Strata Design for Variance Reduction in Stochastic Simulation

Jaeshin Park and Eunshin Byon*
Department of Industrial and Operations Engineering,
University of Michigan, Ann Arbor, MI 48109
and
Young Myoung Ko†
Department of Industrial and Management Engineering,
Pohang University of Science and Technology,
Pohang, Gyeongbuk 37673, Republic of Korea
and
Sara Shashaani ‡
Department of Industrial and Systems Engineering,
North Carolina State University, Raleigh, NC 27695

Abstract

Stratified sampling is one of the powerful variance reduction methods for analyzing system performance, such as reliability, with stochastic simulation. It divides the input space into disjoint subsets, called strata, to draw samples from each stratum. Partitioning the input space properly and allocating greater computational effort to crucial strata can help accurately estimate system performance with a limited computational budget. How to create strata, however, has yet to be thoroughly examined. Strata design faces the curse of dimensionality and data scarcity as the input dimension increases. We analytically derive the optimal stratification structure that minimizes the estimation variance for univariate problems. Further, reconciling the optimal stratification into decision trees, we devise a robust algorithm for multi-dimensional problems. Numerical experiments and a wind turbine case study demonstrate the superiority of the proposed method in terms of variance reduction, leading to computational efficiency and scalability.

Keywords: Monte Carlo simulation, reliability, stratified sampling, variance reduction

*This work was supported in part by the National Science Foundation (grant no.: CMMI- 2226348)

†This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (grant no. NRF-2021R1A2C1094699 and NRF-2021R1A4A1031019)

‡This work was supported in part by the National Science Foundation (grant no.: CMMI- 2226347)

1 Introduction

The design or operation of physical or social systems has benefited from the widespread use of stochastic simulation as a mathematical modeling and analytical tool (Wycoff et al., 2022). Stochastic simulation is useful for estimating system performance, especially when observational data are insufficient or unavailable. With increasing computing power, modern computer models employed in simulations offer a better understanding of stochastic system behavior under various input conditions.

Stochastic simulation research generally falls into two main categories: Monte Carlo sampling (MCS)-based and surrogate modeling-based approaches. Studies in surrogate modeling focus on approximating the underlying response surface across the input space to predict responses at unseen data points within the input domain. For example, in the Gaussian Process (GP), one of the most popular methods for surrogate modeling, there is an emphasis on leveraging the correlation between data points to estimate the response at unseen inputs, accompanied by the uncertainty quantification capability.

On the other hand, MCS methods are designed to estimate the overall statistical properties of the response variable. MCS methods generate a series of input vectors and use each to run a computer model, producing corresponding simulated outputs. They rely on the data samples to approximate the response variable's distribution without requiring assumptions about correlations among responses, as each sample is considered independent of the others. The critical assumption here is the independence of the simulated outputs, which is intrinsic to the MCS approach. When using MCS to estimate system performance, the estimation variance gets reduced with an increased computational budget. However,

even with a limited budget, one can still decrease the estimation variance through a class of techniques known as variance reduction techniques. Among various variance reduction techniques, importance sampling (IS) has been widely studied in the literature (Neddermeyer, 2009), with its vulnerability to the curse of dimensionality being notable even for an input dimension of three or more (Cao and Choe, 2019).

This study focuses on the use of stratified sampling in stochastic simulation, which is another variance reduction method that enhances computational efficiency. Stratified sampling divides the input space into multiple disjoint regions, or strata, and draws random samples from each one. The approach follows a specific rule for allocating the total sample size–or budget–among the strata, ensuring that the estimator remains unbiased. By targeting more samples from strategically important strata, stratified sampling can substantially improve estimation accuracy. However, the computational efficiency of stratified sampling depends on both the design of the strata–how the input domain is divided–and the allocation of the budget across these strata–deciding how many samples to take from each one, when the strata design is not previously known.

This study devises new algorithms for designing strata that effectively lead to significant reductions in estimation variance. Our approach adheres to the *parsimonious principle* that most physical or social systems are governed by a small number of key input variables. By identifying and utilizing these critical variables to stratify the input space, our method aims to ensure that each stratum displays a homogeneous response pattern. Specifically, we first derive the analytical solution for finding splitting locations that minimize the estimation variance in univariate problems. We then leverage this analytical insight as a guiding

principle to determine influential splitting variables and their partitioning points for multivariate problems. For the scalable strata design, we recursively partition the input domain and develop a binary decision tree. We further offer guidelines for determining the strata size to ensure the robustness of our stratification approach. We call the proposed approach *Optimization-guided and Tree-based Stratified Sampling*, or *OptiTreeStrat* for short. Our numerical experiments, which include synthetic examples and a wind turbine case study, confirm that our method effectively reduces the estimation variance through effective strata design.

In the remainder of the paper, Section 2 reviews relevant studies. Section 3 presents the proposed methodology. Sections 4 and 5 implement the approach with synthetic numerical examples and the wind turbine case study, respectively. Finally, Section 6 concludes.

2 Literature Review

2.1 Stratification

Several studies investigated finding a stratification structure for improving simulation efficiency. Tipton (2013) constructed strata using clustering analysis based on the closeness between observations of multivariate input variables. Etoré et al. (2011) devised an analytical procedure when input variables obey the Gaussian distribution, where the estimate of the objective function gradient sequentially updates the stratification direction. This approach may be less scalable and converge slowly when the input dimension is large and the response surface is complex because it is based on first-order optimization.

Some studies have explored methods for determining an appropriate stratification strategy. Cochran (1977) utilized linear regression to estimate a lower bound of variance for a given number of strata in survey sampling. While this approach assists in deciding the strata size, the linearity assumption could be unrealistic in many problems, and it addresses univariate inputs. Further, stratification is solely made based on the input distribution, constructing equi-probable strata that overlook the input-output relationship. Recently, Pettersson and Krumscheid (2022) presented an adaptive stratified sampling method, resembling batch sequential designs. Their approach constructs strata through successive equi-probable bi-partitions. As a new batch of data is collected, an equi-probable split for each input variable is contemplated across all existing strata. The specific stratum and input variable that offer the most substantial variance reduction are then selected for further partitioning. Although this approach accommodates multi-dimensional inputs, its reliance on equi-probable partitioning might not always be efficient. Particularly, if the initial divisions are poorly made, such inefficiencies can accumulate in subsequent partitions.

Shields (2016) combined Latin hypercube sampling (LHS) with stratified sampling by establishing strata boundaries based on LHS stratification. This stratification method may be less effective as it relies on the distribution of input variables and does not exploit how the output reacts to different inputs. Mease and Bingham (2006) proposed Latin hyperrectangle sampling that allows unequal probabilities of the strata. They assume the output variance within each stratum is proportional to the cell size, which may not necessarily hold in many applications.

Another approach involves the use of control variables which may consist of either a

subset of input variables or response variables different from the output of interest (Cannamela et al., 2008). By employing low dimensional control variables, it can reduce the problem's dimensionality. However, finding a suitable choice of control variables may be difficult or even unattainable in some circumstances.

In summary, existing approaches in stratified sampling either do not fully exploit the problem structure or make strong assumptions. Additionally, determining the strata size with multivariate inputs remains an area that has not been comprehensively explored.

2.2 Other variance reduction techniques

In the literature, sequential design approaches have been studied for the estimation of failure probabilities in deterministic computational models. These methodologies predominantly hinge on the utilization of surrogate models—most notably, GP —which iteratively refine the surrogate model with the acquisition of incremental data samples. Bect et al. (2012) devised a new acquisition function tailored to accommodate the uncertainties in failure probability estimations for choosing next design points. Cole et al. (2023) proposed a new acquisition function, called the Entropy-based Contour Locator (ECL), to adaptively update the GP. They subsequently utilized the refined GP within the framework of IS as an instrumental density function. In these sequential designs, the computational burden of constructing and updating the GP rapidly increases as the sample size increases. High-dimensional input spaces further intensify this challenge, increasing the time required for GP updates and complicating the optimization of the acquisition function.

While the majority of IS research has focused on deterministic computer models that

produce fixed output for a specific input (Neddermeyer, 2009; Kurtz and Song, 2013), Choe et al. (2015) derived the optimal IS density, referred to as stochastic IS (SIS), for stochastic computer models. Due to the unknown quantity within the optimal density, the direct application of SIS is not straightforward. Cao and Choe (2019) employed Cross Entropy (CE) as a metric to approximate the optimal SIS density, a method termed CE-SIS. The CE-SIS's efficiency dramatically declines with three or more input variables. Li et al. (2021) addressed the challenges inherent in multivariate input settings, particularly when there is significant interaction between input variables, through the application of a Weighted Additive Multiplicative Kernel (Lee et al., 2015). This approach, referred to as WAMK-SIS, has been shown to achieve notable variance reduction—superior to that of CE-SIS—especially in cases where the input dimensions range from three to four. Nonetheless, as the input dimension grows, WAMK-SIS still faces challenges posed by the curse of dimensionality, because it constructs a bivariate kernel for each pair of input variables.

