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.

PLAUSIBLE INFERENCE WITH A PLAUSIBLE LIPSCHITZ CONSTANT

Gregory Keslin[1], Daniel W. Apley[1], and Barry L. Nelson[1]

[1]Dept. of Industrial Engineering & Management Sciences, Northwestern University, Evanston, IL, USA

ABSTRACT

Plausible inference is a growing body of literature that treats stochastic simulation as a gray box when structural properties of the simulation output performance measures as a function of design, decision or contextual variables are known. Plausible inference exploits these properties to allow the outputs from values of decision variables that *have* been simulated to provide inference about output performance measures at values of decision variables that *have not* been simulated; statements about the possible optimality or feasibility are examples. Lipschitz continuity is a structural property of many simulation problems. Unfortunately, the all-important—and essential for plausible inference—Lipschitz constant is rarely known. In this paper we show how to obtain plausible inference with an estimated Lipschitz constant that is also derived by plausible inference reasoning, as well as how to create the experiment design to simulate.

1 INTRODUCTION

Stochastic simulation is used to explore and compare the performance of complex systems, ideally in a cost-effective and efficient manner. These comparisons can be thought of as determining the set of system configurations that have *acceptable* performance. The two forms of "acceptable" that we consider here are feasible and optimal.

We assume that each system (also called decision, solution, scenario or configuration) is specified by a vector of decision variables, $\boldsymbol{x} \in \mathscr{X} \subset \Re^q$. For any setting of the decision variables, independent and identically distributed (i.i.d.) replications of the primary simulation output, $Y_j(\boldsymbol{x}), j = 1,2,\ldots$, can be generated. The performance measure of interest is the expectation $\mu(\boldsymbol{x}) = \mathbb{E}(Y_j(\boldsymbol{x}))$. We assume that larger performance is better. In feasibility checking, the goal is to find the subset of systems whose expected performance is above some prescribed threshold. Optimality checking seeks to find the system or systems that attain the maximum performance in $\mathscr{X}$.

Because we observe $Y_j(\boldsymbol{x})$ rather than $\mu(\boldsymbol{x})$, it is not possible to determine the acceptable systems with certainty, even if every alternative is simulated. Instead, the goal is to employ simulation to indicate with high confidence whether a system is acceptable or not. If, with high confidence, the system fails to satisfy the acceptability goal then it is excluded from the set of "plausibly acceptable" systems; we call this process *screening*.

Of course the set $\mathscr{X}$ of possible decision variable values may be very large or even infinite. If there is no known structural information to link performance measures from simulated systems to unsimulated ones, then every configuration must be simulated. However, if the computational cost of simulation is high, or there are an infinite number of possible systems, then simulating every one is impractical or impossible. To infer $\mu(\boldsymbol{x})$ at unsimulated systems we exploit structural information about $\mu(\boldsymbol{x})$ as a function of $\boldsymbol{x}$, specifically that it is Lipschitz continuous. Exploiting Lipschitz continuity in this way is not a new idea (see Section 2), but we further assume that the value of the Lipschitz constant, λ, is not known. One of the contributions of this paper is the derivation of an estimator of the Lipschitz constant that is compatible with plausible inference.

Here is a summary of the plausible inference approach: By restricting the set of performance functions $\mu(\cdot)$ to Lipschitz continuous functions with the inferred Lipschitz constant, a hypothesis test on the

performance function at unsimulated values of $\boldsymbol{x}$ given the simulated ones is established. For each $\boldsymbol{x}$, given the data, if there exists a performance function that satisfies the performance constraint for $\boldsymbol{x}$ and is not rejected by the hypothesis test, then $\boldsymbol{x}$ is considered an acceptable system. However, if all performance functions that satisfy the performance constraint for $\boldsymbol{x}$ are rejected by the hypothesis test, then $\boldsymbol{x}$ is screened out. Notice that both simulated and, importantly, unsimulated systems can be screened out. Further, every system is screened with a confidence guarantee on the set of retained systems. This means that the set of screened out systems can be rejected with a user specified confidence level. All of this assumes that λ is known. When it is not—which is our case—we provide a plausible estimator of it that improves as more systems are simulated.

We organize the remainder of the paper as follows: Section 2 is a brief overview of existing work on plausible screening and Lipschitz optimization methods. Section 3 gives a mathematical development of the screening procedure described above. Section 4 derives our estimator of λ that is then employed for screening in the sequel. Section 5 presents methods for sequentially choosing which systems ("design points") to simulate, our second major contribution. By sequentially adding new design points, rather than employing a fixed design, the screening power can be increased. Two numerical illustrations are summarized in Section 6.

2 LITERATURE REVIEW

Screening of systems without exhaustive simulation at all configurations is a nascent field. Plumlee and Nelson (2018) first established the underlying method that is used in this paper. Their work was focused on cases where the performance functions $\mu(\cdot)$ are convex or Lipschitz and acceptable performance is optimality. Eckman et al. (2020) extended these results by considering a wider range of metrics on which to screen. Eckman et al. (2022) created a general framework that allows for a greater variety of functional information than convexity or Lipschitz continuity. Further, they developed optimization formulations to reduce many screening problems to the solution of linear programs. Their framework provides the basis for our screening algorithms described in later sections. Eckman et al. (2021) used gradient estimates to enhance screening under the assumption that the performance functions are convex. Altogether the methodology described above is called "plausible inference."

Importantly, the work on plausible screening has assumed a structure that is fully known, which in the Lipschitz case means that λ is available. While Lipschitz continuity itself is a relatively benign assumption that is applicable to nearly all simulation problems, it is rare to know the Lipschitz constant. In this paper, we propose a method for estimating the Lipschitz constant that is compatible with plausible screening.

The field of global optimization has exploited Lipschitz continuity to aid in the search for optimal solutions when objective functions are evaluated without noise. The work in this area can be partitioned by whether the Lipschitz constant is known or must be estimated. Shubert (1972) and Piyavskii (1972) both proposed optimization algorithms when λ is known. Since then, a significant literature has been dedicated to new algorithms and their properties in the known-Lipschitz-constant case. See Hansen and Jaumard (1995) for a broad overview. When λ is not known, Wood and Zhang (1996) derived an asymptotically valid distribution of the maximum slope for sampled points. Malherbe and Vayatis (2017) provided an algorithm that sequentially updates an estimate of λ. Fazlyab et al. (2019), among others, provided methods for estimating λ tailored to neural networks. These methods assume noiseless observations which are not applicable in our context of stochastic simulation.

3 PROBLEM OVERVIEW

Because our work builds on the framework established by Eckman et al. (2022), we briefly review their methods applied to the feasibility and optimization definitions of acceptability.

