THESIS

WEIGHTED ENSEMBLE: PRACTICAL VARIANCE REDUCTION TECHNIQUES

Submitted by

Mats S. Johnson

Department of Mathematics

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2022

Master's Committee:

Advisor: David Aristoff

Margaret Cheney
Diego Krapf
Olivier Pinaud

ABSTRACT

WEIGHTED ENSEMBLE: PRACTICAL VARIANCE REDUCTION TECHNIQUES

Computational biology and chemistry is proliferated with important constants that are desirable for researchers. The *mean-first-passage time* (MFPT) is one such important quantity of interest and is pursued in molecular dynamics simulating protein conformational changes, enzyme reaction rates, and more. Often, the simulation of these processes is hindered by such events having prohibitively small probability of observation.

For these *rare-events*, direct estimation by Monte Carlo techniques can be burdened by high variance. We analyzed an importance sampling splitting and killing algorithm called *weighted ensemble* to address these drawbacks. We used weighted ensemble in the context of a stochastic process governed by a Markov chain $(X_t)_{t\geq 0}$ with steady state distribution μ to estimate the MFPT. Weighted ensemble works by partitioning the state space into bins and replicating trajectories in an advantageous and unbiased manner. By introducing a recycling boundary condition, we improved the convergence of our problem to steady state and made use of the *Hill relation* to estimate the MFPT. This change allows relevant conclusions to be drawn from simulations that are much shorter in time scale when compared to direct estimation of the MFPT.

After defining the weighted ensemble algorithm, we decomposed the variance of the weighted ensemble estimator in a way that admits simple optimization problems to be posed. We also defined the relevant coordinate, *the flux-discrepancy function*, for splitting trajectories in the weighted ensemble method and its associated variance function. When combined with the variance formulas, the flux-discrepancy function was used to guide parameter choices for choosing binning and replication strategies for the weighted ensemble algorithm.

Finally, we discuss practical implementations of solutions to the aforementioned optimization problems and demonstrate their effectiveness in the context of a toy problem. We found that the

techniques we presented offered a significant variance reduction over a naive implementation of weighted ensemble that is commonly used in practice and direct simulation by naive Monte Carlo. The optimizations we presented correspond to a reduced computational cost for implementing the weighted ensemble algorithm. We further found that our results were applicable even in the case of limited resources which makes their application even more appealing.

ACKNOWLEDGEMENTS

I'd like to thank my adviser Dr. David Aristoff for his guidance and insightful discussions in pursuit of this thesis.

DEDICATION

This thesis is dedicated to my family whose continued belief and support mean the world to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

With the surge of computational power, the scale of systems that can be modeled on a computer is growing significantly. Many of these systems are stochastic processes and are prevalent in studying molecular dynamics. The applications of molecular dynamics are wide-ranging, for instance, charge hopping can be viewed as a random walk [1, 2], protein-association reactions [3] or protein conformational changes [4–7], and the dissociation rate for protein-ligands [8]. To obtain meaningful results, researchers require accurate simulations of the dynamics that underlie these systems which can be extremely complex [9, 10]. Many of these processes occur in or near steady state due to the homeostasis of the biological setting that governs them. Though there are many different quantities of interest in these systems, the *mean-first-passage time* (MFPT) is one of particular interest and will be the focus of this thesis.

MFPT calculations are desirable throughout biochemistry. For example, the Michaelis-Menton equation gives the rate of enzymatic reactions as the inverse MFPT [11] and protein conformational changes are often cast as MFPT problems [12, 13]. Another common problem is in the context of charged ions and attempting to compute the transition rate from between potential wells [14, 15].

Generally, the stochastic process in these problems can be framed as a Markov chain $(X_t)_{t\geq 0}$ with kernel K and steady state distribution p. The MFPT can be considered as a transition from a source set A to a target set B. It is necessary to define ρ_A, the initial distribution in A, as this along with B will define the MFPT for a choice in dynamics. Let τ_B denote the first passage time to B (see Figure 1.1). The MFPT can then be found from

$$\text{MFPT of } X(t) \text{ from } A \text{ to } B = \mathbb{E}[\tau_B \mid X_0 \sim \rho_A] \coloneqq \mathbb{E}^{\rho_A}[\tau_B].$$

Monte Carlo methods are one way of approaching the problem of estimating the probability of events. *Naive Monte Carlo* methods involve independently simulating the dynamics many times

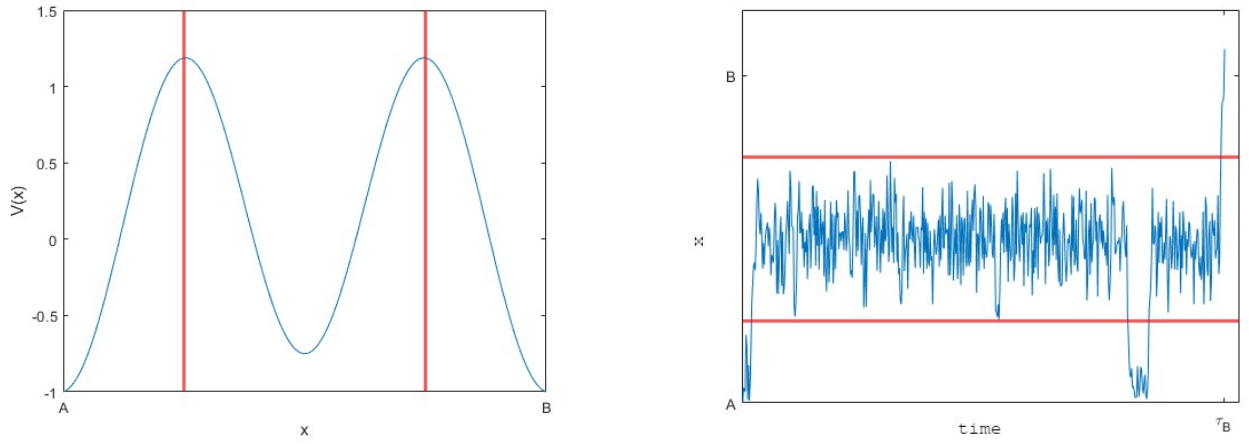


Figure 1.1: Left: A three well potential. Right: Evolution by Brownian motion of a particle starting at $x = A$. The first passage time τ_B denotes the first time $t > 0$ where $x \geq B$. Red lines indicate the the maxima of $V(x)$.

and taking an average of the results. For example, in estimating the MFPT for a charged ion to escape a potential well, a naive Monte Carlo simulation would initialize a particle with respect to ρ_A and count the the time it takes the particle to escape. The estimation would be obtained by averaging the results of these simulations across the number of trials.

More precisely, for a state space Ω, a random variable X with probability distribution $p(X)$, and a bounded real valued function f, N samples $\{X_1, X_2, \ldots, X_N\}$ are drawn according to p. An estimator of $\mathbb{E}[f(X)]$ can be obtained by:

$$\mathbb{E}[f(X)] = \int_\Omega f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i) = \theta_{MC}. \tag{1.1}$$

From the law of large numbers, as $N \to \infty$ the estimator $\theta_{MC} \xrightarrow{a.s.} \mathbb{E}[f(X)]$. Naive Monte Carlo remains useful today though implicit in its construction is the assumption that p may be efficiently sampled.

This assumption is not valid in many of the practices outlined earlier. Indeed, many significant events occur with such small probabilities that direct observation by brute-force is not feasible. Such "rare events" are still important in understanding biochemical problems such as protein folding [16] or ion transfers with a high activation barrier [17]. *Importance sampling* can be used

to generate samples of the target distribution p more effectively, through the proposal of a new distribution q. This introduces a new random variable $f(X)p(X)/q(X)$ in the following way

$$\mathbb{E}^p[f(X)] = \int_\Omega f(x)p(x)dx = \int_\Omega f(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}^q\left[f(X)\frac{p(X)}{q(X)}\right], \tag{1.2}$$

where the superscript denotes the probability distribution of X. A new estimator can then be defined as

$$\mathbb{E}^q\left[f(X)\frac{p(X)}{q(X)}\right] \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i)\frac{p(X_i)}{q(X_i)} = \theta_{IS}, \tag{1.3}$$

and naive Monte Carlo techniques can be used with q as the new distribution.

Choosing q requires some care; for instance, $q(X) \neq 0$ when $p(X) \neq 0$. The variance of θ_{IS} is

$$\mathrm{Var}^q\left[f(X)\frac{p(X)}{q(X)}\right] = \int_\Omega \left(f(x)\frac{p(x)}{q(x)}\right)^2 q(x)dx - \mathbb{E}^p[f(X)]. \tag{1.4}$$

Changing the distribution (and random variable) results in a new variance with the difference in variance given by

$$\mathrm{Var}^p[f(X)] - \mathrm{Var}^q\left[f(X)\frac{p(X)}{q(X)}\right] = \int_\Omega f^2(x)q(x)\left(1 - \frac{p(x)}{q(x)}\right)dx. \tag{1.5}$$

Intuitively, q should be chosen such that the likelihood ratio p/q is greater than small where $|f(X)|p(X)$ is large and large when $|f(X)|p(X)$ is small. Such a choice of q will result in more samples being drawn from important points in the state space. More significantly, this also results in a variance reduction compared to naive Monte Carlo. Importantly, with $q(X) = f(X)p(X)/\mathbb{E}^p[f(X)]$, by (1.4) it can be seen that θ_{IS} is a zero variance estimator.

For a Markov chain $(X_t)_{t\geq 0}$ with stationary distribution μ, estimates of $\mu(f) = \int_\Omega f(x)\mu(x)\,dx$ can be obtained by the trajectory average

$$\mu(f) \approx \frac{1}{T}\sum_{t=0}^{T-1} f(X_t).$$

Such an estimate can be obtained through a variety of Markov Chain Monte Carlo (MCMC) algorithms such as Metropolis-Hastings [18, 19] or Gibbs Sampling [20]. However, MCMC still does not perform well when estimating the probability of rare sets as estimating the MFPT requires prohibitively long simulation times.

Weighted ensemble is an importance sampling method that works by simulating and splitting trajectories. It is useful in estimating the average of some observable with respect to the steady state distribution of a Markov chain. Weighted ensemble works by employing weighted particles that evolve via the underlying Markov kernel. Periodically, the particles are divided into bins and resampled relative to their weights. Following the motivations of importance sampling, the choice of bins and resampling should encourage more particles in important regions of the state space. Weighted ensemble can exhibit a dramatic reduction in variance when compared to naive or direct Monte Carlo.

Weighted ensemble is one of numerous path sampling approaches that have grown in popularity due to their efficiency in simulating rare events. One of the most important uses of weighted ensemble is to estimate the MFPT for a random process to transition between two states A and B. Other approaches such as Adaptive Multi-level Splitting [21–23], Sequential Monte Carlo [24,25], and Markov State Modeling [26–28] are also used in estimating the MFPT. However, weighted ensemble has some key features that make it an attractive choice. Weighted ensemble can be used to define an asymptotically unbiased estimator of the MFTP whose variance does not explode over long time estimates. Weighted ensemble is also highly parallelizable and easily implemented due to the WESTPA package [29–31] that is available.

In the context of rare events, transitions between A and B are also rare and direct simulation is not tractable. Using weighted ensemble and imposing recycling boundary conditions, the MFPT can be expressed in terms of the inverse of the steady state flux into B by the *Hill relation* [32,33]. The Hill relation will allow accurate estimates of the MFPT using paths that are significantly shorter than the true MFPT.

The rest of the thesis is ordered as follows. In Chapter 2, we first give an overview of weighted ensemble in the context of computing the MFPT. In Chapter 3, we then give a detailed mathematical description of the weighted ensemble algorithm and derive some key properties. Chapter 4 finds formulas for the variance and discusses minimization strategies. Finally, Chapter 5 discusses optimization techniques and Chapter 6 demonstrates the effectiveness of these strategies when compared to direct Monte Carlo and traditionally implemented weighted ensemble in the context of a 2D toy problem.

Chapter 2

Weighted Ensemble Introduction

2.1 Weighted Ensemble Framework

Weighted Ensemble is a rare-event sampling method based on the splitting and merging of independently evolving particles. We assign each particle a positive weight such that the total weight is 1. During the splitting and merging process, new weights are chosen so that the total weight remains constant in time. Herein, we assume the Markov kernel K is uniformly geometrically ergodic with respect to its stationary distribution μ. Weighted ensemble will be used to estimate the steady-state average of a bounded real-valued function or observable f.

We will estimate $\int f \, d\mu$ by computing the weighted sum of f on the ensemble of particles at each time point $t \geq 0$. This estimation relies only on information that is available at time t. This means that weighted ensemble does not need to store the entire trajectory of a particle, just its current position and weight.

During the evolution or *mutation* step, we allow particles to independently evolve via the underlying Markov kernel K. In practice, as another measure to minimize the variance, the mutation step evolves the particles from time t to time $t + \Delta t$. This makes K a Δt-skeleton of the underlying diffusion process. Only the positions of the particles are updated in this step, the weights remain fixed. The integrator time-step will only advance during the mutation step.

In the resampling process or *selection* step, we select particles to copy for the next evolution step. We will refer to the particles before selection as *parents* and those after selection as *children*. Each parent may have zero or many children but each child will only have one parent. The children's weight and position will depend on its parent. However, the weights are chosen to conserve the total weight and the evolution of the children will still be independent.

We further require a set of bins $\mathcal{B}$ that define a partition of the state space. These bins may be defined in an initialization step or at a particular time t. As $\mathcal{B}$ is a partition, each particle necessarily

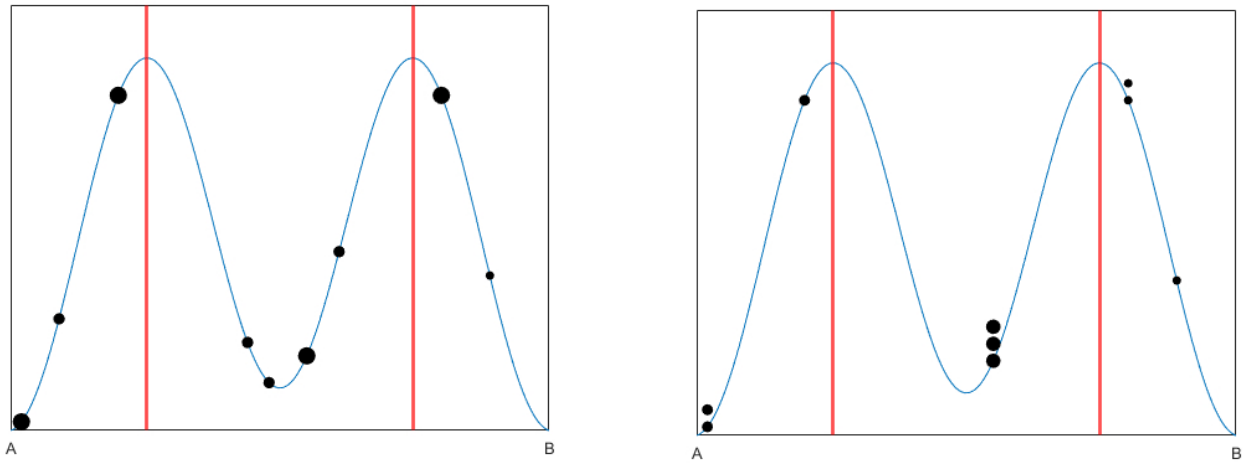


Figure 2.1: For the potential in Figure 1.1, the particles are represented by the black dots with size proportional to their weight. Here, the red lines also denote the division between bins. Left: Before selection, the ensemble of parents. Right: After selection with the same number of children in each bin. The new weights of the children are represented by the size of the dots. See Chapter 3 for a more detailed description.

belongs to some bin $u \in \mathcal{B}$. For each bin, let $N_t(u)$ denote the number of children in bin u at time t. This allocation of particles may be chosen independently of the number of parents in bin u with the following provisions:

1. Unoccupied bins have zero children.

2. Occupied bins have at least one child.

3. The number of children in each bin is chosen such that the total number of particles remains constant in time.

Children are selected from the parents in bin u proportional to the weights of the parents. This selection can performed by a number of different processes. For instance, in Chapter 6, copies are chosen via residual sampling. As any occupied bin u prior to selection will remain occupied following selection, the process of resampling may also be thought of as splitting or merging the trajectories in u.