Although surrogate-based sequential designs and IS techniques have demonstrated significant potential for variance reduction, their application has been largely limited to small datasets or low-dimensional problems. Our study bridges this gap by providing both analytical and practical solutions that can be applied to more complex problems.

3 Methodology

This section discusses the proposed OptiTreeStrat method.

3.1 Problem Description

We consider a black box computer model that generates the output $Y \in \mathbf{R}^1$ at input $\mathbf{X} \in \mathcal{D} \subseteq \mathbf{R}^d$. In stratified sampling, the domain $\mathcal{D}$ of input $\mathbf{X}$ is divided into multiple disjoint regions $\mathbf{\Omega}_i$'s for $i = 1, 2, \cdots, I$ such that $\mathcal{D} = \cup_i \mathbf{\Omega}_i$. Strata are a collection of $\mathbf{\Omega}_i$'s in this context. We consider hyperrectangular stratification (Pettersson and Krumscheid, 2022). Let $p_i = P(\mathbf{X} \in \mathbf{\Omega}_i)$ denote the probability that $\mathbf{X}$ belongs to the ith stratum $\mathbf{\Omega}_i$. Here, p_i can be obtained from the pdf $f(\cdot)$ of $\mathbf{X}$ by computing $\int_{\mathbf{\Omega}_i} f(x)dx$, assuming that $f(\cdot)$ is known. In case where $f(\cdot)$ is not predefined, with the assumption that ample data related to the input variables is accessible, p_i can be estimated by the ratio of the available data that fall into $\mathbf{\Omega}_i$. We are interested in estimating the expectation of a function of Y. Let Z denote a quantity of interest, i.e., $Z = w(Y)$. For instance, in reliability analysis, we can set $Z = \mathbb{I}(Y > l)$ to estimate the failure probability $\mathrm{E}(Z) = P(Y > l)$, where $\mathbb{I}(\cdot)$ denotes an indicator function and l is a resistance level.

Let us consider the stochastic computer model. Stochastic simulation with stochastic computer models often involves a two-level scheme (Choe et al., 2018), where the first level selects inputs and the second entails running the computer model to generate stochastic outputs. This approach unavoidably raises the 'exploration vs replication' trade-off. Ko and Byon (2022) investigate this issue by contrasting two IS estimators: one that employs replication for more thorough exploitation of important regions and another that focuses exclusively on exploration by running the model only once per input. While Binois et al. (2019) showed that replication can be beneficial in the context of GP surrogate modeling, Ko and Byon (2022) suggest that the exploration-focused estimator without replicates

is more advantageous in SIS due to its robustness against sample size rounding errors that might occur in implementing optimal allocation, as well as potential inaccuracies in surrogate models that are sometimes used in SIS. From this perspective, our study also employs an estimator that avoids the replicates at each input.

Suppose we draw n_i inputs of X_{ij} for $j = 1, 2, \cdots, n_i$ at the ith stratum for $i = 1, \cdots, I$. Let Z_{ij} denote an output obtained at X_{ij}. The stratified sampling estimator for $\mathrm{E}(Z)$ can be obtained from the conditional (or local) mean of each stratum as follows:

$$\hat{\mathrm{E}}(Z)_{SS} = \sum_{i=1}^{I} \frac{p_i}{n_i} \sum_{j=1}^{n_i} Z_{ij}, \tag{1}$$

which is an unbiased estimator of $\mathrm{E}(Z)$ for $n_i > 0$, $\forall i$. Its variance is

$$\mathrm{Var}(\hat{\mathrm{E}}(Z)_{SS}) = \sum_{i=1}^{I} \frac{p_i^2}{n_i^2} \mathrm{Var}\left(\sum_{j=1}^{n_i} Z_{ij}\right) = \sum_{i=1}^{I} p_i^2 \frac{\sigma_i^2}{n_i}, \tag{2}$$

where $\sigma_i^2 = \mathrm{Var}(Z_{ij}|\mathbf{X}_{ij} \in \mathbf{\Omega}_i)$ is the conditional (or local) variance of the response, given that the input belongs to the ith stratum $\mathbf{\Omega}_i$.

When the cost of drawing a sample is the same across strata, given the stratification structure $\{\mathbf{\Omega}_i\}_{i=1}^{I}$, the following budget allocation

$$n_i^* = n \frac{p_i \sigma_i}{\sum_{i=1}^{I} p_i \sigma_i} \tag{3}$$

minimizes the variance of $\hat{\mathrm{E}}(Z)_{SS}$ in (2) (Cochran, 1977). This rule allocates a larger budget to $\mathbf{\Omega}_i$, where p_i is large, or when the simulation output changes greatly over $\mathbf{\Omega}_i$. While p_i can be obtained from $f(\cdot)$, the conditional variance σ_i^2 cannot be computed because the conditional distribution of the response is typically unknown due to the black box nature of the computer model and thus, it needs to be estimated, e.g., using sample variance.

When strata are not pre-specified, designing these strata to achieve the maximum variance reduction is yet to be decided. Tying the optimal budget allocation in (3) with strata design together, $\mathrm{Var}(\hat{\mathrm{E}}(Z)_{SS})$ in (2) becomes

$$\mathrm{Var}(\hat{\mathrm{E}}(Z)_{SS,opt}) = \sum_{i=1}^{I} p_i^2 \frac{\sigma_i^2}{n_i^*} = \frac{1}{n}\left(\sum_{i=1}^{I} p_i \sigma_i\right)^2. \tag{4}$$

Minimizing (4) is equivalent to minimizing $\sum_{i=1}^{I} p_i\sigma_i$. *The focus of this study is to design the strata, i.e., to decide the strata size I^* and the corresponding set of strata $\{\mathbf{\Omega}_i^*\}_{i=1}^{I}$* as

$$I^*, \mathbf{\Omega}_1^*, \cdots, \mathbf{\Omega}_{I^*}^* = \underset{\{I, \mathbf{\Omega}_1, \cdots, \mathbf{\Omega}_I\}}{\arg\min} \sum_{i=1}^{I} p_i \sigma_i. \tag{5}$$

Searching for $\{\mathbf{\Omega}_i^*\}_{i=1}^{I^*}$ provides an important implication. The optimal strata design requires minimizing the within-stratum variance σ_i^2. That is, a desirable design would have a similar (or homogeneous) response pattern within each stratum, leading to small local variance σ_i^2, $\forall i$. To illustrate, consider the univariate input case where one wants to estimate the exceedance probability $\mathrm{E}(Z) = P(Y > l) = \int P(Y > l|x) f(x) dx$ with the conditional variance $\sigma_i^2 = P(Y > l | X \in \Omega_i)(1 - P(Y > l | X \in \Omega_i))$. Figure 1 depicts scatter plots with various strata choices. In the left figure, where an equi-distant design is used, the conditional variances (in particular, σ_2^2 and σ_3^2) could be substantially high because simulation outputs larger than l are mixed with those smaller than l within each stratum. In the middle design, the first two strata include more similar responses above and below l, respectively, representing a more effective design. When we subdivide the third stratum into finer strata in the last design, we can further reduce the within-stratum local variance. While we can be easily tempted to increase the strata size, there should be a right strata size because the budget for each stratum is an integer and we need to estimate σ_i^2.

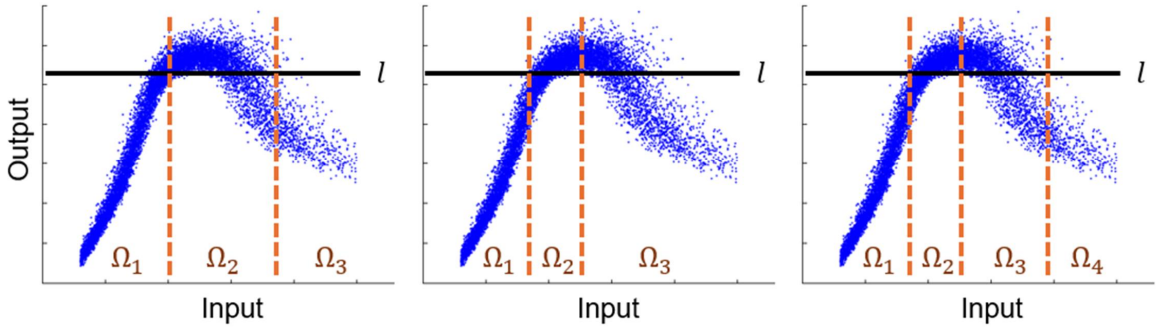


Figure 1: Different strata designs. The dotted vertical lines represent strata boundaries and the horizontal black line denotes the resistance level l.

It should be noted that the optimal strata design can be attained only when the response pattern over the input domain, e.g., $\mathrm{E}[Z|X]$ and $\mathrm{E}[Z^2|X]$, is known. Due to the black-box nature of computer models, one cannot exactly know these quantities. However, we can utilize domain knowledge or take small-scale pilot samples to approximate them. Assuming we can estimate $E[Z|X]$ and $E[Z^2|X]$, we first analytically derive the stratification structure to minimize the estimation variance, given the number of strata I, for univariate problems and extend the result for multivariate problems. Additionally, we would like to highlight that budget allocation and strata design are seamlessly decoupled. Once the optimal stratification is obtained by solving (5), we can allocate computational budgets using (3). Hence, we focus on strata design in the next section.

