*Proceedings of the 2024 Winter Simulation Conference*
*H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.*

# IMPORTANCE SAMPLING IN OPTIMIZATION UNDER UNCERTAINTY USING SURROGATE MODELS

Xiaotie Chen[1], and David L. Woodruff[2]

[1]Dept. of Mathematics, University of California, Davis, CA, USA
[2]Graduate School of Management, University of California, Davis, CA, USA

## ABSTRACT

For the purpose of computing the expected value of a stochastic optimization problem via simulation, we propose a method for efficiently constructing importance sampling distributions using surrogate modeling. A software implementation of the methods called SMAIS is available on github. We use this software in experiments to demonstrate that our method can outperform Monte Carlo simulation. We also show good parallel efficiency for up to 16 processors allowing a speed up of more than 10. Our method uses adaptive sample sizes so it is not very sensitive to sample size parameters.

## 1 INTRODUCTION

Stochastic programming models can serve as indispensable tools for decision-making in uncertain environments. The effective evaluation of the recourse function for a given candidate solution $\widehat{x}$ via simulation is critical in optimizing stochastic programming problems, as seen in methodologies like stochastic dual dynamic programming (SDDP) (Pereira and Pinto 1991) and Benders of the L-shape method (Van Slyke and Wets 1969). It is also needed for computing upper bounds as in Knueven et al. (2023).

Consider an abstract statement of a stochastic programming problem

$$z^* = \min_x E_{p(\xi)}[g(x,\xi)],$$

where constraints are implicitly incorporated into the function values, and the function $g$ can encompass complex structures accounting for future states. Given a candidate solution $\widehat{x}$, our objective is to determine the expected function value:

$$\widehat{z} = E_{p(\xi)} g(\widehat{x},\xi). \tag{1}$$

Evaluating this expectation can be computationally challenging and time-consuming, often involving nested optimization problems within an integral. Moreover, obtaining exact solutions for these integrals is frequently intractable in practical settings. In such situations, alternative methods, such as Monte Carlo simulation, come to play.

Monte Carlo simulation offers a practical method for approximating the recourse function when exact solutions are elusive. It utilizes random sampling to provide estimates that converge to the true value as the number of samples increases. A basic Monte Carlo method involves taking a set of $N$ samples of the random variable $\xi_1,\ldots,\xi_N$ and approximating the integral $\widehat{z}$ with: $\widehat{z}^{MC} = \frac{1}{N}\sum g(\widehat{x},\xi_i)$. However, challenges arise when the function $g$ exhibits high variance. In such cases, the basic Monte Carlo method can require a large number of samples to achieve the desired level of accuracy, with each sample corresponding to a computationally expensive optimization problem.

Rather than relying solely on the basic Monte Carlo method, an alternative approach known as importance sampling can be employed. This method involves a change of variables and the adoption of a different probability distribution, referred to as the importance sampling distribution and denoted as $q(\xi)$. This way, the function value $\widehat{z}$ in (1) can be expressed as: $\widehat{z} = E_{q(\xi)}\frac{g(\widehat{x},\xi)p(\xi)}{q(\xi)}$. With a carefully chosen probability distribution function $q(\xi)$, we can generate samples from this distribution and construct our estimate as follows:

$$\widehat{z}^{IMC} = \frac{1}{N}\sum_{i} \frac{g(\widehat{x},\xi_i)p(\xi_i)}{q(\xi_i)}. \tag{2}$$

The selection of an appropriate $q(\xi)$ can significantly reduce the variance of the estimator. It has been demonstrated that the optimal importance sampling distribution takes the form:

$$q^*(\xi) = \frac{|g(\widehat{x},\xi)|p(\xi)}{E_{p(\xi)}|g(\widehat{x},\xi)|}. \tag{3}$$

In cases where the function $g$ is non-negative, it is well-known that $q^*(\xi) = \frac{g(\widehat{x},\xi)p(\xi)}{\widehat{z}}$, and $q^*(\xi)$ becomes a zero-variance estimator, meaning that a perfect estimate can be obtained with just a single observation. However, it is essential to note that directly obtaining the optimal $q^*(\xi)$ is not feasible without access to an exact value of $\widehat{z}$, which is precisely what we aim to evaluate in the first place.

In this paper, we propose a method for efficiently constructing importance sampling distributions using surrogate modeling. We then utilize these specially designed distributions to estimate the function value $\widehat{z}$. A software implementation of the methods called SMAIS is available on GitHub (Chen and Woodruff 2024). Our approach aims to address these key considerations:

- Efficient use of Function Evaluations: Our methods are designed to minimize the wasted evaluation of the function $g(\widehat{x},\xi)$. Since these evaluations often require the use of optimization algorithms and can be computationally expensive, minimizing their computation is crucial for efficient estimation.
- Parallelism: We aim to exploit parallelism to achieve reductions in run-time.

Our ideas are inspired by techniques in surrogate optimization, see Garud et al. (2017) for an overview. Surrogate optimization deals with situations where function evaluations can be even more computationally demanding compared to our context. Additionally, our work has connections to response surface methodology, where function evaluations sometimes involve physical experiments, but not always.

A response surface methodology (RSM) was proposed in Bailey et al. (1999) for sensitivity analysis of two-stage stochastic programming problems. In particular, the RSM enables efficient identification of senstivity to changes in first-stage variables. The paper uses a quadratic response surface and reports good results. Latin hypercube sampling is advocated as a variance reduction technique.