2.2 Mean First Passage Time

Recall, we are interested in computing the mean first passage time (MFPT) between a metastable source, A, and a sink, B, for a Markov process $X(t)$ with steady state distribution μ. The relevant function in computing the mean first passage time from A to B is the indicator function of the target set:

$$\mathbb{1}_B(X(t)) = \begin{cases} 1 & X(t) \in B \\ 0 & X(t) \notin B \end{cases}. \tag{2.1}$$

The MFPT is affected only by the sets A and B, the distribution of particles ρ_A, and how the particles evolve in time. Particles that reach B are held there until the end of the mutation step and are recycled according to ρ_A at the end of the resampling time Δt. Recycling in this way ensures all flux into B is counted though it does introduce a negligible bias to the Hill relation. This bias can be avoided in practice if the flux is counted at the end of each integrator step and recycling is done immediately as a particle enters B.

The addition of a recycling boundary condition changes $X(t)$ into an irreversible process. The motivation for recycling is two-fold, first it allows the MFPT to be calculated by the *Hill relation* and it causes the system to converge to steady state more quickly. For the latter, consider a simple 3-state system with a transition matrix K given by

$$K = \begin{bmatrix} .999 & .001 & 0 \\ .999 & 0 & .001 \\ 0 & .001 & .999 \end{bmatrix}. \tag{2.2}$$

Let A be the first state and B the third state. The MFPT from A to B is then approximately 10^6 as two successive transitions of probability 10^{-3} are required. However, the convergence of the system to steady-state will be extremely slow using this kernel. To illustrate this, we consider the eigenvalues of K. K is a transition matrix so 1 is an eigenvalue; let $\lambda_2 = |.999|$ denote the eigenvalue of K with the second largest magnitude. For an initial distribution of particles $v \in \mathbb{R}^3$

and $t > 0$, convergence to the equilibrium distribution follows

$$\upsilon K^t \approx \mu + \lambda_2^t v_\upsilon = \mu + .999^t v_\upsilon. \tag{2.3}$$

From (2.3), it is clear that convergence will be slow.

Recycling modifies (2.2) in the following way

$$\tilde{K} = \begin{bmatrix} .999 & .001 & 0 \\ .999 & 0 & .001 \\ 1 & 0 & 0 \end{bmatrix}. \tag{2.4}$$

The eigenvalue of $\tilde{K}$ with the second largest magnitude is now $\tilde{\lambda}_2 = |-.001|$. For $t > 0$, convergence to the steady state distribution now follows

$$\upsilon K^t \approx \mu + \lambda_2^t v_\upsilon = \mu + (-.001)^t v_\upsilon,$$

which will be much faster than (2.3). This example is a single illustration of a simple problem but the principle that recycling improves convergence time holds true in general [34–38].

With these recycling boundary conditions, the Hill relation gives an expression for the MFPT in terms of the inverse of the steady-state flux into B. Recall τ_B denotes the first passage time to B and let N_t denote the number of arrivals in B by time t. The Hill relation then gives:

$$\mathbb{E}\left[\tau_B | X(0) \sim \rho_A\right] = \left(\frac{\mathbb{E}[N_t | X(0) \sim \mu]}{t}\right)^{-1} \tag{2.5}$$

The estimate from (2.5) is useful in reducing the overall simulation time. The intuition for this reduction in computational time is due to shifting the observable from directly computing the MFPT to instead computing the flux into B. The former requires simulations at least as long as the MFPT on average which are quite long due to the low probability of transitioning from A to B.

In the latter case, provided we are in steady state, the simulation time will be significantly shorter to achieve the same result. Furthermore, as discussed above, the time to approach steady state can be much smaller than the MFPT. To see this, note that (2.5) implies that the probability of a transition occurring in a finite time interval Δ_t is simply $\Delta_t / \mathbb{E}^{\rho_A}[\tau_B]$. The significance of (2.5) is that Δ_t may be significantly smaller than the MFPT as the expression is true for any $t > 0$. For systems with a long MFPT, $\left(\mathbb{E}^{\rho_A}[\tau_B]\right)^{-1}$ is extremely small which necessitates weighted ensemble (or another importance sampling technique) to accurately estimate. One immediate concern is whether the computation time to converge to steady-state negates the gain of using the Hill relation. Fortunately, convergence to steady state can be relatively fast with recycling boundary conditions.

Chapter 3

Mathematical Description of Weighted Ensemble

3.1 Precise Description of Weighted Ensemble

We denote the parents at time t by $\xi_t^1, \ldots, \xi_t^N$ and their associated weights by $\omega_t^1, \ldots, \omega_t^N$ where N denotes the population of the ensemble. Similarly, we let $\hat{\xi}_t^1, \ldots, \hat{\xi}_t^N$ be the children with associated weights $\hat{\omega}_t^1, \ldots, \hat{\omega}_t^N$. The individual steps of weighted ensemble then update the position-weight tuples as follows:

$$\{\xi_t, \omega_t\}_{i=1}^N \xrightarrow{Selection} \{\hat{\xi}_t, \hat{\omega}_t\}_{i=1}^N,$$
$$\{\hat{\xi}_t, \hat{\omega}_t\}_{i=1}^N \xrightarrow{Mutation} \{\xi_{t+1}, \omega_{t+1}\}_{i=1}^N.$$

The particles belong to a common state space E which is partitioned into a finite set of m bins $\mathcal{B}$. The set of bins may be defined in an initialization step or during the selection step. At times, it is easier to think of the bins as a partition on the particles rather than explicitly of E. One such case is when using K-means clustering to assign the particles to bins. The result of K-means clustering is a label for each particle rather than a division of the underlying space E.

$\mathcal{B}$ will always be defined such that every particle belongs to one bin $u \in \mathcal{B}$. We can define the weight of a bin at time t as

$$\omega_t(u) = \sum_{i:\xi_t^i \in u} \omega_t^i, \tag{3.1}$$

where any empty bin u is assigned a weight of zero.

In the mutation step, only the positions of the particles are updated by the chosen dynamics of the system. One such choice (as in Chapter 6) is *overdamped Langevin Dynamics*

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\beta^{-1}}dW_t, \tag{3.2}$$

where W_t is standard Brownian motion and β, dt, and $V(x)$ are user-chosen parameters. Between selection steps, the integrator runs for time Δt. As discussed in Chapter 2, the underlying kernel of the simulation $K^{\Delta t}$ is a Δt-skeleton of the true Markov kernel K.

In the selection step, for a population of N particles and a set of m bins $\mathcal{B}$, each bin requires an *allocation* giving the number of children to copy in that bin. We will denote the allocation for bin u at time t by $N_t(u)$. Recall that this allocation must satisfy that no empty bins are assigned children, any occupied bins have at least one child, and that the total population remains fixed. Thus, $N_t(u)$ satisfies the following definition.

Definition 3.1.1. $\{N_t(u)\}_{u\in\mathcal{B}}$ is a valid allocation if for $N_t(u) \in \{0\} \cup \mathbb{N}$, the following hold.

1. If $\omega_t(u) = 0$ then $N_t(u) = 0$.
2. If $\omega_t(u) > 0$ then $N_t(u) \geq 1$.
3. $\sum_{u\in\mathcal{B}} N_t(u) = N$.

$N_t(u)$ can equivalently be understood as the number of copies of the parents or the number of children in bin u at time t. For a specific parent ξ_t^i, we denote the number of children of ξ_t^i by C_t^i.

The weights of the parents define an intra-bin distribution that is used during sampling. The probability of copying a parent $\xi_t^i \in u$ is then

$$\mathbb{P}(\text{copy } \xi_t^i) = \frac{\omega_t^i}{\omega_t(u)}. \tag{3.3}$$

This definition means that a parent may have multiple children but a child will have a unique parent. We will denote the parent of a child by: $\text{par}(\hat{\xi}_t^j) = \xi_t^i$. Note, that it is not necessary that $j = i$.

Only the parent's position is copied. To maintain the total weight, the weight of the children in bin u at time t are given weights

$$\hat{\omega}_t^j = \frac{\omega_t(u)}{N_t(u)}. \tag{3.4}$$

With this definition, the total weight is then conserved as for any $t \geq 0$

$$\sum_{i=1}^{N} \hat{\omega}_t^i = \sum_{u \in \mathcal{B}} \frac{\omega_t(u)}{N_t(u)} N_t(u) = \sum_{u \in \mathcal{B}} \omega_t(u) = 1 \tag{3.5}$$

The children in bin u then are defined by the tuple $\{\hat{\xi}_t^j, \hat{\omega}_t^j\}$ where

$$\begin{aligned} \hat{\xi}_t^j &= \xi_t^i \qquad \text{if par}(\hat{\xi}_t^j) = \xi_t^i, \\ \hat{\omega}_t^i &= \frac{\omega_t(u)}{N_t(u)}, \end{aligned}$$

for all parents $\xi_t^i \in u$.

Finally, it is convenient to define $\mathcal{F}_k$, the σ-algebra generated by weighted ensemble until the k^{th} selection step and $\hat{\mathcal{F}}_k$, the σ-algebra generated by weighted ensemble following the k^{th} selection step. Recall, Δt is a fixed interval between selection steps. Therefore, the k^{th} selection step will occur when $t = k\Delta t$. Thus, we can write

$$\begin{aligned} \mathcal{F}_k &= \sigma\left(\{\xi_s^i, \omega_t^i\}_{0 \leq s \leq t}^{i=1,2,\ldots,N}, \{N_\ell(u)\}_{0 \leq \ell \leq k}^{u \in \mathcal{B}}, \{\mathcal{B}_\ell\}_{0 \leq \ell \leq k}, \{\hat{\xi}_\ell^i, \hat{\omega}_\ell^i\}_{0 \leq \ell \leq k-1}^{i=1,2,\ldots,N}, \{C_\ell^i\}_{0 \leq \ell \leq k-1}^{i=1,2,\ldots,N}\right) \\ \hat{\mathcal{F}}_k &= \sigma\left(\{\xi_s^i, \omega_x^i\}_{0 \leq s \leq t}^{i=1,2,\ldots,N}, \{N_\ell(u)\}_{0 \leq \ell \leq k}^{u \in \mathcal{B}}, \{\mathcal{B}_\ell\}_{0 \leq \ell \leq k}, \{\hat{\xi}_s^i, \hat{\omega}_x^i\}_{0 \leq s \leq t}^{i=1,2,\ldots,N}, \{C_\ell^i\}_{0 \leq \ell \leq k}^{i=1,2,\ldots,N}\right). \end{aligned}$$

We summarize weighted ensemble by the following algorithm.

Algorithm 1. Weighted Ensemble

Initialization

1. Choose initial particles and positive weights $\{\xi_0^i, \omega_0^i\}_{i=1}^N$ such that $\sum_{i=1}^N \omega_0^i = 1$. Choose a collection of bins or binning strategy $\mathcal{B}$, an allocation strategy, a resampling interval Δt, and number of selection steps T. Set the weight of the flux $J = 0$.

For $0 \leq t \leq T$, iterate the following:

Selection step

1. Partition the parents. $\{\xi_t^i\}_{i=1}^N$ according to $\mathcal{B}$

2. For each bin $u \in \mathcal{B}$ perform the following:

   i. Define $N_t(u)$ for each bin $u \in \mathcal{B}$ according to the allocation strategy.

   ii. Sample $N_t(u)$ children from the parents in bin u according to (3.3)

$$\mathbb{P}(\text{copy } \xi_t^i) = \frac{w_t^i}{w_t(u)}.$$

   iii. Set the weights of the children, $\hat{\xi}_t^j \in u$, according to (3.4)

$$\hat{\omega}_t^j = \frac{\omega_t(u)}{N_t(u)}.$$

Mutation Step

1. Evolve the children $\{\xi_t^i, \omega_t^i\}_{i=1}^N$ conditionally independently by $K^{\Delta t}$ to obtain the parents at time $t+1$, $\{\xi_{t+1}^i\}_{i=1}^N$.

2. Keep the weights fixed until the next selection step, $\{\hat{\omega}_t^j\}_{j=1}^N = \{\omega_{t+1}^j\}_{j=1}^N$.

3. After evolving,

   i. $J \leftarrow J + W_t$ where W_t is the weight of all particles that crossed into B.

   ii. Recycle all particles that crossed into B according to ρ_A.

Remark 3.1. In steady state and with sufficient particles and bins, the weights (ω_t^i) of the particles (ξ_t^i) approximate the steady state distribution associated with the particles' positions ($\mu(\xi_t^i)$).

Remark 3.2. In algorithm 1, selection steps are performed conditionally on $\mathcal{F}_t$ and mutation steps are performed conditionally on $\hat{\mathcal{F}}_t$. Additionally,

1. The allocation strategy in practice is typically taken as a uniform allocation where each occupied bin is assigned the same number of children.

2. In practice, bins are usually based on the Root-Mean-Squared Distance from the sink.

3. We discuss other allocation and binning strategies in Chapter 4 and particular implementations in Chapter 5.

4. Weighted ensemble will converge for any valid choice of allocation strategy (satisfying definition 3.1.1) so long as the resampling method is unbiased.

Remark 3.3. Direct Monte Carlo, i.e. independent particles, is a special case of algorithm 1 where we enforce $\mathbb{E}[C_t^i|\mathcal{F}_t] = 1$ for all parents $\{\xi_t^i\}_{i=1}^N$. Equivalently, direct Monte Carlo is weighted ensemble without a selection step.

3.2 Properties of Weighted Ensemble

Recall, we have assumed the Markov kernel K has stationary distribution μ. For a bounded observable f, we compute the weighted sum of f on the ensemble of parents

$$\sum_{i=1}^{N} \omega_t^i f(\xi_t^i). \tag{3.6}$$

From (3.6), we obtain an estimate

$$\theta_T \approx \int f \, d\mu,$$

where

$$\theta_T = \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \omega_t^i f(\xi_t^i). \tag{3.7}$$

Specifically for computing the MFPT, we choose f as in (2.1). In Algorithm 1, when the underlying dynamics enter the target set, they are held there until the end of the resampling interval. As mentioned before, this introduces a small bias but admits nice variance formulas and ensures

all flux into B is counted. The discretization of the underlying continuous process also introduces a bias, though this and the bias from recycling are negligible. θ_T is then the average weight of the particles that have been recycled by time $T-1$. This distinction clarifies that we only need the flux into B with no knowledge of the full trajectory of the particles that cross into B. The estimations obtained from algorithm 1 have two important properties.

Theorem 3.2.1. *(Unbiased Property). In algorithm 1, let g be any bounded measurable function on the state space and $X(t)$ be a Markov chain with kernel K. Assume X_t has an initial distribution ν where*

$$\int g \, d\nu = \mathbb{E}\left[\sum_{i=1}^{N} \omega_0^i g(\xi_0^i)\right].$$

Then for any $t \geq 0$

$$\mathbb{E}\left[\sum_{i=1}^{N} \omega_t^i g(\xi_t^i)\right] = \mathbb{E}\left[g(X(t))\right] = \int K^t g \, d\nu.$$

Theorem 3.2.1 is a direct result of the one-step means, which will be introduced in the following section, and the tower property of conditional expectation.

Theorem 3.2.2. *(Ergodic Theorem). Let g be any bounded measurable function on the state space and $X(t)$ be a geometrically ergodic Markov chain with steady state distribution μ and kernel K. The estimator of weighted ensemble is ergodic in the sense that almost surely*

$$\lim_{T\to\infty} \theta_T = \int g \, d\mu.$$

Theorem 3.2.2 follows from Theorem 3.2.1 and our assumption that K is geometrically ergodic. The proof of the ergodic theorem is beyond the scope of this thesis but can be found in [39].

Recall Remark 3.3, that if no selection steps are performed in algorithm 1, we recover direct Monte Carlo sampling. Thus, for weighted ensemble to be a prudent choice, parameters (such as binning and allocation methods) should be chosen to provide a variance reduction compared to the Monte Carlo estimator. To inform these choices, we will derive variance formulas that show that

the variance can be decomposed into a contribution from each of the three steps of algorithm 1. Briefly, for weighted ensemble to beat direct Monte Carlo, the goal will be to leverage the selection step to sufficiently reduce the variance of the mutation step.