3.2 Univariate stratified sampling

Consider a univariate problem with $X \in \mathbf{R}$ with the ith stratum defined as $\mathbf{\Omega}_i = (a_{i-1}, a_i]$ for $i = 1, 2, \ldots, I$, with $a_0 = -\infty$ and $a_I = +\infty$. If we obtain the optimal strata design $\{\mathbf{\Omega}\}_{i=1}^{I}$, given I, the optimal strata size I^* can be obtained by comparing the variance of the

optimal strata for each I. In (4), p_i and σ_i depend on strata boundaries. Hence, obtaining the optimal partitions in (5) is equivalent to finding $\{a_i\}_{i=1}^{I-1}$. Let $h(x)$ and $t(x)$, respectively, denote the conditional expectations of the response and squared response, given x, i.e., $h(x) = \mathrm{E}(Z|X = x)$ and $t(x) = \mathrm{E}(Z^2|X = x)$. Theorem 1 provides the analytical solution for designing the strata, given I. The proof is available in the supplementary document.

Theorem 1 *Let $X \in \mathbf{R}$ denote a univariate input variable following a known density $f(x)$. Let Z denote a function of simulation output Y, i.e., $Z = w(Y) = w(g(X))$, where $g(X)$ is a stochastic computer model. Assume that $h(x)$, $t(x)$, and $f(x)$ are integrable. Given a strata size I, the optimal boundaries $\{a_i^*\}_{i=1}^{I-1}$ that minimize $\mathrm{Var}(\hat{\mathrm{E}}(Z)_{SS,opt})$ in (4) satisfy*

$$\frac{(t(a_i) + s_i) - 2h(a_i)\mu_i}{\sigma_i} = \frac{(t(a_i) + s_{i+1}) - 2h(a_i)\mu_{i+1}}{\sigma_{i+1}}, \tag{6}$$

for $i = 1, 2, \cdots, I-1$ with $\mu_i = \frac{1}{p_i}\int_{a_{i-1}}^{a_i} h(x)f(x)dx$ representing the conditional mean of the ith stratum, $s_i = \frac{1}{p_i}\int_{a_{i-1}}^{a_i} t(x)f(x)dx$ representing the conditional second moment of the ith stratum, and $p_i = \int_{a_{i-1}}^{a_i} f(x)dx$.

Applying the result in Theorem 1 involves additional considerations. First, due to the unknown conditional density of Y given $X = x$, each term in (6) needs to be estimated. We can approximate $h(x)$ and $t(x)$ using data collected in a pilot run by adopting adequate regression techniques such as spline and kernel regression. Then μ_i, s_i and σ_i can be estimated with numerical integration. Another difficulty is that μ_i, s_i, and σ_i are functions of a_i and a_{i-1}, which may not take closed-forms. Thus, one cannot immediately find the equation's root. To address these issues, we obtain a_i's iteratively, similar to the procedure

in Mease and Bingham (2006). Note that (6) can be rewritten as

$$0 = t(a_i)(k_i - k_{i+1}) - 2h(a_i)(k_i\mu_i - k_{i+1}\mu_{i+1}) + (k_i s_i - k_{i+1}s_{i+1}), \tag{7}$$

with $k_i = \frac{1}{\sigma_i}$. Suppose we have a_i's obtained in the previous iteration. With these, we calculate μ_i, s_i, and k_i. Consequently, given μ_i, s_i, and k_i, the right-hand side (RHS) of (7) becomes a function of a_i's, represented through $h(a)$ and $t(a)$. Lacking a closed-form solution for a_i, we rely on numerical approximation, specifically employing a root-finding algorithm. We use the uniroot() function available in R, following the methodology described in Brent (1971).

Algorithm 1 summarizes the procedure. We approximate $h(a)$ and $t(a)$ with their respective surrogates $\hat{h}(a)$ and $\hat{t}(a)$ using statistical methods in Line #2. While any regression techniques can be used, we use kernel regression (Nadaraya, 1964) in our implementation. Specifically, we obtain $\hat{h}(x)$ and $\hat{t}(x)$ by

$$\hat{h}(x) = \frac{\sum_{j=1}^{n} K(\frac{x-X_j}{h})Z_j}{\sum_{j=1}^{n} K(\frac{x-X_j}{h})}, \quad \hat{t}(x) = \frac{\sum_{j=1}^{n} K(\frac{x-X_j}{h})Z_j^2}{\sum_{j=1}^{n} K(\frac{x-X_j}{h})}, \tag{8}$$

where (X_j, Z_j) is the jth sample data for $j = 1, 2, \ldots, n$, $K(\cdot)$ is a univariate kernel function and h denotes the bandwidth. We use the Gaussian kernel and decide the bandwidth by optimizing the asymptotic mean integrated squared error (Li et al., 2021). In Line #3 of Algorithm 1, we stop the iteration when the splitting points at the tth iteration are sufficiently close to the previous iteration's splitting points.

In the failure probability estimation with $Z = \mathbb{I}(Y > l)$, $h(x)$ and $t(x)$ become equal, both denoting the conditional failure probability at x, i.e., $h(x) = t(x) = P(Y > l|X = x)$. It allows us to simplify (6), as shown in Corollary 1.

Algorithm 1 Univariate strata design in stochastic simulations

1: **Input**: number of strata I. Pilot samples $(X_j, Y_j), X_j \in \mathbf{R}, Y_j \in \mathbf{R}$ for $j = 1, \cdots, n_0$, where n_0 is the pilot sample size.

2: **Initialization**: Set the iteration number $t = 0$. Initialize the splitting points $\mathbf{a}^{(0)} = (a_1^{(0)}, \cdots, a_{I-1}^{(0)})$. Approximate $h(x)$ and $t(x)$ with pilot samples to get $\hat{h}(x)$ and $\hat{t}(x)$.

3: **while** convergence criterion is not satisfied **do**

4: Obtain p_i, μ_i, s_i, k_i using the current splitting points $\mathbf{a}^{(t)}$.

5: Solve (7) using a root-finding algorithm.

6: update $\mathbf{a}^{(t+1)} \leftarrow \mathbf{a}^{(t)}$.

7: set $t \leftarrow t + 1$

8: **end while**

9: Estimate σ_i^2 with sample variance for all $\mathbf{\Omega}_i = (a_{i-1}, a_i],\ i = 1, 2, \ldots, I$

10: Obtain n_i^* using (3).

11: **Output**: $\mathbf{a}^* = (a_1^{(t)}, \cdots, a_{I-1}^{(t)})$, $\mathbf{n}^* = (n_1^*, \cdots, n_I^*)$.

Corollary 1 *Let $X \in \mathbf{R}$ denote a univariate input variable following a known density $f(x)$. Consider $Z = \mathbb{I}(g(X) > l)$ where $g(X)$ is a stochastic computer model. Assume that $h(x)$ and $f(x)$ are integrable. Given a strata size I, the optimal boundaries $\{a_i^*\}_{i=1}^{I-1}$ that minimize $\mathrm{Var}(\hat{\mathrm{E}}(Z)_{SS,opt})$ in (4) satisfy*

$$h(a_i) = \frac{o_i o_{i+1}}{o_i o_{i+1} + 1}, \tag{9}$$

for $i = 1, \cdots, I-1$, where $h(x) = P(Y > l | X = x)$ and $o_i = \sqrt{\mu_i/(1-\mu_i)}$ with $\mu_i = \frac{1}{p_i}\int_{a_{i-1}}^{a_i} h(x)f(x)dx$ and $p_i = \int_{a_{i-1}}^{a_i} f(x)dx$.

The result in (9) shows that $h(a_i)$ at the optimal a_i only depends on the adjacent strata's

conditional means. In addition, the range of $h(a_i)$ is between 0 and 1, which aligns with the fact that $h(a_i)$ denotes the conditional failure probability at $X = a_i$.

Next, Theorem 2 below provides the analytical solution for designing the strata for deterministic computer models.

Theorem 2 *Let $X \in \mathbf{R}$ denote a univariate input variable following a known density $f(x)$. Let Z denote a function of simulation output Y, i.e., $Z = w(Y) = w(g(X))$, where $g(X)$ is a deterministic computer model. Assume that $w(x)$, $g(x)$, and $f(x)$ are integrable. Given a strata size I, the optimal boundaries $\{a_i^*\}_{i=1}^{I-1}$ that minimize $\mathrm{Var}(\hat{\mathrm{E}}(Z)_{SS,opt})$ in (4) satisfy*

$$\frac{(w(g(a_i))^2 + s_i) - 2w(g(a_i))\mu_i}{\sigma_i} = \frac{(w(g(a_i))^2 + s_{i+1}) - 2w(g(a_i))\mu_{i+1}}{\sigma_{i+1}}, \tag{10}$$

for $i = 1, 2, \cdots, I-1$ with $\mu_i = \frac{1}{p_i}\int_{a_{i-1}}^{a_i} w(g(x))f(x)dx$ representing the conditional mean of the ith stratum, $s_i = \frac{1}{p_i}\int_{a_{i-1}}^{a_i} w(g(x))^2 f(x)dx$ representing the conditional second moment of the ith stratum, and $p_i = \int_{a_{i-1}}^{a_i} f(x)dx$.