## 2 BACKGROUND

The Monte Carlo method has long been used in stochastic programming to approximate the function values (Birge and Louveaux 2011; Homem-de Mello and Bayraksan 2014), with most of them focusing mainly on the basic Monte Carlo method (Kleywegt et al. 2002; Shapiro 1991). The literature on importance sampling in the area of stochastic programming starts with series of papers (Dantzig and Glynn 1990; Infanger 1992; Dantzig and Infanger 1993) that introduced an importance sampling scheme, based on strong assumptions. They assumed that the cost surface can be approximated using an additive, separable model that considers marginal costs in each dimension alongside a base cost. Under these assumptions, they constructed importance sampling distributions for each dimension and aggregated the estimates from these dimensions to derive the final approximation.

Parpas et al. (2015) proposed a method that does not rely on the assumptions of additive models. They directly generated samples from the optimal distribution $q^*(\xi)$ in Equation (3) using a Markov Chain Monte Carlo (MCMC) algorithm. Subsequently, they employed these samples to "recover" or reconstruct an approximation of $q^*(\xi)$ using a kernel density estimation (KDE) algorithm, resulting in $\widehat{q}_M$. However, the limitation of the MCMC-IS method (Parpas et al. 2015) is that it often requires a large number of samples generated by the MCMC algorithm to obtain a good approximation of $q^*$. Furthermore, many of the generated samples are discarded, which can be computationally wasteful.

Apart from importance sampling, other techniques for variance reduction techniques has also been applied in the area of stochastic programming, including Antithetic variates, Quasi-Monte Carlo method, and Control Variates (Higle 1998; Koivu 2005; Drew and Homem-de Mello 2006).

The idea of combining surrogate model with importance sampling is previously explored in the area of engineering design optimization, mostly focused on analyzing the failure rate (Yao et al. 2013; Peherstorfer et al. 2016; Song et al. 2021). However, the methods developed for failure rate estimation do not directly translate to stochastic programming, as they primarily focus on generating a distribution for the failure region, while in stochastic programming, the entire region needs to be considered for optimization.

## 3 ADAPTIVE IMPORTANCE SAMPLING WITH SURROGATE MODELING

The core concept of our methodology revolves around the efficient construction of an effective importance sampling distribution, denoted as $q(\xi)$. This function aims to reduce the variance of the Monte Carlo estimator $\widehat{z}^{IMC}$, which in turn allows us to efficiently approximate the function value $\widehat{z}$. Once we have established this constructed $q(\xi)$, we can utilize Equation (2) to obtain a reliable approximation for $\widehat{z}$. The optimal importance sampling distribution is in the form of $q^*$ as defined in Equation (3). However, the construction of the exact optimal $q^*$ requires the knowledge of the value of $E_{p(\xi)}|g(\widehat{x},\xi)|$, which is closely related to what we aim to estimate in the first place. This creates a computational paradox, rendering the direct construction of the optimal $q^*$ infeasible.

To circumvent this limitation, we leverage Surrogate Modeling techniques. Surrogate models are simplified and computationally efficient approximation that mimic the behavior of the original complex and often expensive-to-evaluate models (the function $g$ in our context). They are commonly used in experiments and simulations in engineering design to speed up the decision-making process and reduce the computational burden. Approximation techniques include polynomial response surfaces (PRSMs), radial basis functions (RBFs), kriging (KRG), artificial neural networks (ANNs), and moving least squares (MLS). For a comprehensive exploration of Surrogate Modeling techniques, readers are encouraged to refer to Garud et al. (2017).

To effectively construct an importance sampling distribution that approximate the optimal sampling function $q^*$, we generate $M$ training samples, $\{\xi_i\}_{i=1}^M$, using stratified sampling (e.g. Latin Hypercube Sampling) to create a surrogate model, referred to as $s$, and use the surrogate model $s$ in place of the function $g$ for constructing an importance sampling distribution $q_s$, such that

$$q_s(\xi) = \frac{|s(\xi)|p(\xi)}{\int_\Omega |s(\xi)|p(\xi)}.$$

Notably, $q_s$ exhibits structural similarities to $q^*$ in Equation (3). In essence, $s(\xi)$ acts as a substitute for $g(\widehat{x},\xi)$ in the importance sampling distribution for a fixed candidate solution $\widehat{x}$. Once the surrogate model $s$ has been generated, we can use rejection sampling, as discussed in Frisch and Hanebeck (2022), to sample $N$ points $\{\xi_1,\dots,\xi_N\}$ from the biased distribution $q_s$. The final estimate of $\widehat{z}$ is obtained with (2).

Finding a proper evaluation sample size $N$ that balances accuracy and computational efficiency can be challenging, as it depends on the quality of the surrogate model $s$, which is determined by both the quantity and quality of the initial training samples. Inadequate sampling for constructing the surrogate model $s$ may yield a suboptimal approximation of the function $g$, whereas an excessive number of samples can lead to an inefficient use of computational resources.

Instead of adhering to a fixed number of training and evaluation samples, an adaptive, multi-fidelity approach can be more beneficial. This means that during the surrogate model construction step, we increase the number of samples based on the quality of the current constructed surrogate model, until we build an acceptable one. Similarly, in the final estimation step, we may adaptively draw samples from the constructed importance sampling distribution $q_s$, until the estimation $\widehat{z}^{IMC}$ converges. The idea of adaptively sampling for surrogate modeling has continuously drawn attention in the engineering community (Mackman et al. 2013; Xiao et al. 2018; Nentwich and Engell 2019; Xu et al. 2020), where most of them are designed for specific applications and does not directly apply in the stochastic programming setting. Adaptive sampling unrelated to importance sampling has been explored in stochastic programming, by, e.g., Higle and Sen (1991) and Bayraksan and Pierre-Louis (2012).

