OPTIMIZING WEIGHTED ENSEMBLE SAMPLING OF STEADY STATES

DAVID ARISTOFF† AND DANIEL M. ZUCKERMAN‡

Abstract. We show how to optimize resampling methods for Markov chains based on stratification, our aim being to reduce variance in steady-state computations. We consider collections of replicas of a Markov chain, with each replica carrying a weight, and where the replicas occupy a number of strata, or bins. We refer to this as as weighted ensemble (WE), after a resampling method from the computational chemistry literature. Traditionally, WE refers to resampling based on an ad hoc rule: maintaining a roughly spatially uniform distribution of replicas in the bins. We derive, from first principles, a strategy for optimizing the replica allocation for a pre-selected set of bins. In numerical tests on simple problems, our optimized allocation strategy outperforms uniform allocation and naive sampling. We further propose a synthetic variance calculation that may be used to help choose WE parameters, including the collection of bins.

Key words. Markov chains, resampling, sequential Monte Carlo, weighted ensemble, molecular dynamics, reaction networks, steady state, coarse graining

AMS subject classifications. 65C05, 65C20, 65C40, 65Y05, 82C80

1. Introduction. Weighted ensemble (WE) is a sampling method for stochastic dynamics based on stratification and resampling, originally proposed by Huber [18, 21, 22] and later refined by Zuckerman and co-authors [4, 5, 12, 28, 29, 31]; see [30] for software and a list of related publications. WE consists of periodically resampling from an ensemble of replicas of a Markov process, and adjusting the replica weights accordingly, so that the resulting distribution is unbiased [29]. The resampling is designed to keep a fixed number – usually the same number – of replicas in each stratum, or bin. We refer to this process of enforcing a target number of replicas in each bin as replica allocation.

WE is used to sample rare events [12] as well as steady states [4], where in the latter case the goal is often to compute a mean first passage time: the mean time for a Markov process to reach a certain set in state space. This article concerns replica allocation for computing steady-state averages. We show how to allocate replicas among the bins in a systematic way, using an optimization procedure based on first principles. We call this optimized allocation. By contrast, in uniform allocation the number of replicas in each occupied bin is roughly the same.

Examples of similar stratification methods include Exact Milestoning (EM) [3], Non-Equilibrium Umbrella Sampling (NEUS) [27, 11], Forward Flux Sampling [1], Transition Interface Sampling [24], Trajectory Tilting [25], and Boxed Molecular dynamics [16]. In those works, a low-dimensional projection of state space is divided into bins, and short trajectories create an effective Markov approximation of the bin-to-bin dynamics. Some of those works use the Markov approximation to accelerate convergence to steady state. To our knowledge, however, none of them use it directly to minimize variance, which is our goal here.

Usually in WE and the related methods listed above, the replica allocation is uniform. Though uniform allocation can work quite well, it is ad hoc and far from optimal in many cases. Our main result is a method – defined by (1)-(2) and Al-
Algorithm 2 below – for optimally allocating replicas among the bins. The algorithm uses a bin-to-bin Markov approximation of the dynamics to define the allocation. The variance can be reduced, compared to uniform allocation, even for a crude bin-to-bin Markov model; see the numerical results in Section 4.

Since WE is simply a resampling technique, it is very general and applies in settings beyond those mentioned above. Resampling algorithms for Markov chains have been studied in the context of particle filters, sequential importance sampling, and sequential Monte Carlo (SMC). For a review of SMC, see the textbook [8], the articles [9, 10] or the compilation [15]. To our knowledge, the allocation strategy we derive here is new. An alternative but similar approach can be found in [2].

This article is organized as follows. In Section 2, we sketch our WE algorithm and give a formula for our optimal replica allocation. In Section 3, we introduce a mathematical framework for WE and describe the algorithm in detail. Numerical examples comparing optimized allocation with uniform allocation are in Section 4. In Section 4 we also define a synthetic variance calculation that may be used to guide parameter choices, like the set of bins. We derive our optimal allocation strategy, and explain in what sense it is optimal, in Section 5.

2. Algorithm. A weighted ensemble (WE) consists of a collection of replicas, or particles, with associated weights, evolving between generations via repeated selection and mutation steps. In the resampling or selection step, the collection of particles, referred to as parents, is resampled to produce children, and the children’s weights are adjusted accordingly. In the evolution or mutation step, all the children evolve one step according to an underlying Markov kernel $K$, becoming the parents of the next generation. We assume $K$ has a unique stationary distribution $\mu$. The sum of all the particles’ weights is always 1, and the total number of particles is always $N$.

We are mainly interested in a version of WE for computing stationary or steady-state averages based on stratification. In this setup, each particle belongs to exactly one of a finite number of bins; all the particles in a given bin will be treated the same way, as we describe below. We are interested in computing $\int f \, d\mu$, the steady-state average of some real-valued observable $f$. Below, subscripts $t$ indicate quantities associated with the $t$-th generation of particles and weights. We write w.p.p.t. to denote with probability proportional to.

2.1. Algorithm sketch. We now give a brief overview of our WE algorithm. Some details will be sketched below; a precise explanation is in the next section.

Algorithm 1 (Bin-based WE algorithm).
Choose a total population $N$ and initial parents and weights. For $t = 0, 1, 2, \ldots$:

Selection. In each bin $r$, resample from parents w.p.p.t. their weights to get exactly $N_t(r)$ children; set the weight of all children in bin $r$ equal to $\omega_t(r)/N_t(r)$, where $\omega_t(r)$ is the bin weight, defined below.

Mutation. Evolve the children to get the next generation of parents. Then update the observable time average, set $t \leftarrow t + 1$, and return to Step 1.

Throughout, $\sum_r \omega_t(r) = 1$, $\sum_r N_t(r) = N$, and $N_t(r) \geq 1$ iff $\omega_t(r) > 0$.

The nonnegative integers $N_t(r)$ define the particle allocation: how many children we will sample in bin $r$ at generation $t$. Our optimized allocation is based on a value function $v_t(r)$ that we think of as the value of selecting children in bin $r$. To compute the value function, we use a bin-to-bin Markov model encoded by the probabilities $Q_t(r, s)$ for a child to go from bin $r$ to bin $s$ in Step 3 above, where the subscript $t$
indicates $Q_t$ can be updated with each generation. Notice that $K$ is a Markov kernel on the underlying state space of the particles, while $Q_t$ is a Markov approximation of the bin-to-bin dynamics of these particles. In that sense, $Q_t$ can be viewed as a coarse-grained approximation of $K$. The strategy we describe, however, can be applied whether or not the bin-to-bin dynamics are close to Markovian.

