cannot reveal the effect of correlations between accepted paths when fast exploration of the reaction coordinate's orthogonal directions is crucial. To analyze this effect, we also ran a two-dimensional (2D) membrane permeation system with two

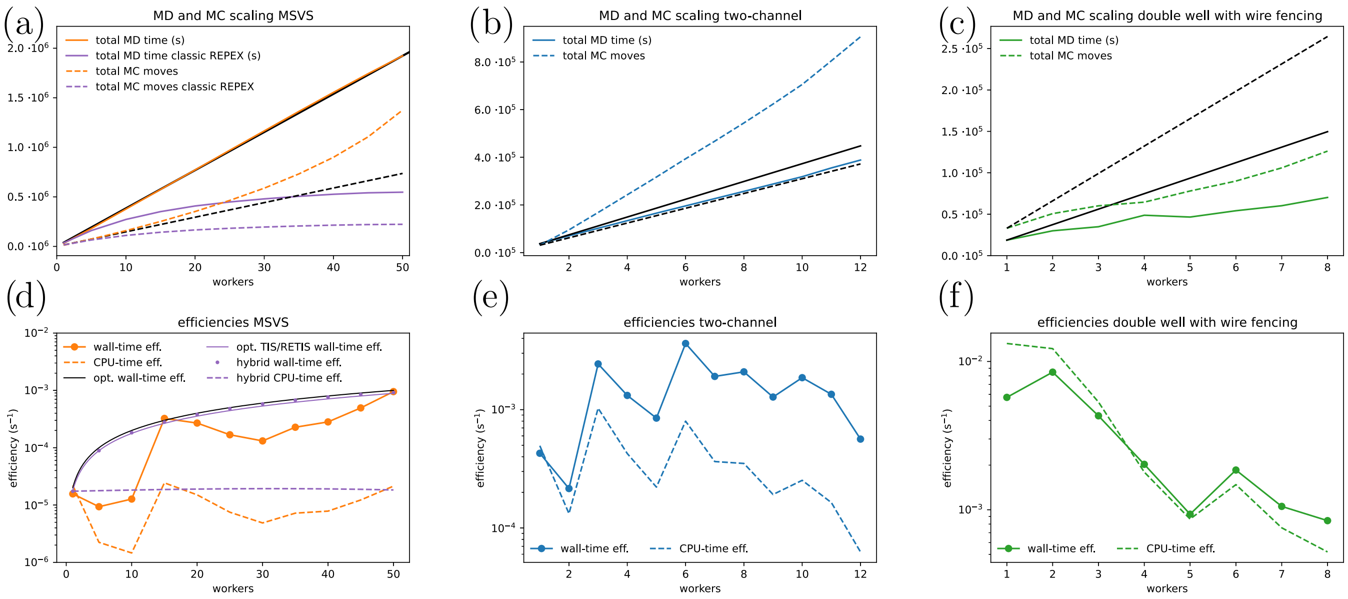


Figure 1. Average scaling of total MD time (cumulative time spent by all of the workers) (solid) and MC moves (dashed) (a–c) and wall-time (solid) and the CPU-time (dashed) efficiencies (d–f) for each number of workers. This is shown for the memoryless single variable stochastic (MSVS) process (a, d, orange), the two-channel system (b, e, blue), and the double well with wire fencing (c, f, green) simulations. Each of the data points is based on five independent simulations. For the scaling plots, the black lines are guides for linear scaling from the 1 worker data-point. The purple lines in the scaling plot for the MSVS simulations (a) show what the scaling would be if we had to wait for the slowest ensemble to finish for each MC move. The black line, purple line, purple dashed line, and points in the efficiency plot of the MSVS process (d) show the optimal, optimal TIS/RETIS, hybrid CPU-time efficiency, and hybrid wall-time efficiency, respectively, as computed in the SI.

slightly asymmetric channels.[31] Finally, to study our algorithm with a more generic W-matrix that needs to be solved via the BBFG formula, we also ran a set of underdamped Langevin simulations of a particle in a double-well potential[32] using the recent wire fencing algorithm with the high-acceptance protocol.[28] All simulation results were performed using five independent runs of 12 h. Errors were based on the standard deviations from these five simulations, except for the MSVS process, where a more reliable statistical error was desired for the comparison with analytical results. Here, block errors were determined on each of the five simulations based on the running average of the overall crossing probability. The block errors were finally combined to obtain the statistical error in the average of the five simulations.

