Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

METHODS OF PLAUSIBLE INFERENCE: THE DEFINITIVE COOKBOOK

Jinbo Zhao[1], Gregory Keslin[2], David J. Eckman[1], and Barry L. Nelson[3]

[1]Dept. of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA
[2]Martin Tuchman School of Management, New Jersey Institute of Technology, Newark, NJ, USA
[3]Dept. of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

ABSTRACT

This tutorial introduces a new cuisine to the simulationist kitchen: plausible inference, the culinary art of output analysis that concocts statistical inferences about possibly unsimulated settings (e.g., solutions, parameters, decision variables) by blending and baking experiment design and problem structure. Tasty applications include screening out settings with undesirable or uninteresting performance or constructing confidence intervals and regions for a setting's measure(s) of performance. This tutorial synthesizes several disparate works on plausible inference into a cohesive, unifying framework with the aim of demonstrating how plausible inference recipes can satisfy a range of analyst appetites. Bon appetit!

1 INTRODUCTION

Stochastic computer simulations are data generators that facilitate analytics for processes, systems and scenarios that do not yet exist. In most applications the simulation is capable of creating data under different settings for the decision variables, parameters or covariates. Since the data are generated by algorithmic code, there is always *some* structural relationship among the results at different settings, yet standard experiment design and analysis typically treats the simulation as a black box, and black-box output analysis methods necessarily require simulations at many settings to reach global conclusions. This feature is particularly limiting when the simulation is computationally expensive.

The toolkit of plausible inference (PI) exploits known structure to leverage the simulations run at a modest number of settings to make statements about unsimulated settings with strong statistical guarantees. From the initial work on identifying plausibly optimal solutions of simulation-optimization problems in Plumlee and Nelson (2018), the toolkit has grown to encompass many classes of problems encountered by simulation practitioners and to exploit various types of structural information, including convexity, monotonicity, Lipschitz continuity and bounds. *This paper brings together state-of-the-art results on PI in one definitive, unified and filling tutorial.*

Among the simulation-inference tasks addressed by PI and covered here are simulation optimization (single and multiple objective), screening, feasibility checking, interval estimation and uncertainty quantification; it is both a generic approach and specific methods. The PI framework can even be used to estimate the Lipschitz constant when Lipschitz continuity is expected but the constant is not known. PI is naturally parallelizable which is useful when investigating many feasible, but unsimulated, settings. This tutorial equips readers with both the theoretical foundations and computational tools needed to exploit PI in their research or applications.

The tutorial is organized as follows. Section 2 previews PI methods through an extended example. In Sections 3 and 4 we lay out the elements of PI methods and show how they can be combined to deliver insight in Section 5. Several straightforward applications are detailed in Section 6, with more advanced applications in Section 7 and conclusions in Section 8.

979-8-3315-8726-0/25/$31.00 ©2025 IEEE

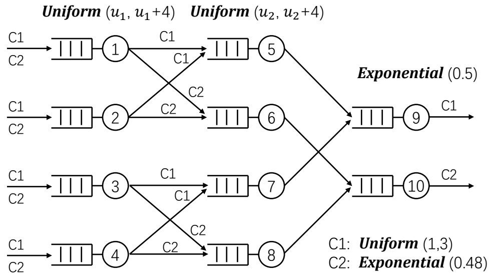


Figure 1: A two-class queueing network with finite buffers and interarrival times and service times that are either uniformly distributed on an interval (a,b) or exponentially distributed with a given mean.