3.3 Weighted Ensemble One Step Means

To begin, explicitly define the one-step mean for a particle in the following way. As a standing assumption, g is a bounded measurable function on the state space.

Lemma 3.3.1. *For each* $1 \leq i \leq N$ *and* $t \geq 0$*,*

$$\mathbb{E}\Big[\omega_{t+1}^i g(\xi_{t+1}^i) \;\Big|\; \hat{\mathcal{F}}_t\Big] = \hat{\omega}_t^i K g(\hat{\xi}_t^i). \tag{3.8}$$

And, for each bin $u \in \mathcal{B}$*,*

$$\mathbb{E}\left[\sum_{i:\hat{\xi}_t^i \in u} \hat{\omega}_t^i g(\hat{\xi}_t^i) \;\middle|\; \mathcal{F}_t\right] = \sum_{i:\xi_t^i \in u} \omega_t^i g(\xi_t^i). \tag{3.9}$$

Proof. First, for (3.8), recall particles evolve via K by $\xi_{t+1}^i = K(\hat{\xi}_t^i, \cdot)$ and $g(\hat{\xi}_t^i)$ is $\hat{\mathcal{F}}_t$ measurable. Then $\mathbb{E}\Big[g(\xi_{t+1}^i) \;\Big|\; \hat{\mathcal{F}}_t\Big] = Kg(\hat{\xi}_t^i)$. As the weight of a particle is constant in the evolution step, we obtain:

$$\mathbb{E}\Big[\omega_{t+1}^i g(\xi_{t+1}^i) \;\Big|\; \hat{\mathcal{F}}_t\Big] = \mathbb{E}\Big[\hat{\omega}_t^i g(\xi_{t+1}^i) \;\Big|\; \hat{\mathcal{F}}_t\Big] = \hat{\omega}_t^i \,\mathbb{E}\Big[g(\xi_{t+1}^i) \;\Big|\; \hat{\mathcal{F}}_t\Big] = \hat{\omega}_t^i K g(\hat{\xi}_t^i).$$

Next, to prove (3.9), define the expected number of children for a parent ξ_t^i in bin u:

$$\mathbb{E}\big[C_t^i \;\big|\; \mathcal{F}_t\big] = N_t(u)\frac{w_t^i}{w_t(u)}, \tag{3.10}$$

where $w_t(u)$ is the total weight in bin u at time t. By the weight update formula (3.4), the weight of the children in bin u at time t is given by: $\hat{\omega}_t^j = \frac{w_t(u)}{N_t(u)}$.

By summing over the parents in bin u at time t, we find:

$$\begin{aligned}
\mathbb{E}\left[\sum_{i:\hat{\xi}_t^i \in u} \hat{\omega}_t^i g(\hat{\xi}_t^i) \;\middle|\; \mathcal{F}_t\right] &= \sum_{i:\xi_t^i \in u} \mathbb{E}\left[\sum_{j:\text{par}(\hat{\xi}_t^j)=\xi_t^i} \hat{\omega}_t^j g(\hat{\xi}_t^j) \;\middle|\; \mathcal{F}_t\right] \\
&= \sum_{i:\xi_t^i \in u} \mathbb{E}\big[C_t^i \mid \mathcal{F}_t\big] \frac{w_t(u)}{N_t(u)} g(\xi_t^i) \\
&= \sum_{i:\xi_t^i \in u} N_t(u) \frac{w_t^i}{w_t(u)} \frac{w_t(u)}{N_t(u)} g(\xi_t^i) \\
&= \sum_{i:\xi_t^i \in u} \omega_t^i g(\xi_t^i).
\end{aligned}$$

□

We can define the one-step means for the ensemble by summing (3.8) over all particles and (3.9) all bins at time t.

Corollary 3.3.2. *(Ensemble one-step means)*

$$\mathbb{E}\left[\sum_{i=1}^N \omega_{t+1}^i g(\xi_{t+1}^i) \;\middle|\; \hat{\mathcal{F}}_t\right] = \sum_{i=1}^N \hat{\omega}_t^i K g(\hat{\xi}_t^i). \tag{3.11}$$

$$\mathbb{E}\left[\sum_{i=1}^N \hat{\omega}_t^i g(\hat{\xi}_t^i) \;\middle|\; \mathcal{F}_t\right] = \sum_{i=1}^N \omega_t^i g(\xi_t^i). \tag{3.12}$$

The first consequence we will prove with Corollary 3.3.2 is the unbiased property of Theorem 3.2.1. Recall, we wish to show that for a Markov chain $X(t)$ with kernel K and any $t \geq 0$

$$\mathbb{E}\left[\sum_{i=1}^N \omega_t^i g(\xi_t^i)\right] = \mathbb{E}\left[g(X(t))\right] = \int K^t g \, d\nu,$$

given that

$$\mathbb{E}\left[\sum_{i=1}^N \omega_0^i g(\xi_0^i)\right] = \int g \, d\nu.$$

Proof. It suffices to show that for any time $t \geq 0$

$$\mathbb{E}\left[\sum_{i=1}^{N} \omega_t^i g(\xi_t^i) \;\middle|\; \mathcal{F}_0\right] = \sum_{i=1}^{N} \omega_0^i K^t g(\xi_0^i), \tag{3.13}$$

as

$$\mathbb{E}\left[\sum_{i=1}^{N} \omega_0^i K^t g(\xi_0^i)\right] = \int K^t g \, d\nu. \tag{3.14}$$

As $\mathcal{F}_t$ and $\hat{\mathcal{F}}_t$ are filtrations, by the tower property and Corollary 3.3.2,

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N} \omega_{t+1}^i g(\xi_{t+1}^i) \;\middle|\; \mathcal{F}_t\right] &= \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{N} \omega_{t+1}^i g(\xi_{t+1}^i) \;\middle|\; \hat{\mathcal{F}}_t\right] \;\middle|\; \mathcal{F}_t\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N} \hat{\omega}_t^i K g(\hat{\xi}_t^i) \;\middle|\; \mathcal{F}_t\right] \\
&= \sum_{i=1}^{N} \omega_t^i K g(\xi_t^i).
\end{aligned}$$

Repeating this process yields (3.13). □

3.4 Weighted Ensemble Variance

We assume the population N is finite and find exact formulas for the variance of weighted ensemble. To accomplish this, some precise definitions are required. For time $t \geq 0$ and a bin u, define the intra-bin distributions based on parents weights

$$\eta_t^u = \sum_{i:\xi_t^i \in u} \frac{\omega_t^i}{\omega_t(u)} \delta_{\xi_t^i},$$

where $\delta_{\xi_t^i}$ is the Dirac delta distribution centered at ξ_t^i. We also introduce

$$h_{t,T} = \sum_{s=0}^{T-t-1} K^s f. \tag{3.15}$$

We let

$$\nu(g) = \int g \, d\nu, \qquad \mathrm{Var}_\nu g = \nu\left(g^2\right) - \left(\nu(g)\right)^2,$$

for a probability distribution ν and bounded measurable function g.

From Corollary 3.3.2, we can define the Doob martingales

$$D_t = \mathbb{E}\left[\sum_{s=0}^{T-1}\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) \;\middle|\; \mathcal{F}_t\right]$$
$$\hat{D}_t = \mathbb{E}\left[\sum_{s=0}^{T-1}\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) \;\middle|\; \hat{\mathcal{F}}_t\right].$$

These definitions facilitate decomposing the variance into separate variances for the initial condition, selection steps, and mutation steps. This decomposition will lead to nice formulas for each variance term which, in turn, inspire optimizations in the choice of the binning and allocation strategies of algorithm 1.

To begin, we decompose each term in the Doob martingales in the following manner.

Proposition 3.4.1. *For* $0 \le t \le T-1$,

$$D_t = \sum_{s=0}^{t}\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) + \sum_{i=1}^{N} \omega_t^i K h_{t+1,T}(\xi_t^i) \tag{3.16}$$

$$\hat{D}_t = \sum_{s=0}^{t}\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) + \sum_{i=1}^{N} \hat{\omega}_t^i K h_{t+1,T}(\hat{\xi}_t^i) \tag{3.17}$$

Proof. For (3.16), given $\mathcal{F}_t$, we can split the expectation on information contained in the σ-algebra until time t and information unavailable at time t. We obtain

$$D_t = \mathbb{E}\left[\sum_{s=0}^{T-1}\sum_{i=1}^{N} \omega_x^i f(\xi_s^i) \;\middle|\; \mathcal{F}_t\right] = \sum_{s=0}^{t}\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) + \mathbb{E}\left[\sum_{s=t+1}^{T-1}\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) \;\middle|\; \mathcal{F}_t\right]. \tag{3.18}$$

The expectation in the right-hand side of (3.18) can be expressed as

$$\sum_{s=t+1}^{T-1} \mathbb{E}\left[\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) \;\middle|\; \mathcal{F}_t\right]. \tag{3.19}$$

Similar to the proof of Theorem 3.2.1, from Corollary 3.3.2 and the tower property, (3.19) can be written as

$$\sum_{s=t+1}^{T-1} \mathbb{E}\left[\sum_{i=1}^{N} \omega_s^i f(\xi_s^i) \;\middle|\; \mathcal{F}_t\right] = \sum_{s=t+1}^{T-1} K^s \sum_{i=1}^{N} \omega_t^i f(\xi_t^i) = \sum_{i=1}^{N} \omega_t^i \sum_{s=t+1}^{T-1} K^s f(\xi_t^i). \tag{3.20}$$

Therefore, from (3.15)

$$D_t = \sum_{s=0}^{t} \sum_{i=1}^{N} \omega_s^i f(\xi_s^i) + \sum_{i=1}^{N} \omega_t^i K h_{t+1,T}(\xi_t^i).$$

The proof of (3.17) is similar. □

We can now define the variance of θ_T by decomposing the Doob martingales. Notice that

$$\theta_T = \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \omega_t^i f(\xi_t^i) = \frac{1}{T} D_{T-1}.$$

Theorem 3.4.2. *(Variance Decomposition) For each $T > 0$,*

$$\begin{aligned}
\mathrm{Var}(\theta_T) = & \\
& \frac{1}{T^2}\mathrm{Var}(D_0) && (3.21)\\
& + \frac{1}{T^2} \sum_{t=0}^{T-2} \mathbb{E}\Big[(\hat{D}_t - D_t)^2 \;\Big|\; \mathcal{F}_t\Big] && (3.22)\\
& + \frac{1}{T^2} \sum_{t=0}^{T-2} \mathbb{E}\Big[(D_{t+1} - \hat{D}_t)^2 \;\Big|\; \hat{\mathcal{F}}_t\Big], && (3.23)
\end{aligned}$$

where (3.21) *is the variance due to the initial condition,* (3.22) *is the variance due to the selection steps, and* (3.23) *is the variance due to the mutation steps.*

Proof. To compute the variance, notice that D_{T-1} may be written as

$$\begin{aligned} D_{T-1} &= D_{T-1} - \hat{D}_{T-2} + \hat{D}_{T-2} - D_{T-2} + D_{T-2} - \cdots + D_1 - \hat{D}_0 + \hat{D}_0 - D_0 + D_0 \\ &= (D_{T-1} - \hat{D}_{T-2}) + (\hat{D}_{T-2} - D_{T-2}) + (D_{T-2} - \cdots + (D_1 - \hat{D}_0) + (\hat{D}_0 - D_0) + D_0 \end{aligned}$$

In the multinomial expansion of D_{T-1}^2, the martingale differences are uncorrelated and are $\mathcal{F}_t$ or $\hat{\mathcal{F}}_t$ measurable for $0 \leq t \leq T-2$. Therefore

$$\mathbb{E}\left[D_{T-1}^2\right] = D_0^2 + \sum_{t=0}^{T-2} \mathbb{E}\Big[(\hat{D}_t - D_t)^2 \;\Big|\; \mathcal{F}_t\Big] + \sum_{t=0}^{T-2} \mathbb{E}\Big[(D_{t+1} - \hat{D}_t)^2 \;\Big|\; \hat{\mathcal{F}}_t\Big]. \tag{3.24}$$

Finally, we can write

$$\mathrm{Var}(\theta_T) = \frac{1}{T^2}\left(\mathbb{E}\left[D_{T-1}^2\right] - \mathbb{E}[D_{T-1}]^2\right). \tag{3.25}$$

Note that $\mathbb{E}[D_{T-1}] = \mathbb{E}[D_0]$ from the martingale property. From (3.24), (3.25) can expressed as

$$\begin{aligned} \mathrm{Var}(\theta_T) &= \frac{1}{T^2}\mathbb{E}\left[(D_{T-1} - \mathbb{E}[D_0])^2\right] \\ &= \frac{1}{T^2}(D_0^2 - \mathbb{E}[D_0]^2) \\ &+ \frac{1}{T^2}\sum_{t=0}^{T-2} \mathbb{E}\Big[(\hat{D}_t - D_t)^2 \;\Big|\; \mathcal{F}_t\Big] \\ &+ \frac{1}{T^2}\sum_{t=0}^{T-2} \mathbb{E}\Big[(D_{t+1} - \hat{D}_t)^2 \;\Big|\; \hat{\mathcal{F}}_t\Big] \end{aligned}$$

to complete the proof. □

Chapter 4

Variance Minimization

Theorem 3.4.2 gives a clear expression of the variance of weighted ensemble in terms of the constituent steps of algorithm 1. To make use of these formulas, we rewrite the expressions in terms of $h_{t,T}$. This choice will prove critical as we will show in this section that $h_{t,T}$ defines one of the key coordinates for controlling the variance. From [39], we can write general expressions for the selection and mutation variance as in Theorem 4.0.1.

Theorem 4.0.1. *The selection variance at time t can be written*

$$\mathbb{E}\left[\left(\hat{D}_t - D_t\right)^2 \,\middle|\, \mathcal{F}_t\right] = \mathbb{E}\left[\mathrm{Var}\left(\sum_{i=1}^{N} \hat{\omega}_t^i K h_{t+1,T}(\hat{\xi}_t^i)\right) \,\middle|\, \mathcal{F}_t\right]. \tag{4.1}$$

Further, the mutation variance at time t can be written

$$\mathbb{E}\left[\left(D_{t+1} - \hat{D}_t\right)^2 \,\middle|\, \hat{\mathcal{F}}_t\right] = \sum_{i=1}^{N} \left(\hat{\omega}_t^i\right)^2 \mathrm{Var}_K\left(h_{t+1,T}(\hat{\xi}_t^i)\right). \tag{4.2}$$

Remark 4.1. There are many suitable choices for the sampling method. This choice will affect the exact form of (4.1) as it will dictate how children are selected. However, the mutation variance will remain as (4.2) regardless of the sampling method.

Notice that for $0 \leq t \leq T-2$ and for each bin $u \in \mathcal{B}$, (4.2) depends on the distribution of the children in u. The number of children in bin u will depend explicitly on the allocation $N_t(u)$. It therefore makes sense to define the mutation variance conditionally on $\mathcal{F}_t$. From the expected number of children given in (3.10), (4.2) can be expressed as

$$\mathbb{E}\left[\left(D_{t+1} - \hat{D}_t\right)^2 \,\middle|\, \mathcal{F}_t\right] = \mathbb{E}\left[\sum_{u\in\mathcal{B}} \frac{\omega_t(u)}{N_t(u)} \sum_{i:\xi_t^i \in u} \omega_t^i \mathrm{Var}_K(h_{t+1,T}\left(\xi_t^i\right)) \,\middle|\, \mathcal{F}_t\right]. \tag{4.3}$$

To minimize the variance of weighted ensemble as presented in algorithm 1, we seek strategies that minimize equations (4.1) and (4.3). Intuitively, (4.1) is controlled by the variance in how the children will evolve. This means that only trajectories which are in similar in their expected behavior, that is $h_{t+1,T}(\xi_t^i) \approx h_{t+1,T}(\xi_t^j)$, should be merged in this step. Only trajectories in the same bin can be merged, thus we use binning to control the variance in the selection step. Minimizing the variance of the mutation step is achieved by carefully choosing the allocation $N_t(u)$. Broadly, bins that have a high weighted variance in expected evolution should have more particles allocated to that bin.