Recall that the simulated systems are indexed by $\boldsymbol{x} \in \mathscr{X}$. The functions $\mu(\boldsymbol{x})$ and $\sigma^2(\boldsymbol{x})$ are the mean and variance, respectively of the output replications $Y_j(\boldsymbol{x})$. The space, $\mathscr{X}$, can be continuous or

discrete. In the optimization screening problem, we define the set of acceptable systems to be $\mathscr{A} = \{\boldsymbol{x} \in \mathscr{X} | \mu(\boldsymbol{x}) = \max_{\boldsymbol{x}' \in \mathscr{X}} \mu(\boldsymbol{x}')\}$. In the feasibility screening problem, we define the set of acceptable systems to be $\mathscr{A} = \{\boldsymbol{x} \in \mathscr{X} | \mu(\boldsymbol{x}) \geq c\}$, for some specified constant c. Consistent with Eckman et al. (2022), we assume that $Y_j(\boldsymbol{x}) \sim N(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$, and outputs are independent across replications and across $\boldsymbol{x} \neq \boldsymbol{x}'$. While we assume normality, the theorems in the paper only require that the sample means of the replications generated from each $\boldsymbol{x}$ to be approximately normal. This is often satisfied due to the Central Limit Theorem for non-normal $Y_j(\boldsymbol{x})$.

Consider the feasibility case. Let M_λ be the set of all Lipschitz continuous functions with Lipschitz constant λ and let $\lambda^\star$ denote the true Lipschitz constant of $\mu(\cdot)$. If $\lambda^\star$ is known then we can assess whether any $\boldsymbol{x} \in \mathscr{X}$ is acceptable by only considering the acceptability of $\boldsymbol{x}$ for functions in $M_{\lambda^\star}$. Define $G_{\lambda^\star}(\boldsymbol{x}) = \{\phi \in M_{\lambda^\star} | \phi(\boldsymbol{x}) \geq c\}$ to be the set of candidate functions for $\mu(\cdot)$ in which $\boldsymbol{x}$ is acceptable.

To determine the acceptability of $\boldsymbol{x}$, simulations are performed at a collection of decision-variable values, $\mathscr{D}_k = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$, which we call the experiment design or design points. The elements of $\mathscr{D}_k$ may or may not be randomly chosen from $\mathscr{X}$. Then for each $\boldsymbol{x}_i \in \mathscr{D}_k$, $n_k(\boldsymbol{x}_i)$ replications $Y_j(\boldsymbol{x}_i), j = 1, 2, \ldots, n_k(\boldsymbol{x}_i)$ are generated. For any $\phi \in M_{\lambda^\star}$, if the discrepancy between the output values and the values of ϕ at the design points is sufficiently large, then the function ϕ is considered "implausible". If all functions $\phi \in G_{\lambda^\star}(\boldsymbol{x})$ are considered implausible, then it is unlikely that $\boldsymbol{x}$ is acceptable and it is screened out.

The choice of discrepancy measure and cutoff for when a function is considered implausible is further explored in Eckman et al. (2022). For simplicity, we choose the discrepancy measure to be the standardized absolute value of the differences between ϕ and the sample means of the replications. Specifcally, let $\bar{\mathbf{Y}} = \{\bar{Y}(\boldsymbol{x}_1), \ldots, \bar{Y}(\boldsymbol{x}_k)\}$ be the set of sample means at the design points and let $\widehat{\sigma} = \{\widehat{\sigma}(\boldsymbol{x}_1), \ldots, \widehat{\sigma}(\boldsymbol{x}_k)\}$ be the corresponding set of sample standard deviations. We define the standardized discrepancy metric as

$$d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) = \sum_{i=1}^{k} \frac{\sqrt{n_k(\boldsymbol{x}_i)}\,|\phi(\boldsymbol{x}_i) - \bar{Y}(\boldsymbol{x}_i)|}{\widehat{\sigma}(\boldsymbol{x}_i)}.$$

For an appropriately chosen discrepancy cutoff, $D_{1-\alpha}$, any $\boldsymbol{x} \in \mathscr{X}$ is classified as acceptable only if

$$\inf_{\phi \in G_{\lambda^\star}(\boldsymbol{x})} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) \leq D_{1-\alpha}. \tag{1}$$

Controlling the probability of incorrectly classifying $\boldsymbol{x}$ as unacceptable if $\boldsymbol{x} \in \mathscr{A}$ requires choosing $D_{1-\alpha}$ such that

$$\sup_{\boldsymbol{x} \in \mathscr{A}} \mathbb{P}\left(\inf_{\phi \in G_{\lambda^\star}(\boldsymbol{x})} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) > D_{1-\alpha} \right) \leq \alpha.$$

Evaluating the probability on the left-hand side of the inequality above requires knowledge of μ. However, if $Y_j(\boldsymbol{x}) \sim N(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$, then for the true mean vector μ, $d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma})$ follows a known distribution: the sum of the absolute value of k independent t distributions with respective degrees of freedom $n_k(\boldsymbol{x}_i) - 1$. Therefore, if $D_{1-\alpha}$ is the $1-\alpha$ quantile of $d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma})$, then

$$\begin{aligned} \sup_{\boldsymbol{x} \in \mathscr{A}} \mathbb{P}\left(\inf_{\phi \in G_{\lambda^\star}(\boldsymbol{x})} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) > D_{1-\alpha} \right) &\leq \sup_{\boldsymbol{x} \in \mathscr{A}} \mathbb{P}\left(d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma}) > D_{1-\alpha} \right) \\ &= \mathbb{P}\left(d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma}) > D_{1-\alpha} \right) = \alpha. \end{aligned}$$

Thus, the screening procedure defined above controls the pointwise Type I error rate of classifying any acceptable $\boldsymbol{x}$ as unacceptable. Of course, evaluation of the infimum in Equation (1) requires knowledge of $\lambda^\star$. As discussed in the introduction, it is typically easier to use the structural information of the problem to determine whether $\mu(\cdot)$ is Lipschitz continuous than it is to determine the Lipschitz constant of $\mu(\cdot)$. The estimation of $\lambda^\star$ is discussed in the next section.

Remark: There is a corresponding development for the optimality screening case, and we can employ any of the discrepancy measures in Eckman et al. (2022), including those that allow simulation with common random numbers.

4 ESTIMATION OF $\lambda^\star$

Methods for estimating $\lambda^\star$ typically assume that $\mu(\boldsymbol{x})$ can be evaluated without noise for any $\boldsymbol{x} \in \mathscr{X}$. Such methods often use the maximum slope of the observed values of any pair of design points, $\max_{i\neq j}(\mu(\boldsymbol{x}_i) - \mu(\boldsymbol{x}_j))/s(\boldsymbol{x}_i,\boldsymbol{x}_j)$, where $s(\cdot,\cdot)$ is the distance metric. For example, Wood and Zhang (1996) derive the asymptotic distribution of the maximum slope to provide an upper confidence bound on $\lambda^\star$. In this section, we propose an estimator that is applicable to the case when $\mu(\cdot)$ is observed with and without noise. When there is no noise, our proposed estimator reduces to this maximal slope estimator.

When only noisy estimates of $\mu(\boldsymbol{x})$ are available, estimating the Lipschitz constant becomes a more difficult task. A naïve method is to use $\max_{i\neq j}(\bar{Y}(\boldsymbol{x}_i) - \bar{Y}(\boldsymbol{x}_j))/s(\boldsymbol{x}_i,\boldsymbol{x}_j)$ as the estimator. However, as more design points are chosen, $\lim_{k\to\infty} \min_{i\neq j} s(\boldsymbol{x}_i,\boldsymbol{x}_j) \to 0$; this leads to a badly behaved and inconsistent estimator whose variance increases to infinity as $k \to \infty$.