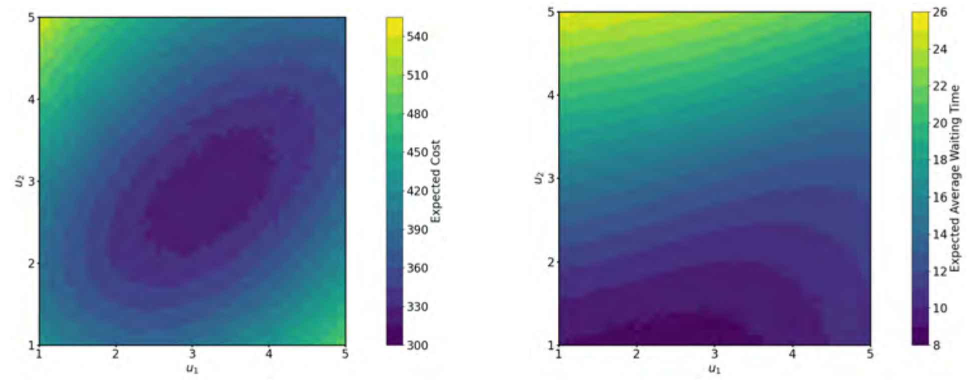
2 MOTIVATING EXAMPLE

We illustrate the capabilities of plausible inference on a queueing network model adapted from Patsis et al. (1997); see Figure 1. The network consists of 10 single-server, first-in-first-out stations arranged in three stages, and each station has an intake buffer of size 2. The network serves two customer classes: C1 with uniform interarrival times and C2 with exponential interarrival times. Customers of both classes are equally likely to arrive at any of the four entry stations (numbered 1–4) and subsequently follow fixed routes through the three stages. A station becomes blocked if both the server and its two buffer slots are occupied, forcing blocked customers to wait upstream, or be rejected and never enter at the entry station.

A decision maker is exploring the possibility of reducing the processing times in the first two stages by investing in server training or equipment improvements. These process improvements translate to shifting the processing time distributions to the left; hence, the problem has two design parameters (decision variables), denoted by u_1 and u_2, corresponding to the lower bounds of the Stage 1 and Stage 2 processing time distributions, as shown in Figure 1. The status quo (no investment) value of both parameters is 5 and they can each be reduced to as low as 1. The training cost associated with a given (u_1,u_2) is $10(5-u_1)^2+10(5-u_2)^2$. The decision maker is interested in two objectives: the expected average waiting time of served customers and an expected cost metric that sums customer rejection loss and training cost, where each rejected customer incurs a loss of 1 unit.

We can extract some useful structural information from the mathematical model alone. First, the training cost is convex by construction, and the total expected cost can, by extension, be reasoned to be convex due to the diminishing marginal effect that reducing processing times has on the number of rejected customers. Second, the expected average waiting time is likely not convex because of complicated blocking effects; e.g., increasing the Stage 1 processing time could incidentally reduce congestion in Stages 2 and 3 by causing more customers to be rejected, potentially lowering the overall waiting time. However, the expected average waiting time is believed to be Lipschitz continuous over the feasible region $[1,5]^2$ because the feasible region is bounded and the expectation operator is assumed to have a smoothing effect. Although an exact Lipschitz constant λ is unknown, one might be derived analytically or estimated from a limited simulation experiment (see Section 7.1). Unlike black-box methods, PI methods leverage these kinds of problem-specific structural information. The painstakingly generated contour plots in Figures 2a–2b illustrate these properties.

We estimate the two objectives via simulation where a single replication simulates the network for a warm-up period of 2,000 time units before collecting statistics over the next 1,000 time units. Suppose that we decide to conduct an exploratory simulation experiment to learn more about how the objectives change with the design parameters. We consider an experiment design (a set of design-parameter pairs to simulate) produced by generating a quasi-Monte Carlo sequence of length 32 in $[1,5]^2$ and rounding them to the nearest grid point on an 81×81 lattice (with spacing 0.05), as shown in Figure 3a. Each setting in the experimental set is simulated independently for 100 replications. Using the experiment results, we construct



(a) Expected total cost.

(b) Expected average waiting time.

Figure 2: Empirical contours of the two objectives for the queueing network problem.

a 95% joint confidence region for the objectives of the simulated settings by crossing $(0.95)^{1/32} \times 100\%$ confidence regions for each setting's objectives, which are individually shown in Figure 3b.