We provide a general framework for Adaptive Importance Sampling with Surrogate Modeling (SM-AIS) in the following algorithm, and discuss the criteria for assessing the efficacy of the constructed surrogate model and for determining the appropriate stopping rule in later subsections.

1. **Initialization**:
   - Initialize the iteration index, $k = 0$
   - Generate the initial training sample set $\Xi^0 = \{\xi_i\}_{i=1}^M$ (e.g., using Latin hypercube sampling).
   - Evaluate $y_i = g(\widehat{x}, \xi_i)$ for each training sample.
2. **Construct Surrogate Model**:
   - Construct a surrogate model $s^k$ mapping $\xi_i$ to $y_i = g(\widehat{x}, \xi_i)$ for each sample in the training set $\Xi^k$. The $\{y_i\}$ values has already been obtained in previous step(s).
3. **Construct Importance Sampling Distribution $q_s^k$**:
   - Construct the current importance sampling distribution as:

$$q_s^k(\xi) = \frac{|s^k(\xi)|p(\xi)}{\frac{1}{K}\sum_j |s^k(\widetilde{\xi}_j)|p(\widetilde{\xi}_j)}. \tag{4}$$

   Here $\{\widetilde{\xi}_j\}$ are some random samples drawn from the domain of the random variable $\xi$ via regular Monte Carlo method or Quasi Monte Carlo method. The denominator in the expression of $q_s^k(\xi)$ serves as an approximation of the integral $\int_\Omega |s^k(\xi)|p(\xi)$ that determines the proper scaling of the importance sampling distribution. Although an accurately estimated integral of $\int_\Omega |s^k(\xi)|p(\xi)$ is essential, this estimation, involving only the surrogate model $s^k$, is generally not computationally intensive compared to calculations involving $g$.
4. **Assess Surrogate Model Quality**:
   - Assess the quality of the surrogate models as discussed in Section 3.1.
   - If the quality of the surrogate model satisfies the predetermined stopping criteria, proceed to Step 6: Final Estimation.
   - Otherwise, continue to the next step: Refine Surrogate Model.
5. **Refine Surrogate Model**:
   - Choose some additional training samples. Add these to the training set $\Xi^k$ to form a new training set $\Xi^{k+1}$. The choice of the additional training samples is discussed in Section 3.2.
   - Increase the iteration index: $k = k + 1$
   - Return to step 2.
6. **Final Estimation**:
   - Adaptively sample from $q_s^k$ using rejection sampling to collect evaluation samples $\{\xi_i\}$ and periodically estimate the function value $\hat{z}$ with Equation (2). Continue until convergence is achieved, as per a defined stopping criterion. stopping criteria are detailed in Section 3.3.

This method follows the principles of multi-fidelity approaches. Initially, we generate a low-fidelity estimate of the importance sampling distribution $q_s$ to expedite the computational process. Subsequently, for our final estimation (2), we leverage a high-fidelity assessment to ensure an unbiased approximation. It is known that as long as the support of the original probability $p$ is adequately covered by the support of the importance sampling distribution $q_s$, and the variance of the weighted function $\frac{g(\widehat{x},\xi)p(\xi)}{q_s(\xi)}$ is finite, using Equation (2) will yield a consistent estimate (Robert, Casella, and Casella 1999). The concept of multi-fidelity models has seen application in different areas (Narayan et al. 2014; Peherstorfer et al. 2018). More recently, Alsup and Peherstorfer (2023) discusses the theoretical aspect of multi-fidelity importance sampling.

### 3.1 Surrogate Model Assessment

Evaluating the quality of surrogate models is crucial in their development and refinement, as it directly impacts the accuracy and reliability of the estimates. We propose two approaches for assessing the quality of the surrogate models. The first approach relies on quantifying the error of the surrogate model, offering a direct measure of the model's accuracy in replicating the true responses. The second approach is predicated on estimating the variance of the final estimator, which indicates the consistency and stability of the estimator.

#### 3.1.1 Direct Error-Based Assessment

To construct an effective surrogate model, the primary goal is to ensure that its associated importance sampling distribution in (4) aligns with the optimal sampling function in (3). Given this objective, a direct and natural approach for surrogate model assessment involves evaluating the maximum absolute error, or $L^{\inf}$ norm, between the surrogate function $s$ and the true function $g$ for the given candidate solution $\widehat{x}$. More specifically, we focus on quantifying the discrepancy between $|s^k(\xi)|p(\xi)$ and $|g(\widehat{x},\xi)|p(\xi)$, so we use $\varepsilon_s = \sup_{\xi\in\Omega}\left||s^k(\xi)|p(\xi) - |g(\widehat{x},\xi)|p(\xi)\right|$, as a measure of the surrogate model's accuracy. Our selection of absolute error over relative error conforms to the principles of importance sampling. This choice permits a greater tolerance for errors in areas of lesser significance, i.e. areas with lower $q$ values, thereby focusing our computational resources and efforts on areas with high $q$ value, ensuring our surrogate model achieves a high degree of precision in important regions.

To estimate the error $\varepsilon_s$, we draw a manageable set of samples, and compute the corresponding absolute error between $|s(\xi_j)|p(\xi_j)$ and $|g(\widehat{x},\xi_j)|p(\xi_j)$ for each sampled $\xi_j$. The composition of these samples is twofold: a portion (of cardinality $M_q$) is drawn in accordance with the surrogate model's importance sampling distribution $q_s$, ensuring that the surrogate model's approximations are precise in areas it identifies as crucial. The remainder of the samples (of cardinality $M_r$) is drawn from a uniform distribution across the domain of the random variable $\xi$, which helps verify the model's overall performance and guards against over-fitting to the training samples.