Huang et al. (2023) propose a consistent estimator that separates the decision variable space into a number of disjoint hypercubes. For each hypercube, a linear regression is run to "approximate" the Lipschitz constant over the hypercube. The maximum slope among the linear regressions is used as the Lipschitz constant. This method is clearly cumbersome; we propose a different method that exploits the plausible inference framework.

For a given value of λ, recall that M_λ is the set of Lipschitz continuous functions with Lipschitz constant λ. To assess the feasibility of any λ, we measure how well the observed data adheres to M_λ. Specifically, for λ to be feasible, we require that $\inf_{\phi\in M_\lambda} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}^2)$ be sufficiently small. Let $\beta \in [0,1]$ and define $\widehat{\lambda}_{1-\beta}$, the $1-\beta$ plausible Lipschitz estimator as,

$$\widehat{\lambda}_{1-\beta} = \inf_{\lambda} \text{ s.t } \inf_{\phi\in M_\lambda} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}^2) \leq D_{1-\beta}.$$

In words, $\widehat{\lambda}_{1-\beta}$, is the smallest λ for which there exists a λ-Lipschitz continuous function having discrepancy with respect to the observed simulation data below the specified quantile.

Theorem 1 proves that, under some mild assumptions, $\widehat{\lambda}_{1-\beta}$ is a $(1-\beta)100\%$ lower confidence bound on the true Lipschitz constant of $\mu(\cdot)$. Because $\lambda^\star$ is unknown, for $\boldsymbol{x} \in \mathscr{X}$ that are not in $\mathscr{D}_k$ the values of $\mu(\boldsymbol{x})$ at these unobserved $\boldsymbol{x}$'s can be arbitrarily large and small, and therefore $\lambda^\star$ can be arbitrarily larger than the maximum slope of the mean values of the design points. Thus, we can only obtain a *lower* confidence bound.

To prove Theorem 1, we assume the following:

Assumption 1 $\mu(\cdot)$ is a Lipschitz continuous function with constant $\lambda^\star$ and $Y_j(\boldsymbol{x}) \sim N(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$, independent across replications j and across configurations $\boldsymbol{x} \neq \boldsymbol{x}'$.

Theorem 1 Under Assumption 1, $\mathbb{P}\left(\widehat{\lambda}_{1-\beta} > \lambda^\star\right) \leq \beta$.

Proof. Because $Y_j(\boldsymbol{x}) \sim N(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$, $d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma}^2) = \sum_{i=1}^k \sqrt{n_k(\boldsymbol{x}_i)}\,|\mu(\boldsymbol{x}_i) - \bar{Y}(\boldsymbol{x}_i)|/\widehat{\sigma}(\boldsymbol{x}_i)$ is the sum of the absolute value of k independent t distributed random variables. Therefore, because $D_{1-\beta}$ is the $1-\beta$ quantile of the sum of the absolute value of k independent t random variables, $\mathbb{P}\left(d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma}^2) > D_{1-\beta}\right) \leq \beta$. Under Assumption 1, if $\widehat{\lambda}_{1-\beta} > \lambda^\star$ then $\inf_{\phi\in M^\star_\lambda} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}^2) > D_{1-\beta}$. Since $\inf_{\phi\in M^\star_\lambda} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}^2) \leq d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma}^2)$, $\mathbb{P}\left(\widehat{\lambda}_{1-\beta} > \lambda^\star\right) \leq \mathbb{P}\left(d^1(\mu, \bar{\mathbf{Y}}, \widehat{\sigma}^2) > D_{1-\beta}\right) \leq \beta$. □

Notice that if $\beta = 1$, then $D_{1-\beta} = 0$ and $\widehat{\lambda}_0 = \max_{i\neq j}(\bar{Y}(\boldsymbol{x}_i) - \bar{Y}(\boldsymbol{x}_j))/s(\boldsymbol{x}_i,\boldsymbol{x}_j)$, the naïve estimator. On the other hand, if $\beta = 0$, then $D_{1-\beta} = \infty$ and $\widehat{\lambda} = 0$. Thus, β can be used to tune the estimator.

While $\widehat{\lambda}_{1-\beta}$ is a $1-\beta$ lower confidence bound on $\lambda^\star$, we next show that under some additional assumptions it converges to $\lambda^\star$ as the number of design points $k \to \infty$.

Assumption 2 We assume the following.

(a) For all $\boldsymbol{x} \in \mathscr{X}$, $0 < \underline{\sigma} \leq \sigma(\boldsymbol{x}) \leq \overline{\sigma} < \infty$.
(b) For all k and $\boldsymbol{x} \in \mathscr{D}_k$, $n_k(\boldsymbol{x}) > 3$.
(c) There exists λ' such that for all k and $\boldsymbol{x}, \boldsymbol{x}' \in \mathscr{D}_k$, $\max_{\boldsymbol{x},\boldsymbol{x}'} |n_k(\boldsymbol{x}) - n_k(\boldsymbol{x}')| \leq \lambda' s(\boldsymbol{x}, \boldsymbol{x}')$ and $n_k(\boldsymbol{x})$ is a non-decreasing function in k.
(d) $\mathscr{X}$ is compact.
(e) As the number of design points k increases, each $\boldsymbol{x}_i$ is independently generated from probability measure π with π having positive support in the entirety of $\mathscr{X}$.

Assumption 2(c) ensures there are not both design points with no new replications added and design points with their number of replications increasing to infinity as k does; this implies that no design points are ignored. If only a finite number of replications are allocated to all design points, Assumption 2(c) is satisfied. Assumption 2(e) ensures that $\lambda^\star$ is identifiable, because if no $\boldsymbol{x}$ and $\boldsymbol{x}'$ that achieve the maximal slope of $\lambda^\star$ are ever simulated then the estimator cannot converge to $\lambda^\star$. This assumption is sufficient, but not necessary; it is a simple way to ensure that the design does not place increasing weight on only a subset of $\mathscr{X}$ resulting in parts of $\mathscr{X}$ being ignored asymptotically.