2.2. The value function. Let $Q_t = Q_t(r,s)$ be the stochastic matrix just described, and let $u(r)$ approximate the value of $f$ on bin $r$. Suppose there is a unique stationary distribution of $Q_t$, i.e. a column vector $\pi_t$ satisfying

$$
\pi_t^T Q_t = \pi_t^T, \quad \sum_r \pi_t(r) = 1.
$$

Assume also there is a unique column vector solution, $\phi_t$, to the Poisson equation

$$
\left( I - Q_t + \frac{\pi_t \pi_t^T}{\|\pi_t\|_2^2} \right) \phi_t = u - \pi_t^T u \mathbb{I}
$$

where $\mathbb{I}$ is the all 1’s column vector and $\| \cdot \|_2$ is the usual $L^2$ norm. Define

$$
v_t(r) = \text{rth entry of } \sqrt{Q_t(\phi_t^2) - (Q_t \phi_t)^2}
$$

where the square root and squares are entrywise. Above, $\pi_t$ and $\phi_t$ exist and are unique provided $Q_t$ is Harris ergodic [14], which is a consequence of the usual assumptions, e.g. aperiodicity and recurrence; see [20]. Below we will assume $\pi_t$ and $\phi_t$ exist and are unique for each $t \geq 0$. We think of $v_t(r)$ as the value of selecting particles in bin $r$; a derivation is in Section 5 below. Observe that

$$
v_t(r)^2 = \text{Var}_{Q_t(r,\cdot)}(\phi_t).
$$

Thus, there is a high value in allocating particles in bins with high variance with respect to our bin-to-bin Markov evolution $Q_t$ and the function $\phi_t$. We may assume $u$ is centered, i.e. $\pi_t^T u = 0$, without loss of generality, in which case $\phi_t = \sum_{s=1}^{\infty} Q_s^t u$.

Note the similarity of $\phi_t$ with the time average of $f$ that we aim to estimate.

2.3. Optimal particle allocation. Our optimized allocation is

$$
N_t(r) \approx \frac{N \omega_t(r) v_t(r)}{\sum_r \omega_t(r) v_t(r)}.
$$

Thus, the number of children in a bin is roughly proportional to the bin weight times the bin selection value. This formula will be derived in Section 5. Recall that uniform allocation means $N_t(r)$ takes roughly the same value in each occupied bin. This corresponds to taking $v_t(r) = 1/\omega_t(r)$ if $\omega_t(r) > 0$, and otherwise $v_t(r) = 0$, in (2).

2.4. Algorithm details. We elaborate on the steps in Algorithm 1:

- **Initialization:** There are $N$ initial particles, with weights summing to 1. We also begin with a bin-to-bin Markov transition matrix $Q_0$. Both $Q_0$ and the initial particles and weights could be obtained via, for example, a preliminary non-optimized WE simulation, though there are other possibilities. We discuss this more below.
Selection step: At the \( t \)-th selection step, we allocate \( N_t(r) \) particles to bin \( r \), as follows. We start by allocating 1 child to each occupied bin, where bin \( r \) is occupied if \( \omega_t(r) > 0 \). This defines the allocation of exactly \( M \) particles. The remaining \( N - M \) particles are assigned to bins with labels \( r \) drawn w.p.p.t. \( \omega_t(r) \)\( \nu_t(r) \). Note that this enforces \( \sum_r N_t(r) = N \), along with \( N_t(r) \geq 1 \) if and only if \( \omega_t(r) > 0 \). We define the bin weight as

\[
\omega_t(r) = \text{sum of weights of parents in bin } r.
\]

An alternative, which leads to a fixed point method that has been used in EM and NEUS, is to take \( \omega_t(r) = \pi_t(r) \). As with many fixed point methods, however, a good initial condition may be needed for convergence to the correct value. By contrast, our WE algorithm converges correctly regardless of the initialization, though our optimization scheme benefits from a good initial condition. Notice also our definition of bin weight enforces \( \sum_r \omega_t(r) = 1 \).

Mutation step: The children evolve independently according to the underlying Markov kernel \( K \), carrying weights assigned in the selection step. The weighted sum of \( f \) over the current WE generation is added to the observable time average. Notice \( Q_t \) could also be updated during this step by counting bin-to-bin transitions; we discuss this more below.

3. Mathematical framework. In this section we introduce mathematical notation for WE and describe Algorithm 1 in precise detail. In Algorithm 1, the \( t \)-th generation of particles and weights will be written, respectively,

\[ \xi^1_t, \ldots, \xi^N_t \text{ and } \omega^1_t, \ldots, \omega^N_t. \]

After selection, we denote the particles and weights with a “hat” symbol,

\[ \hat{\xi}^1_t, \ldots, \hat{\xi}^N_t \text{ and } \hat{\omega}^1_t, \ldots, \hat{\omega}^N_t. \]

The following diagram illustrates the evolution of a WE as in Algorithm 1:

\[
\begin{align*}
\{\xi^j_t\}_{j=1,...,N} & \xrightarrow{\text{selection}} \{\hat{\xi}^j_t\}_{j=1,...,N} \xrightarrow{\text{mutation}} \{\xi^j_{t+1}\}_{j=1,...,N}, \\
\{\omega^j_t\}_{j=1,...,N} & \xrightarrow{\text{selection}} \{\hat{\omega}^j_t\}_{j=1,...,N} \xrightarrow{\text{mutation}} \{\omega^j_{t+1}\}_{j=1,...,N}.
\end{align*}
\]

We write \( \text{par}(\hat{\xi}^j_t) = \xi^j_t \) to indicate \( \xi^j_t \) is the parent of \( \hat{\xi}^j_t \). This is a slight abuse of notation as the parent index, \( j \), is important: a child always has exactly one parent, even if two parents occupy the same point in state space.

We assume the particles belong to a standard Borel state space that has been divided into a finite collection \( R \) of disjoint bins. All subsets of this space will be assumed measurable, and all functions bounded and measurable. We write \( \text{bin}(\xi) = r \) if particle \( \xi \) is in bin \( r \in R \), and define the weight of the \( r \)th bin as

\[
\omega_t(r) = \sum_{j: \text{bin}(\xi^j_t) = r} \omega^j_t.
\]

Our aim is to use WE to estimate

\[
\int f \, d\mu \approx \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^N \omega^j_t f(\xi^j_t) =: \theta_T,
\]

where
where we start the sum from \( t = 1 \) for notational simplicity and so that it agrees with the update formula in (11) below. All of the analysis that follows, particularly in Section 5, can be applied to more general, time dependent functions of the underlying Markov process, without much extra work [2].

We will need the following notation. For a distribution \( \eta \) and random object \( X \), \( X \sim \eta \) means \( X \) is distributed as \( \eta \). For a finite set \( S \), the number of elements of \( S \) is \( \#S \). We write \( /BD_{A} \) for the Boolean variable that equals 1 if \( A \) is true, and 0 otherwise.