The decision to refine the surrogate model $s$ further is guided by a stopping criterion centered on whether the maximum absolute error across the assessment samples exceeds a predefined threshold. This threshold may be expressed either as a fixed absolute number or, as we have found effective in our experiments, in the form of $c_\beta \times \max|g(\widehat{x},\xi_j)|p(\xi_j)$, where $c_\beta$ represents a predetermined scaling factor. Implementing a threshold in this manner simplifies the adjustment of the threshold value, aligning with our goal to develop an importance sampling distribution that emphasizes accuracy in critical regions characterized by high $q_s$ values.

#### 3.1.2 Variance-Based Assessment

An alternative method for evaluating the surrogate model focuses on examining the variance of the final estimator, $\widehat{z}^{IMC}$. This strategy is based on the understanding that a high-quality surrogate model will yield a final estimator with lower variance. Consequently, assessing this variance serves as an indirect but effective way to gauge the surrogate model $s$ accuracy and to inform its further refinement. Specifically, with the surrogate model $s$ and the associated importance sampling distribution $q_s$, the convergence rate of the final estimator $\widehat{z}^{IMC}$ in Eq. (2) is expressed as $O\left(\frac{\sigma_s}{\sqrt{N}}\right)$, where the variance $\sigma_s^2$ is determined by

$$\sigma_s^2 = E_{q_s}\left(\frac{g(\widehat{x},\xi)p(\xi)}{q_s(\xi)}\right)^2 - \left(E_{q_s}\frac{g(\widehat{x},\xi)p(\xi)}{q_s(\xi)}\right)^2. \tag{5}$$

Hence we can develop a stopping criterion for the refinement of the surrogate model $s$, predicated on the variance $\sigma_s^2$ descending below a specified threshold.

To this end, we draw modest collection of samples $\{\tilde{\xi}_i\}_{i=1}^{M_e}$ from $q_s$ (of cardinality $M_q$). The variance $\sigma_s^2$ is then estimated either by direct computation:

$$\tilde{\sigma}_s^2 = \frac{1}{M_q - 1} \sum_i \left( \frac{g(\widehat{x}, \tilde{\xi}_i) p(\tilde{\xi}_i)}{q(\tilde{\xi}_i)} - \tilde{z} \right)^2, \tag{6}$$

where $\tilde{z} = \frac{1}{M_q} \sum_i \frac{g(\widehat{x}, \tilde{\xi}_i) p(\tilde{\xi}_i)}{q(\tilde{\xi}_i)}$.

It should be emphasized that this approach may be more prone to over-fitting the training samples, making it appropriate only when a substantial initial training sample set is utilized, where there is sufficient coverage of regions considered potentially significant. In practical applications, combining variance-based evaluation with direct error-based assessment by properly setting thresholds on both the error and the variance can yield the most effective results.

### 3.2 Additional Training Sample Selection

The selection of additional training samples is inherently informed by the assessments of the surrogate model's quality. To maximize computational efficiency, our approach emphasizes the reuse of samples that have already contributed to the evaluation of the surrogate model. It's important to highlight that adding extra samples inevitably incur some computational cost for retraining the surrogate model $s$. However, we avoid the need for repeated evaluations of the function value $g$, which can be costly, for these new samples, since these evaluations are already completed during the assessment phase.

Reusing all samples from the assessment phase brings unnecessary retraining costs for the surrogate model $s$ and risks overfitting in well-performing areas. To allocate our computational resources efficiently and achieve significant accuracy improvements, we have explored two primary approaches for selecting additional training samples. The first involves selecting a fixed number or proportion of samples, identified during the assessment phase, that demonstrate the largest errors. The second method focuses on samples whose errors surpass a specific threshold in the form of $c_\beta \times \max |g(\widehat{x}, \xi_i)| p(\xi_i)$, as mentioned in Section 3.1.1. Through practical application, we have observed that prioritizing samples with errors above a predetermined threshold, thereby excluding those with minimal errors, results in a more balanced surrogate model.

### 3.3 Adaptive Sampling for Final Estimation

In the final estimation step, we employ a dynamic process for sampling from the importance sampling distribution $q_s$, and periodically estimate the function value based on the collected samples. The estimator is given in (2).

Note that this approach allows for the simultaneous construction of confidence intervals for the true function value $\widehat{z}$ around the estimate $\widehat{z}^{IMC}$ using Central Limit Theorem. These intervals are expressed as $[\widehat{z}^{IMC} - t_{1-\alpha/2}\widetilde{\sigma}_s, \widehat{z}^{IMC} + t_{1-\alpha/2}\widetilde{\sigma}_s]$, where $t_{1-\alpha/2}$ refers to the student-t distribution.

At the same time, it also facilitates the estimation of the total number of samples required to meet a specified error tolerance. Since the estimator $\widehat{z}^{IMC}$ is a sample average estimator, according to the central limit theorem, it converges to a normal distribution with mean $\widehat{z}$ and variance $\sigma_s^2/N$, at a rate of $O\left(\frac{\sigma_s}{\sqrt{N}}\right)$. Here $\sigma_s^2$ is the associated variance as defined in (5).