4.1 Coordinates for Variance Reduction

To minimize the variance in using weighted ensemble, we focus on the choice of binning and allocation strategies. In addition to the initial distribution of particles, these are the only weighted ensemble parameters. The motivation of this choice is that these two strategies determine where and how many children are created which directly impacts the variance. Recalling Remark 3.3, weighted ensemble can achieve a lower mutation variance by incurring a cost in the selection step. The primary goal of choosing binning and allocation strategies is to over-sample regions in the state space that which contribute volatility to the flux. A secondary objective will be to minimize the cost incurred in the selection step by identifying particles that are in some way similar and grouping them into bins.

To quantify different regions of state space as similar in terms of estimating the MFPT, we define the *flux discrepancy function*. The discrepancy function is a mapping of the state space $E \to \mathbb{R}$ and measures the difference in flux from starting at a point x in the state space compared to the steady-state distribution, μ. Let N_T denote the number of crossings into B in the time interval $[0, T])$. We then define the discrepancy function as

$$h(x) = \lim_{T\to\infty} \left(\mathbb{E}[N_t \mid x_0 \sim x] - \mathbb{E}[N_t \mid x_0 \sim \mu]\right). \tag{4.4}$$

Two points in state space will be considered similar under the discrepancy function if h evaluated at each point is similar. During the resampling step, merging particles at these two points would result in minimal information loss. h then describes particles that make similar contributions to the flux and makes it an ideal coordinate for managing merging.

As we have discretized our problem, we are considering a Markov chain that evolves according to the kernel $K^{\Delta t}$ where Δt is the resampling time. More generally weighted ensemble computes the average value of some observable f in practice. Therefore, (4.4) may more appropriately be described by

$$h_{\Delta t}(x) = \lim_{T\to\infty}\left(\mathbb{E}\left[\sum_{k=0}^{T} f(X(k\Delta t)) \;\middle|\; X_0 \sim x\right] - \mathbb{E}\left[\sum_{k=0}^{T} f(X(k\Delta t)) \;\middle|\; X_0 \sim \mu\right]\right). \tag{4.5}$$

In the limit as $\Delta t \to 0$ and f as the indicator function of the set B, through rescaling by Δt it can be shown that

$$h(x) = \lim_{\Delta t\to 0} \Delta t \cdot h_{\Delta t}(x).$$

Here, the requirement that recycling only occurs at the end of the resampling time, Δt, is important. This counts all flux into B in (4.5) before the particles are recycled according to ρ_A.

As we are interested in long time averages, in the limit that $\Delta t \to 0$, h can be interpreted as $\lim_{T\to\infty} h_{t,T}$. This allows the simplifications

$$\lim_{T\to\infty} \mathrm{Var}_\eta h_{t+1,T} = \mathrm{Var}_\eta h$$
$$\lim_{T\to\infty} \mathrm{Var}_\eta K h_{t+1,T} = \mathrm{Var}_\eta K h.$$

These formulas give simpler formulas for the selection variance (4.1)

$$\mathbb{E}\left[\left(\hat{D}_t - D_t\right)^2 \;\middle|\; \mathcal{F}_t\right] = \mathbb{E}\left[\mathrm{Var}\left(\sum_{i=1}^{N} \hat{\omega}_t^i K h(\hat{\xi}_t^i)\right) \;\middle|\; \mathcal{F}_t\right]. \tag{4.6}$$

and the mutation variance (4.3)

$$\mathbb{E}\left[\left(D_{t+1}-\hat{D}_t\right)^2 \;\middle|\; \mathcal{F}_t\right] = \mathbb{E}\left[\sum_{u\in\mathcal{B}} \frac{\omega_t(u)}{N_t(u)} \sum_{i:\xi_t^i\in u} \omega_t^i \mathrm{Var}_K(h\left(\xi_t^i\right)) \;\middle|\; \mathcal{F}_t\right]. \tag{4.7}$$

The beauty in the discrepancy function is that it is able to address both our goals. Particles are similar if the value of h is approximately the same at each region in the state space. h allows a tangible means of creating bins that collect similarly behaving particles in the long-time limit. This will minimize the variance term in (4.6) and mitigate the cost of merging trajectories.

The discrepancy function also addresses the goal of identifying regions of volatility. Regions of state space where h has a high variance would indicate areas that have drastically variable contributions to the flux into B. At low temperatures, h is roughly constant along metastable sets and varies greatly in the regions between these sets. These regions of space are precisely those which need to be over-sampled to reduce the mutation variance. For one mutation step, we can quantify this idea by introducing the *variance function*

$$v_{\Delta t}(x)^2 = \mathbb{E}\big[h^2(X(\Delta t) \;\big|\; x_0\sim x\big] - \mathbb{E}[h(X(\Delta t) \mid x_0\sim x]^2. \tag{4.8}$$

The relevance of (4.8) is that the mutation variance (4.2) can be rewritten use $v_{\Delta t}$.

Proposition 4.1.1. *The mutation variance at time t prior to selection satisfies*

$$\lim_{T\to\infty} T^2\,\mathbb{E}\left[\left(D_{t+1}-\hat{D}_t\right)^2 \;\middle|\; \mathcal{F}_t\right] = \sum_{u\in\mathcal{B}} \frac{(\omega_t(u))^2}{N_t(u)} \mathbb{E}^{\eta_t^u}\left[v_{\Delta t}^2\right].$$

Proof. From the formula for mutation variance (4.2) and the weight update formula (3.4)

$$\lim_{T\to\infty} T^2\,\mathbb{E}\left[\left(D_{t+1}-\hat{D}_t\right)^2 \;\middle|\; \mathcal{F}_t\right] = \sum_{u\in\mathcal{B}} \left(\frac{\omega_t(u)}{N_t(u)}\right)^2 \mathbb{E}\left[\sum_{i:\hat{\xi}_t^i\in u} \mathrm{Var}_{K^{\Delta t}} h_{t+1}(\hat{\xi}_t^i) \;\middle|\; \mathcal{F}_t\right]. \tag{4.9}$$

Recall $\mathbb{E}\left[C_t^i \mid \mathcal{F}_t\right] = N_t(u)\omega_t^i/\omega_t(u)$. Then (4.9) can be split as a sum over the parents to find

$$\begin{aligned}(4.9) &= \sum_{u\in\mathcal{B}} \left(\frac{\omega_t(u)}{N_t(u)}\right)^2 \sum_{i:\xi_t^i\in u} N_t(u)\frac{w_t^i}{w_t(u)} v_{\Delta t}(\xi_t^i)^2 \\ &= \sum_{u\in\mathcal{B}} \frac{(\omega_t(u))^2}{N_t(u)} \sum_{i:\xi_t^i\in u} \frac{w_t^i}{w_t(u)} v_{\Delta t}(\xi_t^i)^2. \end{aligned} \tag{4.10}$$

The sum over the parents in (4.10) can then be seen as a weighted average of $v_{\Delta t}^2$ over the intra-bin distribution at time t. □

We can interpret (4.8) as the variance in future cumulative flux for a particle starting at x. Again, it will be convenient to consider the result of the rescaled limit

$$v = \lim_{\Delta t\to 0} \frac{v_{\Delta t}}{\Delta t}.$$

We can connect the overdamped Langevin dynamics discretization

$$dX_t = -\nabla V(X_t)dt + \sqrt{2\beta^{-1}}dW_t$$

to the continuous underlying process to address the rescaled limits above. We can more precisely define the infinitesimal generator L in the case of overdamped Langevin dynamics by

$$Lg = -\nabla V \cdot \nabla g + \beta^{-1}\Delta g$$

with appropriate boundary conditions. L is a second order operator that describes the evolution of the Markov process. From the definition of an infinitesimal generator

$$Lh(x) = \lim_{\Delta t\to 0}\lim_{t\to\infty} \frac{\mathbb{E}^x[h(X_{\Delta t})] - h(x)}{\Delta t}, \tag{4.11}$$

where $\mathbb{E}^x[h(X_{\Delta t}] = \mathbb{E}[h(X_{\Delta t})|X_0 \sim x]$. From (4.11) and (4.8), it can be shown that

$$\begin{aligned}\lim_{\Delta t\to 0}\frac{v_{\Delta t}(x)^2}{\Delta t} &= \lim_{\Delta t\to 0}\frac{\mathbb{E}^x[h^2(X_{\Delta t})] - \mathbb{E}^x[h(X_{\Delta t})]^2}{\Delta t}\\ &= Lh^2(x) - 2h(x)Lh(x)\\ &= 2\beta^{-1}|\nabla h(x)|^2\end{aligned}$$

Furthermore, $h_{\Delta t}$ satisfies the poisson equation

$$(Id - K)h_{\Delta t} = f - \int f\,d\mu, \qquad \int h_{\Delta t}\,d\mu = 0. \tag{4.12}$$

In practice, h and v^2 are unknown but can be estimated by uninformed implementations of weighted ensemble [40]. Rough estimates of K can be used to solve (4.12) to obtain estimates of h and v^2.

4.2 Minimization Strategies

Adapting the observations of the previous section, we can outline our approach for minimizing the variance of weighted ensemble. The driving idea will be to allocate particles in volatile regions, meaning regions with large ∇h values, to minimize the mutation variance and to choose bins of similar particles, in the sense of having similar h values, to minimize the selection variance. To simplify the formulas involved, we will assume that N is fixed and let $T \to \infty$. With the latter assumption, recall

$$\begin{aligned}\lim_{T\to\infty}\mathrm{Var}_\eta h_{t+1,T} &= \mathrm{Var}_\eta h\\ \lim_{T\to\infty}\mathrm{Var}_\eta Kh_{t+1,T} &= \mathrm{Var}_\eta Kh.\end{aligned}$$

We further assume that at time t, we employ both our binning and allocation strategy. This means that allocation is performed with knowledge available before selection.

To minimize the *mutation variance*, recall Proposition 4.1.1. The advantage to writing the mutation variance in this form is it allows for a simple optimization question to be posed. As we

are only interested in choosing an allocation at time t, we can consider the following problem:

$$\text{minimize} \sum_{u \in \mathcal{B}} \frac{(\omega_t(u))^2}{N_t(u)} \mathbb{E}^{\eta_t^u} \left[v_{\Delta t}^2 \right] \tag{4.13}$$

subject to $N_t(u) \in \mathbb{R}^+$ satisfying $\sum_{u \in \mathcal{B}} N_t(u) = N$. The solution to the optimization problem posed in (4.13) can be found from Lagrange multipliers to be

$$N_t(u) \propto \sqrt{(\omega_t(u))^2 \cdot \mathbb{E}^{\eta_t^u} \left[v_{\Delta t}^2 \right]} \approx \omega_t(u) \cdot \mathbb{E}^{\eta_t^u} \left[v_{\Delta t} \right]. \tag{4.14}$$

Notice we have relaxed our requirements for the allocation as found in 3.1.1. The solution presented in (4.14) is then an approximate solution in terms of implementation. In practice, enforcing $\mathbb{N}_t \in \mathbb{Z}^{\geq 0}$ and 3.1.1 is not difficult, [41] outlines one such algorithm.

An algorithm for choosing an optimal set of m bins, $\mathcal{B}$ is also outlined in [41]. We will adopt a similar approach, attempting to choose bins to minimize the variance of h according to the intra-bin distributions ν_u. That is

$$\text{minimize} \sum_{u \in \mathcal{B}} \text{Var}_{\nu_u} Kh \tag{4.15}$$

for partitions $\mathcal{B}$ of size m. The motivation for this approach is to consider the selection variance given by (4.6). Recall that after selection, the weight of the children $\hat{\omega}_t^i$ is equal by (3.4) and the evolution of particles in separate bins is uncorrelated. Therefore, for $t \geq 0$ and a choice of bins u with allocations $N_t(u)$, the selection variance (4.6) is equivalent to

$$\mathbb{E}\Big[(\hat{D}_t - D_t)^2 \;\Big|\; \mathcal{F}_t \Big] = \mathbb{E}\left[\sum_{u \in \mathcal{B}} \left(\frac{w_t(u)}{N_t(u)} \right)^2 \text{Var} \left(\sum_{i : \hat{\xi}_t^i \in u} Kh(\hat{\xi}_t^i) \right) \;\middle|\; \mathcal{F}_t \right]. \tag{4.16}$$

In the following chapter, we will outline three successful strategies that are based on these principles.

4.3 Comparison against Naive Methods

On a final note, in the limit that the number of bins $m \to \infty$ and the population $N \to \infty$ it is clear that the selection variance will vanish. This is a consequence of $\mathrm{Var}_\nu(Kh) \to 0$ for an intra-bin distribution ν. However, this is not a practical choice. Simulations in molecular dynamics typically come with great computational cost providing further restrictions on the number of particles and bins that can be employed.

In complex problems of interest, this cost is largely dictated by the evolution of the particles which places a limit on N. From Remark 3.3, the number of bins should be chosen such that Kh does not vary too greatly and still allows over-sampling of important regions. No comprehensive rigorous analysis has been completed on this idea for fixed finite N, but intuitively the number of fixed spatial bins should not exceed N by too much. In the case of adaptive bins that directly partition the particles, it is advisable that the number of bins is strictly less than N.

It has been shown that the mutation variance of weighted ensemble with positive selection variance is bounded above by the mutation variance of direct Monte Carlo [39]. In problems where v is nearly constant, there are no regions of the state space that will be over-sampled. Therefore, we would expect weighted ensemble to perform on par with direct Monte Carlo. However, for problems where v is highly variable, weighted ensemble can achieve a large reduction in variance by optimal allocation. A myriad of practitioners have demonstrated that a variance reduction is achievable in practice [40, 42–45]. We will later demonstrate a variance reduction in the context of a toy problem.

To further illustrate the potential gain over direct Monte Carlo, consider overdamped Langevin dynamics on $[0, 1]$ with $A = 0$, $B = 1$ and recycling boundary conditions. Let $V(x)$ be the potential and define

$$\Delta V = \max_{x\in[0,1]} V(x) - \min_{x\in[0,1]} V(x).$$

Assume that there is a potential barrier between A and B. As h is roughly constant in metastable regions, v will vary greatly only if that potential barrier is large, that is if ΔV is large. The variance

improvement factor follows

$$\text{VIF} = \frac{\text{dimensionless MCMC variance}}{\text{dimensionless optimal WE variance}} = \frac{\mu(v^2)}{\mu(v)^2} \sim \exp(\beta \Delta V),$$

where β is an inverse temperature parameter. Therefore, the reduction in variance from using weighted ensemble can be significant for problems where v is far from a constant.

As noted in Remark 3.2, weighted ensemble is not usually accompanied by these minimization practices. These naive implementations of weighted ensemble can provide improved stability in the estimation of rare-events. We will show that by implementing allocation and binning strategies that address (4.13) and (4.15), a further reduction in variance can be achieved.

Chapter 5

Methods

5.1 Binning Methods

We will present static and adaptive binning methods which are inspired by the selection variance formula 4.16. Static binning methods are methods where the bins are set in during the initialization of the weighted ensemble simulation and remain fixed in time. The adaptive binning method we cover will change in time where particles are partitioned at the beginning of each selection step. In Chapter 4, we found h to be of great importance for minimizing the selection variance. The discrepancy function h is used in each optimized method below as it gives a 1D representation of an $n-$dimensional problem.

Remark 5.1. In the following discussion with binning and allocation methods, we will adopt the notation that bold characters (**u**, **a**, **s**, **v**) will denote binning strategies and script characters ($\mathfrak{u}$, $\mathfrak{w}$, $\mathfrak{h}$, $\mathfrak{v}$) will denote allocation strategies. This distinction is made to clarify the difference between uniform binning, **u**, and uniform allocation, $\mathfrak{u}$.

5.1.1 Static Binning Method

The first static method we present is where bins are created uniformly in space. This approach does not use any of the minimization techniques outlined in Chapter 4 but is by far the most common weighted ensemble scheme [44, 46–48]. Bins can easily be computed from the root-mean-square deviation (RMSD) to the sink and subsequently evenly dividing the RMSD space between A and B (see Figure 6.6). The parents $\{\xi_t^i\}$ are assigned a bin based on the RMSD to B. We will denote this method as **u**.