Based on the simulation experiment results, the decision maker may be interested in questions such as the following: What is the value of expected average waiting time over the feasible region (Figure 4a)? For a given setting (potentially unsimulated), is its performance acceptable (Figure 4b)? And what are the objective function values along the Pareto front of the two objectives (Figure 5)?

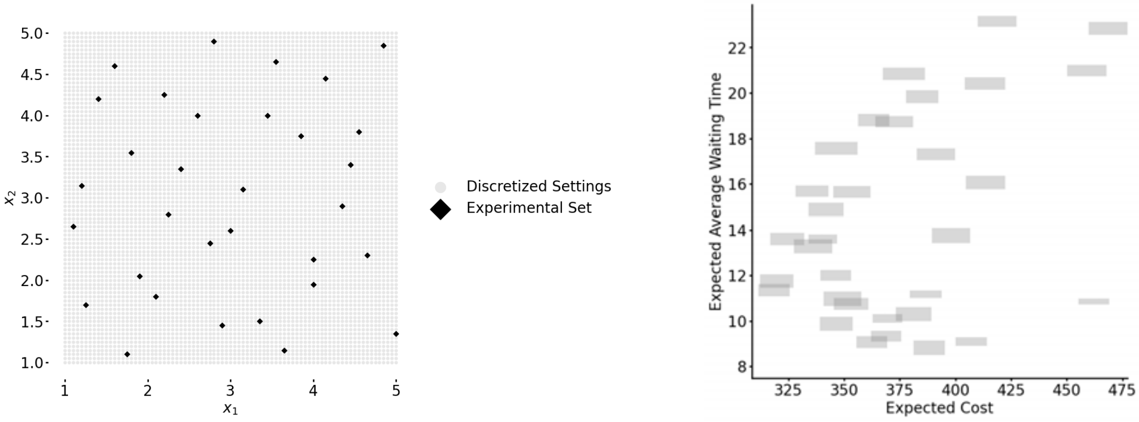
Here we exhibit how PI methods can provide answers to these questions using only the simulation outputs from the experimental set and the assumed structural properties. First, we can construct a plausible band with probability at least 95% containing the two true objective functions over the entire feasible region. In Figure 4a, we plot a cross-section of this plausible band for the expected average waiting time at $u_2 = 3$. Second, for any given setting, we can infer whether it is acceptable. In this example, we define a setting to be "acceptable" if it is Pareto optimal with respect to the aforementioned two objectives. We perform inference on all 81×81 settings over $[1,5]^2$, spaced at intervals of 0.05. We use the term "screening" to refer to the process of checking whether each candidate setting is acceptable. As shown in Figure 4b, we screened out 2152 of the total 6561 settings. Plausible inference can also give a confidence region for a target region in either the setting space or the objective space. Figure 4b is actually an inner approximation of a confidence region for the Pareto optimal settings. In Figure 5 we show a confidence region for the Pareto front, i.e., the objective values of all Pareto optimal settings.

3 THE PLAUSIBLE INFERENCE PANTRY

Methods of PI are built from a few fundamental ingredients that can be combined in a variety of ways to make inference. The strength of the framework is its versatility, both in how users can substitute these ingredients and still achieve valid inference and in how many forms of inference can be delivered. At a very high level, a common set of steps is followed when employing methods of PI:

Ingredients List: The user prepares the necessary ingredients. These include (a) *functional properties* that the unknown objective function is asserted to satisfy, such as convexity, monotonicity, or Lipschitz continuity; (b) a *definition of acceptability* that specifies the performance requirements under which a setting is regarded as acceptable; and (c) a *simulation experiment* that encompasses both the simulation outputs and the experimental design, including which settings were simulated, how many replications were performed, and whether techniques such as common random numbers (CRN) were used.