Theorem 2 Under Assumptions 1–2, for any $\varepsilon > 0$

$$\lim_{k\to\infty} \mathbb{P}\left(\widehat{\lambda}_{1-\beta} \leq \lambda^\star - \varepsilon\right) = 0.$$

Proof. Sketch of the proof: To establish Theorem 2, a uniform law of large numbers is proven for the set of Lipschitz continuous functions. Then, the expected discrepancy is shown to be larger by a fixed constant than the expected discrepancy of the true mean function for any Lipschitz continuous function with constant smaller than $\lambda^\star - \varepsilon$. Combining the two statements yields the result. □

To compute the numerical value of $\widehat{\lambda}_{1-\beta}$ we use the plausible inference framework of Eckman et al. (2022) to note that $\widehat{\lambda}_{1-\beta}$ is the solution to the following linear program:

$$\begin{aligned} \min_{m_1,\ldots,m_k,\lambda} \quad & \lambda \\ \text{s.t.} \quad & \sum_{i=1}^{k} \frac{\sqrt{n_k(\boldsymbol{x}_i)}\,|m_i - \bar{Y}(\boldsymbol{x}_i)|}{\widehat{\sigma}(\boldsymbol{x}_i)} \leq D_{1-\beta}, \\ & m_i - m_j \leq \lambda s(\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ for all } 1 \leq i, j \leq k, \end{aligned}$$

where $\boldsymbol{x}_i$, $n_k(\boldsymbol{x}_i)$, $\bar{Y}(\boldsymbol{x}_i)$, $\widehat{\sigma}(\boldsymbol{x}_i), i = 1, 2, \ldots, k$, and $D_{1-\beta}$ are all observed or computed. The variables to be optimized are $m_1, m_2, \ldots, m_k$ and λ, with $\widehat{\lambda}_{1-\beta}$ taken to be the optimal value of λ. Notice that the sum constraint above can be made linear.

5 SEQUENTIAL DESIGN

Given knowledge of $\lambda^\star$ or its estimator $\widehat{\lambda}_{1-\beta}$, the choice of the design set, $\mathscr{D}_k$, remains an important problem. Depending on the choice of $\mathscr{D}_k$, a significantly different number of systems may be screened out. Sequential designs use previously simulated data to better decide which additional design points will be most informative. In this section we propose a method for sequentially choosing design points in the optimization and feasibility settings. At one extreme our fully sequential design adds design point $\boldsymbol{x}_{k+1}$ to obtain $\mathscr{D}_{k+1}$ as a function of the locations and observed outputs from $\mathscr{D}_k$, that is, $\{(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k), (\bar{Y}(\boldsymbol{x}_1), \bar{Y}(\boldsymbol{x}_2), \ldots, \bar{Y}(\boldsymbol{x}_k)), (\widehat{\sigma}(\boldsymbol{x}_1), \widehat{\sigma}(\boldsymbol{x}_2), \ldots, \widehat{\sigma}(\boldsymbol{x}_k))\}$. Theorem 3 shows that a fully sequential design does not affect the Type I error rate guarantees in the feasibility and optimization settings.

Theorem 3 Under assumptions (A)–(B), for all sequential design choices to obtain $\mathscr{D}_{k+1}$, including those based on $\mathscr{D}_k$,

$$\sup_{\boldsymbol{x}\in\mathscr{A}} \mathbb{P}\left(\inf_{\phi\in G_{\lambda^\star}(\boldsymbol{x})} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) > D_{1-\alpha} \right) \leq \alpha.$$

Theorem 3 allows significant freedom in how the sequential design is chosen while maintaining the Type I error rate. Specifically, even when the entirety of the simulation output is used to choose a new design point, the pointwise guarantee is maintained. In addition to fully sequential designs, we also consider batch-sequential designs in which we add $b > 1$ design points at each iteration. Both strategies use upper and lower confidence bounds provided by plausible inference at potential design points.

5.1 Fully Sequential Design

We assume that after a design point, $\boldsymbol{x}_{k+1}$ is chosen, we generate $n_{k+1}(\boldsymbol{x}_{k+1})$ replications at $\boldsymbol{x}_{k+1}$: $Y_j(\boldsymbol{x}_{k+1}), j = 1,2,\ldots,n_{k+1}(\boldsymbol{x}_{k+1})$. Then combining the new outputs, and all previous outputs, to form $\mathscr{D}_{k+1}$, the next design point $\boldsymbol{x}_{k+2}$ is chosen, and so on.

Let $\mathscr{D}_{k_0}$ be an initial design consisting of k_0 design points. For example, $\mathscr{D}_{k_0}$ may be a space-filling design in $\mathscr{X}$. Then, given $\mathscr{D}_k$, $\bar{\mathbf{Y}}$ and $\widehat{\sigma}$, vectors that grow with k, the $(k+1)$st design point, $\boldsymbol{x}_{k+1}$ is chosen. Similar to standard methods in Bayesian optimization, we propose using $1-\alpha$ upper and lower confidence bounds for $\mu(\boldsymbol{x})$ at each $\boldsymbol{x}\in\mathscr{X}$ as the basis for choosing $\boldsymbol{x}_{k+1}$. The $1-\alpha$ upper confidence bound for $\mu(\boldsymbol{x})$ is

$$U_{1-\alpha}(\boldsymbol{x}, \mathscr{D}_k, \bar{\mathbf{Y}}, \widehat{\sigma}, \widehat{\lambda}_{1-\beta}) = \max_{\phi\in M_{\widehat{\lambda}_{1-\beta}}} \phi(\boldsymbol{x}) \text{ s.t. } d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) \leq D_{1-\alpha}.$$

Similarly, the $1-\alpha$ lower confidence bound is

$$L_{1-\alpha}(\boldsymbol{x}, \mathscr{D}_k, \bar{\mathbf{Y}}, \widehat{\sigma}, \widehat{\lambda}_{1-\beta}) = \min_{\phi\in M_{\widehat{\lambda}_{1-\beta}}} \phi(\boldsymbol{x}) \text{ s.t. } d^1(\phi, \bar{\mathbf{Y}}, \widehat{\sigma}) \leq D_{1-\alpha}.$$

Notice that in the bounds above, if $\lambda = \lambda^\star$ then the upper and lower confidence bounds are pointwise $1-\alpha$ confidence bounds for all $\mu(\boldsymbol{x})$. When $\lambda^\star$ is unknown, we use $\widehat{\lambda}_{1-\beta}$ as a plug-in estimator. These bounds represent a range of plausible values for $\mu(\boldsymbol{x})$ given the already simulated design points and Lipschitz structural information. For ease of notation, we omit the arguments $\mathscr{D}_k, \bar{\mathbf{Y}}, \widehat{\sigma}$ and $\widehat{\lambda}_{1-\beta}$ and denote the upper and lower bounds at $\boldsymbol{x}$ by $U_{1-\alpha}(\boldsymbol{x})$ and $L_{1-\alpha}(\boldsymbol{x})$, respectively.

Based on the upper and lower confidence bounds, an acquisition function that evaluates the suitability of any $\boldsymbol{x}$ as the next design point, denoted $g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x}))$, can be defined. This acquisition function is used to select the next design point. For example, in the feasibility setting, where the feasibility threshold is $\mu(\boldsymbol{x}) \geq c$, we employ

$$g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x})) = I(c \in [L_{1-\alpha}(\boldsymbol{x}), U_{1-\alpha}(\boldsymbol{x})])\,(U_{1-\alpha}(\boldsymbol{x}) - L_{1-\alpha}(\boldsymbol{x})).$$

Here, $g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x}))$ is the confidence interval width at $\boldsymbol{x}$ if the interval covers c and zero otherwise. This choice of acquisition function targets decision variables which have both high uncertainty about their mean performance and are plausible to obtain feasibility. Likewise, in the optimization setting we employ the standard upper confidence bound strategy,

$$g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x})) = U_{1-\alpha}(\boldsymbol{x})$$

which reflects the potential optimality of $\boldsymbol{x}$. The $(k+1)$st design point is then chosen as

$$\boldsymbol{x}_{k+1} = \text{argmax}_{\boldsymbol{x}\in\mathscr{X}} g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x})).$$

Because the upper and lower $1-\alpha$ confidence bounds for any $\boldsymbol{x} \in \mathscr{X}$ can be computed using a linear program (Eckman et al. 2022; Qiao et al. 2024), evaluation of $g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x}))$ requires solving a linear program. If $|\mathscr{X}|$ is finite, one can iterate through each $\boldsymbol{x} \in \mathscr{X}$ to find $\boldsymbol{x}_{k+1}$. If $\mathscr{X}$ is a hypercube in $\Re^d$, then it can be shown that $\boldsymbol{x}_{k+1}$ is the solution to a quadratically constrained quadratic program.