We now present two optimized static methods. Recall, we are choosing bins to minimize the selection variance in (4.16). Selection may also be understood as splitting and merging trajectories of particles in the same bin. Inherent in the process of merging is the introduction of a correlation

in the ensemble of trajectories. This correlation is a direct result of merging two or more particles who now share the same trajectory and results in an increase in variance. In an extreme case, consider a weighted ensemble simulation with only one bin. Particles that are at the source are likely to have a larger weight than those near the sink (as $\omega \sim \mu$) and therefore are more likely to be copied in the selection step. This would very likely result in a loss of flux and a larger overall variance. From (4.4), these particles would have drastically different values of h and should not be merged.

When the resampling interval Δt and the integrator time step δt are small, $Kh(\xi_t^i) \approx h(\xi_t^i)$. Therefore, to minimize the selection variance, particles should be binned to minimize the variance in $\sum_{i:\xi_t^i \in u} h(\hat{\xi}_t^i)$.

During selection, we would like to limit this cost involved in merging. In this case, minimal information is lost. Limiting the impact of these correlations is the goal of the method **s** (see Figure 6.7 (a)). We have established that h is the relevant coordinate for merging but do not yet have an understanding on how to choose the size of the bins.

From Remark 3.1, the optimal allocation (4.14) can be interpreted as

$$\sum_{i:\xi_t^i \in u} \mu(\xi_t^i) \cdot \mathbb{E}^{\eta_t^u}\left[v_{\Delta t}(\xi_t^i)\right] \approx \sum_{i:\xi_t^i \in u} \mu(\xi_t^i) \cdot v(\xi_t^i). \tag{5.1}$$

The advantage to (5.1) is that it can be computed during initialization by choosing a set of points uniformly distributed through the state space and computing the values of μ and v at each point. This mesh is representative of a weighted ensemble algorithm with infinite particles in the limit $T \to \infty$. Furthermore, it can be used to define bins that are sampled equally while still oversampling relevant parts of the state space. To reduce the impact of correlation, bins $\mathcal{B}$ can be chosen to such that the optimal allocation is approximately uniform across $\mathcal{B}$.

Combining these two ideas results in the method we denote as **s** (see Figure 6.7 (a)). First, we consider sorted h space, $\mathcal{H}$, to create bins with similar h values. Then, we consider bins

$\mathcal{B} = \{u_1, u_2, \ldots, u_m\}$ that partition $\mathcal{H}$ and satisfy for any u_i, u_j

$$\int\limits_{x:h(x)\in u_i} \mu(x)v(x)\,dx \approx \int\limits_{x:h(x)\in u_j} \mu(x)v(x)\,dx.$$

As **s** does not consider the resulting intra-bin distributions, we propose an improvement which we will denote as **v** (see Figure 6.7 (b)). Retaining the framework of **s**, we introduce the added idea that the bin-size modulated variance of the intra-bin distribution μv should be minimized. This can be accomplished in the following way. Let $\{H_i\}_{i=1}^{m-1}$ be level sets in $\mathcal{H}$ corresponding to the bins we seek. Consider the sets

$$\Omega_1 \subset \Omega_2 \subset \ldots \subset \Omega_m,$$

where

$$\Omega_i = \int\limits_{x:h(x)\leq H_i} \mu(x)v(x)\,dx.$$

Let $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_3\}$. We can then solve

$$\underset{\{\Omega_k\}_{k=1}^m \in \Omega}{\operatorname{argmin}} \; |\Omega_k| \mathrm{Var}(\Omega_k) \tag{5.2}$$

by k-means clustering on Ω. Recall, we assumed a mesh that discretized our state space which allows this computations in practice. Therefore, **v** may be understood as performing k-means clustering on the CDF of the optimal allocation distribution (that follows from (5.1)).

To motivate the addition of bin size in the minimization problem, consider the selection variance (4.16) using the optimal allocation (4.14). As $N \to \infty$ and $\Delta t \to 0$, the intra-bin distribution η_t^u follows the optimal allocation distribution and $Kh(\hat{\xi}_t^i) \approx h(\hat{\xi}_t^i)$. Then the selection variance is given by

$$\mathbb{E}\left[\sum_{u\in\mathcal{B}} \left(\frac{1}{\mathbb{E}^{\eta_t^u}\left[v_{\Delta t}\right]}\right)^2 \mathrm{Var}_{\eta_t^u}\left(\sum_{i:\hat{\xi}_t^i\in u} h(\hat{\xi}_t^i)\right) \,\middle|\, \mathcal{F}_t\right]. \tag{5.3}$$

The expectation of $v_{\Delta t}$ term is the average value of $v_{\Delta t}$ in bin u. The inverse of this term would correlate to (optimized) bin size in the following way, regions where $v_{\Delta t}$ is large would have

smaller bins and vice versa. Hence, the selection variance follows bins size times a variance in evolution term which agrees with (5.2).

Remark 5.2. Example code for the procedures in **s** and **v** is included in the appendix, sections A.1 and A.2 respectively.

5.1.2 Adaptive Binning Method

For adaptive bins, it is more natural to consider $\mathcal{B}$ as a partition of the particles rather than the state space. Consider the parents $\{\xi_t^i\}_{i=1}^N$ at time t. Taking ν as a uniform distribution, we can solve (4.15) by k-means clustering on the set $\{h(\xi_t^i)\}$ with m centroids. This will result in a labeling for the parents that can be understood as partition $\mathcal{B}$. We will denote this method as **a**.

5.2 Allocation Methods

Across the set of bins $\mathcal{B} = \{u_i\}_{i=1}^N$ we tested four allocation methods. Let δ_u^t represent if a bin at time t, u, is occupied,

$$\delta_u^t := \begin{cases} 1 & \exists \xi_i^t \in u \\ 0 & \text{otherwise} \end{cases}.$$

Also, let N_u^t be the allocation for bin u at time t. From Definition 3.1.1, it is implicit that if $\delta_u^t = 1$ then $N_u^t \geq 1$. In each strategy, approximate allocations are presented. The requirement that $N_t(u) \in \mathbb{Z}^{\geq 0}$ means that care must be taken to ensure the population remains fixed at N. We will perform sampling by *residual sampling* as defined in Algorithm 8.1 of [41] which also discusses how to maintain the population in Algorithm 5.1 (of [41]). The choice of residual sampling is not required and several other valid choices such as multinomial sampling are possible.

Summarizing, denote the distributions below as $d_t(u)$, then, each bin u is allocated $\lfloor N d_t(u) \rfloor$. Let $n_t(u) = N - \sum_u \lfloor N d_t(u) \rfloor$ be the number of samples needed to maintain the population and $\epsilon_t(u) = N d_t(u) - \lfloor N d_t(u) \rfloor$. Then $n(t)$ samples can be drawn via multinomial sampling from the

distribution

$$\frac{\epsilon_t(u)}{\sum_{u\in\mathcal{B}}\epsilon_t(u)}.$$

Let $\mathfrak{u}$ denote the uniform allocation strategy, where each occupied bin is assigned approximately the same number of children. Define

$$d_t(u) = \frac{\delta_u^t}{\sum_{p\in\mathcal{B}}\delta_p^t}.$$

Let $\mathfrak{w}$ denote the strategy where a bin u is assigned a number of children proportional to its relative weight. Define

$$d_t(u) = \sum_{i:\xi_t^i\in u} w_i^t.$$

Similarly $\mathfrak{h}$ assigns a bin u a number of children proportional to the product of the weights and the absolute value of h of the parents $\xi_t^i \in u$. This is an approximation of the optimal allocation as bins with relatively high $w\cdot|h|$ would indicate a region that should be over-sampled. Define

$$d_t(u) = \frac{\sum_{i:\xi_t^i\in u} w_i^t|h(\xi_i^t)|}{\sum_{i=1}^N w_i^t|h(\xi_i^t)|}.$$

Finally, let $\mathfrak{v}$ denote the optimal allocation, as found in (4.14). Define

$$d_t(u) = \frac{\sum_{i:\xi_t^i\in u} w_i^t\, v(\xi_i^t)}{\sum_{i=1}^N w_i^t\, v(\xi_i^t)}.$$

Chapter 6

Numerical Example Setting

We will demonstrate practical implementations of the variance reduction techniques as outlined in Chapter 4. In this example, we consider overdamped Langevin dynamics

$$dX(t_t) = -\nabla V(X_t) + \sqrt{2\beta^{-1}} dW_t,$$

where $(W_t)_{t\geq 0}$ is standard Brownian motion with the 2D potential (see Figure 6.1)

$$\begin{aligned} V(x,y) = &\exp(-(50.5(x-0.25)^2 + 50.5(y-0.75)^2 + 2\cdot 49.5(x-0.25)(y-0.75))) \\ &+ \exp(-10^5(x^2(1-x)^2y^2(1-y)^2)) \\ &+ 0.5\exp(-(51x^2 + 51y^2 - 2\cdot 49xy)). \end{aligned} \tag{6.1}$$

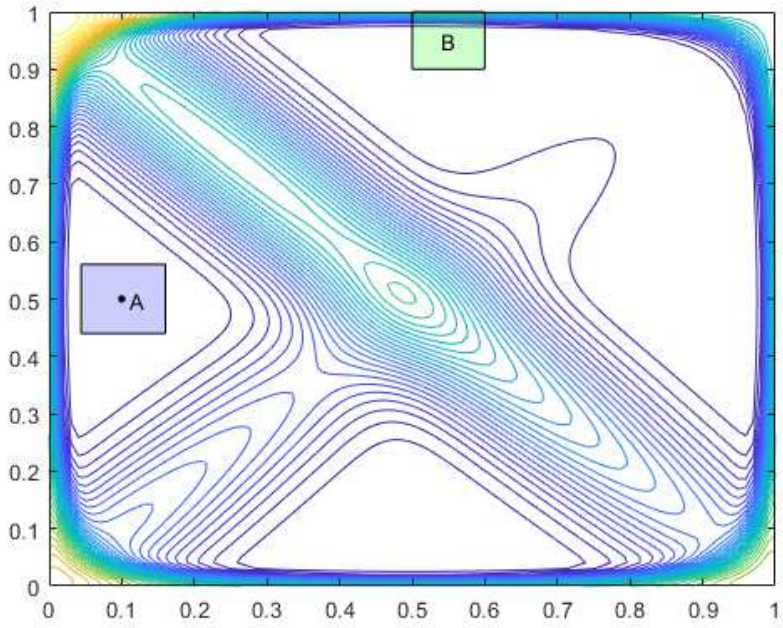


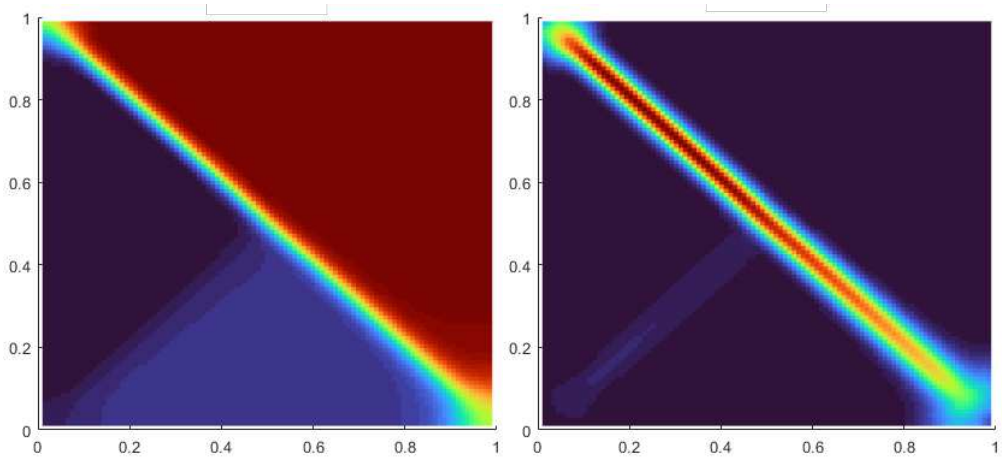
Figure 6.1: Contour plot for the potential $V(x,y)$. The source A is highlighted blue and the sink set B is highlighted green.

We initially set $\beta = 15$ and will briefly comment on results for $\beta = 30$. Higher values of β correspond to lower temperatures. This means that simulations will take significantly longer to converge for larger β. Simulations were performed using 28 cores and in the simplistic setting of this toy problem, to run 1000 trials for one set of parameters at $\beta = 15$ takes ~ 5 hours compared to ~ 100 hours at $\beta = 30$. Thus, the majority of our results use $\beta = 15$ to cover the most ground. However, at this temperature, the difference between direct Monte Carlo and the optimized weighted ensemble methods is not significant (for example, Figure 6.10 or Figure 6.13). For this reason, we include results for $\beta = 30$ where we see a more pronounced difference.

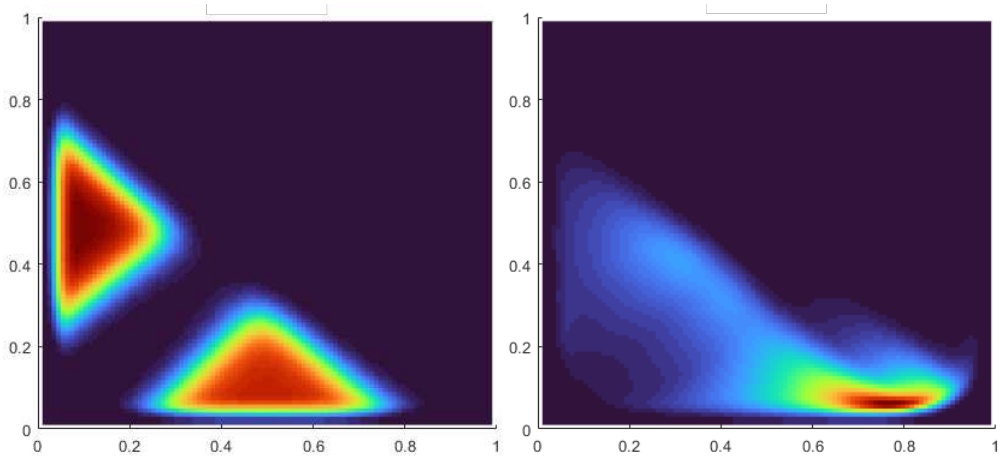
Weighted ensemble will be used to compute the MFPT between the source set $A = (.1, .5)$ to the sink set $B = [.5, .9] \times [.6, 1]$. Particles are initialized from a normal distribution centered at A and recycled immediately to A during the mutation step. $V(x, y)$ constrains the particles to $[0, 1]^2$ naturally so additional boundary conditions are not required. The advantage to working with a toy problem is that it is relatively quick to estimate the true MFPT, the Markov kernel K, and the steady state distribution μ. We take $f = \mathbb{1}_B$, the indicator function of the set B, and can numerically solve the poisson equation (4.12) to find estimates of h and v (Figure 6.2 (a) and (b) respectively).

The metastable regions of the potential, 6.1, are clear in Figure 6.2 (c). Notice in Figure 6.2 (a), that h is roughly constant inside the metastable regions. The basin containing B would also be metastable were it not for the sink and h is roughly constant here as well. The transitions between these regions is the only place where v is highly non-constant. From Section 4.2, the optimal allocation will follow $\mu(x, y) \cdot v(x, y)$. This allocation is given in Figure 6.2 (d) which indicates that particles will be placed preferentially near the transitions between the metastable states.