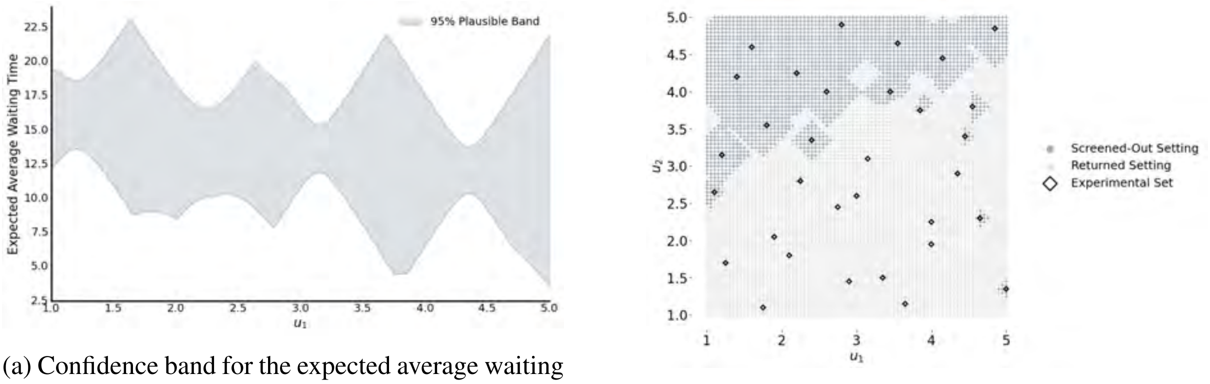
Mise en Place: Based on the assumed functional properties (Ingredient (a)) and the specified definition of acceptability (Ingredient (b)), we can access a set of functions satisfying the given functional properties



(a) Experiment design.

(b) Confidence regions.

Figure 3: The 32-setting design of a modest simulation experiment for the queueing network and a projected visualization of 95% joint confidence regions for the objectives of all *simulated* settings. Each gray box is the confidence region for the two objectives of a simulated setting.



(a) Confidence band for the expected average waiting time.

(b) Screening with functional structure.

Figure 4: The slice of the confidence band for the expected average waiting at time $u_2 = 3$ and screening results for the discretized settings.

and for which a given setting is acceptable. We project those functions to the real space corresponding to the objective values of the settings in the experimental set. A confidence region for the objective values of the settings in the experimental set must also be constructed using the simulation outputs obtained from the simulation experiment (Ingredient (c)).

Assembly and Baking: For each setting at which inference is desired, a small number of mathematical programs are formulated and solved, depending on the desired form of inference, and the optimal values are assessed to derive the inference.

In the remainder of this section, we provide the mathematical foundation for describing each of the ingredients in the PI pantry before presenting the preparations done in the Mise en Place step and the various mathematical programs that can arise in the Assembly and Baking step in Section 5. Sample recipes will be given in Section 6.

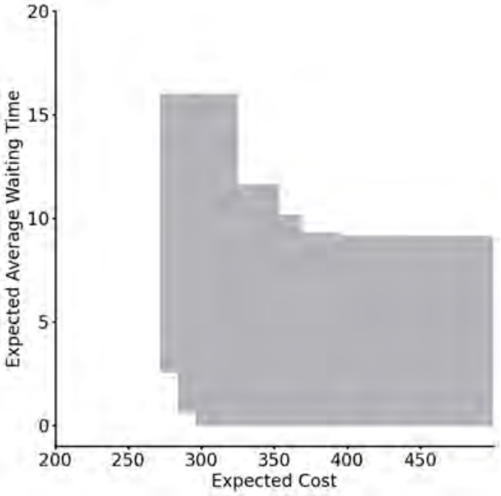


Figure 5: 95% confidence region for the Pareto front.

3.1 Simulation Model

Although not listed among the ingredients, a stochastic simulation model is essential for carrying out PI. We regard the simulation model as a mapping that takes fixed inputs describing a system design and returns random outputs measuring the system's performance on a single replication. Let $\mathbb{X} \subseteq \mathbb{R}^s$ denote the space of settings under consideration, where a setting is described by a vector $x \in \mathbb{X}$. The machinery of PI works the same whether $\mathbb{X}$ is finite, discrete, or continuous, but these distinctions have some implications on the assumed functional properties (see Section 3.2) and how inference is carried out. For example, when $\mathbb{X}$ consists of infinitely many settings, a user is more likely to invoke PI methods to assess quality at select settings, such as those observed in online-monitoring settings or those visited by optimization algorithms. When $\mathbb{X}$ consists of finitely many settings, or can be adequately discretized, one can make inference at all settings.