This way, To determine the sample size $N_{s,\varepsilon}$ necessary for achieving an error tolerance $\varepsilon$, we use the formula $N_{s,\varepsilon} = \left\lceil \left( \frac{Z_{1-\alpha/2} \cdot \sigma_s}{\varepsilon} \right)^2 \right\rceil$, where $Z_{1-\alpha/2}$ is the z-score associated with a predefined confidence level. That is, with a sample size of $N_{s,\varepsilon}$ and probability of $1-\alpha$, the error in the final estimator $\widehat{z}^{IMC}$ will be less than $\varepsilon$.

While we do not have direct access to the variance $\sigma_s$, we may derive an initial estimation of the variance, $\widetilde{\sigma}_s$, from an early subset of samples drawn from the importance sampling distribution $q_s$. This preliminary variance estimation gives us a projection of the approximate number of samples, $\widetilde{N}_{s,\varepsilon}$, to achieve our target precision.

As the adaptive sampling process progresses, each additional sample drawn incrementally refines our estimates of $\widetilde{\sigma}_s$ and $\widetilde{N}_{s,\varepsilon}$. Simultaneously, it updates the confidence intervals for the true function value, $\widehat{z}$.

### 3.4 Parallelization of the Algorithm

The main algorithm lends itself well to parallelization, which can significantly reduce overall execution time. Here is an outline of the components of the algorithm that are amenable to parallel execution:

- **Initialization:** The generation of the initial training sample set $\Xi^0 = \{\xi_i\}_{i=1}^M$ and the evaluation of their function values $g(\widehat{x}, \xi_i)$ can be easily parallelized, as the generation of each sample and the evaluation of the function value are independent of other samples.
- **Construct Surrogate Model:** The potential for parallelizing this surrogate model training process largely depends on the type of surrogate model used and its parallelization capabilities. For instance, neural networks and certain implementations of Kriging can be trained in parallel by leveraging multiple processing units to handle batches of data or to compute model parameters concurrently.
- **Construct Importance Sampling Distribution $q_s^k$:** Constructing the importance sampling distribution $q_s^k$ requires the evaluation of the value $|s^k(\tilde{\xi}_j)| * p(\tilde{\xi}_j)$ over a reasonably large set of samples, which can be processed in parallel as each calculation is independent. In our parallelization experiments, we allocated the computation of $|s^k(\tilde{\xi}_j)| \times p(\tilde{\xi}_j)$ among all accessible processors, subsequently gathering the outcomes to obtain the proper scaling of the importance sampling distribution $q_s^k$.
- **Assess Surrogate Model Quality:** The surrogate model's quality assessment involves the use of rejection sampling from $q_s^k$, a technique well-suited for parallel execution. In our experiments, each worker is tasked with examining $B_a$ samples per batch, determining whether each sample should be accepted or rejected. This workflow is maintained across all workers until a collective total of $M_q$ samples has been drawn from $q_s^k$.
- **Final Estimation:** The final estimation phase adopts a similar approach to the quality assessment step, employing parallelized rejection sampling from the importance sampling distribution, complemented by function evaluations. In our implementation, each worker is assigned the task of evaluating $B_e$ samples within a batch to decide their acceptance. For those samples that are accepted, the subsequent step involves calculating the weighted function value $\frac{g(\widehat{x},\xi_i)p(\xi_i)}{q_s(\xi_i)}$.

## 4 EXPERIMENTAL RESULTS

The candidate solutions used in our experiments are generated randomly by solving a sample average approximation of the original stochastic programming problems with a limited number of scenarios. We did not observe significant differences in estimation accuracy among the various candidate solutions. Each candidate solution $\hat{x}$ was used to evaluate the performance of our proposed adaptive importance sampling with surrogate models (SM-AIS) method in comparison with two other approaches.

The first comparative method is the regular Monte Carlo (MC) method, wherein sampling is conducted according to the original probability distribution denoted as $p$. We adopted rejection sampling with Sobol sequences to work with distribution $p$, allowing us to handle various types of distribution functions.

The second method is the Markov Chain Monte Carlo Importance Sampling (MCMC-IS) Method, as proposed by Parpas et al. (2015). This method involves generating samples from the optimal distribution $q^*(\xi)$ using MCMC and then approximating this distribution with kernel density estimation (KDE). We use normal distribution as our proposal distribution, experimenting with both constant variance and adaptive Metropolis techniques (Haario et al. 2001). Preliminary parameter tuning was conducted for the fixed variance approach, and the best-performing setup, was selected for method comparison.

In the experimental results reported below, we focus on direct error-based assessment, as it has proven effective for our tested problem. We adopt the kriging method because it leverage both the sampled data and the inherent correlation patterns between the samples. Hence, it's able to approximate nonlinear functions effectively. Other surrogate models are also available in SMAIS.

### 4.1 Problem Examples

We conducted experiments over four different example problems.

The NewsVendor problem involves finding the optimal order quantity to maximize profit or minimize costs while considering the balance between lost sales and excess inventory. In our experiment we used the two-stage stochastic program example provided in Lubin et al. (2023), where the uncertain demand follows a triangular distribution.

CVaR, or Conditional Value at Risk, is a risk measure used in finance and statistics to quantify the potential loss in the worst-case scenarios of an investment or portfolio. In our experiment we solve a $(1-\alpha)$-level CVaR problem as in Lam and Qian (2018): $\min_x \left\{x+\frac{1}{a}E\left[\max(\xi-x,0)\right]\right\}$, where $a=0.1$ and $\xi$ is a drawn from a standard normal distribution.