4.1. Memoryless Single Variable Stochastic (MSVS) Process.

Table 1 reports the overall crossing probabilities and their statistical errors for a system with 50 interfaces and 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50 workers. All values are within a 50% deviation from the true value of 10^{-50} with the more accurate estimates for the simulations having a large number of workers. Also, the true value is within one standard deviation of the reported averages for 70% of the data points, as is expected from the standard Gaussian confidence intervals. Figure 1a shows the scaling of the MD time (solid lines) and number of MC moves (dashed lines) of the MSVS simulations (orange) compared to linear scaling (black) and the expected scaling for standard replica exchange (REPEX) in which ensembles are updated in cohort (purple).

Although the number of "MD steps" and MC moves quickly levels off to a nearly flat plateau in the standard approach due to workers being idle as they need to wait for the slowest worker, the replica exchange approach developed in this article shows a perfect linear scaling with respect to the MD time. The number of MC moves in the new method shows an even better than linear scaling due to the fact that the ensembles with shorter "path lengths" get simulated relatively more often with more workers, resulting in more MC moves per second. This in itself does not necessarily mean that the simulations converge much faster because the additional computational effort may not be targeted to the sampling where it is needed. If we neglect the fact that path ensemble simulations are correlated via the replica exchange moves, we can write that the relative error in overall crossing probability ϵ follows from the relative errors in each path ensemble ϵ_i via: $\epsilon^2 = \sum_i \epsilon_i^2$. It is henceforth clear that additional computational power should not aim to lower the error in a few path ensembles that were already low compared to other path ensembles. We therefore measure the effectiveness of the additional workers by calculating computational efficiencies. The efficiency of a specific computational method is here defined as the inverse computer time, CPU- or wall-time, to obtain an overall relative error equal to 1: $\epsilon = 1$.

In Figure 1d, the efficiencies based on wall-time (solid) and CPU-time (dashed) are plotted for the MSVS process. These plots depend on the ability of computing reliable statistical errors in the overall crossing probability that is an extremely small number, 10^{-50}. The somewhat fluctuating behavior of these curves should hence be viewed as statistical noise as the confidence interval of these efficiencies depends on the statistical error of this error. Despite that, clear trends can be observed in which the CPU-time efficiency is more or less constant within statistical fluctuations, while the wall-time efficiency shows an upward trend. If we neglect the effect of replica exchange moves on the efficiency, we can relate these numerical results with theoretical ones[22,33] for any possible division of a fixed total CPU-time over the different ensembles. A common sense approach would be to aim for the same error ϵ_i in each ensemble (which implies doing the same number of MC moves per ensemble) or to divide the total CPU-time evenly over the ensembles. These two strategies correspond to the case $K = 1$ or standard RETIS and $K = N$ or standard TIS, respectively. Ref 33 showed that these two strategies provide the same efficiency, and in the SI, we derive that this leads to a wall-time efficiency as a function of the number of workers (K) equal to $K/56250$, which is denoted by the continuous purple line in Figure 1d (the hypothetical wall-time efficiency for a parallel simulation that uses the CPU hours equally efficient as TIS and serial RETIS, see eq 50 in the SI). The optimum division, however, would give a slightly better wall-time efficiency equal to $K/50{,}000$ which is the continuous black line in this figure. Also shown in Figure 1d are the expected theoretical efficiencies based on the numerical distribution of MC moves in each ensemble. This hybrid numerical/theoretical result is shown by the small purple dots and the dashed purple line. This shows that ∞RETIS, at least for a system in which the path length grows linearly with the ensemble's rank, naturally provides a division of the computational resources that is even better than TIS ($K = N$) or RETIS ($K = 1$). Yet, due to statistical inaccuracies this is only evident for the $K = 15$ case if we base our analysis on the numerical block errors.