Naive weighted ensemble simulations of the MFPT for this problem simulate transitions directly from A to B. However, this does not accurately simulate the dynamics of the system at low temperatures. In the low temperature regime (large β), particles will transition to the intermediary meta stable set in the bottom before crossing the potential barrier in the bottom right. As naive



(a) $h(x, y)$ **(b)** $v(x, y)$

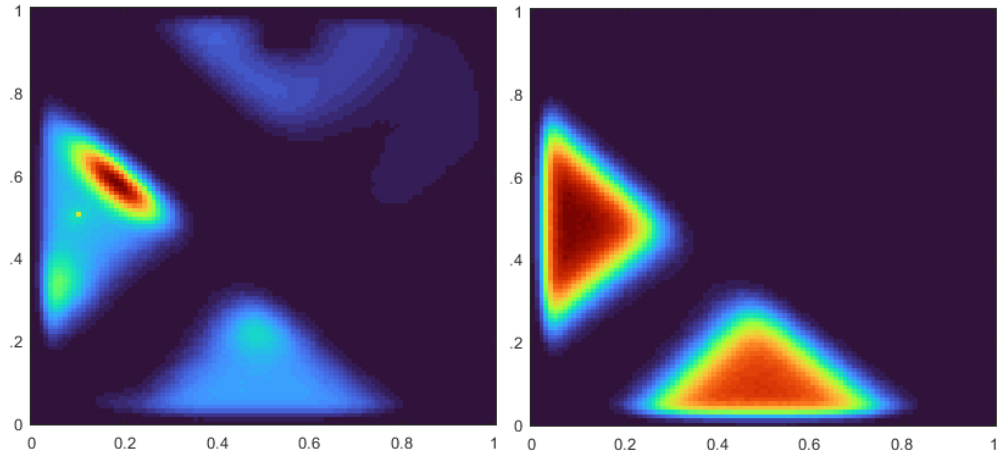


(c) $\mu(x, y)$ **(d)** $\mu(x, y) \cdot v(x, y)$

Figure 6.2: Numerical solutions for the relevant splitting coordinates h (a) and v (b) and the steady state distribution μ (c). The optimal allocation $\mu \cdot v$ (d) indicates particles will be placed along the correct transition pathway. Brighter (redder) areas correspond to higher values.

simulations do not capture this behavior, this leads to an instability in the estimate of the MFPT and, consequentially, higher variance.

Figure 6.3, which plots relative occupancy of space for weighted ensemble simulations, shows this behavior well. In Figure 6.3 (a), a naive weighted ensemble is shown to have most of the particles in the first metastable state. As the particles are binned without using properties of the underlying problem (for instance without using h), they are attempting to transition directly over the large potential barrier. Furthermore, once particles make the transition many particles remain "trapped" near the sink due to selection. In Figure 6.3 (b), naive Monte Carlo simulations again



(a) Naive WE **(b)** NMC

(c) Adaptive Optimal **(d)** Static Optimal

Figure 6.3: Comparison of naive weighted ensemble (a) and naive Monte Carlo (b) with optimized strategies (c) and (d). Each figure plots relative occupancy over a set of weighted ensemble simulations after relaxing to steady state, with brighter (redder) areas having higher occupancy. The correct dynamics are shown in (c) and (d) where particles transition to the metastable state at the bottom before crossing the large potential barrier.

evolve without knowledge of h. As there is no selection step, the particles spend the majority of the simulation time in the metastable sets without any clear progress.

In comparison to these uninformed simulations, Figure 6.3 (c) and Figure 6.3 (d) depict optimized binning schemes that show the correct dynamics clearly. In Figure 6.3 (c), an adaptive binning scheme is employed with optimal allocation. Here, the particles can clearly be seen transitioning in the bottom right with a large proportion in the second metastable set. In Figure 6.3

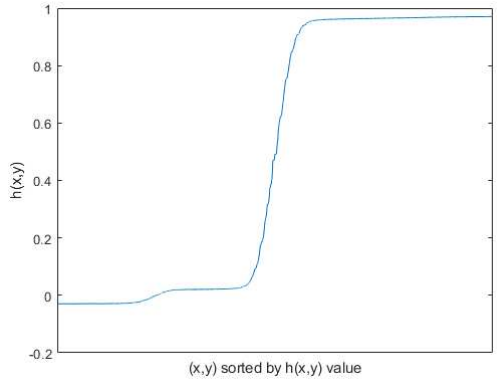


Figure 6.4: The sorted discrepancy function h. The first non-constant part roughly corresponds to a transition from metastable state on the left to the bottom. The second, larger transition is from the bottom metastable region to the basin containing B.

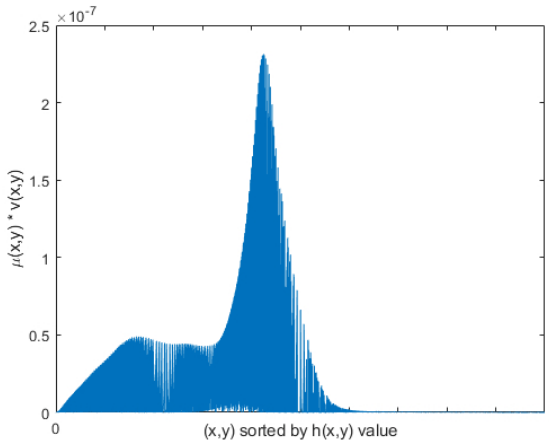


Figure 6.5: μv sorted by the discrepancy function. Notice the sharp spike in the distribution falls at the same place as the large barrier in h (Figure 6.4)

(d), a static binning scheme is employed with optimal allocation. Here, the particles clearly show a tendency to cluster where the transitions will most likely occur and can be seen transitioning in the bottom right as well. From Chapter 5, Figure 6.3 (c) depicts binning method **a** and Figure 6.3 (d) depicts binning method **v**.

Remark 6.1. h increases when transitioning between the left to the bottom metastable regions to the basin containing B with the latter transition corresponding to a much greater increase. This is clearly seen when sorting h as in Figure 6.4. It will be convenient to apply this sorting to other coordinates such as $\mu \cdot v$ to define bins in practice. The greatest variance in $\mu \cdot v$ is seen near the large barrier in h (Figure 6.5).

6.1 Bin Visualizations

Using the methods discussed in Chapter 5, we will apply these strategies to this toy problem. We initially set the number of bins to 6 for visualization and will vary this parameter later to explore the impact on the variance.

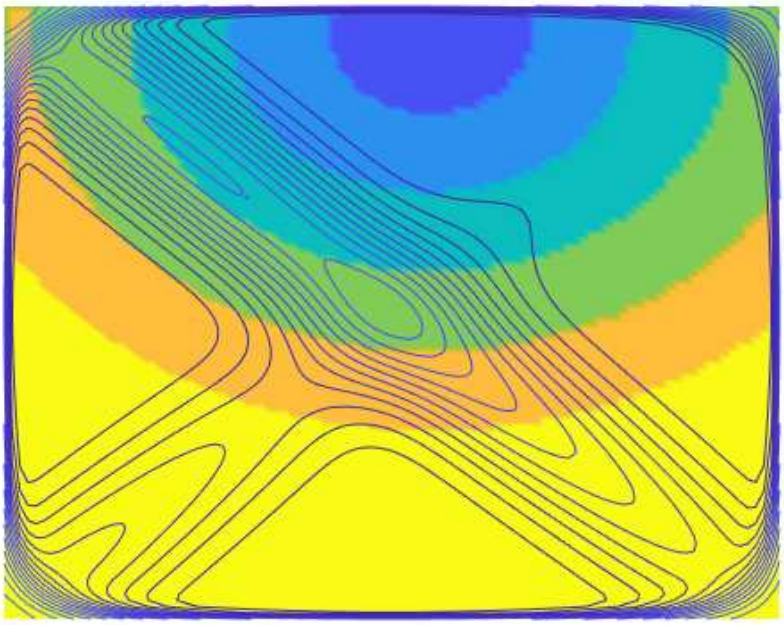


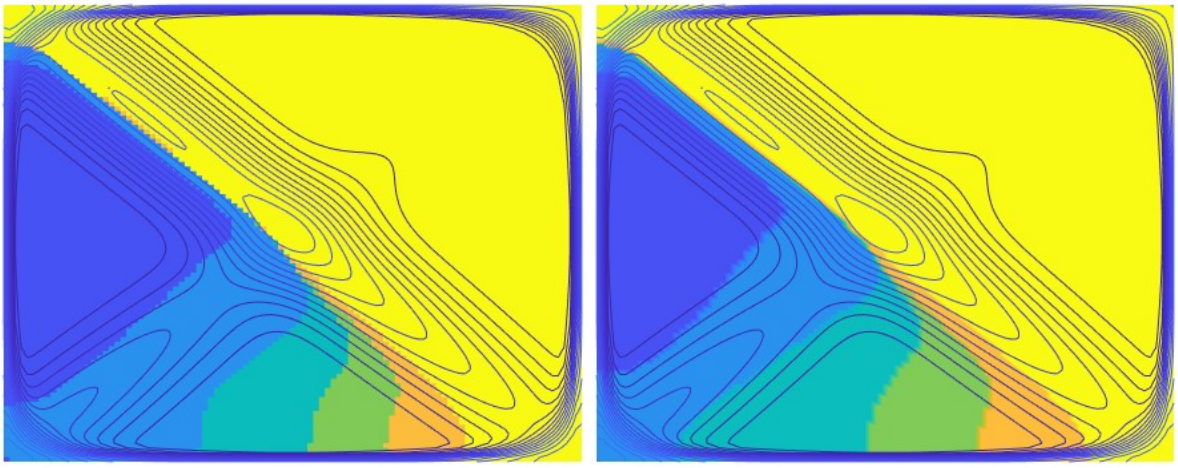
Figure 6.6: Strategy **u**. Different colored regions represent different bins.

The first static method presented in Section 5.1, **u**, is defined by bins that are created uniformly in RMSD space, Figure 6.6. This method does not use any knowledge about the problem though is commonly used in practice.

Next, two static optimized methods created level sets in sorted h space, $\mathcal{H}$ along with the CDF of the optimal allocation distribution

$$\int_{x:h(x)\in\mathcal{H}} \mu(x)v(x)\delta_x, \tag{6.2}$$

where δ_x is the delta function centered at the point $x \in \mathbb{R}^2$. By $x : h(x) \in \mathcal{H}$, we mean that we take the integral in (6.2) over $x \in \mathbb{R}^2$ such that $h(x)$ is increasing. The first of these methods, **s**, was defined using (6.2) and setting level sets in h space such that this distribution was uniform across all bins. The second, **v**, used (6.2) to phrase a version of (4.15) that was solved using k-means clustering. The bins produced from these methods are shown in Figure 6.7.



(a) Strategy **s** **(b)** Strategy **v**

Figure 6.7: Comparison of binning methods **s** and **v**. Different colored regions correspond to different bins.

Unsurprisingly, the bins of **s** and **v** are remarkably similar, though the difference between the two strategies is more explicit (Figure 6.8) when we consider the intra-bin distributions defined by

$$\gamma_t(u) \propto \sum_{i:\xi_t^i \in u} \mu(\xi_t^i) v(\xi_t^i) \delta_{\xi_t^i}.$$

Particularly for the bins near the large jump in h, it is clear that **v** has produced bins with smaller variance with respect to μv. In Section 6.3, **v** will provide a greater reduction in the variance of weighted ensemble estimations.

Lastly, recall the adaptive method **a** which creates bins by k-means clustering on the set $\{h(\xi_t^i)\}_{i=1}^N$.

6.2 Allocations

In Section 5.2, several allocation strategies were laid out. Shortly summarizing, $\mathfrak{u}$ denotes the uniform allocation where each occupied bin is assigned the same number of children. In $\mathfrak{w}$, the allocation in each bin is proportional to the weight of the bin and in $\mathfrak{h}$ it is proportional to $\sum_{i:\xi_t^i \in u} \omega_t^i |h(\xi_t^i)|$. Finally, $\mathfrak{v}$ denotes the optimal allocation from (4.14) where the allocation follows $\sum_{i:\xi_t^i \in u} \omega_t^i v(\xi_t^i)$.

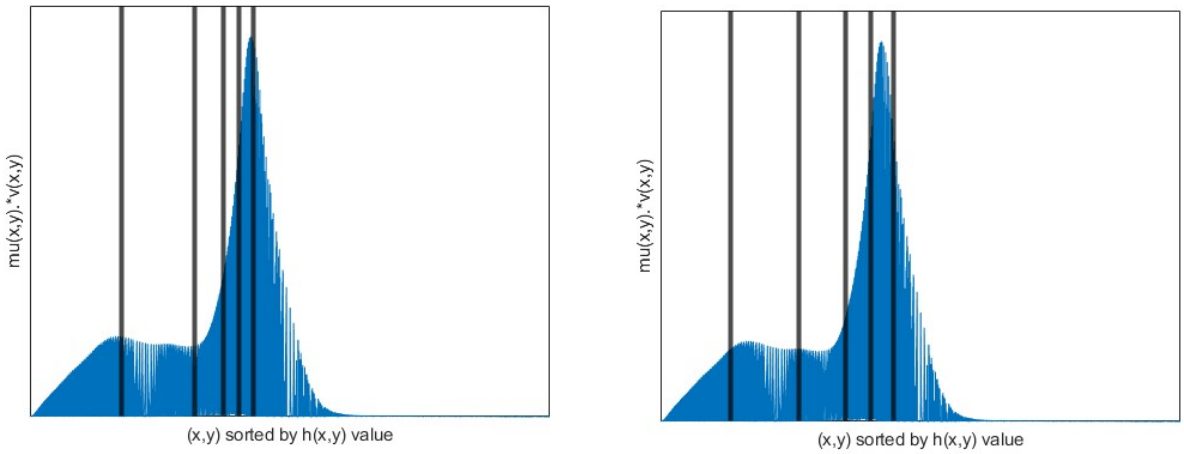


Figure 6.8: Left: binning by **s**. Right: binning by **v**. Plots depict μv values across sorted by h with the levels sets of the bins represented by the black bars.

6.3 Results

It is common in practice to uniformly bin and allocate particles in spatial bins that are separated by level sets of RMSD to B [44, 46–48]. We will refer specifically to this method as RMSD weighted ensemble (or RMSD WE). We will also refer to the special case outlined in Remark 3.3 as direct sampling or direct/naive Monte Carlo.

We will begin by showing that the optimized methods provide a variance reduction over RMSD weighted ensemble at $\beta = 15$ and discuss the impact of the parameters N and m. We then conclude by showing that at $\beta = 30$, the performance of RMSD weighted ensemble is still worse than direct sampling. However, at this temperature, direct sampling is significantly worse than the optimized method: binning by **v** and allocating by $\mathfrak{v}$ (see Figure 6.9 and Table 6.5).

Remark 6.2. In each set of trials, the number of selection steps T was chosen large enough to ensure that all trials converged to the estimation of the exact inverse MFPT, $\mathcal{J}$ (Figure 6.10). A numerical estimation of $\mathcal{J}$ was found by relaxation and confirmed by direct Monte Carlo. Then, for $\mathcal{N}$ trials

$$\frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \theta_T^i \approx \mathcal{J}.$$

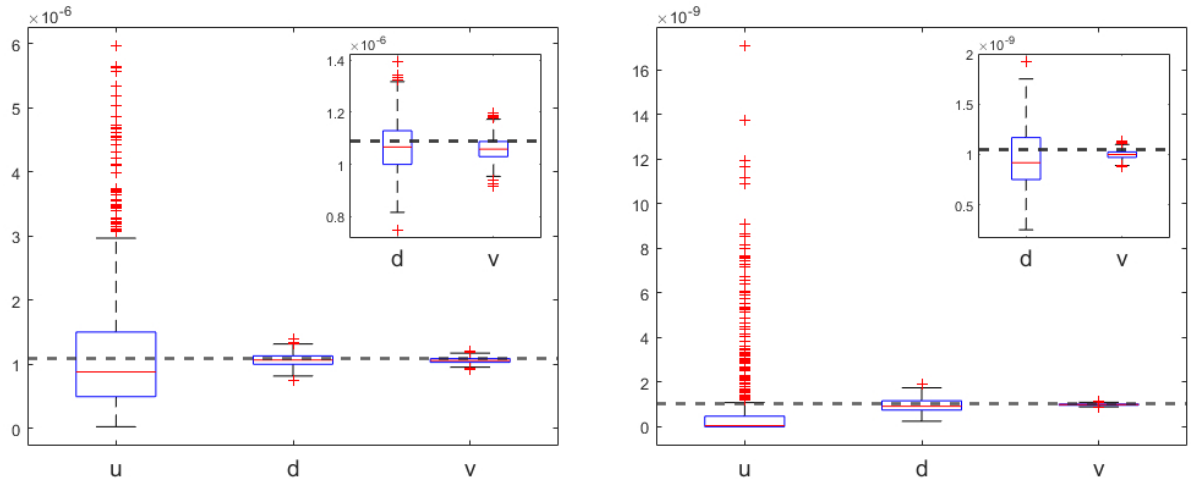


Figure 6.9: Box and whisker plots on the estimation of the rate constant in Table 6.4 comparing the performance of RMSD WE **u**, direct MC **d**, and **v** with optimal allocation. The estimation of the exact average is depicted by the dashed line. Left: $\beta = 15$, no significant gain over direct MC. Right: $\beta = 30$, a significant variance reduction is observed and summarized in Table 6.5.