Algorithm 1 is the pseudocode implementation of the fully sequential design for the feasibility problem. An analogous algorithm, requiring only minor modification, can be used for the optimization problem.

Algorithm 1 Feasibility Checking with Fully Sequential Design

Require: $c, \alpha, \beta, k_0, K, \mathscr{D}_{k_0}, \bar{\mathbf{Y}} = (\bar{\mathbf{Y}}_1, \ldots, \bar{\mathbf{Y}}_{k_0})$ and $\widehat{\boldsymbol{\sigma}} = (\widehat{\boldsymbol{\sigma}}_1, \ldots, \widehat{\boldsymbol{\sigma}}_{k_0})$

$D_{1-\alpha} \leftarrow$ The $1-\alpha$ quantile of the sum of the absolute value of k_0 independent t random variables

$D_{1-\beta} \leftarrow$ The $1-\beta$ quantile of the sum of the absolute value of k_0 independent t random variables

$\widehat{\lambda}_{1-\beta} \leftarrow \inf_{\lambda}$ s.t $\inf_{\phi \in M_\lambda} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\boldsymbol{\sigma}}) \leq D_{1-\beta}$

for $k \in \{k_0+1, \ldots, K\}$ **do**

$\boldsymbol{x}_k \leftarrow \text{argmax}_{\boldsymbol{x} \in \mathscr{X}} g(\boldsymbol{x}, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x}))$ ▷ Choose the next design point

Generate $n_k(\boldsymbol{x}_k)$ i.i.d. replications, $Y_1(\boldsymbol{x}_k), \ldots, Y_{n_k(\boldsymbol{x}_k)}(\boldsymbol{x}_k)$

$\bar{Y}(\boldsymbol{x}_k) \leftarrow \sum_{j=1}^{n_k(\boldsymbol{x}_k)} Y_j(\boldsymbol{x}_k)/n_k(\boldsymbol{x}_k), \widehat{\sigma}_k \leftarrow \sqrt{\sum_{j=1}^{n_k(\boldsymbol{x}_k)} (Y_j(\boldsymbol{x}_k) - \bar{Y}_k(\boldsymbol{x}_k))^2/(n_k(\boldsymbol{x}_k)-1)}$

$\mathscr{D}_k \leftarrow (\mathscr{D}_{k-1}, \boldsymbol{x}_k), \bar{\mathbf{Y}} \leftarrow (\bar{\mathbf{Y}}, \bar{Y}(\boldsymbol{x}_k)), \widehat{\boldsymbol{\sigma}} \leftarrow (\widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\sigma}}(\boldsymbol{x}_k))$

Update $D_{1-\alpha}$ and $D_{1-\beta}$ based on k random variables

$\widehat{\lambda}_{1-\beta} \leftarrow \inf \lambda$ s.t $\inf_{\phi \in M_\lambda} d^1(\phi, \bar{\mathbf{Y}}, \widehat{\boldsymbol{\sigma}}) \leq D_{1-\beta}$ ▷ Update the Lipschitz Estimator

end for

Any $\boldsymbol{x}_0 \in \mathscr{X}$ is declared unacceptable if $U_{1-\alpha}(\boldsymbol{x}_0) < c$

5.2 Batch-Sequential Design

In the formulation above, $\boldsymbol{x}_{k+1}$ was chosen based on all information obtained from design $\mathscr{D}_k$. However, it is frequently advantageous to choose the next set of design points as a batch, say $\boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+2}, \ldots, \boldsymbol{x}_{k+b}$ prior to simulating any of them. For example, given a batch of b design points they can all be simulated in parallel.

A naïve approach for choosing a batch of design points is to find b unique design points in $\mathscr{X}$ which attain acquisition values at least as large as any other points in $\mathscr{X}$ that have not already been included. A problem with this approach is that if $\mathscr{X}$ is continuous, then finding these may not be possible. Further, even if $|\mathscr{X}|$ is finite, it is likely that the chosen points will be clustered together because they do not take into account the increased information from simulating the new design points. To address these issues we adapt the "constant liar" heuristic used in Gaussian Process regression (Ginsbourger et al. 2010).

The idea is straightforward: We build the batch in a fully sequential manner—that is, adding one design point at a time until the batch is complete—by imputing outputs for the not-yet-simulated design points in the batch. We illustrate the approach for the feasibility problem. The imputed value used in Ginsbourger et al. (2010) imputes the posterior mean for each batch design point. However, in the feasibility setting, there exists no distribution to help choose an imputed value. Instead, we impute the feasibility threshold, c. Theorem 4 below shows that in the noiseless case employing c for the imputed value does not affect the feasibility of the remaining design points.

We now discuss the selection of, say, $\boldsymbol{x}_{k+i}$ given the design points that have been simulated, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ and the $k+i-1$ already chosen batch design points, $\boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_{k+i-1}$. To select $\boldsymbol{x}_{k+i}$, we define the upper and lower confidence bounds using mean and standard deviation vectors consisting of both the sample mean and standard deviation estimates of simulated design points together with the imputed values of the unsimulated design points already chosen for the batch. For instance, the length $k+i-1$ estimated mean vector is $\tilde{\mathbf{Y}} = (\bar{Y}(\boldsymbol{x}_1), \ldots, \bar{Y}(\boldsymbol{x}_k), c, \ldots, c)$ and the length $k+i-1$ estimated standard deviation vector is

$\tilde{\boldsymbol{\sigma}} = (\widehat{\sigma}(\boldsymbol{x}_1), \ldots, \widehat{\sigma}(\boldsymbol{x}_k), \tilde{\sigma}, \ldots, \tilde{\sigma})$. The standard deviation estimate $\tilde{\sigma}$ can simply be $\sum_{i=1}^{k} \widehat{\sigma}(\boldsymbol{x}_i)/k$, or could come from a fitted metamodel for $\sigma^2(\cdot)$ to obtain more refined "lies."