The result of Theorem 2 is a special case of Theorem 1. For a deterministic computer model, we have $h(x) = E(Z|X = x) = E(w(g(X))|X = x) = w(g(x))$ and $t(x) = E(Z^2|X = x) = E(w(g(X))^2|X = x) = w(g(x))^2$, because there is no randomness inside the computer model. Plugging these $h(x)$ and $t(x)$ into (6), we obtain (10).

3.3 Multivariate stratified sampling

Multivariate stratified sampling is prone to the curse of dimensionality, making it challenging to develop an effective strata design. The analytical approach presented in the previous section requires numerical integration over input variables, which grows computationally

costly and becomes numerically unstable or even intractable as the input dimension increases. Additionally, data sparsity may be a problem in estimating the terms in (6) when some strata have few data points.

Nevertheless, the analytical findings for the univariate case set the stage for successfully building strata with multivariate inputs. We note that the well-known parsimonious principle is present in many social, scientific, and engineering problems. It encourages us to use those crucial variables while dividing the input space, if they can be discovered. To achieve this goal, our strategy is to recursively identify the most crucial variable and segment the input domain one at a time and repeat the procedure. Our idea is similar to constructing a decision tree such as a classification and regression tree (CART) (James et al., 2013). Consider a binary tree where the strata are determined by the tree's terminal nodes. To build a tree, we use a greedy approach to select the best terminal node in the current tree, along with the best splitting variable and its partitioning point, that achieves the maximum variance reduction. Our approach, while similar to CART, contains a number of unique elements because of the distinct features of stratified sampling and our efforts to best utilize the analytical results obtained for the univariate context. It also decides the strata design $\{\Omega_i\}_{i=1}^{I}$ and strata size in an integrated way.

3.3.1 Recursive strata expansion

Suppose we have d input variables $\mathbf{X} \in \mathbf{R}^d$. Let us consider the ith terminal node in the current tree, with $\mathbf{\Omega}_i$ standing for its associated stratum. We grow a tree by subdividing it in hyperplane and adding two children nodes whose domains are specified by a splitting

variable and its splitting point with their associated two children strata. We choose the splitting variable and its partitioning point based on how much variance they can reduce. Recall that our objective function for the variance minimization problem is reduced to $\sum_i p_i \sigma_i$ as shown in (5). Among all possible input variables and their splitting points within $\mathbf{\Omega}_i$, we get the splitting variable and its partitioning point as

$$X^i_{s^*}, a^i_{s^*} = \underset{X_s, a^i_s \ \ s=1,\ldots,d}{\arg\min} \left[p_{i,l}(X_s, a^i_s)\sigma_{i,l}(X_s, a^i_s) + p_{i,r}(X_s, a^i_s)\sigma_{i,r}(X_s, a^i_s) \right], \tag{11}$$

where $p_{i,l}(\cdot)$ and $p_{i,r}(\cdot)$ are the probability of inputs belonging to $\mathbf{\Omega}_{i,l}(X_s, a^i_s) = \{\mathbf{X} | X_s \leq a^i_s, \mathbf{X} \in \mathbf{\Omega}_i\}$ and $\mathbf{\Omega}_{i,r}(X_s, a^i_s) = \{\mathbf{X} | X_s > a^i_s, \mathbf{X} \in \mathbf{\Omega}_i\}$, respectively, and $\sigma_{i,l}(\cdot)$ and $\sigma_{i,r}(\cdot)$ denote the corresponding conditional standard deviations of the response.

To illustrate, let us consider a 3-dimensional problem shown in Figure 2. At the ith node, we consider potential splits based on each of the three variables. For a split on variable X_s, we obtain the splitting point a^i_s using Theorem 1. We then evaluate the combined variance of the resulting left and right child nodes as $[p_{i,l}(X_s, a^i_s)\sigma_{i,l}(X_s, a^i_s) + p_{i,r}(X_s, a^i_s)\sigma_{i,r}(X_s, a^i_s)]$ for each variable. After examining all three variables, we select the variable that minimizes this combined variance measure, e.g., X_3 as illustrated in Figure 2.

Here, the key element is that the analytical procedure discussed in Section 3.2 is adopted to find the splitting point $a^i_{s^*}$ in (11). Specifically, we employ the result in Theorem 1 with $I = 2$ to obtain a^i_s for each input X_s for $s = 1, \cdots, d$. Then we choose the best variable $X^i_{s^*}$ and its splitting point $a^i_{s^*}$ for terminal node i. We apply this procedure to each terminal node and identify the terminal node that results in the greatest variance reduction when it is branched. Suppose the tree currently consists of T terminal nodes. We determine the node to further subdivide, tying together with its splitting variable and partitioning point,

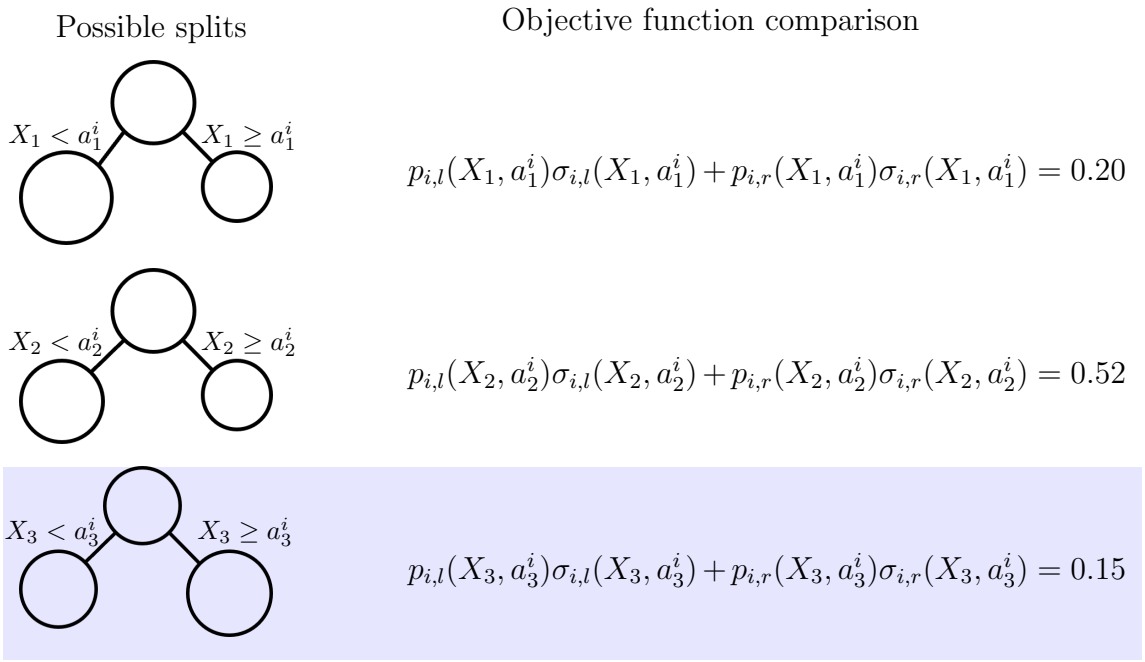


Figure 2: Selection of a splitting variable in the ith node

that achieves the minimum estimation variance as follows.

$$i^* = \arg\min_i \left[\sum_{i'=1, i' \neq i}^{T} p_{i'} \sigma_{i'} + \min_{X_s, a_s^i} \left[p_{i,l}(X_s, a_s^i)\sigma_{i,l}(X_s, a_s^i) + p_{i,r}(X_s, a_s^i)\sigma_{i,r}(X_s, a_s^i) \right] \right]. \quad (12)$$

3.3.2 Determination of strata size

Now we discuss how to decide the strata size I. Theoretically, we can achieve a greater variance reduction with a larger tree (Pettersson and Krumscheid, 2022). However, this is the case when σ_i's are exactly known, and the optimal budget distributions across strata just so happen to be positive integers. In practice, σ_i's must be estimated with data, and the allocations in (3) are frequently rounded. The estimation accuracy would decline as the tree size increases because fewer samples are taken from each stratum (terminal node). As a result, we must stop growing a tree when the variance reduction is barely noticeable.