We categorize the performance of a weighted ensemble implementation by

$$\sigma = \sigma_T / \sqrt{\mathcal{N}}$$

where σ_T is the sample standard deviation.

In Table 6.1, we begin by comparing RMSD weighted ensemble to the optimization methods detailed previously. These preliminary tests demonstrate a significant reduction in variance is possible. By design, both **s** and **v** have similar performance for uniform and optimal allocations. Interestingly, **a** performs best when allocating proportionally to the weights or to $|h|$.

Relative to our problem, $N = 500$ is still a large population. To simulate a more restrictive case, we also consider $N = 40$ to demonstrate that a similar variance reduction is still achievable. Additionally, the number of bins will affect the performance of the binning methods differently. Intuitively, we would expect static methods to suffer compared adaptive methods at low bin counts as there isn't enough resolution in the initialization step to accurately predict the dynamics of the problem. This intuition is supported by Tables 6.2 and 6.3.

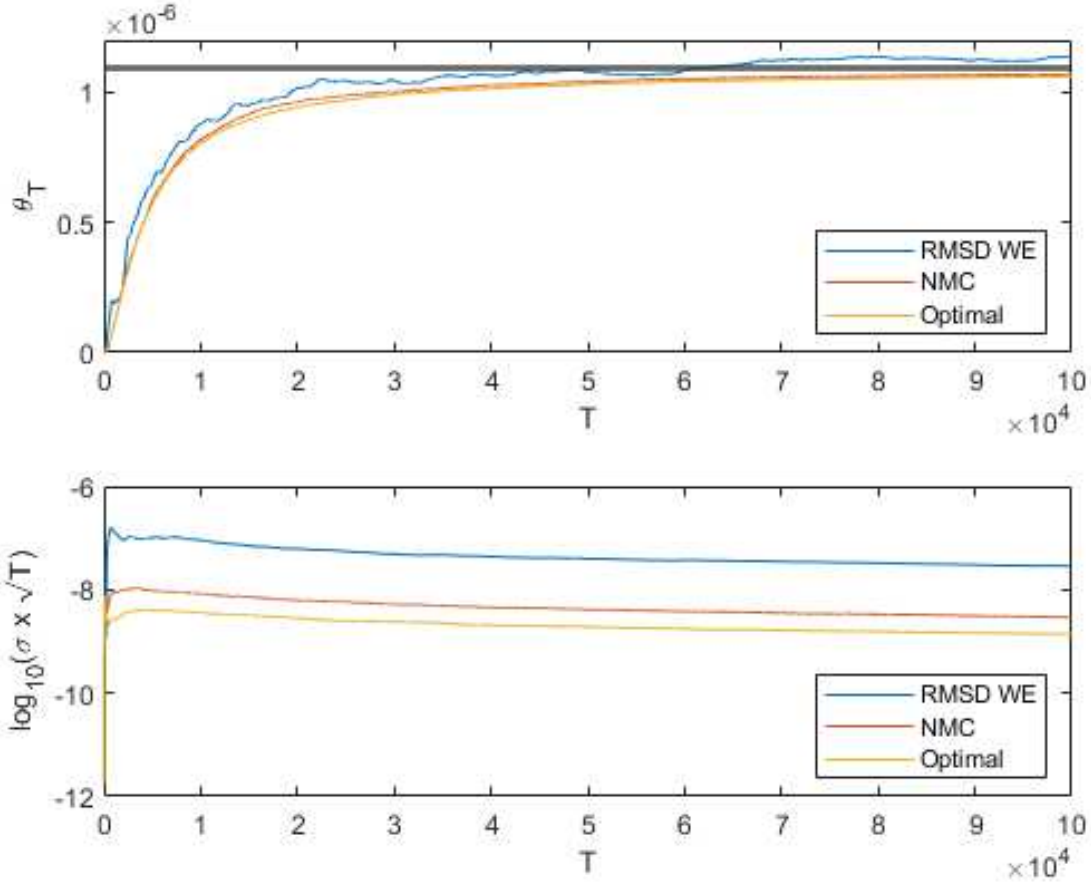


Figure 6.10: Example of convergence of weighted ensemble estimates for RMSD weighted ensemble (RMSD WE), direct sampling or naive Monte Carlo (NMC), and binning by $\mathbf{v}$ with allocation $\mathfrak{v}$ (Optimal). Top: Plotted as the number of selection steps T vs the estimate θ_T, the estimate of exact average is represented by the black bar. Bottom: Convergence of the standard deviation $\sigma_T/\sqrt{\#\text{ Trials}}$ scaled by $\sqrt{T}$.

These results also suggest that static binning methods will begin to outperform adaptive methods for higher bin counts. As noted above, this may be an artifact of the toy problem under consideration and we do not claim that this result holds in higher dimensions. Though, it is certainly true that more bins may be used in static methods. This is a direct consequence of the adaptive method **a** being indistinguishable from direct sampling for $m \geq N$. Figure 6.11 gives an example how the number of bins affects the performance for the optimized binning methods. As more bins will reduce the selection variance, static binning methods may remain an effective choice in higher dimensions if computational resources allow, so long as the number of bins does not significantly exceed the number of particles. However, it is possible that in higher dimensions, adaptive binning

Table 6.1: The results above were obtained over 1000 trials with $N = 500$, $m = 6$, $\beta = 15$, $\delta t = .001$, $\Delta t = 10$, and 10^5 selection steps. Values reported as $\sigma_T/\sqrt{1000} \times 10^{-9}$.

Strategy	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{h}$	$\mathfrak{v}$
u	20.6	12.6	14.9	15.7
a	7.30	1.06	.928	1.81
s	1.39	4.06	1.52	1.64
v	.624	2.17	1.03	.776

schemes will be more robust. For this problem at least, Figure 6.11 suggests that there is an optimal number of bins to choose. This was not investigated and in general, choosing the best number of bins remains largely heuristic.

Table 6.4 gives a stronger idea of the improvement of optimized methods over RMSD weighted ensemble. We set $N = 100$ and performed small samples to determine that $m = 10$ gave the lowest variance. For the optimized binning methods we chose the optimal allocation $\mathfrak{v}$. Table 6.4 shows a variance reduction by a factor of 900 when binning by **v** for our problem. Figure 6.12 also shows the stark performance difference. RMSD weighted ensemble's large variance is due to its inability to capture the dynamics of the underlying system correctly (Figure 6.3).

This inability to accurately capture the underlying dynamics can lead to a problem beyond performing worse than even direct Monte Carlo. In practice, RMSD weighted ensemble may appear to converge to an incorrect estimate if not enough time steps are allowed. From Chapter 3, the estimator θ_T is unbiased and ergodic so RMSD weighed ensemble remains valid though care must be taken to ensure convergence.

Another significant question to address is whether these improvements over RMSD weighted ensemble offer similar improvements over direct sampling. At $\beta = 15$, there is marginal improvement but direct sampling also significantly outperformed RMSD weighted ensemble at this temperature. A smaller test at $\beta = 30$ (Table 6.5) found the performance of direct sampling is still better than RMSD weighted ensemble but that at this temperature, the best performing optimal

Table 6.2: The results above were obtained over 1000 trials with $N = 40$, $\beta = 15$, $\delta t = .001$, $\Delta t = 10$, and 10^5 selection steps. Values reported as $\sigma_T/\sqrt{1000} \times 10^{-9}$.

	$m = 2$			$m = 4$			$m = 6$			$m = 8$		
	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$
u	35.0	$*$	$*$	38.9	$*$	$*$	37.1	$*$	$*$	36.3	$*$	$*$
a	9.89	6.89	7.95	9.83	3.45	6.26	10.4	3.13	5.16	10.9	2.95	3.99
s	26.8	27.0	27.4	11.5	13.0	11.0	5.53	7.69	4.61	3.98	5.91	2.48
v	25.3	25.1	24.4	7.88	11.0	7.73	3.06	5.09	1.86	2.46	4.18	1.40

Table 6.3: The results above were obtained over 1000 trials with $N = 500$, $\delta t = .001$, $\beta = 15$, $\Delta t = 10$, and 10^5 selection steps. Values reported as $\sigma_T/\sqrt{1000} \times 10^{-9}$.

	$m = 2$			$m = 4$			$m = 6$			$m = 8$		
	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$	$\mathfrak{u}$	$\mathfrak{w}$	$\mathfrak{v}$
u	15.3	$*$	$*$	17.9	$*$	$*$	20.6	$*$	$*$	23.1	$*$	$*$
a	6.91	5.09	5.45	6.92	1.20	2.38	7.30	1.06	1.81	7.22	1.00	1.26
s	14.1	16.2	13.0	3.47	8.12	3.92	1.39	4.06	1.64	.853	2.78	1.08
v	11.1	15.1	11.0	2.34	6.70	2.92	.624	2.17	.776	.504	1.53	.552

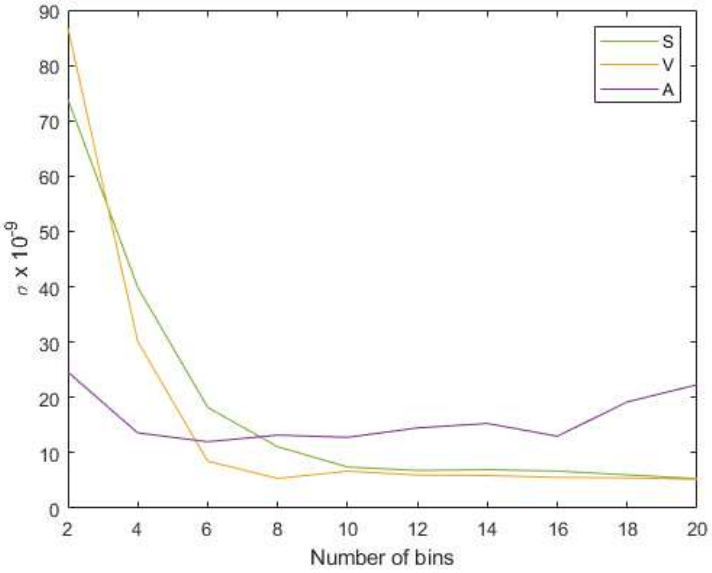


Figure 6.11: Performance of optimized strategies against the number of bins over 100 trials. S denotes binning by **s** with optimal allocation, V denotes binning by **v** with optimal allocation, and A denotes binning by **a** with allocation proportional to weights. 20 particles were used with $\delta t = .001$, $\Delta t = 10$, and 10^5 selection steps.

Table 6.4: RMSD weighted ensemble **u** against binning by **s** and **v** with optimal allocation. 10^4 trials with $N = 100$, $m = 10$, $\beta = 15$, $\delta t = .001$, $\Delta t = 10$, and 10^5 selection steps.

	$\sigma_T/\sqrt{10^4} \times 10^{-10}$
u	85.42
s	4.790
v	2.843

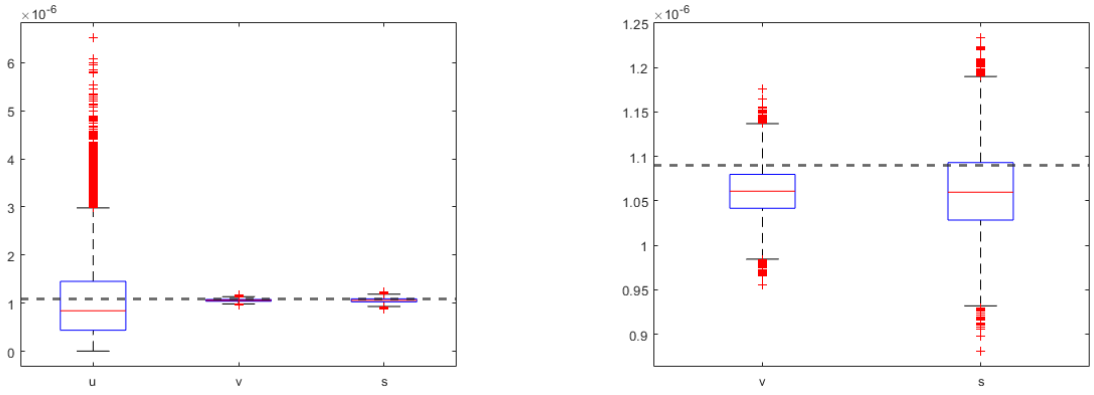


Figure 6.12: Box and whisker plots on the estimation of the rate constant in Table 6.4 versus the strategy employed. The estimation of the exact average is depicted by the dashed line. Left: RMSD weighted ensemble versus optimized implementations. Right: Comparison of **v** and **s** (both with optimal allocation).

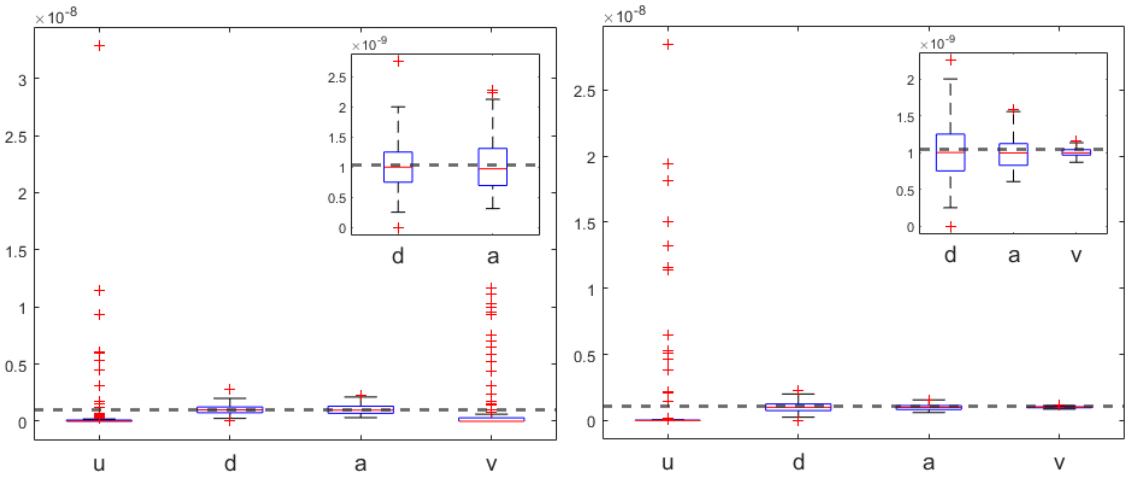
method achieves a significant reduction in variance. These tests converge significantly slower and out of convenience we limited our discussion above to the $\beta = 15$ case.

Table 6.5: Results of 500 trials for $\beta = 30$. The results report $\sigma_T/\sqrt{500}$ for $N = 120$, $m = 6$, $\delta t = .001$, $\Delta t = 10$, and 10^7 selection steps for RMSD WE **u**, direct MC **d**, and binning and allocating by **v**.

	σ
u	1.86×10^{-10}
d	1.24×10^{-11}
v	1.89×10^{-12}

We also investigated the impact of the number of bins on the performance of weighted ensemble at $\beta = 30$ and for a low population of $N = 40$. Similar to Tables 6.2 and 6.3, with a small number of bins, **a** has the best performance and **v** performs poorly; with more bins, **v** performs the best (see Figure 6.13). With 2 bins, both RMSD weighed ensemble and **v** had many samples that recorded no flux which highlights the potential benefit to using an adaptive binning scheme.