The imputed upper and lower confidence bounds are then

$$\begin{aligned} \tilde{U}_{1-\alpha}(\boldsymbol{x}) &= \max_{\phi \in M_{\widehat{\lambda}_{1-\beta}}} \phi(\boldsymbol{x}) \text{ s.t. } d^1(\phi, \tilde{\mathbf{Y}}, \tilde{\boldsymbol{\sigma}}) \leq D_{1-\alpha} \\ \tilde{L}_{1-\alpha}(\boldsymbol{x}) &= \min_{\phi \in M\widehat{\lambda}_{1-\beta}} \phi(\boldsymbol{x}) \text{ s.t. } d^1(\phi, \tilde{\mathbf{Y}}, \tilde{\boldsymbol{\sigma}}) \leq D_{1-\alpha}. \end{aligned}$$

Using these imputed confidence bounds in the previously defined acquisition functions, the next chosen design point is $\boldsymbol{x}_{k+i} = \text{argmax}_{\boldsymbol{x} \in \mathscr{X}} g(\boldsymbol{x}, \tilde{U}_{1-\alpha}(\boldsymbol{x}), \tilde{L}_{1-\alpha}(\boldsymbol{x}))$. We continue until the batch is complete.

Why might this "constant lie" be useful? As evidence, consider the noiseless case where $\sigma^2(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathscr{X}$. Theorem 4 shows that if c is chosen as the value for the constant liar, then the addition of the newly selected design point does not alter the feasibility of any other decision-variable value. Because of this, a sequence of design points can be chosen without worrying about altering the apparent feasibility of yet other design points.

Specifically, we define the set of apparently acceptable decision-variable values to be $\widehat{\mathscr{A}} = \{\boldsymbol{x} \in \mathscr{X} | U_{1-\alpha}(\boldsymbol{x}) \geq c\}$. The upper confidence bound, $U_{1-\alpha}$, used to construct $\widehat{\mathscr{A}}$ is based on $\mathscr{D}_k$ and its corresponding simulated outputs. When b batched design points are added to $\mathscr{D}_k$, we define the set of acceptable decision variables to be $\tilde{\mathscr{B}} = \{\boldsymbol{x} \in \mathscr{X} | \tilde{U}_{1-\alpha}(\boldsymbol{x}) \geq c\}$. Recall that $\tilde{U}_{1-\alpha}(\boldsymbol{x})$ employs the imputed value c. Theorem 4 demonstrates that the acceptable $\boldsymbol{x}$ values remain unchanged when adding new design points so long as their imputed mean values are c.

Theorem 4 Suppose $\sigma^2(\boldsymbol{x}) = 0, \ \forall \boldsymbol{x} \in \mathscr{X}$. Then for any $b > 0$, $\widehat{\mathscr{A}} = \tilde{\mathscr{B}}$.

6 EMPIRICAL ILLUSTRATIONS

In this section, we present results from applying fully sequential and two-stage batch-sequential designs to two different feasibility checking problems. In both problems we compare results when $\widehat{\lambda}_{1-\beta}$ is used as a plug-in estimator to results that use the true Lipschitz constant, $\lambda^\star$.

The sequential procedures require the specification of the following parameters: the Type I error rate, α, used for screening, the lower confidence bound confidence level, $1-\beta$, used for $\widehat{\lambda}_{1-\beta}$, the initial design size, k_0, the ending design size, K, the feasibility threshold, c, and the acquisition function, $g(\cdot)$. In both problems, we set $\alpha = 0.05$ and $1-\beta = 0.5$. We set α to be small to protect against Type I error. On the other hand, we set $1-\beta$ to be relatively large because $\widehat{\lambda}_{1-\beta}$ is a $1-\beta$ lower confidence bound for $\lambda^\star$. Setting the confidence level for this to be 50% helps $\widehat{\lambda}_{1-\beta}$ to act more like an estimator than a lower confidence bound. Our distance metric is Euclidean.

For the two feasibility problems, we use the acquisition function,

$$g(x, U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x})) = I\left(c \in [U_{1-\alpha}(\boldsymbol{x}), L_{1-\alpha}(\boldsymbol{x})]\right)\left(U_{1-\alpha}(\boldsymbol{x}) - L_{1-\alpha}(\boldsymbol{x})\right),$$

the confidence bound gap if it includes the feasible threshold, c. In practice, we have found this acquisition function to perform well across a wide range of problems. Finally, we set the feasible threshold, c, to be the 90th percentile of the mean performance measure across an integer grid of decision variables which span $\mathscr{X}$. Specifically, let T be an integer grid of $\mathscr{X}$. Then c is the 90th percentile of $\{\mu(t) | t \in T\}$.

To evaluate the performance of each procedure, we employ two metrics. The first is the power, defined as the probability of correctly screening out infeasible systems. The second is the Type I error rate, defined as the probability of incorrectly screening out feasible systems. Because these two metrics depend on the possibly random initial design chosen by the procedure and the replications generated at each design point, we run 100 macroreplications of the procedures. Across all 100 macroreplications, we define the set of decision variables used in the power calculation to be $\mathscr{A}^c = \{t \in T | \mu(t) < c\}$ and

the set of feasible decision variables to be $\mathscr{A} = \{t \in T | \mu(t) \geq c\}$. Therefore, we report the estimated expected power as $\widehat{\text{Power}} = \sum_{r=1}^{100} \sum_{t \in \mathscr{A}^c} I_r(U_{1-\alpha}(t) < c)/(100 \cdot |\mathscr{A}^c|)$, and the expected Type I error rate as $\widehat{\text{Error}} = \sum_{r=1}^{100} \sum_{t \in \mathscr{A}} I_r(U_{1-\alpha}(t) \geq c)/(100 \cdot |\mathscr{A}|)$. When estimated Lipschitz values are used, we also report the final sample average Lipschitz value.

6.1 Newsvendor Problem

The first problem considered is the single-period Newsvendor problem in Eckman et al. (2020). In this problem, the newsvendor must choose an appropriate order quantity, x, of goods to sell to maximize expected profit. The unknown demand for the goods, ξ, follows a Weibull distribution with scale parameter 50 and shape parameter 2. The purchase of each good incurs cost, c_{order}, and the sale of each good incurs revenue, p_{sales}. If the newsvendor orders more goods than are demanded, the extra goods are sold at a per-unit revenue of p_{salvage}. If the newsvendor orders fewer goods than are demanded, the unmet demand incurs a per-unit cost of c_{shortage}. The mean performance of decision variable x is,

$$\mu(x) = \mathbb{E}\left(p_{\text{sales}} \min(x, \xi) + p_{\text{salvage}} \max(0, x - \xi) - c_{\text{order}} x - c_{\text{shortage}} \max(0, \xi - x)\right).$$

In our experiment, we set $p_{\text{sales}} = 9$, $p_{\text{salvage}} = 1$, $c_{\text{order}} = 3$ and $c_{\text{shortage}} = 1$. For these given values, $\mu(\cdot)$ is a Lipschitz continuous function with $\lambda^\star = 7$.

We set the initial design size to be $k_0 = 15$. The initial design is a space-filling design of 15 equally spaced design points across $\mathscr{X}$. The final design size is $K = 30$. The space of decision variables is $\mathscr{X} = [0, 200]$. The variance of the replications generated from the decision variables, $\sigma^2(x)$, is relatively large with typical values as high as $36,000$. Because of this, we set a relatively large number of replications, $n_k(\boldsymbol{x}_i) = 300$, for all design points, $\boldsymbol{x}_i$. This results in standard errors of around 10 for the sample means of each design point. Finally, we used $c = 192.7$.