A scalable version of the farmer example in Birge and Louveaux (2011) is implemented in Knueven et al. (2023). For our experiments we used the original three crops, and introduced a new feature, denoted as “yield-cv," which represents the coefficient of variation of the crop yields.

The multi-knapsack problem originates from the stochastic programming problem discussed in Vaagen and Wallace (2008) and also explored in King and Wallace (2012). It can be regarded as a multidimensional newsvendor issue that includes substitution effects. In our analysis, we focus on the scenario involving the sale of six products, each subject to a uniform substitution rate of 10%.

### 4.2 Parameter Values

In the interest of reproducibility, Tabe 1 shows the parameter values used in experiments except when otherwise noted.

Table 1: Parameter values used in experiments.

| Symbol | Name in SMAIS | NewsVendor | CVaR | Farmer | Knapsack |
|---|---|---|---|---|---|
| $M$ | initial_sample_size | 20 | 80 | 40 | 80 |
| $M_q$ | assess_size | 20 | 20 | 40 | 40 |
| $M_r$ | additional_sample_size | 20 | 20 | 40 | 40 |
| $K$ | sp_integral_size | 2000 | 2000 | 10,000 | 1,000,000 |
| $c_\beta$ | adaptive_error_threshold_factor | 0.1 | 0.1 | 0.1 | 0.1 |
| $N$ | evaluation_N | 500 | 500 | 500 | 500 |

### 4.3 Experimental Results Summary Tables

For the CVaR problem, the effectiveness was assessed by measuring the deviation of the estimator for the generated function value from the benchmark and the width of the generated confidence intervals for $\widehat{z}$, across various predetermined cutoff times. The benchmark value achieved by the conventional Monte Carlo method upon convergence. The methods were implemented in serial, without parallelization. We will discuss potential performance improvements for the SM-AIS method through parallelization in a later section. Due to the necessity for an initial runtime to establish its importance sampling distribution, the MCMC Importance Sampling method does not provide estimations at early cutoff times. When the relative error is less than 0.1%, we consider it as negligibly small.

Table 2: Summary table for CVaR, with the benchmark function value $\hat{z} \approx 2.1155$ obtained using numerical integration. The MCMC-IS method run with 2500 samples for hastings method. The surrogate model construction for the SM-AIS method meets the stopping criteria within 6 iterations.

| Cut-off Time (Seconds) | MC | | MCMC-IS | | SM-AIS | |
|---|---|---|---|---|---|---|
| | Error | CI Width | Error | CI Width | Error | CI Width |
| 100 | 0.22 | 0.99 | - | - | 0.013 | 0.092 |
| 200 | 0.081 | 0.27 | 0.015 | 0.11 | negligible | 0.024 |
| 400 | 0.050 | 0.20 | 0.015 | 0.05 | negligible | 0.015 |

Table 2 demonstrates the varied effectiveness of sampling methods in addressing the CVaR problem. Due to the significant disparity between the important region for CVaR estimation and the original distribution $p$, the regular MC method can be extremely inefficient, and an effective importance sampling distribution is necessary for fast estimation of the function value.

While the MCMC-IS method's slower development of the importance sampling distribution might seem a drawback, it still surpasses the basic MC method due to the usage of the importance function. The SM-AIS method stands out for its rapid creation of an efficient importance sampling distribution, leading to minimal errors and significantly tighter confidence intervals.

For the rest of the example problems, the efficiency of the MCMC-IS method is constrained by the limited utility of the importance sampling distribution in relatively simple scenarios. The generation of the importance sampling distribution via the MCMC-IS approach is time-consuming and does not offer sufficient evaluation efficiency improvements for certain problems. This challenge is particularly evident for some multi-dimensional optimization problem, where optimally adjusting the MCMC-IS method's parameters proves difficult due to the Kernel Density Package available for use in our Python implementation. Hence for the rest of the examples, we primarily report the time complexity advantage of our approach over the regular MC method

Table 3: Summary table for NewsVendor, with the benchmark function value $\hat{z} \approx 558.6$ obtained using regular MC with $20k$ samples. The surrogate model construction for the SM-AIS method meets the stopping criteria within 2 iterations.

| Cut-off Time (Seconds) | MC | | SM-AIS | |
|---|---|---|---|---|
| | Error | CI Width | Error | CI Width |
| 20 | 2.7 | 16.7 | 0.3 | 0.4 |
| 100 | 1.3 | 8.2 | negligible | 0.1 |
| 200 | 0.6 | 5.7 | negligible | $< 0.1$ |
| 400 | 0.8 | 4.0 | negligible | $<0.1$ |

As shown in Tables 3 and 4, in the NewsVendor problem and the farmer problem, the regular Monte Carlo (MC) method achieves efficient convergence, yet the Adaptive Importance Sampling with Surrogate Models (SM-AIS) method shows faster convergence and significantly narrower confidence intervals.

For multi-dimensional problems like the multi-knapsack problem, constructing the importance sampling distribution in Step 3 requires a substantial number of samples, denoted by $K$, to accurately estimate the integral in the denominator. However, since the integral calculation solely involves evaluations of the surrogate model $s$, the process remains relatively cost-effective, despite the requirement for extensive sampling. Table 5 suggests that our method effectively reduces the variance of the final estimator, resulting in a more precise confidence interval. In Section 4.4, we demonstrate how incorporating parallel computing strategies can markedly decrease the computational time necessitated by high-dimensional integration.