For brevity, statements written for a particular variable or index are assumed to apply to all admissible variables and indices, unless otherwise specified. For instance, in (4) we implicitly mean this equation holds for all \( t \geq 0 \) and \( r \in \mathbb{R} \).

3.1. Resampling. For \( \{q_{i} : i \in I\} \) a finite set of nonnegative real numbers,

\[
\{n_{i} : i \in I\} = \text{resample}\left(\{q_{i} : i \in I\}, n\right)
\]

indicates that

\[
\sum_{i \in I} n_{i} = n, \quad \mathbb{E}[n_{i}] = \frac{nq_{i}}{\sum_{i \in I} q_{i}}, \quad \left| n_{i} - \frac{nq_{i}}{\sum_{i \in I} q_{i}} \right| < 1, \quad i \in I.
\]

If \( n \leq 0 \) then by convention we set \( n_{i} = 0, i \in I \). We will use systematic resampling, which typically leads to smaller variance compared to other resampling techniques [13]. For the reader’s convenience we briefly describe it. Let \( I = \{1, \ldots, m\} \), let \( U \) be a uniform in \([0, 1)\) random variable, and let

\[
U_{j} = U + \frac{j - 1}{n} \mod 1, \quad j = 1, \ldots, n.
\]

Then systematic resampling is obtained via

\[
n_{i} = \# \left\{ j : U_{j} \in \left[ \frac{\sum_{k=1}^{i-1} q_{k}}{\sum_{k=1}^{m} q_{k}}, \frac{\sum_{k=1}^{i} q_{k}}{\sum_{k=1}^{m} q_{k}} \right] \right\}
\]

Note that only one random number is needed to resample \( n \) times from \( \{q_{i} : i \in I\} \). In practice, less randomness in resampling typically leads to smaller variance [13].

3.2. WE algorithm. We now describe Algorithm 1 in precise detail.

**Algorithm 2** (Precise description of Algorithm 1).

- Initialization: Choose \( N \) initial points and weights with \( \sum_{i=1}^{N} \omega_{0}^{i} = \sum_{r \in \mathcal{R}} \omega_{0}(r) = 1, \) and an initial transition matrix \( Q_{0} \). Set the initial observable average \( \theta_{0} = 0 \). Choose a total number \( T \) of generations. Then iterate for \( t = 0, 1, \ldots, T \):

  - Selection step: In the \( t \)-th selection step, do the following. Pick a value function \( v_{t} = v_{t}(r) \) which may depend on the information from the algorithm up to the current generation. If \( v_{t}(r) \) is defined as in (1), we say the allocation is optimized. If \( v_{t}(r) = 1/\omega_{t}(r) \) when \( \omega_{t}(r) > 0 \) and otherwise \( v_{t}(r) = 0 \), we
say the allocation is uniform. Set

\[ \{ \tilde{N}_t(r) : r \in \mathcal{R} \} = \text{resample} \left( \{ \omega_t v_t(r) : r \in \mathcal{R} \}, N - \sum_{r=1}^{R} \mathbb{1}_{\omega_t(r) > 0} \right) \]

and set the number of children in each bin as

\[ N_t(r) = \mathbb{1}_{\omega_t(r) > 0} + \tilde{N}_t(r). \]

Let

\[ \{ C_i^t : \text{bin}(\xi_i^t) = r \} = \text{resample} \left( \{ \omega_i^t : \text{bin}(\xi_i^t) = r \}, N_t(r) \right) \]

define the number of children of each parent. That is,

\[ C_i^t = \# \left\{ i : \text{par}(\xi_i^t) = \xi_j^t \right\}. \]

Notice children are selected w.p.p.t. their parents’ weights. Set

\[ \hat{\omega}_i^t = \frac{\omega_t(r)}{N_t(r)} \text{ if bin}(\xi_i^t) = r, \]

as the children’s weights.

- Mutation step: In the \( t \)-th mutation step, do the following. Let the next generation \( \{ \xi_j^{t+1} \}_{j=1}^{N} \) of parents be independent, conditionally on the \( t \)-th selection step, with

\[ \xi_j^{t+1} \sim K(\xi_j^t, \cdot). \]

Let the corresponding weights be the ones defined during selection:

\[ \omega_j^{t+1} = \hat{\omega}_i^t. \]

Next, update the observable time average via

\[ \theta_{t+1} = \left( 1 - \frac{1}{t+1} \right) \theta_t + \frac{1}{t+1} \sum_{j=1}^{N} \omega_j^{t+1} f(\xi_j^{t+1}). \]

Finally, update the bin-to-bin transition matrix \( Q_{t+1}(r, s) \). We describe a possible method for this in Section 3.3 below. After the mutation step, update \( t \leftarrow t + 1 \) and return to the selection step.

We have found that our selection step, which preserves total weight and number of particles, with minimal randomness – only \( 1 + \#\mathcal{R} \) random numbers are required for the resampling – leads to well controlled variance even as \( T \) and the number of resampling steps become large. This is in significant contrast with standard resampling techniques that do not preserve total weight \([8, 9, 10]\). Note that after selection, all the weights in a given bin are the same \([6, 7]\).
3.3. Updating the bin-to-bin transition matrix. In this section we describe one possible way to update $Q_t = Q_t(r, s)$. Fix a decreasing sequence $\{\epsilon_n\}_{n \geq 0}$ such that $\sum_{n=1}^{\infty} \epsilon_n = \infty$ and $\sum_{n=1}^{\infty} \epsilon_n^2 < \infty$. In the $t$-th mutation step of Algorithm 2, initialize $\tilde{Q}(r, s) = Q_t(r, s)$. Then for $j = 1, \ldots, N$, if $\text{bin}(\tilde{\xi}_j^t) = r$, write

$$n(r) = \# \left\{ \tilde{\xi}_i^t : \text{bin}(\tilde{\xi}_i^t) = r, \ 0 \leq s < t \text{ or } s = t, \ i \leq j \right\}$$

for the total number of previously selected particles in bin $r$, and update

$$\tilde{Q}(r, s) \leftarrow (1 - \epsilon_n(r))\tilde{Q}(r, s) + \epsilon_n(r) I_{\text{bin}(\xi_{i+1}^t) = s}.$$  

Finally set $Q_{t+1}(r, s) = \tilde{Q}(r, s)$. The form of (12), and the conditions on $\{\epsilon_n\}_{n \geq 0}$, come from stochastic approximation theory [19]. Note that $\epsilon_n = 1/(n + 1)$ corresponds to simple averaging. If desired, the count in $n(r)$ can include particles from an initialization step that defines $Q_0$. We comment on our choice of $\{\epsilon_n\}_{n \geq 0}$ in more detail in Section 4.