The output of a single replication at a setting x is denoted by a random vector $Y(x) \in \mathbb{R}^p$, where each component of $Y(x)$ represents a real-valued performance measure such as the total cost or the average waiting time in the queueing network example. A setting x is associated with a vector of objectives $\theta(x) \in \mathbb{R}^d$, where each objective is a functional of one or more performance measures, e.g., an expectation, a quantile, or a CoVaR (Tobias and Brunnermeier 2016). The number of objectives need not equal the number of performance measures; for example, one may be interested in both the mean and the variance of total cost—two functionals of a single performance measure. Plausible inference methods do not usually produce statements about the simulation outputs, $Y(x)$, but instead concern the objectives $\theta(x)$ and estimates thereof.

In the queueing network example, $\mathbb{X} = [1,5] \times [1,5]$, a setting x is described by the pair (u_1, u_2), $Y(x)$ is a vector containing the associated total cost and the average waiting time, and $\theta(x)$ is a vector containing the expectations of these two performance measures.

3.2 Functional Properties

Simulation experiments are usually conducted at only a modest number of settings. Except in extreme scenarios where every setting is simulated, there inevitably exist settings whose performance must be inferred rather than directly estimated. Functional structures serve as prior information that link the performance of observed settings to those of unobserved settings. These functional properties can be ascertained, verified, detected, or estimated, depending on the context. For example, in the queueing network problem, one may conclude that decreasing the processing times of a certain node reduces the number of blocked customers, reflecting monotonicity. The set of all objective functions possessing the asserted structural properties is denoted by $\mathscr{M}$. For example, if convexity is assumed, then $\mathscr{M}$ represents the set of all convex functions over the setting space. When there are multiple objectives, $\mathscr{M}$ is a set of vector-valued functions mapping from $\mathbb{X}$ to $\mathbb{R}^d$. We assume that $\theta(x) \in \mathscr{M}$.

3.3 Acceptability

Acceptability captures whether a setting would be chosen or retained by the decision maker, assuming full knowledge of its true objective value and the entire objective function. By incorporating acceptability, PI cleanly supports a variety of decision-making contexts, such as optimization and feasibility determination. Some definitions of acceptability are relative, involving comparisons across settings; for example, optimality (having the smallest objective value), top-m (being among the settings having the m smallest objective values), or Pareto optimality are possible definitions. Others are absolute, such as requiring performance to exceed a fixed threshold or lie within a target range. For a given setting x_0, acceptability specifies the set of objective functions, denoted by $\mathscr{A}(x_0)$, for which x_0 is considered acceptable.

3.4 Simulation Experiment

Plausible inference supposes that the user has conducted an experiment wherein a modest number of settings have been simulated. The design of this experiment includes which settings to simulate, how many replications to take at each setting, and how to use pseudo-random numbers when executing the replications.

The experimental set, denoted by $\mathsf{X} = \{x_1, x_2, \ldots, x_k\} \in \mathbb{X}$, consists of the settings that are to be simulated. The construction of X may be influenced by the structural assumptions or the definition of acceptability or may simply be space filling. Let $\mathsf{n} = (n_1, n_2, \ldots, n_k)$ be the number of replications taken at each simulated setting. Given a fixed overall simulation budget $\sum_{i=1}^{k} n_i$, there is a tradeoff between obtaining more precise estimates of the objectives at the simulated settings and estimating the objective functions at more settings. The choices of X and n influence the results of PI methods in nontrivial ways, which we do not explore further here.