Figure 1 is an illustration of a single sample path of the initial design. The response sample mean at each initial design point (red bullets) are shown together with the mean performance function, $\mu(\cdot)$ (black curve) and c (black horizontal line). Figure 2 plots a single path of the fully sequential and two-stage designs side by side. To help with readability, only the subset of $\mathscr{X}$ that contains the sequential design points is plotted. Notice that for the two-stage design all 15 sequential design points were chosen in one batch, but they were still chosen in a specific order using the constant liar heuristic. The numbering of the points represents that order.

A comparison of the performance of the fully sequential and two-stage designs is given in Table 1. As a benchmark for comparison, we also show results from a non-sequential (one-shot) space-filling design of the same total design size ($K = 30$). In both the estimated and known Lipschitz constant cases, the fully sequential design outperforms the two-stage design, which outperforms the one-shot space-filling design. In all cases, the Type I error rate is zero.

When using the estimated Lipschitz constant, it is difficult to isolate the screening performance of each design procedure from the design procedure's impact on the estimated Lipschitz constant. However, in the known Lipschitz case, all three procedures utilize the same Lipschitz constant value of 7. In this case, we still see that the fully sequential design outperforms the two-stage design which outperforms the space-filling design.

6.2 (S,s) Inventory Problem

The second problem we consider is the multi-period (s,S) inventory problem. In this problem, inventory is stocked according to an (s,S) policy stipulating that when inventory falls below the level s, the inventory is replenished to the higher level S. Therefore, there are two decision variables which are represented as $\boldsymbol{x} = (s, S-s)$.

In each time period, the unknown demand for goods, ξ, follows a Poisson distribution with mean 25. If demand falls below the current inventory level, a per-unit holding cost of c_{holding} is incurred. On the

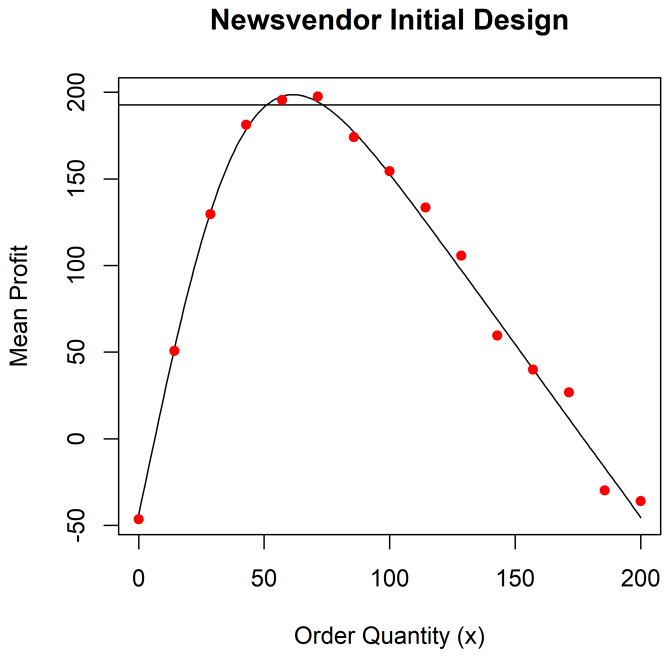

Figure 1: Sample means (red bullets) at the initial design points for a typical macroreplication, together with the true function $\mu(x)$ (black curve).

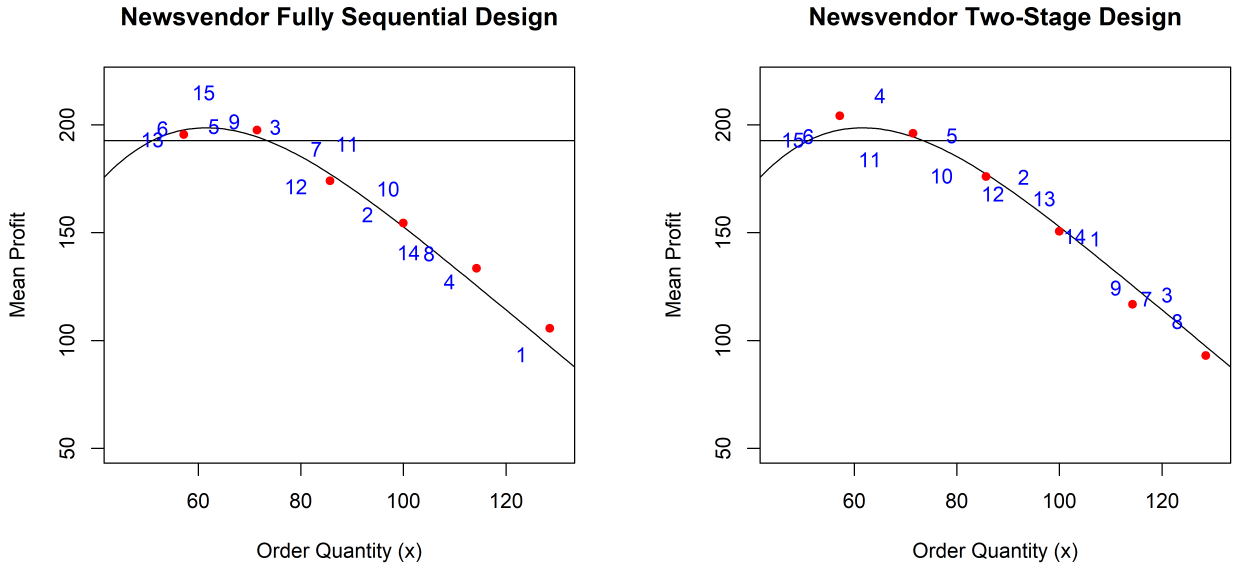

Figure 2: A typical example of the sequential design procedures. The sample means for the points chosen by the fully sequential design (left panel) and the points chosen by the two-stage design (right panel) are numbered in the order they were chosen in blue. Red points are the sample means for the initial one-shot space-filling design points.

Table 1: Newsvendor Experiment Results.

Lipschitz Constant	Design Type	$\widehat{\text{Power}}$	$\widehat{\text{Error}}$	Ave $\widehat{\lambda}_{1-\beta}$
Estimated	Fully Sequential	0.782 (0.002)	0	5.353 (0.016)
	Two-Stage Sequential	0.780 (0.002)	0	5.320 (0.013)
	Space-Filling	0.725 (0.002)	0	5.629 (0.010)
Known	Fully Sequential	0.601 (.002)	0	7
	Two-Stage Sequential	0.545 (.003)	0	7
	Space-Filling	0.534 (.001)	0	7

other hand, if demand is above the current inventory level, a per-unit shortage cost of c_{shortage} is incurred. When inventory is restocked, a flat fee of c_{flat} is charged along with a per-unit cost of c_{order}. Let q_i be the inventory level at time period i with $q_1 = S$. Set τ to be the total number of time periods. Then the mean performance at decision variable $\boldsymbol{x} = (s, S-s)$ is the negative of the average cost incurred by the (s,S) policy across the first τ time periods,

$$\mu(\boldsymbol{x}) = -\frac{1}{\tau}\mathbb{E}\left(\sum_{i=1}^{\tau} I(q_i < x_1)(c_{\text{flat}} + (x_2 + x_1 - q_i)c_{\text{cost}}) + \max(0, q_i - \xi)c_{\text{holding}} + \max(0, \xi - q_i)c_{\text{shortage}}\right).$$

In our experiment, we set $c_{\text{flat}} = 32$, $c_{\text{cost}} = 3$. $c_{\text{holding}} = 1$, $c_{\text{shortage}} = 5$ and $\tau = 30$. For these given values, $\mu(\cdot)$ is a Lipschitz continuous function with $\lambda = 4.7$.