4. Numerical examples: mean first passage time computation. Consider a Markovian stochastic dynamics $(Y_t)_{t \geq 0}$, and let $\tau_R = \inf\{t \geq 0 : Y_t \in R\}$ be the first time it reaches a set $R$ in state space. Below, we consider a fixed binning of state space, and $R$ is one of the bins. We are interested in computing

$$\text{mean first passage time to } R := \mathbb{E}[\tau_R|Y_0 \sim \rho],$$

starting at some distribution $\rho$ with support disjoint from $R$. This can be done as follows. Let $\tau_{\text{bin}} = \min\{t > 0 : \text{bin}(Y_t) \neq \text{bin}(Y_0)\}$ be the first time for $(Y_t)_{t \geq 0}$ to reach a new bin. Let $\Delta t > 0$ be a parameter and define $\tau_{\text{loc}} = \min\{\tau_{\text{bin}}, \Delta t\}$ as a
Fig. 2. Data from Algorithm 2 for the example in Section 4.1. Left: $\omega_T(r)$, $v_T(r)$, and a rescaled version of $V$ plotted against $r$. Note that $v_T$ favors selection in regions between basins of attraction of $V$. Right: $\omega_T(r)$ and rescaled versions of $N_T(r)$ for optimized and uniform allocation. The solid dots indicate the bins corresponding to $\rho$ and $R$. Notice that optimized allocation mostly ignores bins $r$ with $r \leq 50$. The values in all plots are obtained from averaging 96 independent instances of Algorithm 2.

resampling time. In Algorithm 2 we use the Markov kernel

$$K(\xi, d\xi') = \begin{cases} P[Y_{\tau_{loc}} \in d\xi | Y_0 = \xi], & \xi \notin R \\ P[Y_{\tau_{loc}} \in d\xi | Y_0 \sim \rho], & \xi \in R. \end{cases}$$

Let $(\xi_t)_{t \geq 0}$ be a Markov chain with kernel $K$ and let $\sigma_R = \min\{t \geq 0 : \xi_t \in R\}$ be the first time it reaches $R$. Assume $E[\sigma_R | Y_0 \sim \rho] < \infty$ and $P[\sigma_R < \infty | \xi_0 = x] = 1$ for almost every initial condition $x$ for $(\xi_t)_{t \geq 0}$. Then it can be shown [2] that $K$ has a unique stationary distribution $\mu$ and

$$E[\tau_R | Y_0 \sim \rho] = \frac{E[\tau_{loc} | Y_0 \sim \mu]}{\mu(R)}.$$

We estimate $E[\tau_R | Y_0 \sim \rho]$ by using Algorithm 2 to simultaneously compute the numerator and denominator of the RHS of (13). Specifically, we use (11) to estimate $\mu(R)$ and adapt (11) to estimate $E[\tau_{loc} | Y_0 \sim \mu]$ in the obvious way, and then take the ratio as in (13) to obtain an estimate of $E[\tau_R | Y_0 \sim \rho]$. Abusing notation somewhat, in Figures 1-4, we write $\theta_t$ for the estimate of the mean first passage time obtained this way from the first $t$ generations in Algorithm 2.

We optimize the allocation to minimize variance in the denominator $\mu(R)$ of (13), which is important when the mean first passage time is large. Thus, in the formulas defining the value function $v_t$ in Section 2.2, we let $u$ be a vector indexed by $R$ with a 1 in the $R$th spot and 0’s elsewhere. In other words, $u$ is a bin-based approximation of the indicator or characteristic function of $R$.

Equation (13) is similar to what is known as the Hill relation [17] in computational chemistry. Initial conditions are important when using (13) to compute the mean first passage time. In Algorithm 2, they should be as close as possible to $\mu$ — subject of course to computational constraints — to reduce the number of WE generations needed for convergence. Good initial conditions can also improve the optimized allocation. This initialization problem is not unique to WE: it is common to many of the methods for sampling non-equilibrium steady states mentioned in the Introduction.
To initialize Algorithm 2, we run short WE simulations with uniform particle allocation to obtain $Q_0$ and our initial particles and weights. These short simulations each have 5% as many steps, or generations, as our full length WE simulations. There are several other possible approaches to initialization. The most general may be to construct a crude Markov State Model on the bins $\mathbb{B}$, using a clustering analysis on some initial brute force or WE simulations. This crude model then defines $Q_0$, which may be used to choose the initial points and weights. When $(Y_t)_{t \geq 0}$ is Langevin dynamics $[23]$, some other possibilities arise from the fact that $\mu$ can be close or related in some way to the Boltzmann distribution $[23]$. Ideas in this direction have been proposed for Milestoning $[2, 3, 26]$ as well as for WE $[6]$. In Milestoning, $K$ is defined slightly differently, but our ideas and algorithm still apply.

We took $\epsilon_n = 1/n^\alpha$ where $\alpha = 3/5$ is a stochastic approximation parameter. We choose $\alpha$ to allow $Q_t$ to converge quickly out of the transient regime, at the cost of more noisy values $[19]$. Other choices are of course possible. Fast convergence of $Q_t$ may be especially desirable when using the fixed point method mentioned in Section 2.3. See $[11, 27]$ for a closely related stochastic approximation.

4.1. Brownian dynamics in 1D landscape. Here we take $(Y_t)_{t \geq 0}$ solving

$$dY_t = -V'(Y_t) \, dt + \sqrt{2\beta^{-1}} \, dW_t$$

with reflecting boundary conditions on $[0, 1]$, with $\beta = 3$. The potential energy is $V(\xi) = 1 + 25(\xi - 1/2)^2$ for $\xi < 1/2$, and otherwise $V(\xi) = \cos(12\pi \xi)$. See Figure 2. We used an Euler integrator with time step $\delta t = 0.0001$. There are 100 equally spaced bins in $[0, 1]$, where bin $r$ is $[(r-1)/100, r/100]$, and $R = [91/100, 92/100)$ is one of the bins. The initial distribution is $\rho = \delta_{1/2}$. We took $N = 500$ and $\Delta t = 0.0002$. Results are in Figures 1-2. We find that optimized allocation consistently beats uniform allocation, while both outperform naive sampling. Here, naive sampling simply means no resampling.
4.2. Langevin dynamics in Muller-Brown landscape. Here we let \((Y_t)_{t \geq 0} = (q_t, p_t)_{t \geq 0}\) be Langevin dynamics \([23]\) on position \(q_t \in \mathbb{R}^2\) and momentum \(p_t \in \mathbb{R}^2\):

\[
\begin{align*}
\frac{dq_t}{dt} &= m^{-1} p_t dt \\
\frac{dp_t}{dt} &= -\nabla V(q_t) dt - \gamma m^{-1} p_t dt + \sigma dW_t,
\end{align*}
\]

with the potential energy \(V : \mathbb{R}^2 \to \mathbb{R}\) pictured in Figure 4, with parameters \(\gamma = 2\), \(\sigma = 8\), \(m = 1\). We used the BBK integrator \([23]\) with time step \(\delta t = 0.001\). Position space is divided into \(25^2 = 625\) equally sized rectangular bins, and trajectories are confined in the region in Figure 4 via reflecting boundary conditions. Velocity space remains unbinned. We take \(R = F \times \mathbb{R}^2\) and \(\rho(dq, dp) = \delta_{q_0} \times \eta\), where \(q_0\) and \(F\) are as in Figure 4, and \(d\eta \propto \exp(-\gamma |p|^2/(m\sigma^2)) dp\) the Boltzmann distribution on velocities. We use \(N = 3125\) particles and \(\Delta t = 1\). Results, in Figures 3-4, are similar to those from the example in Section 4.1.