As for how the simulation replications are taken, the PI framework makes no explicit assumptions about the distribution of $Y(x)$ for any x, nor does it require independence across replications at a given setting or across different settings. Hence, the framework does not preclude the use of variance reduction techniques that induce dependence across replications at a given setting (e.g., stratified sampling, antithetic sampling) or across settings (e.g., CRN). The sole requirement is that one can construct a valid or asymptotically valid confidence region for the objectives at the simulated settings based on the outputs of the initial experiment. We discuss this further in Section 4.2.

In this tutorial, we mostly focus on the context in which the methods of PI are applied to the results of a static designed experiment. We remark on the adaptive design setting in Section 7.2.

In the queueing network example, the experimental set is generated by filling the feasible region with a quasi-Monte Carlo sequence to produce even coverage across the feasible region. The total simulation budget is evenly allocated across all settings, and each setting is simulated independently to accommodate various forms of inference in this paper.

4 PREP WORK

In this section, we describe the two preparatory tasks in the Mise en Place step: deriving projections of sets of functions and constructing confidence regions for the objectives of the simulated settings.

4.1 Projecting Sets of Functions

Methods of PI are derived by assessing the feasibility of, or optimizing over, intersections of sets of functions. Working directly with set of functions becomes unwieldy, as doing so gives rise to infinite-dimensional optimization problems. For the sake of computational tractability, we will instead work with certain projections of these sets of functions.

Let $\mathscr{S} \subseteq \mathscr{F}^d$ be some generic set of functions, where $\mathscr{F}^d$ denotes the set of all functions mapping from $\mathbb{X} \to \mathbb{R}^d$. We define the projection of $\mathscr{S}$ onto a finite set $\widetilde{\mathsf{X}} = \{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_l\} \subseteq \mathbb{X}$—corresponding to the values $\mathscr{S}$ takes at the settings $\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_l$—as

$$\mathrm{proj}_{\widetilde{\mathsf{X}}}(\mathscr{S}) = \left\{ \mathbf{M} \in \mathbb{R}^{l \times d} \colon \text{there exists } m \in \mathscr{S} \text{ such that } m(\widetilde{\mathsf{X}}) = \mathbf{M} \right\},$$

where $m(\widetilde{\mathsf{X}})$ denotes the restriction of m to $\widetilde{\mathsf{X}}$. In other words, $\mathrm{proj}_{\widetilde{\mathsf{X}}}(\mathscr{S})$ is the set of all matrices containing the potential objectives of settings in $\widetilde{\mathsf{X}}$ for which there exists an interpolating function belonging to $\mathscr{S}$.

Methods of PI typically deploy the following four projections, where x_0 is some setting of interest:

- $\mathbb{M} \equiv \mathrm{proj}_{\mathsf{X}}(\mathscr{M})$ projects functions satisfying the functional properties onto the experimental set.
- $\mathbb{M}^+(x_0) \equiv \mathrm{proj}_{\mathsf{X} \cup \{x_0\}}(\mathscr{M})$ projects functions satisfying the functional properties onto the experimental set *and* x_0.
- $\mathbb{M}(x_0) \equiv \mathrm{proj}_{\mathsf{X}}(\mathscr{M} \cap \mathscr{A}(x_0))$ projects functions satisfying the functional properties, and for which setting x_0 is acceptable, onto the experimental set.
- $\mathbb{M}^+_A(x_0) \equiv \mathrm{proj}_{\mathsf{X} \cup \{x_0\}}(\mathscr{M} \cap \mathscr{A}(x_0))$ projects functions satisfying the functional properties, and for which setting x_0 is acceptable, onto the experimental set *and* x_0.