For molecular dynamics, the evolution step will consume the majority of computational resources. This means that k-means clustering, a potentially expensive algorithm, will not be a burden on the simulation time. However, for this problem with 10 bins, **v** significantly outperforms all methods, $\sigma_{\boldsymbol{v}} = 6.10 \times 10^{-12}$, compared to the adaptive method $\sigma_{\boldsymbol{a}} = 2.15 \times 10^{-11}$.



(a) $m = 2$ **(b)** $m = 10$

Figure 6.13: Box and whisker plots for the estimation of the rate constant with 100 trials, $N = 40$, $\beta = 30$, $\delta t = .001$, $\Delta t = 10$, and 10^7 selection steps for RMSD WE **u**, direct MC **d**, **a** with $\mathfrak{w}$, and **v** with $\mathfrak{v}$.Left: 2 bins. Right: 10 bins.

Chapter 7

Conclusion

This thesis presented the framework for the weighted ensemble algorithm in the context of computing the mean first passage time. We showed that the variance of weighted ensemble can be decomposed in terms of the steps of the algorithm, the initialization, selection, and mutation steps. Using these formulas, the key coordinates for controlling the selection variance were identified. From these coordinates, variance reduction strategies were constructed and implemented in a numerical example. In this example, we found that a significant reduction in variance was achievable in comparison to both naive Monte Carlo estimation and the industry standard RMSD weighted ensemble. The significance of this result lies in reducing the computation time of estimating the MFPT.

Notably, the variance minimization theories were derived in the large N and T regime. In practice, the particle count may be severely limited, though we showed that the techniques can still improve the variance in the estimation. It remains to be seen if there are better strategies for problems with few particles and bins. Extending the results of the toy problem to problems in molecular dynamics would also be a next step. It is possible that a variation of k-means, the MinMax k-means algorithm [49], may provide additional benefits. The MinMax k-means algorithm would minimize the worst intra-bin variance rather than the cumulative variance.

Beyond simply seeing success with the minimization techniques at finite N, the variance improvement factor over RMSD weighted ensemble stays roughly the same in our tests. Therefore, simulations with relatively few particles may still see similar reductions in variance when implementing the strategies we have proposed. It should be noted that the choice of allocation scheme may depend on the binning method. In particular, adaptive binning on h-space through k-means clustering performs best when allocating by the weights of the bins or the weights scaled by h.

To reach a desired accuracy, increasing the population will allow a greater number of bins which will further reduce the variance. Our tests show that there may be an optimal number of

bins to choose for a particular population but do not offer any guidance on how to choose this number.

We also assumed a constant resampling time of Δt throughout. This parameter will affect the selection variance and its best choice does not have rigorous mathematical guidance. There is a balance in choosing Δt due to how the selection and mutation variances are affected. As the selection variance will depend on the similarity of the particles being merged, longer Δt lag times will result in larger variance, so long as some resampling steps occur. Instead, Δt should be short enough to take advantage of the weighted ensemble algorithm. Through binning and selection, particles in the weighed ensemble algorithm can "ratchet" their way over large potential barriers which reduces the mutation variance. In general, the best choice of parameters for weighted ensemble remains open.

Bibliography

[1] Jenny Nelson and Rosemary E. Chandler. Random walk models of charge transfer and transport in dye sensitized systems. *Coordination Chemistry Reviews*, 248(13-14):1181–1194, July 2004.

[2] David R. Evans, Hyunwook S. Kwak, David J. Giesen, Alexander Goldberg, Mathew D. Halls, and Masahito Oh-e. Estimation of charge carrier mobility in amorphous organic materials using percolation corrected random-walk model. *Organic Electronics*, 29:50–56, February 2016.

[3] G.A. Huber and S. Kim. Weighted-ensemble brownian dynamics simulations for protein association reactions. *Biophysical Journal*, 70(1):97–110, January 1996.

[4] Bibhab Bandhu Majumdar, Vera Prytkova, Eric K. Wong, J. Alfredo Freites, Douglas J. Tobias, and Matthias Heyden. Role of conformational flexibility in monte carlo simulations of many-protein systems. *Journal of Chemical Theory and Computation*, 15(2):1399–1408, January 2019.

[5] Matthew C. Zwier, Adam J. Pratt, Joshua L. Adelman, Joseph W. Kaus, Daniel M. Zuckerman, and Lillian T. Chong. Efficient atomistic simulation of pathways and calculation of rate constants for a protein–peptide binding process: Application to the MDM2 protein and an intrinsically disordered p53 peptide. *The Journal of Physical Chemistry Letters*, 7(17):3440–3445, August 2016.

[6] Anu Nagarajan, Juan R. Perilla, and Thomas B. Woolf. Understanding the conformational changes in ca-APTase using coarse-grained and all-atom simulations with dynamic importance sampling. *Biophysical Journal*, 96(3):430a, February 2009.

[7] Lillian T Chong, Ali S Saglam, and Daniel M Zuckerman. Path-sampling strategies for simulating rare events in biomolecular systems. *Current Opinion in Structural Biology*, 43:88–94, April 2017.

[8] Ivan Teo, Christopher G. Mayne, Klaus Schulten, and Tony Lelièvre. Adaptive multilevel splitting method for molecular dynamics calculation of benzamidine-trypsin dissociation time. *Journal of Chemical Theory and Computation*, 12(6):2983–2989, May 2016.

[9] David E. Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, Michael P. Eastwood, Joseph A. Bank, John M. Jumper, John K. Salmon, Yibing Shan, and Willy Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, October 2010.

[10] Gongpu Zhao, Juan R. Perilla, Ernest L. Yufenyuy, Xin Meng, Bo Chen, Jiying Ning, Jinwoo Ahn, Angela M. Gronenborn, Klaus Schulten, Christopher Aiken, and Peijun Zhang. Mature HIV-1 capsid structure by cryo-electron microscopy and all-atom molecular dynamics. *Nature*, 497(7451):643–646, May 2013.

[11] W. W. Cleland. Partition analysis and concept of net rate constants as tools in enzyme kinetics. *Biochemistry*, 14(14):3220–3224, July 1975.

[12] Vishal Singh and Parbati Biswas. Estimating the mean first passage time of protein misfolding. *Physical Chemistry Chemical Physics*, 20(8):5692–5698, 2018.

[13] Nicholas F. Polizzi, Michael J. Therien, and David N. Beratan. Mean first-passage times in biology. *Israel Journal of Chemistry*, 56(9-10):816–824, August 2016.

[14] Derek Y. C. Chan and Donald A. McQuarrie. Mean first passage times of ions between charged surfaces. *Journal of the Chemical Society, Faraday Transactions*, 86(21):3585, 1990.

[15] B. Shotorban. First passage time in multistep stochastic processes with applications to dust charging. *Physical Review E*, 101(1), January 2020.

[16] Joshua L. Adelman and Michael Grabe. Simulating rare events using a weighted ensemble-based string method. *The Journal of Chemical Physics*, 138(4):044105, January 2013.

[17] Anders Lervik and Titus S. van Erp. Gluing potential energy surfaces with rare event simulations. *Journal of Chemical Theory and Computation*, 11(6):2440–2450, May 2015.

[18] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[19] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[20] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[21] Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, February 2007.

[22] Charles-Edouard Bréhier, Tony Lelièvre, and Mathias Rousset. Analysis of adaptive multilevel splitting algorithms in an idealized case. *ESAIM: Probability and Statistics*, 19:361–394, 2015.

[23] Frédéric Cérou, Arnaud Guyader, and Mathias Rousset. Adaptive multilevel splitting: Historical perspective and recent results. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(4):043108, April 2019.

[24] Piere Del Moral and Arnaud Doucet. Particle methods: An introduction with applications. *ESAIM: Proceedings*, 44:1–46, January 2014.

[25] N. Kantas, A. Doucet, S.S. Singh, and J.M. Maciejowski. An overview of sequential monte carlo methods for parameter estimation in general state-space models. *IFAC Proceedings Volumes*, 42(10):774–785, 2009.

[26] S. Chandrasekhar. Stochastic problems in physics and astronomy. *Reviews of Modern Physics*, 15(1):1–89, January 1943.

[27] Timothy DelSole. A fundamental limitation of markov models. *Journal of the Atmospheric Sciences*, 57(13):2158–2168, July 2000.

[28] Brooke E. Husic and Vijay S. Pande. Markov state models: From an art to a science. *Journal of the American Chemical Society*, 140(7):2386–2396, February 2018.

[29] M. C. Zwier, J. L. Adelman, J. W. Kaus, A. J. Pratt, K. F. Wong, N. S. Rego, E. Suárez, S. Lettieri, D. W. Wang, M. Grabe, D. M. Zuckerman, and L. T. Chong. Westpa. https://westpa.github.io/westpa/, 2015.

[30] Matthew C. Zwier, Joshua L. Adelman, Joseph W. Kaus, Adam J. Pratt, Kim F. Wong, Nicholas B. Rego, Ernesto Suárez, Steven Lettieri, David W. Wang, Michael Grabe, Daniel M. Zuckerman, and Lillian T. Chong. WESTPA: An interoperable, highly scalable software package for weighted ensemble simulation and analysis. *Journal of Chemical Theory and Computation*, 11(2):800–809, January 2015.

[31] Anthony T. Bogetti, Barmak Mostofian, Alex Dickson, AJ Pratt, Ali S. Saglam, Paige O. Harrison, Joshua LL. Adelman, Max Dudek, Paul A. Torrillo, Alex J. DeGrave, Upendra Adhikari, Matthew C. Zwier, Daniel M. Zuckerman, and Lillian T. Chong. A suite of tutorials for the WESTPA rare-events sampling software [article v1.0]. *Living Journal of Computational Molecular Science*, 1(2), 2019.

[32] Terrell Hill. *Free Energy Transduction and Biochemical Cycle Kinetics*. Dover, Mineola, NY:, 2005.

[33] M. Torchala, P. Chelminiak, and P. A. Bates. Mean first-passage time calculations: comparison of the deterministic hill's algorithm with monte carlo simulations. *The European Physical Journal B*, 85(4), April 2012.

[34] Iddo Ben-Ari and Ross G. Pinsky. Spectral analysis of a family of second-order elliptic operators with nonlocal boundary condition indexed by a probability measure. *Journal of Functional Analysis*, 251(1):122–140, October 2007.

[35] Iddo Ben-Ari and Ross G. Pinsky. Ergodic behavior of diffusions with random jumps from the boundary. *Stochastic Processes and their Applications*, 119(3):864–881, March 2009.

[36] Ilie Grigorescu and Min Kang. Ergodic properties of multidimensional brownian motion with rebirth. *Electronic Journal of Probability*, 12(none), January 2007.

[37] Yuk J. Leung, Wenbo V. Li, and Rakesh. Spectral analysis of brownian motion with jump boundary. *Proceedings of the American Mathematical Society*, 136(12):4427–4436, June 2008.

[38] Ross Pinsky. Spectral analysis of a class of nonlocal elliptic operators related to brownian motion with random jumps. *Transactions of the American Mathematical Society*, 361(9):5041–5060, April 2009.

[39] David Aristoff. An ergodic theorem for the weighted ensemble method. *Journal of Applied Probability*, pages 1–15, January 2022.

[40] Divesh Bhatt, Bin W. Zhang, and Daniel M. Zuckerman. Steady-state simulations using weighted ensemble path sampling. *The Journal of Chemical Physics*, 133(1):014110, July 2010.

[41] David Aristoff and Daniel M. Zuckerman. Optimizing weighted ensemble sampling of steady states. *Multiscale Modeling & Simulation*, 18(2):646–673, January 2020.

[42] Ernesto Suárez, Joshua L. Adelman, and Daniel M. Zuckerman. Accurate estimation of protein folding and unfolding times: Beyond markov state models. *Journal of Chemical Theory and Computation*, 12(8):3473–3481, July 2016.

[43] Bin W. Zhang, David Jasnow, and Daniel M. Zuckerman. Efficient and verified simulation of a path ensemble for conformational change in a united-residue model of calmodulin. *Proceedings of the National Academy of Sciences*, 104(46):18043–18048, November 2007.

[44] Matthew C. Zwier, Joseph W. Kaus, and Lillian T. Chong. Efficient explicit-solvent molecular dynamics simulations of molecular association kinetics: Methane/methane, na+/cl-, methane/benzene, and k+/18-crown-6 ether. *Journal of Chemical Theory and Computation*, 7(4):1189–1197, February 2011.

[45] Ali S. Saglam and Lillian T. Chong. Highly efficient computation of the basal k_{on} using direct simulation of protein–protein association with flexible molecular models. *The Journal of Physical Chemistry B*, 120(1):117–122, December 2015.

[46] Ernesto Suárez, Steven Lettieri, Matthew C. Zwier, Carsen A. Stringer, Sundar Raman Subramanian, Lillian T. Chong, and Daniel M. Zuckerman. Simultaneous computation of dynamical and equilibrium information using a weighted ensemble of trajectories. *Journal of Chemical Theory and Computation*, 10(7):2658–2667, March 2014.

[47] Joshua L. Adelman and Michael Grabe. Simulating current–voltage relationships for a narrow ion channel using the weighted ensemble method. *Journal of Chemical Theory and Computation*, 11(4):1907–1918, March 2015.

[48] Ernesto Suárez, Adam J. Pratt, Lillian T. Chong, and Daniel M. Zuckerman. Estimating first-passage time distributions from weighted ensemble simulations and non-markovian analyses. *Protein Science*, 25(1):67–78, September 2015.

[49] Grigorios Tzortzis and Aristidis Likas. The MinMax k-means clustering algorithm. *Pattern Recognition*, 47(7):2505–2516, July 2014.

[50] David B Hitchcock. A history of the metropolis–hastings algorithm. *The American Statistician*, 57(4):254–257, November 2003.

[51] Grigorios A. Pavliotis. *Stochastic Processes and Applications*. Springer New York, 2014.

[52] Bin W. Zhang, David Jasnow, and Daniel M. Zuckerman. The "weighted ensemble" path sampling method is statistically exact for a broad class of stochastic processes and binning procedures. *The Journal of Chemical Physics*, 132(5):054107, February 2010.

[53] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

Appendix A

Supplementary Code

A.1 Code: Levels for binning strategy s

(Using MATLAB functions) - Recall, h, μ, v are vectors representing the exact function value at even points in $[0,1]^2$ space and m is the number of bins.

Data: h, μ, v
[SH, inds] $\leftarrow$ sort(h) ; /* sort h and store permutation */
alloc $\leftarrow$ μ(inds). $*$ v(inds) ; /* get allocation distribution */
cs_alloc $\leftarrow$ cumsum(alloc) ; /* roughly $\int \mu \cdot v$ */
cs_bins $\leftarrow$ linspace(cs_alloc(1), cs_alloc(end), $m+1$);
; /* cs_bins = bins with constant $\int \mu \cdot v$ */
cs_bins $\leftarrow$ (levels(2 : end -1)) ; /* set upper/lower bound to ∞ */
;
; /* convert μv bins in h space */
levels $\leftarrow$ zeros($m-1, 1$);
for *i=1:m-1* **do**
  levels(i) $\leftarrow$ find(cs_alloc $>$ cs_bins(i), 1); /* find 1^{st} instance of logical */
end

A.2 Code: Levels for binning strategy v

(Using MATLAB functions) - Recall, h, μ, v are vectors representing the exact function value at even points in $[0,1]^2$ space and m is the number of bins.

Data: h, μ, v
[SH, inds] $\leftarrow$ sort(h) ; /* sort h and store permutation */
alloc $\leftarrow$ μ(inds). $*$ v(inds) ; /* get allocation distribution */
cs_alloc $\leftarrow$ cumsum(alloc) ; /* roughly $\int \mu \cdot v$ */

k_bins $\leftarrow$ kmeans(cs_alloc, m);
; /* kmeans to minimize the intrabin variance */
;
; /* convert μv bins in h space */
levels $\leftarrow$ zeros(m, 1);
for *i=1:m-1* **do**
  levels(i) $\leftarrow$ SH(find(k_bins $== i$, 1)); /* find 1^{st} instance of logical */
end
levels $\leftarrow$ (sort(levels))(2 : end)