4.3. Gain over uniform allocation. We find that optimized allocation outperforms uniform allocation over a wide range of parameters. See Figures 1 and 3.

The variance reduction from optimized allocation, compared to uniform allocation, can be crudely estimated as follows. Consider the subset, \(S\), of bins \(r\) where \(\omega_t(r) v_t(r) / \sum_{r \in \mathcal{R}} \omega_t(r) v_t(r) \gg 0\), and let \(N_{\text{unif}}\) be the typical number of particles in \(S\) under uniform allocation. Then the variance can be reduced by a factor of \(\approx N / N_{\text{unif}}\) by using optimized instead of uniform allocation.
Intuitively, this means that if the “important” regions of state space, identified by \( v_t(r) \) as just described, are much smaller than the regions of state space accessible by uniform allocation – see the right of Figure 2 and bottom of Figure 4 – then optimizing the allocation results in a significant variance reduction. We make a more rigorous comparison in the next section.

4.4. Synthetic variance and parameter choice. Our analysis in Section 5 below suggests how we might estimate the variance of \( \theta_T \) in an instance of Algorithm 1, before actually running it, once initialization is complete. Define

\[
\sigma^2_{\text{syn}} = \sum_{r \in R} \omega_0(r)^2 \frac{N_0(r)}{v_0(r)}^2, \quad \omega_0(r) := \pi_0(r),
\]

where \( v_0(r) \) and \( N_0(r) \) are defined as in Algorithm 2, and we recall \( \pi_0 \) is the stationary vector of \( Q_0 \). We will derive (14) from a variance analysis in Section 5.4 below.

We think of \( \sigma^2_{\text{syn}} \) as a synthetic approximation, up to a constant, of the variance in our WE estimate \( \theta_T \) of \( \int f \, d\mu \). Of course the value of \( \sigma^2_{\text{syn}} \) depends on the parameters chosen, such as the value function, resampling times, bins, number \( N \) of particles, etc. Ratios of \( \sigma^2_{\text{syn}} \) corresponding to different parameter choices are useful when \( Q_0 \) and \( \omega_0 \) are approximations, even if crude, of the steady-state bin-to-bin transition matrices and bin weights. One can compute a synthetic variance even when \( Q_0 \) and \( \omega_0 \) are not close to steady state, but the formula is more complicated than (14).

As a consequence, one could do a random search or simulated annealing in a full or reduced parameter space, computing \( \sigma^2_{\text{syn}} \) for each choice of parameters, and then choosing the parameters that minimize it. We leave this parameter study to future work. One can also compare the synthetic variance of different particle allocation schemes, like our optimized and uniform allocation, by modifying \( N_0(r) \) accordingly. We find this can lead to useful benchmarks; see the far right of Figures 1 and 3.

5. Mathematical analysis. Consider a time \( T > 0 \) and let

\[ \theta := T \theta_T, \]

with \( \theta_T \) as in (5) or (11). Let \( \mathcal{F}_t \) be the \( \sigma \)-algebra generated by the random objects from Algorithm 1 up to, but not including, the \( t \)-th selection step. Similarly, let \( \hat{\mathcal{F}}_t \) be the \( \sigma \)-algebra generated by the random objects from Algorithm 1 up to and including the \( t \)-th selection step. Below, \( g \) will denote a generic bounded measurable real-valued function defined on the state space of the particles.

5.1. Doob martingale. Consider the Doob martingale

\[
D_0, \hat{D}_0, D_1, \hat{D}_1, D_2, \hat{D}_2, \ldots \quad \text{where} \quad D_t = \mathbb{E} [\theta | \mathcal{F}_t], \quad \hat{D}_t = \mathbb{E} [\theta | \hat{\mathcal{F}}_t].
\]

Of course \( D_t \) depends on \( T \), but we suppress this since \( T \) is mostly fixed below.

Proposition 3. We have

\[
D_t = \sum_{j=1}^{N} \left( \sum_{s=1}^{t-1} \omega_s^j f(\xi_s^j) + \sum_{s=0}^{T-t} \omega_s^j K^sf(\xi_s^j) \right),
\]

\[
\hat{D}_t = \sum_{j=1}^{N} \left( \sum_{s=1}^{t} \omega_s^j f(\hat{\xi}_s^j) + \sum_{s=1}^{T-t} \hat{\omega}_s^j K^sf(\hat{\xi}_s^j) \right).
\]
Proof. By definition, \( \theta = \sum_{t=1}^{T} \sum_{j=1}^{N} \omega_{t} f(\xi_{t}^j) \). So evidently (15) holds when \( t = T \). We proceed by induction. Suppose (15) holds for a given \( t \leq T \). By definition (9)-(10) of the mutation step, it is straightforward to see that for any \( g \),

\[
E \left[ g(\xi_{t}^j) \mid \tilde{F}_{t-1} \right] = Kg(\xi_{t-1}^j).
\]

Thus,

\[
\dot{D}_{t-1} = E \left[ \theta \mid \tilde{F}_{t-1} \right] = E \left[ D_{t} \mid \tilde{F}_{t-1} \right] = \\
= \sum_{j=1}^{N} \left( \sum_{s=1}^{t-1} \omega_{s} f(\xi_{s}^j) + E \left[ \sum_{s=0}^{T-t} \omega_{s} K^s f(\xi_{t}^j) \mid \tilde{F}_{t-1} \right] \right) \\
= \sum_{j=1}^{N} \left( \sum_{s=1}^{t-1} \omega_{s} f(\xi_{s}^j) + \tilde{\omega}_{t-1} E \left[ \sum_{s=0}^{T-t} K^s f(\xi_{t}^j) \mid \tilde{F}_{t-1} \right] \right) \\
= \sum_{j=1}^{N} \left( \sum_{s=1}^{t-1} \omega_{s} f(\xi_{s}^j) + \sum_{s=1}^{T-t+1} \tilde{\omega}_{t-1} K^s f(\xi_{t-1}^j) \right),
\]

which shows that (16) holds for \( t - 1 \). In particular, by this and the base case, (16) holds when \( t = T - 1 \). Let \( \tilde{F}_{t} := \sigma \left( F_{t} \cup \{ N_{t}(r) : r \in \mathcal{R} \} \right) \). By (7),

\[
E \left[ \# \{ i : \text{par}(\xi_{t}^i) = \xi_{t}^j \} \mid \tilde{F}_{t} \right] = N_{t}(r) \frac{\omega_{t} f(\xi_{t}^j)}{\omega_{t}(r)}, \quad \text{if } \text{bin}(\xi_{t}^j) = r,
\]

while by (8),

\[
\tilde{\omega}_{t} \text{ is } \tilde{F}_{t}-\text{measurable}, \quad \tilde{\omega}_{t} = \frac{\omega_{t}(r)}{N_{t}(r)}, \quad \text{if } \text{bin}(\xi_{t}^j) = r.
\]