Table 4: Summary table for Farmer, with the actual function value $\widehat{z} \approx 151,600$ obtained with $65k$ samples. The surrogate model construction for the SM-AIS method meets the stopping criteria within 2 iterations.

| Cut-off Time (Seconds) | MC | | SM-AIS | |
|---|---|---|---|---|
| | Error | CI Width | Error | CI Width |
| 40 | 2126 | 4,125 | 87 | 418 |
| 200 | 575 | 1995 | negligible | 164 |
| 400 | 347 | 1397 | negligible | 117 |
| 800 | 104 | 1018 | negligible | 83 |

Table 5: Summary table for Multi-Knapsack, with the actual function value $\widehat{z} \approx 24,860$ obtained with $20k$ samples. The surrogate model construction for the SM-AIS method meets the stopping criteria within 2 iterations.

| Cut-off Time (Seconds) | MC | | SM-AIS | |
|---|---|---|---|---|
| | Error | CI Width | Error | CI Width |
| 500 | 220 | 1,198 | 117 | 433 |
| 1000 | 53 | 862 | 41 | 121 |
| 1500 | 37 | 713 | negligible | 95 |

### 4.4 Parallel Speed Up

The SM-AIS method offers the advantage of straightforward parallelization, leading to considerable performance enhancements. To demonstrate, we present a comparison between the parallelized and non-parallelized implementations of the SM-AIS method when applied to the multi-knapsack problem. For this purpose, we employ a fixed set of parameters and evaluate the run-time performance of the SM-AIS method across a variety of process counts.

Our implementation of parallelization was focused on three critical phases of SM-AIS: In Step 3, parallel processing was employed to evaluate the denominator when constructing the importance sampling distribution $q_s^k$; Step 4 leveraged parallelism for performing rejection sampling for evaluating the surrogate model's quality through error-based assessment; and in Step 6, parallel processes were utilized once more for drawing samples from $q_s^k$ for the final estimation. Further discussion of the parallel implementation can be found in Section 3.4. In our experiments, the batch sizes were set to $B_a = 200$ for the quality assessment phase and $B_e = 100$ for the final estimation phase.

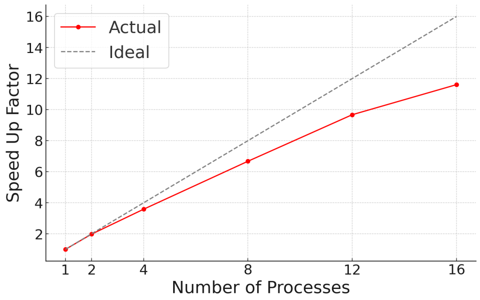

Figure 1: The speed up factor for SM-AIS with increasing parallel processes for multi-knapsack example. The red line indicates the actual speed up of the algorithm as the number of processes increases, while the gray dashed line represents the ideal linear speed up.

Figure 1 shows the scaling efficiency by comparing the speed-up factor with the number of processors, illustrating the effectiveness of parallelization in reducing runtime. With up to 16 processors, it achieves a speed-up factor greater than 10.

## 5 CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

To conduct simulations for computing the expected value of a stochastic optimization problem, we propose a method for efficiently constructing importance sampling distributions using surrogate modeling. The method is shown to offer a significant improvement over Monte Carlo sampling even when augmented by importance sampling because the surrogate model allows for much faster evaluation than is required to solve an optimization problem. Our methods strive to reduce the need for evaluation of the function $g(\widehat{x},\xi)$ because these evaluations often require the use of optimization algorithms and can be computationally expensive. We demonstrate good parallel efficiency up to 16 processors.

Our software implementation supports multiple surrogate model types. It remains as future research to invent new surrogate models for this application. A related area that could be explored in future research is to incorporate prior knowledge in constructing the surrogate. The parallel efficiency is starting to degrade at 16 processors, so improvements in that area would be a good avenue for exploration. An implementation of the method is available on github to researchers who want consider improvements or who want to test it against new methods.

## REFERENCES

Alsup, T. and B. Peherstorfer. 2023. “Context-Aware Surrogate Modeling for Balancing Approximation and Sampling Costs in Multifidelity Importance Sampling and Bayesian Inverse Problems”. *SIAM/ASA Journal on Uncertainty Quantification* 11(1):285–319.

Bailey, T. G., P. A. Jensen, and D. P. Morton. 1999. “Response surface analysis of two-stage stochastic linear programming with recourse”. *Naval Research Logistics (NRL)* 46(7):753–776.

Bayraksan, G. and P. Pierre-Louis. 2012. “Fixed-width sequential stopping rules for a class of stochastic programs”. *SIAM Journal on Optimization* 22(4):1518–1548.

Birge, J. R. and F. Louveaux. 2011. *Introduction to stochastic programming*. Springer Science & Business Media.

Chen, Xiaotie and Woodruff, David L 2024. “SMAIS: Adaptive Importance Sampling with Surrogate Modeling”. https://github.com/DLWoodruff/SMAIS.git, accessed 18th August 2024.

Dantzig, G. B. and P. W. Glynn. 1990. “Parallel processors for planning under uncertainty”. *Annals of Operations research* 22(1):1–21.

Dantzig, G. B. and G. Infanger. 1993. “Multi-stage stochastic linear programs for portfolio optimization”. *Annals of Operations Research* 45(1):59–76.

Drew, S. S. and T. Homem-de Mello. 2006. “Quas-monte carlo strategies for stochastic optimization”. In *Proceedings of the 2006 winter simulation conference*, 774–782. IEEE.

Frisch, D. and U. D. Hanebeck. 2022. “Rejection Sampling from Arbitrary Multivariate Distributions Using Generalized Fibonacci Lattices”. In *2022 25th International Conference on Information Fusion (FUSION)*, 1–7. IEEE.