We fix an initial design size of $k_0 = 16$ and an ending design size of $K = 49$. We set $\mathscr{X} = [10,40]^2$. The initial design, $\mathscr{D}_{k_0}$, is the two dimensional grid of evenly spaced design points, $[10,20,30,40] \times [10,20,30,40]$. In this problem, the variance of simulated response values $Y_j(\boldsymbol{x})$ averaged across all integer-valued decision variables in $\mathscr{X}$ is approximately 18. Therefore, we set $n_k(\boldsymbol{x}) = 5$. The feasible threshold is set to $c = -110$.

Table 2 compares the (s,S) problem using the estimated versus known Lipschitz constant. For each case, the fully sequential, two-stage and one-shot space-filling designs are compared. When the Lipschitz constant is estimated, performances of the three different designs are similar. While the one-shot design performs slightly better, this is most likely due to its estimated Lipschitz constant being smaller. When the Lipschitz constant is the same across all three designs in the known-constant case, the one-shot design performs worst. Importantly, the Lipschitz constant estimates in all three designs are significantly smaller than the true Lipschitz constant of 4.7. This is the likely reason for the much better performance of all three design procedures when the estimated Lipschitz constant is used: good Lipschitz estimates are, for the most part, only needed locally at decision variable values whose mean performance is close to the feasible threshold. If a decision variable's mean performance is much larger or smaller than the feasible threshold, it is unlikely that a too-small Lipschitz constant estimate will lead to a misclassification. For the decision variables that have mean performance close to the feasible threshold, more accurate estimates are needed to prevent misclassification. While these estimates need to be accurate, they only need to represent slopes in a local region instead of attaining global accuracy. Using the true Lipschitz constant, a global maximum slope, leads to overly conservative screening results. In contrast, because our Lipschitz estimator only uses outputs from the design points, which are typically clustered in regions whose decision variables are on the borderline of feasibility, it is more reflective of an aggregation of local slope values instead of a globally maximum value.

The downside of using the true Lipschitz constant, an extremely conservative value, is apparent from the results of Table 2: all power is lost when the true Lipschitz constant is used. None of the design methods are able to screen any of the space. While other problems such as the newsvendor problem do not suffer such a large decrease in power when the known Lipschitz constants are employed, using the estimated Lipschitz constant instead of the known Lipschitz constant led to uniform increases in power and no change in the Type I error rate.

Table 2: (S,s) Experiment Results.

Lipschitz Constant	Design Type	$\widehat{\text{Power}}$	$\widehat{\text{Error}}$	Ave $\widehat{\lambda}_{1-\beta}$
Estimated	Fully Sequential	0.561 (0.005)	0	1.050 (0.011)
	Two-Stage Sequential	0.547 (0.006)	0	1.045 (0.009)
	Space-Filling	0.614 (0.003)	0	0.962 (0.008)
Known	Fully Sequential	0.002 (0)	0	4.7
	Two-Stage Sequential	0.001 (0)	0	4.7
	Space-Filling	0 (0)	0	4.7

ACKNOWLEDGEMENTS

This research was partially supported by National Science Foundation Grant No. CMMI-2206973.

REFERENCES

Eckman, D. J., M. Plumlee, and B. L. Nelson. 2020. "Revisiting Subset Selection". In *2020 Winter Simulation Conference (WSC)*, 2972–2983 https://doi.org/10.1109/WSC48552.2020.9383921.

Eckman, D. J., M. Plumlee, and B. L. Nelson. 2021. "Flat Chance! Using Stochastic Gradient Estimators to Assess Plausible Optimality for Convex Functions". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715288.

Eckman, D. J., M. Plumlee, and B. L. Nelson. 2022. "Plausible Screening using Functional Properties for Simulations with Large Solution Spaces". *Operations Research* 70(6):3473–3489.

Fazlyab, M., A. Robey, H. Hassani, M. Morari and G. Pappas. 2019. "Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks". In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Volume 32. Vancouver, Canada: Curran Associates, Inc.

Ginsbourger, D., R. Le Riche, and L. Carraro. 2010. "Kriging Is Well-Suited to Parallelize Optimization". In *Computational Intelligence in Expensive Optimization Problems*, edited by Y. Tenne and C.-K. Goh, 131–162. Berlin: Springer.

Hansen, P. and B. Jaumard. 1995. *Lipschitz Optimization*. New York: Springer.

Huang, J. W., S. J. Roberts, and J.-P. Calliess. 2023. "On the Sample Complexity of Lipschitz Constant Estimation". *Transactions on Machine Learning Research* 9:1–45.

Malherbe, C. and N. Vayatis. 2017, August. "Global Optimization of Lipschitz Functions". In *Proceedings of the 34th International Conference on Machine Learning*, edited by D. Precup and Y. W. Teh, Volume 70, 2314–2323. Sydney, Australia: International Conference on Machine Learning.

Piyavskii, S. 1972. "An Algorithm for Finding the Absolute Extremum of a Function". *USSR Computational Mathematics and Mathematical Physics* 12(4):57–67.

Plumlee, M. and B. L. Nelson. 2018. "Plausible Optima". In *2018 Winter Simulation Conference (WSC)*, 1981–1992 https://doi.org/10.1109/WSC.2018.8632297.

Qiao, T., D. Eckman, and B. L. Nelson. 2024. "Plausible Intervals: Global Inference from Limited Simulation of Structured Problems". Technical report, Department of Industrial & Systems Engineering, Texas A&M University.

Shubert, B. O. 1972. "A Sequential Method Seeking the Global Maximum of a Function". *SIAM Journal on Numerical Analysis* 9(3):379–388.

Wood, G. and B. Zhang. 1996. "Estimation of the Lipschitz Constant of a Function". *Journal of Global Optimization* 8:91–103.

AUTHOR BIOGRAPHIES

GREGORY KESLIN is a Ph.D. student in the Department of Industrial Engineering & Management Sciences at Northwestern University. His research interests are simulation optimization, plausible inference and statistics. His e-mail address is gregorykeslin2025@u.northwestern.edu.

DANIEL W. APLEY is a Professor in the Department of Industrial Engineering & Management Sciences at Northwestern University. He is a Fellow of the American Statistical Association and has served as Editor of *Technometrics* and of the *Journal of Quality Technology*. His email address is apley@northwestern.edu.

BARRY L. NELSON is the Walter P. Murphy Professor Emeritus in the Department of Industrial Engineering & Management Sciences at Northwestern University. He is a Fellow of INFORMS and IISE. His e-mail address is nelsonb@northwestern.edu.