Thus, for any \( g \),

\[
E \left[ \sum_{i: \text{bin}(\xi_{t}^j) = r} \tilde{\omega}_{t} g(\xi_{t}^i) \right] = \sum_{j: \text{bin}(\xi_{t}^j) = r} E \left[ \sum_{i: \text{par}(\xi_{t}^i) = \xi_{t}^j} \tilde{\omega}_{t} g(\xi_{t}^i) \right] \\
= \sum_{j: \text{bin}(\xi_{t}^j) = r} g_{t}(\xi_{t}^j) E \left[ \sum_{i: \text{par}(\xi_{t}^i) = \xi_{t}^j} \tilde{\omega}_{t} \right] \mid \tilde{F}_{t} \right] \\
= \sum_{j: \text{bin}(\xi_{t}^j) = r} g_{t}(\xi_{t}^j) \left[ \frac{\omega_{t}(r)}{N_{t}(r)} \right] E \left[ \# \{ i : \text{par}(\xi_{t}^i) = \xi_{t}^j \} \right] \mid \tilde{F}_{t} \right] \\
= \sum_{j: \text{bin}(\xi_{t}^j) = r} g_{t}(\xi_{t}^j) \left[ \frac{\omega_{t}(r) N_{t}(r) \omega_{t} f(\xi_{t}^j)}{N_{t}(r) \omega_{t}(r)} \right] \mid F_{t} \right] \\
= \sum_{j: \text{bin}(\xi_{t}^j) = r} \omega_{t} g_{t}(\xi_{t}^j).
\]

It follows that for any \( g \) and \( t \geq 0 \),

\[
\sum_{j=1}^{N} \omega_{t} g_{t}(\xi_{t}^j) = \sum_{j=1}^{N} \omega_{t} g_{t}(\xi_{t}^j),
\]
which is another way of saying the selection step is unbiased. Now, inductively assume that (16) holds for a given \( t \leq T - 1 \). Then (19) implies

\[
D_t = \mathbb{E}[\theta | F_t] = \mathbb{E}[\hat{D}_t | F_t]
\]

\[
= \sum_{j=1}^{N} \left( \sum_{s=1}^{t} \omega_j^s f(\xi_j^s) + \mathbb{E} \left[ \sum_{s=1}^{T-t} \tilde{\omega}_j^s K^s f(\xi_j^s) | F_t \right] \right)
\]

\[
= \sum_{j=1}^{N} \left( \sum_{s=1}^{t-1} \omega_j^s f(\xi_j^s) + \sum_{s=0}^{T-t} \tilde{\omega}_j^s K^s f(\xi_j^s) \right),
\]

which shows that (15) holds for \( t \). We have shown that (15) holds for \( t = T \), which implies (16) holds for \( t = T - 1 \), which in turn shows (15) holds for \( t = T - 1 \), which means that (16) holds for \( t = T - 2 \), and so on. By induction we are done.

Recall \( \theta_T \equiv \theta/T \) is our WE estimate of our observable of interest: we expect

\[
\lim_{T \to \infty} \theta_T \equiv \lim_{T \to \infty} \frac{D_T}{T} \text{ a.s.} = \int f \, d\mu.
\]

When does (20) hold? Suppose the initial WE generation satisfies

\[
\sum_{j=1}^{N} \omega_j^0 g(\xi_j^0) = \int g \, d\nu, \quad \text{for all } g.
\]

Then (19) and (9)- (10) show that the \( t \)-th generation satisfies

\[
\sum_{j=1}^{N} \omega_j^t g(\xi_j^t) = \int g \, d(\nu K^t), \quad \text{for all } g,
\]

which is a way of saying WE is unbiased [2, 29]. Under appropriate ergodicity assumptions, \( \lim_{t \to \infty} \nu K^t = \mu \). This suggests that (20) should hold under a condition about sufficient independence between WE generations. A rigorous proof is technical and beyond the scope of this article. We refer the reader, however, to [8] for rigorous arguments in this direction.

### 5.2. Doob decomposition.

Our principal interest will be in a scheme to minimize the variance in our WE estimate \( \theta_T = D_T/T \approx \int f \, d\mu \). To this end we consider the following Doob decomposition.

**Proposition 4.** We have

\[
(\theta - \mathbb{E}[\theta])^2 = B_T + (D_0 - \mathbb{E}[\theta])^2
\]

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

where \( \mathbb{E}[B_T] = 0 \).

**Proof.** The Doob decomposition of

\[
(D_0 - \theta)^2, (\hat{D}_0 - \theta)^2, (D_1 - \theta)^2, (\hat{D}_1 - \theta)^2, (D_2 - \theta)^2, (\hat{D}_2 - \theta)^2, \ldots
\]

with respect to the filtration \( \mathcal{F}_0, \hat{\mathcal{F}}_0, \mathcal{F}_1, \hat{\mathcal{F}}_1, \mathcal{F}_2, \hat{\mathcal{F}}_2, \ldots \) satisfies

\[
(D_t - \theta)^2 = (D_0 - \theta)^2 + A_t + B_t,
\]
with predictable part \((A_t)_{t \geq 0}\) given by
\[
A_t = \sum_{s=0}^{t-1} \left( \mathbb{E} \left[ (D_{s+1} - \theta)^2 \mid \mathcal{F}_s \right] - (\hat{D}_s - \theta)^2 \right) + \mathbb{E} \left[ (\hat{D}_s - \theta)^2 \right] - (D_s - \theta)^2
\]
(22)
\[
= \sum_{s=0}^{t-1} \left( \mathbb{E} \left[ (D_{s+1} - \hat{D}_s)^2 \mid \mathcal{F}_s \right] + \mathbb{E} \left[ (\hat{D}_s - D_s)^2 \mid \mathcal{F}_s \right] \right),
\]
and martingale part \((B_t)_{0 \leq t \leq T}\) with \(B_0 = 0\).