In any case, it shows that the possible concern that additional CPU resources in parallel ∞RETIS runs may not be properly targeted due to oversampling of the ensembles with the shorter paths is unfounded. On the contrary, even the CPU efficiency seems to improve slightly compared to TIS and RETIS that have the same CPU efficiency. This is maybe not so surprising since the division of CPU hours over the different ensembles in ∞RETIS is somewhat in between the divisions what one gets with TIS and RETIS, and the optimum[33] is also in between TIS and RETIS. In addition, we leave the number of interfaces and their locations unchanged in our analysis. However, the flexibility of ∞RETIS makes it very easy to add additional interfaces. By placing a higher density of interfaces high on the barrier, it is also possible to target more CPU to the expensive ensembles. This higher density has the additional benefit that the local crossing probabilities in this area are increased so that fewer paths are needed to calculate them.

Coming back to Figure 1d, we see that the best wall-time efficiency is obtained for the case $K = N$, which is essentially equivalent of running independent TIS simulations (i.e., without doing any replica exchange moves). We do not expect this to apply to more complex systems where the replica exchange move is a proven weapon for efficient sampling.

4.2. Two-Channel Simulations.

In the middle column of Table 1, we report the calculated crossing probabilities and permeabilities for five simulations for every number of workers. All simulations are somewhat higher, though still in good agreement with the previous simulation from ref 31. We also evaluated the approximate result based on Kramers' theory (see the SI), which seems to confirm the results obtained in this paper. Figure 1b shows the scaling of the MD time (solid lines) and number of MC moves (dashed lines) of the two-channel simulations (blue) compared to linear scaling (black). We see a slightly worse than linear scaling of the MD time, which might just be due to a small positive fluctuation of the 1 worker data-point. We also see a similar more than linear scaling in the number of MC moves as with the MSVS simulations, for the same reason. In Figure 1e, the efficiencies based on wall-time (solid) and CPU-time (dashed) are plotted for the two-channel system. The CPU-time efficiency is more or less flat until 8 workers after which it starts to drop off. The wall-time efficiency shows an upward trend until 10 workers after which it starts to drop off as well. We assign this drop to the reduction of replica exchange moves which is an essential aspect for sampling this system efficiently.[31] This is tangible from Figure S1 in the SI where we plot fraction of trajectories, passing through λ_{M-1}, that are in the lower barrier channel. While from the average fraction it still looks like the simulations sampled both channels for any number of workers, 4 out of the 5 simulations in the $K = N = 12$ case solely visited one of the two channels. This is in agreement with previous TIS results.[31] The $K = 11$ case already provides a dramatic improvement, but is still expected to be suboptimal due to the relatively low frequency of replica exchange moves compared to $K < 11$. From this 2D system, it would indicate that having $K \approx N/2$ is a safe bet for optimum efficiency.

4.3. Double-Well 1D Barrier Using Wire Fencing.

In the right column of Table 1, we report the calculated crossing probabilities and rates for the underdamped Langevin particle in the 1D double-well potential. All simulations are in reasonable agreement with each other and the results of refs 28 and 32, as well as the approximate value based on Kramers' theory. However, while these results confirm the soundness of the method, the scaling and efficiency are less convincing. Figure 1c shows a significantly worse than linear scaling. On further inspection, we found the average time per MC move was significantly smaller than our infinite swapping time (1 s) when the simulation was run with more than two workers. This results in a bottleneck on how many MC moves can be started per second, which is the reason for the observed bad scaling. It still is slightly positive instead of flat as the infinite swapping procedure becomes quicker with more workers due to the smaller W-matrix. The same bottleneck can be seen in Figure 1f where both efficiencies plummet with more than two workers. The reported scaling deficiency is of little significance for actual molecular systems where the creation of a full path takes minutes to hours rather than subseconds.

5. CONCLUSIONS

We developed a new generic replica exchange method that is able to effectively deal with MC moves with varying CPU costs, for instance, due to the algorithmic complexity of the MC moves. An essential aspect of the method is that the number of workers, who execute the ensemble's specific MC moves in parallel, is less than the number of ensembles. Once a worker is finished with its move, replica exchange moves are carried out solely between those ensembles that are not occupied by a worker. This implies that the ensembles are updated at irregular intervals and a different number of MC moves will be executed for each ensemble. As a result, the conceptual viewpoint in which the set of replicas are viewed as a single superstate is no longer valid and the existence of some kind of detailed-balance relation is no longer trivial. To prove the exactness of our approach, we introduced new conceptual views on the replica exchange methodology that is different from the common superstate principle. Instead, we show that the distributions in the new approach are conserved for each ensemble individually via a twisted detailed-balance relation in which the other ensembles constitute an environment that is potentially actively involved in the MC move of the considered ensemble. In addition, the method can be combined with an infinite swapping approach without factorial scaling based on a mathematical reformulation using permanents.