Choosing the appropriate strata size is analogous to deciding the tree size in CART (Liu et al., 2022). However, there are a few vital differences. In CART, a tree is typically fully grown until each terminal node contains the minimum required number of samples prior to applying a pruning procedure. However, growing a whole tree could incur non-negligible computational overhead in stratified sampling where computational efficiency is important because we need to solve the univariate design problem for each variable at each terminal node. Thus, we stop growing the tree when the reduced variance obtained by adding two more children nodes becomes negligible in comparison to the variance obtained by the tree so far (up to the point of splitting), i.e., when the following reduction rate (RR) falls below a predetermined threshold, e.g., 0.05, denoted by l_{RR}.

$$RR = \frac{p_{i^*}\sigma_{i^*} - (p_{i^*,l}(X_{s^*}^{i*}, a_{s^*}^{i*})\sigma_{i^*,l}(X_{s^*}^{i*}, a_{s^*}^{i*}) + p_{i^*,r}(X_{s^*}^{i*}, a_{s^*}^{i*})\sigma_{i^*,r}(X_{s^*}^{i*}, a_{s^*}^{i*}))}{\sum_{i=1}^{T} p_i\sigma_i}, \tag{13}$$

where the numerator implies the amount of reduction when the tree grows by splitting the node i^*, which is determined in (12), and T is the current tree size.

Once we construct a sufficiently large tree, we prune it to avoid overfitting. We use the following complexity cost, which seeks to balance variance reduction with tree complexity, similar to the complexity parameter in CART (James et al., 2013):

$$C_\alpha(\mathbf{\Omega}') = \sum_{i \in V(\mathbf{\Omega}')} p_i\sigma_i + \alpha|\mathbf{\Omega}'|, \tag{14}$$

for a subtree $\mathbf{\Omega}'$ in $\mathbb{\Omega}'$, where $\mathbb{\Omega}'$ indicates a set of all possible subtrees, $|\mathbf{\Omega}'|$ is the number of terminal nodes in $\mathbf{\Omega}'$, $V(\cdot)$ denotes the set of terminal nodes and α is a tuning parameter.

Given α, we identify the subtree $\mathbf{\Omega}_\alpha$ that produces the smallest $C_\alpha(\cdot)$. Finding a good strata size requires making an adequate choice of α. Suppose that we consider M different

α's and for each α, we get the best subtree $\mathbf{\Omega}_\alpha$. A commonly used approach to decide α is assessing the performance of $\mathbf{\Omega}_\alpha$ with validation sets. However, blindly employing the traditional validation approach can lead to incorrect selection of α in the context of stratified sampling. Consider M candidate subtrees, each decided from M choices of α's. To evaluate the performance of each subtree, samples need to be chosen based on the optimal allocation rule in (3). Thus, M different validation sets should be constructed to evaluate the subtree structure $\mathbf{\Omega}_\alpha$, which increases computational overheads substantially because computer models should generate the necessary data.

To alleviate the computational burden, we modify the original validation set with resampling. When the data has been split into a training set and a validation set, we utilize the training set to build a suitably sized tree using the reduction rate in (13). Then, for each subtree, we resample data from the validation set according to the subtree's optimal allocation rule, compute the total variance of the subtree with that resample, and choose α with the smallest variance. Since the same samples can be taken, our resampling strategy might not totally solve the overfitting problem, but it avoids excessive computing overhead while determining the right strata size using a separate validation set.

Algorithm 2 provides a summary of the steps for determining the strata design with multivariate inputs in the OptiTreeStrat approach. The procedure can be generalized to 5-fold (or 10-fold) cross-validation with resampling but at the expense of increased computation. Additionally, the process for determining the tree size is illustrated in Figure 3. Consider the initial tree in the upper panel, obtained from Phase 1. During Phase 2, we evaluate three candidate subtrees and select the optimal subtree through a validation pro-

cedure. Specifically, for each α, we identify the candidate subtree $\mathbf{\Omega}_\alpha$ that exhibits the lowest cost-complexity $C_\alpha(\mathbf{\Omega}')$ among the three subtrees (Step 1 of Phase 2 in Figure 3, corresponding to line 11 in Algorithm 2). Subsequently, we perform resampling within the validation set, where the sample sizes for each node are defined by the optimal allocation in (3) for each candidate subtree (Step 2 of Phase 2 in Figure 3, corresponding to lines 12-15 in Algorithm 2). Finally, we determine the appropriate α and its corresponding subtree, which together yield the minimal objective function value (Step 3 of Phase 2 in Figure 3, corresponding to line 17 in Algorithm 2).

Note that in line 15 of Algorithm 2, computing p_i could be challenging if the ith stratum $\mathbf{\Omega}_i$ spans multiple interdependent variables. However, our strategy, inspired by the parsimonious principle, would identity the important variables and focuses on stratifying the input space based upon them. Thus, the computation of p_i becomes manageable and maintains high accuracy. Further investigation will be needed to address high-dimensional situations for which the parsimonious principle does not apply.

4 Numerical Examples

4.1 Problem Setting

We implement OptiTreeStrat with three numerical examples that represent stochastic computer models with the following data-generating structures (Li et al., 2021).

$$\mathbf{X}_m \sim N(\mathbf{0}, \mathbf{I}_m), \qquad Y_m|\mathbf{X}_m \sim N(\mu_m(\mathbf{X}_m), 1),$$

Algorithm 2 Multivariate strata design in OptiTreeStrat

Input: Pilot samples $(\mathbf{X}_j, Y_j), \mathbf{X}_j \in \mathbf{R}^d, Y_j \in \mathbf{R}$ for $j = 1, \cdots, n_0$, where n_0 is the pilot sample size, the tree size threshold l_{RR}, a set of pruning parameters $\mathbb{A} = \{\alpha_1, \cdots, \alpha_M\}$, reconstructed validation set size n_B.

Initialization: Divide the pilot dataset into training and validation sets. Set $\Omega^{'}$, $\Omega^{'}_{\mathbb{A}}$, $\mathbf{\Omega}^{branch}$, $\mathbf{\Omega}^{left}$, and $\mathbf{\Omega}^{right}$ as empty sets. Set $RR = 1$ and $\mathbf{\Omega} = \mathcal{D}$.

Phase 1: Obtaining a suitably-sized strata with training set:

while $RR \geq l_{RR}$ **do**

  Update $\mathbf{\Omega} \leftarrow \{\mathbf{\Omega} \backslash \{ \mathbf{\Omega}^{branch} \}\} \cup \{\mathbf{\Omega}^{left}, \mathbf{\Omega}^{right} \}$ and $\Omega^{'} \leftarrow \Omega^{'} \cup \mathbf{\Omega}$.

  for $i = 1, \cdots, |\mathbf{\Omega}|$ **do**

    Get $X^i_{s^*}$ and $a^i_{s^*}$ by solving (11) using Algorithm 1 with $I = 2$ for $s = 1, \cdots, d$,

  end for

  Solve (12) to get i^* and calculate RR in (13)

  Set $\mathbf{\Omega}^{branch} = \mathbf{\Omega}_{i^*}$, $\mathbf{\Omega}^{left} = \mathbf{\Omega}_{i^*,l}(X^{i*}_{s^*}, a^{i*}_{s^*})$, and $\mathbf{\Omega}^{right} = \mathbf{\Omega}_{i^*,r}(X^{i*}_{s^*}, a^{i*}_{s^*})$

end while

Phase 2: Finding the best strata with validation set:

For each $\alpha \in \mathbb{A}$, find $\mathbf{\Omega}_\alpha$ that minimizes $C_\alpha(\mathbf{\Omega}^{'})$ in (14), $\forall \mathbf{\Omega}^{'} \in \Omega^{'}$ and $\Omega^{'}_{\mathbb{A}} \leftarrow \Omega^{'}_{\mathbb{A}} \cup \mathbf{\Omega}_\alpha$.

for $\mathbf{\Omega}_\alpha \in \Omega^{'}_{\mathbb{A}}$ **do**

  Obtain n^*_i in (3) for $i = 1, \cdots, |\mathbf{\Omega}_\alpha|$ with $n = n_B$.

  Resample data from the validation set to generate n^*_i samples.

  Obtain p_i and $\hat{\sigma}_i$ from resampled data and compute $U_\alpha = \sum_i p_i \hat{\sigma}_i$.

end for

Output: $\mathbf{\Omega}^*_\alpha = \arg\min_{\mathbf{\Omega}_\alpha \in \Omega^{'}_{\mathbb{A}}} U_\alpha$

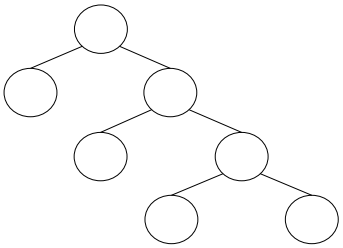
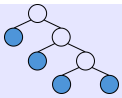
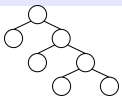
Phase 1: Creating suitably sized strata with training data			
			
Phase 2			
Step 1: Choosing the best subtree for each α with training data		Step 2: Resampling in validation data according to allocation rule in (3)	Step 3: Choosing the best tree
Subtree ($\Omega^{'}$)	Complexity cost ($C_\alpha(\Omega^{'})$)	Objective function (U_α)	Best tree (Ω^*_α)
α_1 	0.22		
	0.20		
	0.19	0.22	
α_2 	0.24		
	0.23	0.17	Choosing as the best tree
	0.25		

Figure 3: Illustration to determine the tree size.

for $m = 1, 2, 3$, where m corresponds to an example index with $\mathbf{X}_1 = (X_1, X_2, X_3)$, $\mathbf{X}_2 = (X_1, X_2, X_3, X_4)$, $\mathbf{X}_3 = (X_1, X_2, \cdots, X_{10})$, and $\mathbf{I}_m$ indicates an identity matrix.