We think of
\[
\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \mathcal{F}_t \right], \quad \mathbb{E} \left[ (\hat{D}_t - D_t)^2 \mid \mathcal{F}_t \right]
\]
as the variances at step \(t\) of Algorithm 2 from, respectively, mutation and selection. Following [2], we will derive a strategy that minimizes, in a certain sense, the variance from mutation. Heuristically, once we decide on a target number of children in each bin, the variance from selection is minimized when the resampling mechanism has as little randomness as possible [2]. This motivates the selection step of Algorithm 2. In Section 5.5 below, we show the variance from selection in Algorithm 2 is exactly zero in a special case.

5.3. Variance minimization in WE Algorithm. Our allocation strategy is based on an idea first proposed in [2]: minimizing the variance from mutation in the Doob decomposition of Proposition 4. We start with a formula for this variance:

**Proposition 5.** The variance from the \(t\)-th mutation step of Algorithm 2 is
\[
\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \mathcal{F}_t \right] = \sum_{i=1}^{N} (\hat{\omega}_i)^2 \left[ K(h_{t+1,T}(\hat{\xi}_i)) - (Kh_{t+1,T}(\hat{\xi}_i))^2 \right]
\]
where
\[
h_{t,T}(\xi) := \sum_{s=0}^{T-t} K^s f(\xi).
\]

**Proof.** Calculations similar to those in the proof of Proposition 3 show that
\[
\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \mathcal{F}_t \right]
\]
\[
= \mathbb{E} \left[ \left( \sum_{j=1}^{N} \sum_{s=0}^{T-1} \omega_{t+1}^j K^s f(\xi_{t+1}^j) \right)^2 \mid \mathcal{F}_t \right] - \left( \sum_{i=1}^{N} \sum_{s=0}^{T-t} \hat{\omega}_i K^s f(\hat{\xi}_i) \right)^2.
\]
By definition (9)-(10) of the mutation step,
\[
\mathbb{E} \left[ \omega_{t+1}^i \omega_{t+1}^j K^m f(\xi_{t+1}^i) K^m f(\xi_{t+1}^j) \mid \mathcal{F}_t \right]
\]
\[
= \begin{cases} 
\hat{\omega}_i \hat{\omega}_j K^{m+1} f(\hat{\xi}_i) K^{n+1} f(\hat{\xi}_i), & i \neq j \\
(\hat{\omega}_i)^2 K(K^m f(\hat{\xi}_i) K^n f(\hat{\xi}_i)), & i = j
\end{cases}.
\]
Combining the last two displays gives
\[
\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \mathcal{F}_t \right]
\]
\[
= \sum_{i=1}^{N} \sum_{m,n=0}^{T-t-1} (\hat{\omega}_i)^2 \left[ K(K^m f(\hat{\xi}_i) K^n f(\hat{\xi}_i)) - K^{m+1} f(\hat{\xi}_i) K^{n+1} f(\hat{\xi}_i)) \right].
\]
The result follows by inspecting the definition of $h_{t,T}$.

The result of Proposition 5 is quite general and, in particular, does not depend on any of the structure of the selection step, including choice of the $N_i(r), r \in \mathcal{R}$. We take advantage of this fact to define the selection step in a way that minimizes the variance from mutation. To do this, we use the approximation

$$h_{t+1,T}(\xi) = \sum_{s=0}^{T-t-1} K^s f(\xi) \approx \sum_{s=0}^{T-t-1} Q_t^s u(r), \quad \text{if bin}(\xi) = r. \quad (23)$$

Let $\tilde{F}_t = \sigma(\mathcal{F}_t \cup \{N_i(r) : r \in \mathcal{R}\})$ and observe that by Proposition 5 and (23),

$$\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \bigg| \tilde{F}_t \right] = \mathbb{E} \left[ \mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \bigg| \tilde{F}_t \right] \bigg| \tilde{F}_t \right] \quad (24)$$

$$= \sum_{i=1}^{N} \mathbb{E} \left[ (\hat{\omega}_i)^2 \left[ K(h_{t+1,T}(\hat{\xi}_i)^2) - (Kh_{t+1,T}(\hat{\xi}_i)) \right] \bigg| \tilde{F}_t \right] \quad (25)$$

$$= \sum_{r \in \mathcal{R}} \frac{\omega_t(r)^2}{N_t(r)} \mathbb{E} \left[ \sum_{i : \text{bin}(\xi_i) = r} K(h_{t+1,T}(\hat{\xi}_i)^2) - (Kh_{t+1,T}(\hat{\xi}_i)) \bigg| \tilde{F}_t \right] \quad (26)$$

$$\approx \sum_{r \in \mathcal{R}} \frac{\omega_t(r)^2}{N_t(r)} \left[ Q_t \left( \sum_{s=0}^{T-t-1} Q_t^s u(r) \right)^2 - \left( Q_t \sum_{s=0}^{T-t-1} Q_t^s u(r) \right)^2 \right]. \quad (27)$$

Minimizing (27) over $\{N_i(r) : r \in \mathcal{R}\}$, subject to $\sum_r N_r(r) = N$, gives

$$N_t(r) = \frac{N \omega_t(r) \sqrt{Q_t \left( \sum_{s=0}^{T-t-1} Q_t^s u(r) \right)^2 - \left( Q_t \sum_{s=0}^{T-t-1} Q_t^s u(r) \right)^2}}{\sum_{r \in \mathcal{R}} \omega_t(r) \sqrt{Q_t \left( \sum_{s=0}^{T-t-1} Q_t^s u(r) \right)^2 - \left( Q_t \sum_{s=0}^{T-t-1} Q_t^s u(r) \right)^2}} \quad (28)$$

Taking $T \to \infty$ in (28) leads to our optimal allocation strategy (2). In the preceding argument, we condition on the $N_i(r), r \in \mathcal{R}$, so that we could arrive at a closed formula for these numbers. Of course the optimal $N_t(r)$ defined this way are noninteger in general; we account for this carefully in Algorithm 2 to mitigate variance from selection. Taking $T \to \infty$ isn’t needed, though we prefer the simple formula that results. Indeed we could use (28) in place of (2), which may be better for small $T$.

Observe that at step $t$ we only minimize –approximately – one term in the Doob decomposition of the variance. On the other hand, under certain assumptions, terms from different generations in the Doob decomposition should become independent in the limit $N \to \infty$. This result may require more independence in the selection step of Algorithm 2 than what we obtain from systematic resampling, though. Anyway we do not pursue arguments in this direction, instead referring the reader to [8] for closely related results.