We applied the novel replica exchange technique on a path sampling algorithm, RETIS, which is a prototype of algorithm where the costs for a Monte Carlo move can vary enormously. The resulting new path sampling algorithm, coined ∞RETIS, was thereafter tested on three model systems. The results of these simulations show that the number of MD steps increases linearly with the number of workers invoked as long as the ensemble's MC move has a lower computational cost than the replica exchange move carried out by the scheduler. The number of executed MC moves shows an even better than linear scaling. Moreover, the efficiency increases linearly with the number of workers for a low-dimensional system in which the replica exchange move has little effect, while it has an optimum in more complex systems as the number of successful replica exchange moves decreases when the number of workers is close to the number of ensembles.

In summary, the replica exchange method discussed in this paper has a clear potential to accelerate present path sampling simulations, but can also be combined with many other complex algorithms including those that are yet to be invented. With the continuing trend to run progressively more massively parallel computing jobs, our algorithm is likely to gain importance. In the case of ∞RETIS, we envision many applications in the fields of nucleation, self-assembly, chemical reactions, enzymatic catalysis, membrane permeation, protein folding, and other conformational changes in biomolecules. The ∞RETIS method and the noncohort replica exchange method, in general, are therefore expected to open up new avenues in the field of molecular simulations and maybe even beyond.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpca.2c06004.

> Details of the simulations, proof that the replica exchange method with cost unbalanced replicas conserves the equilibrium distribution at the individual ensemble level; $O(n^2)$ algorithm for computing the P-matrix from a W-matrix for the case that the W-matrix consists of rows having a series with ones, followed by zeros; derivations of the theoretical results on the crossing probabilities, rate constant and permeability via Kramers' theory that are shown in Table 1; discussion on the computational efficiencies with the derivations for the most optimal efficiencies; some additional simulation results on the relative transition probabilities through the lower and higher barrier channel for the two-channel system; our current code with which the model system calculations were done and the project started for advanced simulations (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Titus S. van Erp − *Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway;* orcid.org/0000-0001-6600-6657; Email: titus.van.erp@ntnu.no

Authors

Sander Roet − *Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway;* orcid.org/0000-0003-0732-545X

Daniel T. Zhang − *Department of Chemistry, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpca.2c06004

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Raffaela Cabriolu and Anders Lervik for their critical comments. They acknowledge funding from the Research Council of Norway through FRINATEK Project No. 275506.

■ REFERENCES

(1) Metropolis, N.; Rosenbluth, A.; Rosenbluth, M.; Teller, A.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087−1092.

(2) Hastings, W. K. Monte-Carlo Sampling methods using Markov chains and their Applications. *Biometrika* **1970**, *57*, 97−109.

(3) Swendsen, R. H.; Wang, J. S. Replica Monte-Carlo simulation of spin-glasses. *Phys. Rev. Lett.* **1986**, *57*, 2607−2609.

(4) Marinari, E.; Parisi, G. Simulated tempering - a new Monte-Carlo scheme. *Europhys. Lett.* **1992**, *19*, 451−458.

(5) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141−151.

(6) Siepmann, J. I.; Frenkel, D. Configurational bias Monte-Carlo - A new sampling scheme for flexible chains. *Mol. Phys.* **1992**, *75*, 59−70.

(7) Vlugt, T. J. H.; Krishna, R.; Smit, B. Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite. *J. Phys. Chem. B* **1999**, *103*, 1102−1118.

(8) Frenkel, D.; Smit, B.*Understanding Molecular Simulations from Algorithms to Applications*; Academic Press: San Diego, California, U.S.A., 2002.

(9) Dellago, C.; Bolhuis, P. G.; Csajka, F. S.; Chandler, D. Transition path sampling and the calculation of rate constants. *J. Chem. Phys.* **1998**, *108*, 1964.

(10) Swendsen, R. H.; Wang, J.-S. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.* **1987**, *58*, 86−88.

(11) Peters, E. A. J. F.; de With, G. Rejection-free Monte Carlo sampling for general potentials. *Phys. Rev. E* **2012**, *85*, No. 026703.

(12) Michel, M.; Kapfer, S. C.; Krauth, W. Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps. *J. Chem. Phys.* **2014**, *140*, No. 054116.