The conditional mean of Y, given $\mathbf{X}$, of each example is as follows.

$$\mu_1(\mathbf{X}_1) = 65 - 40e^{-0.2\sqrt{(X_1^2+X_2^2)/2}} - 20e^{-0.2|X_1|} - 5e^{-0.2\sqrt{(X_2^2+X_3^2)/2}}$$
$$- \sum_{1\le i<j\le 3} e^{\cos(2\pi X_i X_j)} - e^{\cos(2\pi X_1 X_2 X_3)}$$

$$\mu_2(\mathbf{X}_2) = 65 - 40e^{-0.2\sqrt{(X_1^2+X_2^2)/2}} - 20e^{-0.2|X_1|} - 5e^{-0.2\sqrt{(X_2^2+X_3^2+X_4^2)/3}} - \sum_{1\le i<j\le 3} e^{\cos(2\pi X_i X_j)}$$

$$\mu_3(\mathbf{X}_3) = 65 - 40e^{-0.2\sqrt{(X_1^2+X_2^2)/2}} - 20e^{-0.2|X_1|} - 5e^{-0.2\sqrt{(X_2^2+X_3^2)/2}}$$
$$- 0.1(e^{-0.2\sqrt{(X_4^2+X_5^2)/2}} + e^{-0.2\sqrt{(X_6^2+X_7^2)/2}} + e^{-0.2\sqrt{(X_8^2+X_9^2+X_{10}^2)/3}}) - \sum_{1\le i<j\le 3} e^{\cos(2\pi X_i X_j)}$$

In all examples, X_1 is the most significant input variable and X_2 comes next, whereas other variables are less significant. We are interested in estimating the failure probability $P(Y > l)$, with $l = 17.9$, 18.9, and 18.74, respectively, for Examples 1, 2, and 3. These resistance levels roughly correspond to the failure probability level $P_t = 0.01$.

To implement the proposed OptiTreeStrat, we first draw pilot samples. Conclusive theoretical results to determine the appropriate pilot sample size have not been established yet in the literature. From a wide range of experiments, we derive an empirical rule for determining the required pilot sample size n_0 for the failure probability estimation as $n_0 \approx 10\sqrt{d(1-P_t)/P_t}$. Noting that the coefficient of variation (CoV) of crude Monte Carlo (CMC) estimator with n_0 samples is $\sqrt{(1-P_t)/(n_0 P_t)}$, this rule suggests that a larger pilot sample size is necessary when the estimator's CoV increases. In our implementation, based on the rule, we draw 170, 200, and 340 pilot samples using the orthogonal array-based LHS (abbreviated as OA-LHS) in each example. The detailed explanation of how to construct the OA in OA-LHS can be found in the supplementary document. Applying this rule in

practical settings would require careful consideration, as P_t is typically unknown prior to the analysis. In situations where even a rough guess of P_t is unavailable, an alternative approach is to iteratively increase the pilot sample size until certain criteria, such as sample CoV, are satisfied (Liu et al., 2022). Please refer to the supplementary material for more details about deciding the pilot size.

We consider several benchmark methods, including stratified sampling with equi-distanced strata design (Equi-SS), CMC, and WAMK-SIS (Li et al., 2021), as well as the approach in Pettersson and Krumscheid (2022), termed Adaptive-SS. The Equi-SS divides the range of each input into two evenly-spaced intervals. CMC draws samples from the original input distribution $f(x)$. Additionally, we consider LHS and its variants, including rank-based LHS (hereafter, referred to as rank LHS) (Stein, 1987) and OA-LHS (Tang, 1993). We use the same seed for implementing these competing methods in each experiment.

We use the $1,200$ total computation budgets in all approaches, including pilot samples. In OptiTreeStrat and WAMK-SIS, the rest of the budget–after allocating the pilot budget–is distributed evenly across five iterations to implement batch sequential design. For more discussion on deciding the batch size and number of batches, please refer to the supplementary material. In computing the failure probability, we exclude pilot samples because they are primarily intended for initial learning rather than for minimizing estimation variance, similar to the approach in Li et al. (2021).

4.2 Implementation Results

The results from 100 experiments are summarized in Table 1. Given the computational resources, Equi-SS could not solve Example 3, since some strata lack sufficient data with extensive strata size 2^{10}. Adaptive-SS does not achieve a substantial variance reduction over CMC, because it relies on an equi-probable split, leading to an ineffective stratification. Consequently, this inhibits the method's ability to target key regions that are critical for variance reduction. The three LHS methods demonstrate enhanced variance reduction capabilities when compared to Equi-SS, CMC, and Adaptive-SS. Nonetheless, they generally underperform relative to OptiTreeStrat.

As an illustration, Figure 4 depicts the final stratification structure realized in one of our experiments for Example 1. Our approach uses the most important variable X_1 as the splitting variable in most cases, resulting in an efficient estimator with a smaller SE. Additionally, the left plot of Figure 5 tracks the SE of 100 failure probability estimates across five different sample sizes (or batches). In the case of OA-LHS, sampling is conducted using an OA of a size similar to each sample size. The right box plot presents the distribution of failure probability estimates derived from 100 experiments. Our method generally shows the smallest SEs and interquartile ranges across all examples. Similar figures for Examples 2 and 3 are provided in the supplementary document.

We would like to comment on the computational efficiency of WAMK-SIS. Even though it generates reasonably good performance, its computational cost increases quickly as the input dimension increases. With d input variables, it constructs $d(d-1)/2$ bivariate kernels. The computation required to obtain all kernel bandwidths with large d increases

Table 1: Comparison results from 100 experiments to estimate the failure probability (note: * marking indicates a statistically significant difference at the 5% significance level in the F-test, compared to OptiTreeStrat.)

	Example 1		Example 2		Example 3	
	Mean	SE	Mean	SE	Mean	SE
OptiTreeStrat	**0.0099**	**0.0015**	0.0103	0.0021	**0.0100**	**0.0013**
Equi-SS	0.0096	0.0030*	0.0102	0.0036*	NA	NA
CMC	0.0099	0.0026*	0.0101	0.0027*	0.0106	0.0029*
Adaptive-SS	0.0096	0.0026*	0.0092	0.0029*	0.0101	0.0029*
LHS	0.0102	0.0023*	0.0104	0.0020	0.0102	0.0021*
Rank-LHS	0.0102	0.0022*	0.0100	0.0021	0.0102	0.0021*
OA-LHS	0.0100	0.0019*	**0.0095**	**0.0020**	0.0103	0.0022*
WAMK-SIS	0.0101	0.0017	0.0106	0.0022	0.0097	0.0024*

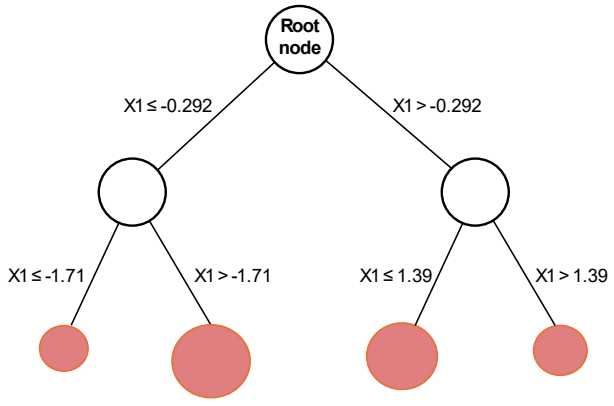


Figure 4: Example of stratification structures in Example 1.

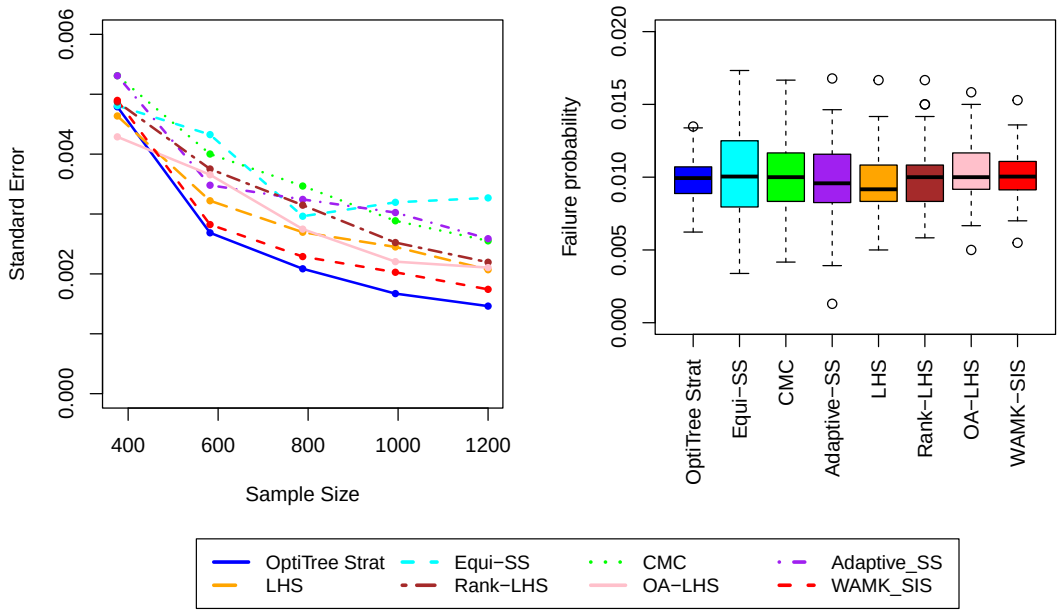


Figure 5: Comparison results of Example 1 from 100 experiments. Left: SE across five different sample sizes marked by solid circles. Right: box plots of failure probability estimates