The approximation (23) that leads to our optimized allocation strategy is uncontrollable in general. However, our allocation strategy need not be truly optimal in order to comfortably beat uniform allocation, as we show in Section 4. Our results in Section 5.5 below suggest that our allocation is truly optimal when $N, T \to \infty$ and each bin contains exactly one point of state space.
5.4. Synthetic variance. The calculations in the previous section suggest how the variance of a WE simulation can be estimated before running the simulation. Suppose the WE begins near steady state at generation \( t = 0 \) in the sense that

\[
Q_0(r, s) \approx \int K(\xi, d\xi') \mathbb{1}_{\text{bin}(\xi) = r, \text{bin}(\xi') = s} \mu(d\xi) \quad \text{and} \quad \omega_0(r) \approx \pi_0(r).
\]

Then in the limit \( T \to \infty \), the estimate (24)-(27) can be replaced with

\[
\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \tilde{F}_t \right] \approx \sum_{r \in \mathcal{R}} \frac{\omega_0(r)^2}{N_0(r)} \left[ Q_0 \left( \sum_{s=0}^{\infty} Q_0^s u(r) \right)^2 - \left( Q_0 \sum_{s=0}^{\infty} Q_0^s u(r) \right)^2 \right]
\]

The term square brackets on the right hand side of the last display is exactly \( v_0(r) \), as described in Section 2.2. This leads us to define a synthetic variance,

\[
\sigma_{\text{syn}}^2 = \sum_{r \in \mathcal{R}} \frac{\omega_0(r)^2}{N_0(r)} v_0(r)^2, \quad \omega_0(r) := \pi_0(r),
\]

as a crude a priori measure of the variance of a WE simulation. Observe that we have replaced \( \omega_0 \) with \( \pi_0 \), where \( \pi_0 \) is the stationary distribution of \( Q_0 \). Thus, the synthetic variance can be computed from \( Q_0 \) alone. The same remarks above – about the uncontrolled approximations leading to this formula – still apply. However, we find that the ratios of \( \sigma_{\text{syn}}^2 \) corresponding to different parameter choices can be close to the corresponding empirical variance ratios. See Figures 1 and 3 in Section 4 above for supporting numerical results.

5.5. Exact variance minimization in a simple case. We begin by showing that (2) is consistent with minimizing mutation variance, in the following sense:

**Proposition 6.** Suppose that each bin contains exactly 1 point in state space. Thinking of the transition kernel \( K \) as a square matrix, and the function \( f \) and stationary probability mass function \( \mu \) as column vectors, let \( g \) be the column vector solving the Poisson equation

\[
\left( I - K + \frac{\mu^T \mu}{\| \mu \|^2} \right) g = f - \mu^T f \|, \quad \text{where} \; \| \cdot \|_2 \text{ is the usual } l^2 \text{ norm. Define}
\]

\[
v(r) = r\text{'th entry of } \sqrt{K(g^2) - (Kg)^2}, \quad \text{where the squaring is entrywise. Let } \tilde{F}_t = \sigma(\mathcal{F}_t \cup \{N_i(r) : r \in \mathcal{R}\}). Then
\]

\[
\lim_{T \to \infty} \mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \tilde{F}_t \right] = \sum_{r \in \mathcal{R}} \frac{\omega_i(r)^2}{N_i(r)} v(r)^2.
\]

**Proof.** Following (24)-(26),

\[
\mathbb{E} \left[ (D_{t+1} - \hat{D}_t)^2 \mid \tilde{F}_t \right] = \sum_{r \in \mathcal{R}} \frac{\omega_i(r)^2}{N_i(r)} \left[ Kh_{t+1,T}^2(r) - (Kh_{t+1,T})^2(r) \right].
\]

Since \( \lim_{T \to \infty} Kh_{t+1,T}^2 - (Kh_{t+1,T})^2 = K(g^2) - (Kg)^2 \), the result follows. \( \square \)}
Last we show that, in this setting, Algorithm 2 has zero variance from selection.

**Proposition 7.** Suppose that each bin contains exactly 1 point in state space. Then for \( t \geq 0 \), the variance from the \( t \)-th selection step in Algorithm 2 is exactly zero. More precisely, for each \( t \geq 0 \),

\[
E\left[ (\hat{D}_t - D_t)^2 \middle| \mathcal{F}_t \right] = 0.
\]

**Proof.** Calculations similar to those in the proof of Proposition 3 show

\[
E\left[ (\hat{D}_t - D_t)^2 \middle| \mathcal{F}_t \right] = E\left[ \left( \sum_{i=1}^{N} \sum_{s=1}^{T-t} \hat{\omega}_i^t K^s f(\hat{\xi}_t^i) \right)^2 \middle| \mathcal{F}_t \right] - \left( \sum_{i=1}^{N} \sum_{s=1}^{T-t} \omega_i^t K^s f(\xi_t^i) \right)^2.
\]

Again defining \( h_{t,T}(\xi) := \sum_{s=0}^{T-1} K^s f(\xi) \), we have

\[
\left( \sum_{j=1}^{N} \sum_{s=1}^{T-t} \omega_j^t K^s f(\xi_t^j) \right)^2 = \sum_{r,r' \in \mathcal{R}} \sum_{i : \text{bin}(\xi_t^i) = r} \sum_{j : \text{bin}(\xi_t^j) = r'} \omega_i^t \omega_j^t K_{h_{t+1,T}(\xi_t^i)} K_{h_{t+1,T}(\xi_t^j)}.
\]

Observe that \( \# \{ i : \text{bin}(\xi_t^i) = r \} = N_t(r) \) is \( \tilde{F}_t \)-measurable. Thus by (18),

\[
E\left[ \left( \sum_{i=1}^{N} \sum_{s=1}^{T-t} \hat{\omega}_i^t K^s f(\hat{\xi}_t^i) \right)^2 \middle| \mathcal{F}_t \right] = E\left[ \left( \sum_{i=1}^{N} \omega_i^t K_{h_{t+1,T}(\hat{\xi}_t^i)} \right)^2 \middle| \mathcal{F}_t \right]
\]

\[
= \sum_{r,r' \in \mathcal{R}} \sum_{i : \text{bin}(\xi_t^i) = r} \sum_{j : \text{bin}(\xi_t^j) = r'} \omega_i(r) \omega_j(r') K_{h_{t+1,T}(r)} K_{h_{t+1,T}(r')}
\]

\[
= \sum_{r,r' \in \mathcal{R}} \sum_{i : \text{bin}(\xi_t^i) = r} \sum_{j : \text{bin}(\xi_t^j) = r'} \omega_i^t \omega_j^t K_{h_{t+1,T}(\xi_t^i)} K_{h_{t+1,T}(\xi_t^j)}.
\]

This result does not depend on how we define the mutation step or \( \{ N_t(r) : r \in \mathcal{R} \} \) in Algorithm 2. The zero variance result above applies to the allocation strategy resulting from any choice of value function; in particular it applies to uniform allocation. Thus, our resampling scheme may be of interest outside the particular setting of this article.

**Acknowledgements.** D. Aristoff thanks Peter Christman, Tony Lelièvre, Jos- selin Garnier, Matthias Rouset, Gideon Simpson, and Gabriel Stoltz for helpful suggestions and discussions.

**References**