(13) Plattner, N.; Doll, J. D.; Dupuis, P.; Wang, H.; Liu, Y.; Gubernatis, J. E. An infinite swapping approach to the rare-event sampling problem. *J. Chem. Phys.* **2011**, *135*, 134111.

(14) Plattner, N.; Doll, J. D.; Meuwly, M. Overcoming the Rare Event Sampling Problem in Biological Systems with Infinite Swapping. *J. Chem. Theory Comput.* **2013**, *9*, 4215−4224.

(15) Yu, T.-Q.; Lu, J.; Abrams, C. F.; Vanden-Eijnden, E. Multiscale implementation of infinite-swap replica exchange molecular dynamics. *Proc. Natl. Acad. Sci. U.S.A* **2016**, *113*, 11744−11749.

(16) Lu, J.; Vanden-Eijnden, E. Methodological and Computational Aspects of Parallel Tempering Methods in the Infinite Swapping Limit. *J. Stat. Phys.* **2019**, *174*, 715−733.

(17) Balasubramanian, K.Combinatorics and Diagonals of Matrices. Ph.D. thesis, Loyola College, Madras, India, 1980.

(18) Bax, E.Finite-difference Algorithms for Counting Problems. Ph.D. thesis, California Institute of Technology: Pasadena, United States, 1998.

(19) Bax, E.; Franklin, J.*A finite-Difference Sieve to Compute the Permanent*; CalTech-CS-TR1996; pp 96−04.

(20) Glynn, D. G. The permanent of a square matrix. *Eur. J. Comb.* **2010**, *31*, 1887−1891.

(21) van Erp, T. S. Reaction rate calculation by parallel path swapping. *Phys. Rev. Lett.* **2007**, *98*, No. 268301.

(22) Cabriolu, R.; Refsnes, K. M. S.; Bolhuis, P. G.; van Erp, T. S. Foundations and latest advances in replica exchange transition interface sampling. *J. Chem. Phys.* **2017**, *147*, 152722.

(23) Dellago, C.; Bolhuis, P. G.; Chandler, D. Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements. *J. Chem. Phys.* **1998**, *108*, 9236−9245.

(24) van Erp, T. S.; Moroni, D.; Bolhuis, P. G. A Novel Path Sampling Method for the Sampling of Rate Constants. *J. Chem. Phys.* **2003**, *118*, 7762−7774.

(25) Swenson, D. W. H.; Prinz, J.-H.; Noe, F.; Chodera, J. D.; Bolhuis, P. G. OpenPathSampling: A Python Framework for Path Sampling Simulations. 2. Building and Customizing Path Ensembles and Sample Schemes. *J. Chem. Theory Comput.* **2019**, *15*, 837−856.

(26) Riccardi, E.; Lervik, A.; Roet, S.; Aarøen, O.; van Erp, T. S. PyRETIS 2: An improbability drive for rare events. *J. Comput. Chem.* **2020**, *41*, 370−377.

(27) Riccardi, E.; Dahlen, O.; van Erp, T. S. Fast decorrelating Monte Carlo moves for efficient path sampling. *J. Phys. Chem. Lett.* **2017**, *8*, 4456−4460.

(28) Zhang, D. T.; Riccardi, E.; van Erp, T. S.Enhanced path sampling using subtrajectory Monte Carlo moves. e-Print archive. https://doi.org/10.48550/arXiv.2210.07026.

(29) Au, S.-K.; Beck, J. L. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Eng. Mech.* **2001**, *16*, 263−277.

(30) Torrie, G.; Valleau, J. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella samping. *J. Comput. Phys.* **1977**, *23*, 187.

(31) Ghysels, A.; Roet, S.; Davoudi, S.; van Erp, T. S. Exact non-Markovian permeability from rare event simulations. *Phys. Rev. Res.* **2021**, *3*, No. 033068.

(32) Van Erp, T. S.Dynamical Rare Event Simulation Techniques for Equilibrium and Nonequilibrium Systems. In *Advances in Chemical Physics*; John Willey & Sons Inc., 2012; Vol. *151*, p 27.

(33) van Erp, T. S. Efficiency analysis of reaction rate calculation methods using analytical models I: The two-dimensional sharp barrier. *J. Chem. Phys.* **2006**, *125*, 174106.