significantly. On the contrary, in our approach, even though we employ nonparametric regression to approximate $h(x)$ and $t(x)$ for each input, we do so with a univariate kernel, which increases the computational overhead linearly. In Examples 1 and 2, for example, WAMK-SIS takes two to three times longer than OptiTreeStrat. With ten variables in Example 3, WAMK-SIS takes six times longer. The increased efficiency and the reduced computational cost of our method are further underscored by a comparative analysis against the sequential design strategy known as ECL, as introduced in Cole et al. (2023). The ECL method's execution time extends over several days, which is significantly longer by an order of magnitude than that required by our approach. More detailed results are included in the supplementary document.

Additionally, we assess the method's effectiveness across a broad spectrum of scenarios, including various computational budgets, diverse input distributions, high-dimensional case, heterogeneous variance, multi-modal response, and various response types. In most cases, our approach demonstrates superior performance over benchmark methods. Comprehensive comparison results are provided in the supplementary material.

5 Wind turbine case study

5.1 Problem setting

We conduct a case study for evaluating wind turbine reliability with the National Renewable Energy Laboratory (NREL)'s aeroelastic stochastic computer models, including TurbSim (Jonkman, 2009) and FAST (Jonkman et al., 2005). Given the input wind condition, Turbsim and FAST generate stochastic load responses in blades, towers, and other locations. One of the significant load responses is the 10-minute maximum tip deflection, so we employ it as the output variable Y in our study (Li et al., 2021; Choe et al., 2016).

Based on the international design standard (International Electrotechnical Commission, 2005) and literature, we consider five input variables. First, to describe wind speed V, we utilize a truncated Rayleigh distribution with a scale parameter of $10\sqrt{2/\pi}$ on the interval of [3, 25] (m/s), as recommended in Moriarty (2008). Next, for turbulence intensity TI, we use the Normal Turbulence Model Class B, which is one of the most commonly used models (Jonkman, 2009). We assume that TI follows a log-normal distribution as $TI|V \sim Lognormal\left(\log\left(\frac{\mu_{TI}^2(V)}{\sqrt{\mu_{TI}^2(V)+\sigma_{TI}^2}}\right), \log\left(1+\frac{\sigma_{TI}^2}{\mu_{TI}^2(V)}\right)\right)$ with $\mu_{TI}(V) = 0.14(0.75V+5.6)/V$ and

$\sigma_{TI} = 0.05$. Similarly, the wind shear S, given V, is assumed to follow the lognormal distribution as $S|V \sim N(\mu_S(V), \sigma_S^2(V))$ with $\mu_S(V) = 2.63 \times 10^{-4}V^3 - 1.09 \times 10^{-2}V^2 + 1.285 \times 10^{-1}V - 1.32 \times 10^{-1}$ and $\sigma_S(V) = 7.767 \times 10^{-5}V^3 - 3.43 \times 10^{-3}V^2 + 3.4 \times 10^{-2}V - 1.3 \times 10^{-1}$ (Li et al., 2021; Ding, 2019).

For a vertical angle VA of the wind, the TurbSim user's guide suggests using a small vertical angle and not exceeding 45° to avoid generating unusual values (Jonkman, 2009). In this study, we use the truncated normal distribution with a zero mean and variance 9 within the interval [-10,10]. Lastly, the density of surface roughness length SR is also assumed to obey the truncated normal distribution with its mean 0.03 (the default value in TurbSim), and a small variance 10^{-6} over $[0.01, 0.05]$.

5.2 Implementation results

We consider two input settings—one with the first three variables (V, TI, and S) and another with five variables (V, TI, S, VA, and SR). The first setting is the same as in Li et al. (2021), where VA and SR are fixed at their mean values. We evaluate $P(Y > l)$, the probability of tip deflection exceeding a threshold $l = 2.45$. We consider a total computation budget of 2,600. We draw 170 and 240 pilot samples for each setting. The remaining budget is then evenly allocated across 10 iterations.

We investigate OptiTreeStrat's strata design. Figure 6 provides scatter plots of the tip deflection vs each input variable (left) and stratification construction (right), using the samples collected in previous iterations. We apply OptiTreeStrat to obtain the stratification structure, as shown in the right panel. The scatter plots in the left panel illustrate the

samples within each stratum corresponding to the white branching node in the right panel. Specifically, each of the first, second, and third columns is the scatter plot of wind speed vs response, turbulence intensity vs response, and wind shear vs response. Each vertical line indicates the splitting value of the respective input variable.

The input domain is first partitioned with the wind speed at $V = 12.1$ m/s, because, at wind speeds less than 12.1 m/s, most responses are less than l. The solid line in the top left sub-figure illustrates that wind speed is selected as the first splitting variable. Between the two child nodes of $V \leq 12.1$ and $V > 12.1$ (the two nodes in the second row of the right plot), the high-wind speed node ($V > 12.1$) is chosen. The second split is obtained with $TI = 0.25$, as shown in the middle sub-figure of the second row. Lastly, the node with $V > 12.1$ and $TI \leq 0.25$ is chosen to be further partitioned with the wind speed at $V = 16.4$ m/s, corresponding to the bottom panel of the left scatter plots. In the right panel, the size of each terminal node represents the budget allocation. With the five input variables, our approach generates similar designs, because the importance of V and TI dominates others in both cases. The supplementary document includes more details.

Table 2 compares the performance of our approach with alternatives. Recall that in the numerical examples presented in Section 4 and the extended settings detailed in the supplementary material, both WAMK-SIS and OA-LHS outperform the other methods; consequently, we exclude other benchmarks from our results. Additionally, we consider Latin Hypercube Sampling Dependent (LHSD) which could potentially perform better than LHS when input variables are dependent (Mondal and Mandal, 2020). The results show that our OptiTreeStrat approach, which utilizes carefully planned strata, can significantly

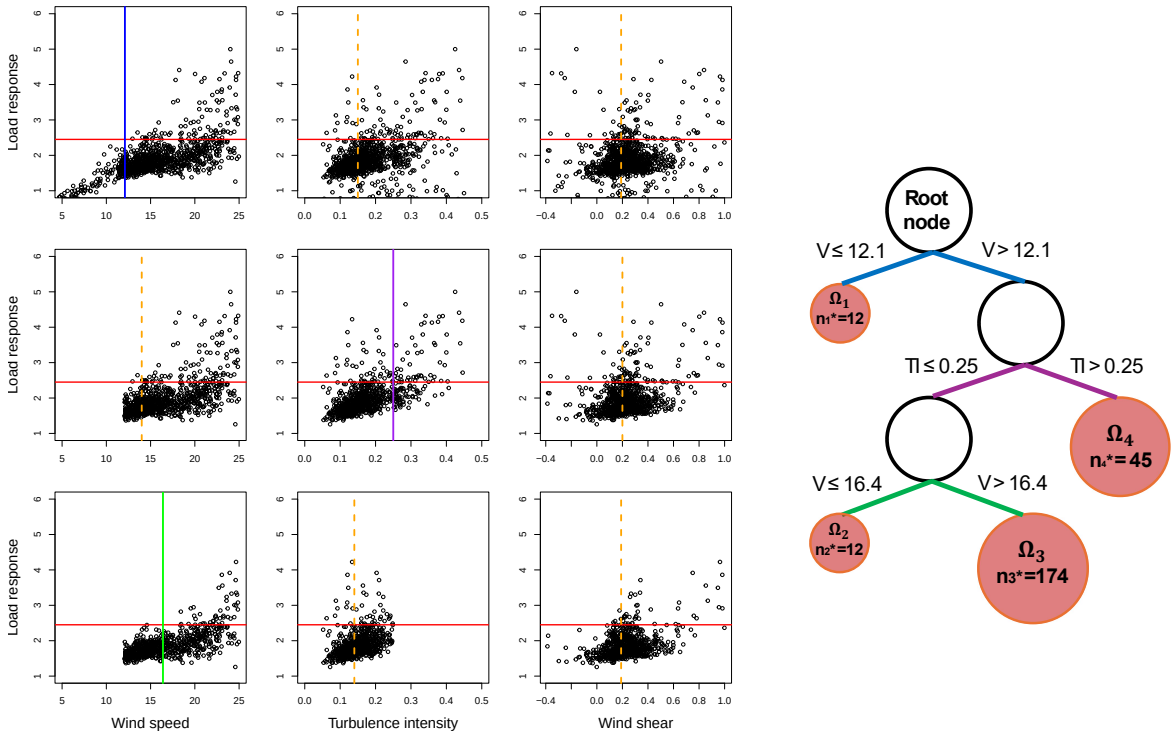


Figure 6: Example of strata design in wind turbine case study. Left: scatter plots of the tip deflection vs each input variable where the horizontal line represents the resistance level l and the vertical lines imply breaking points; Right: stratification construction, where each splitting variable and its corresponding value match those in the left figure.

reduce the estimation variance.