Garud, S. S., I. Karimi, and M. Kraft. 2017. “Smart Sampling Algorithm for Surrogate Model Development”. *Computers & Chemical Engineering* 96:103–114.

Haario, H., E. Saksman, and J. Tamminen. 2001. “An adaptive Metropolis algorithm”. *Bernoulli*:223–242.

Higle, J. L. 1998. “Variance reduction and objective function evaluation in stochastic linear programs”. *INFORMS Journal on Computing* 10(2):236–247.

Higle, J. L. and S. Sen. 1991. “Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse”. *Mathematics of Operations Research* 16(3):650–669.

Homem-de Mello, T. and G. Bayraksan. 2014. “Monte Carlo sampling-based methods for stochastic optimization”. *Surveys in Operations Research and Management Science* 19(1):56–85.

Infanger, G. 1992. “Monte Carlo (importance) sampling within a Benders decomposition algorithm for stochastic linear programs”. *Annals of Operations Research* 39(1):69–95.

King, A. J. and S. W. Wallace. 2012. *Modeling with stochastic programming*. Springer.

Kleywegt, A. J., A. Shapiro, and T. Homem-de Mello. 2002. “The sample average approximation method for stochastic discrete optimization”. *SIAM Journal on optimization* 12(2):479–502.

Knueven, B., D. Mildebrath, C. Muir, J. D. Siirola, J.-P. Watson and D. L. Woodruff. 2023. “A Parallel Hub-and-Spoke System for Large-Scale Scenario-Based Optimization Under Uncertainty”. *Math. Prog. Comp.* 15:591—619.

Koivu, M. 2005. “Variance reduction in sample approximations of stochastic programs”. *Mathematical programming* 103:463–485.

Lam, H. and H. Qian. 2018. “Assessing solution quality in stochastic optimization via bootstrap aggregating”. In *2018 Winter Simulation Conference (WSC)*, 2061–2071. IEEE https://doi.org/10.1109/WSC.2018.8632334.

Lubin, M., O. Dowson, J. Dias Garcia, J. Huchette, B. Legat and J. P. Vielma. 2023. “JuMP 1.0: Recent improvements to a modeling language for mathematical optimization”. *Mathematical Programming Computation*.

Mackman, T. J., C. B. Allen, M. Ghoreyshi, and K. Badcock. 2013. “Comparison of adaptive sampling methods for generation of surrogate aerodynamic models”. *AIAA journal* 51(4):797–808.

Narayan, A., C. Gittelson, and D. Xiu. 2014. “A stochastic collocation algorithm with multifidelity models”. *SIAM Journal on Scientific Computing* 36(2):A495–A521.

Nentwich, C. and S. Engell. 2019. “Surrogate modeling of phase equilibrium calculations using adaptive sampling”. *Computers & Chemical Engineering* 126:204–217.

Parpas, P., B. Ustun, M. Webster, and Q. K. Tran. 2015. “Importance sampling in stochastic programming: A Markov chain Monte Carlo approach”. *INFORMS Journal on Computing* 27(2):358–377.

Peherstorfer, B., T. Cui, Y. Marzouk, and K. Willcox. 2016. “Multifidelity importance sampling”. *Computer Methods in Applied Mechanics and Engineering* 300:490–509.

Peherstorfer, B., K. Willcox, and M. Gunzburger. 2018. “Survey of multifidelity methods in uncertainty propagation, inference, and optimization”. *Siam Review* 60(3):550–591.

Pereira, M. V. and L. M. Pinto. 1991. “Multi-stage stochastic optimization applied to energy planning”. *Mathematical programming* 52:359–375.

Robert, C. P., G. Casella, and G. Casella. 1999. *Monte Carlo statistical methods*, Volume 2. Springer.

Shapiro, A. 1991. “Asymptotic analysis of stochastic programs”. *Annals of Operations Research* 30:169–186.

Song, K., Y. Zhang, X. Zhuang, X. Yu and B. Song. 2021. “Reliability-based design optimization using adaptive surrogate model and importance sampling-based modified SORA method”. *Engineering with Computers* 37:1295–1314.

Vaagen, H. and S. W. Wallace. 2008. “Product variety arising from hedging in the fashion supply chains”. *International Journal of Production Economics* 114(2):431–455.

Van Slyke, R. M. and R. Wets. 1969. “L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming”. *SIAM Journal on Applied Mathematics* 17(4):638–663.

Xiao, N.-C., M. J. Zuo, and C. Zhou. 2018. “A new adaptive sequential sampling method to construct surrogate models for efficient reliability analysis”. *Reliability Engineering & System Safety* 169:330–338.

Xu, H., L. Liu, and M. Zhang. 2020. “Adaptive surrogate model-based optimization framework applied to battery pack design”. *Materials & Design* 195:108938.

Yao, J., Z. Ye, and Y. Wang. 2013. “Efficient importance sampling for high-sigma yield analysis with adaptive online surrogate modeling”. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1291–1296. IEEE.

## AUTHOR BIOGRAPHIES

**XIAOTIE CHEN** is a Ph.D. student in the Department of Mathematics at University of California, Davis. Her research interest lies in uncertainty quantification, Monte Carlo methods, and stochastic optimization. Her email address is xtjchen@ucdavis.edu.

**DAVID L. WOODRUFF** is a Professor of Management at the University of California Davis. His research interests include optimization under uncertainty and software for modeling and solving optimization problems. His email address is DLWoodruff@UCDavis.edu.