6 Conclusion

This study investigates the stratification structure to minimize the estimation variance in stochastic simulation. Based on the analytical finding in a univariate input, a robust algorithm for multivariate inputs by using the concept of the decision tree is introduced. OptiTreeStrat determines the split point with optimization and can avoid the data scarcity

Table 2: Comparison results from 25 experiments in wind turbine case study

	3-dimensional case		5-dimensional case	
	Mean	SE	Mean	SE
OptiTreeStrat	**0.0098**	**0.0007**	**0.0095**	**0.0010**
OA-LHS	0.0100	0.0015	0.0097	0.0013
LHSD	0.0098	0.0019	0.0099	0.0019
WAMK-SIS	0.0099	0.0010	0.0099	0.0019

issue by providing an appropriate number of strata when the input dimension is high, but a few key variables exist. The numerical studies and wind turbine case study present that OptiTreeStrat successfully recognizes important variables to effectively partition the input space and is scalable because it explores variables one by one without considering complicated functions such as multivariate kernels.

For computer models that demand extensive runtime, such as hours or even days for a single run, surrogate-based methods, or experimental designs such as LHS, could be useful. Conversely, OptiTreeStrat is well-suited for computer models with shorter runtime, on the scale of minutes, or when large computational resources are available. The wind turbine simulations utilized in our study have runtimes of a few minutes. The benefit of our approach is most notable when estimating small failure probabilities, especially those within the 10^{-2} to 10^{-3} range.

In the future, we will extend the approach to accommodate more complex cases by exploring multiple splits and addressing multivariate responses. When the input distribution

is uncertain, robust variance reduction techniques will be developed to effectively account for such uncertainties. Additionally, our investigation will extend to establishing more general guidelines for determining the pilot sample size and batch size in the context of batch sequential experiments. We also plan to combine the proposed approach with dimension reduction techniques (Li, 1991; Li and Yin, 2008) to enable a more flexible stratification design. Finally, we will extend the application of our findings to the estimation of extreme quantiles (Pan et al., 2020; Cannamela et al., 2008).

Acknowledgements

The authors thank the editor, the associate editor, and reviewers for their constructive and thoughtful comments on various aspects of this work.

Disclosure Statement

The authors have declared that there are no conflicts of interest.

Funding

This work was partly supported by the U.S. National Science Foundation (Grant No. CMMI-2226348 and CMMI-2226347) and National Research Foundation of Korea (Grant No.: NRF-2021R1A2C1094699 and NRF-2021R1A4A1031019)

ORCID

Jaeshin Park: https://orcid.org/0009-0002-4347-5914

Eunshin Byon: https://orcid.org/0000-0002-2506-1606

Young Myoung Ko: https://orcid.org/0000-0003-0659-6688

Sara Shashaani: https://orcid.org/0000-0001-8515-5877

Supplementary Materials

The supplementary materials contain the following: (i) proofs of the theorems, (ii) discussion on how the pilot sample size and batch size are determined, (iii) details on the stratification structure and bandwidth used in the numerical examples, (iv) extensive comparison results, including additional numerical experiments, and (v) further results from the case study. Additionally, we have included the codes to reproduce the results in Table 1 and Figure 4. For detailed instructions to run the codes, please refer to the 'ReadMe' file.

References

Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. Statistics and Computing, 22:773–793.

Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2019). Replication or explo-

ration? sequential design for stochastic simulation experiments. Technometrics, 61(1):7–23.

Brent, R. P. (1971). An algorithm with guaranteed convergence for finding a zero of a function. The Computer Journal, 14(4):422–425.

Cannamela, C., Garnier, J., and Iooss, B. (2008). Controlled stratification for quantile estimation. The Annals of Applied Statistics, 2(4):1554–1580.

Cao, Q. D. and Choe, Y. (2019). Cross-entropy based importance sampling for stochastic simulation models. Reliability Engineering & System Safety, 191:106526.

Choe, Y., Byon, E., and Chen, N. (2015). Importance sampling for reliability evaluation with stochastic simulation models. Technometrics, 57(3):351–361.

Choe, Y., Lam, H., and Byon, E. (2018). Uncertainty quantification of stochastic simulation for black-box computer experiments. Methodology and Computing in Applied Probability, 20:1155–1172.

Choe, Y., Pan, Q., and Byon, E. (2016). Computationally efficient uncertainty minimization in wind turbine extreme load assessments. Journal of Solar Energy Engineering, 138(4):041012.

Cochran, W. G. (1977). Sampling Techniques. John Wiley & Sons, New York, NY.

Cole, D. A., Gramacy, R. B., Warner, J. E., Bomarito, G. F., Leser, P. E., and Leser, W. P. (2023). Entropy-based adaptive design for contour finding and estimating reliability. Journal of Quality Technology, 55(1):43–60.

Ding, Y. (2019). Data Science for Wind Energy. CRC Press, Boca Raton, FL.

Etoré, P., Fort, G., Jourdain, B., and Moulines, É. (2011). On adaptive stratification. Annals of Operations Research, 189(1):127–154.

International Electrotechnical Commission (2005). Wind turbines—part 1: Design requirements, iec/tc88, 61400-1 (3rd ed.). Technical report, Geneva.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An Introduction to Statistical Learning, volume 112. Springer, New York, NY.

Jonkman, B. J. (2009). TurbSim User's guide: Version 1.50. Technical report, National Renewable Energy Lab.(NREL), Golden, Colorado.

Jonkman, J. M., Buhl, M. L., et al. (2005). FAST User's guide. Technical report, National Renewable Energy Lab.(NREL), Golden, Colorado.

Ko, Y. M. and Byon, E. (2022). Optimal budget allocation for stochastic simulation with importance sampling: exploration vs. replication. IISE Transactions, 54(9):881–893.

Kurtz, N. and Song, J. (2013). Cross-entropy-based adaptive importance sampling using gaussian mixture. Structural Safety, 42:35–44.

Lee, G., Ding, Y., Genton, M. G., and Xie, L. (2015). Power curve estimation with multivariate environmental factors for inland and offshore wind farms. Journal of the American Statistical Association, 110(509):56–67.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. Journal of the American Statistical Association, 86(414):316–327.

Li, L. and Yin, X. (2008). Sliced inverse regression with regularizations. Biometrics, 64(1):124–131.

Li, S., Ko, Y. M., and Byon, E. (2021). Nonparametric importance sampling for wind turbine reliability analysis with stochastic computer models. The Annals of Applied Statistics, 15(4):1850–1871.

Liu, B., Yue, X., Byon, E., and Kontar, R. A. (2022). Parameter calibration in wake effect simulation model with stochastic gradient descent and stratified sampling. The Annals of Applied Statistics, 16(3):1795 – 1821.

Mease, D. and Bingham, D. (2006). Latin hyperrectangle sampling for computer experiments. Technometrics, 48(4):467–477.

Mondal, A. and Mandal, A. (2020). Stratified random sampling for dependent inputs in Monte Carlo simulations from computer experiments. Journal of Statistical Planning and Inference, 205:269–282.

Moriarty, P. (2008). Database for validation of design load extrapolation techniques. Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology, 11(6):559–576.

Nadaraya, E. A. (1964). On estimating regression. Theory of Probability & Its Applications, 9(1):141–142.

Neddermeyer, J. C. (2009). Computationally efficient nonparametric importance sampling. Journal of the American Statistical Association, 104(486):788–802.

Pan, Q., Byon, E., Ko, Y. M., and Lam, H. (2020). Adaptive importance sampling for extreme quantile estimation with stochastic black box computer models. Naval Research Logistics, 67(7):524–547.

Pettersson, P. and Krumscheid, S. (2022). Adaptive stratified sampling for nonsmooth problems. International Journal for Uncertainty Quantification, 12(6).

Shields, M. D. (2016). Refined latinized stratified sampling: A robust sequential sample size extension methodology for high-dimensional latin hypercube and stratified designs. International Journal for Uncertainty Quantification, 6(1):79–97.

Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling. Technometrics, 29(2):143–151.

Tang, B. (1993). Orthogonal array-based latin hypercubes. Journal of the American Statistical Association, 88(424):1392–1397.

Tipton, E. (2013). Stratified sampling using cluster analysis: A sample selection strategy for improved generalizations from experiments. Evaluation review, 37(2):109–139.

Wycoff, N., Binois, M., and Gramacy, R. B. (2022). Sensitivity prewarping for local surrogate modeling. Technometrics, 64(4):535–547.