Concrete examples of these projections will be given in Section 6. For the functional properties we have investigated, the projections $\mathbb{M}$ and $\mathbb{M}^+(x_0)$ can be derived by referring to the definition of $\mathscr{M}$ and keeping only those conditions that pertain to settings in X or $\mathsf{X} \cup \{x_0\}$, respectively. On the other hand, the projections $\mathbb{M}(x_0)$ and $\mathbb{M}^+_A(x_0)$ can include constraints that describe interactions among the functional properties and the definition of acceptability. Outer approximations of these projections can be obtained by projecting $\mathscr{M}$ and $\mathscr{A}(x_0)$ separately and then taking the intersection of their projections, i.e., $\mathbb{M}(x_0) \subseteq \mathrm{proj}_{\mathsf{X}}(\mathscr{M}) \cap \mathrm{proj}_{\mathsf{X}}(\mathscr{A}(x_0))$ and $\mathbb{M}^+_A(x_0) \subseteq \mathrm{proj}_{\mathsf{X} \cup \{x_0\}}(\mathscr{M}) \cap \mathrm{proj}_{\mathsf{X} \cup \{x_0\}}(\mathscr{A}(x_0))$.

4.2 Confidence Regions

The final preparation step is constructing a confidence region for the objectives of the simulated settings. This region describes the uncertainty about $\theta(\mathsf{X})$ due to the simulation error and demarcates a set of objective matrices that are sufficiently aligned with the simulation outputs. This confidence region, which we denote by $\mathbb{C} \subseteq \mathbb{R}^{k \times d}$, is associated with a set of functions

$$\mathscr{C} \equiv \{m \in \mathscr{F}^d \colon m(\mathsf{X}) \in \mathbb{C}\}$$

in which the true objective function θ is believed to reside with some degree of statistical confidence. Because $\mathbb{C} = \mathrm{proj}_{\mathsf{X}}(\mathscr{C})$, the region $\mathbb{C}$ is at the same level of abstraction as the projections introduced in the previous section.

We assume that $\mathbb{C}$ is a $100(1-\alpha)\%$ confidence region for $\theta(\mathsf{X})$ with either finite-sample or asymptotic confidence. Mathematically, this means that $\mathrm{P}(\theta(\mathsf{X}) \in \mathbb{C}) \geq 1-\alpha$ or $\mathrm{P}(\theta(\mathsf{X}) \in \mathbb{C}) \gtrsim 1-\alpha$, where the notation $\gtrsim$ means that for any $\varepsilon > 0$, there exists an N such that for any n for which $\min_{i=1,2\ldots,k} n_i \geq N$, $\mathrm{P}(\theta(\mathsf{X}) \in \mathbb{C}) \geq 1-\alpha-\varepsilon$. Such a confidence region can be constructed in various ways such as bootstrapping or appealing to the Central Limit Theorem. Another approach involves *discrepancy functions*—which measure how well a candidate objective function aligns with the sample data—and appropriately chosen cutoffs. Although earlier work on PI largely built upon a foundation of discrepancy functions (Eckman et al. 2022), confidence regions are more general and are implied by discrepancy functions.

Regardless of the underlying distribution of the simulation outputs, an approach that is generally available to users (but is not required) is to construct $\mathbb{C}$ by combining (crossing) confidence regions for components of $\theta(\mathsf{X})$. When settings in X are simulated independently, $\mathbb{C}$ can be formed by crossing confidence regions for the objectives of each simulated setting, where the error is split multiplicatively over settings. More specifically, if $\mathbb{C}_i$ is a $100(1-\alpha)^{1/k}\%$ confidence region for $\theta(x_i)$ for $i = 1,2,\ldots,k$, then $\mathbb{C} = \mathbb{C}_1 \times \mathbb{C}_2 \times \cdots \times \mathbb{C}_k$ is a valid $100(1-\alpha)\%$ confidence region for $\theta(\mathsf{X})$. Alternatively, if settings in X are simulated dependently, a Bonferroni correction implies that the same cross-product formation of $\mathbb{C}$ can work, but with the error split additively over settings. Moreover, a joint confidence region for a setting's multiple objectives can be formed by crossing marginal confidence intervals for its individual objectives and using suitable inequalities, e.g., the Bonferroni inequality.

5 THE PLAUSIBLE INFERENCE COOKBOOK

In this section, we describe how the raw ingredients—functional properties, acceptability, and simulation experiment results—and the prepared ingredients—projections and confidence regions—introduced earlier are assembled to produce different forms of inference. At a conceptual level, methods of PI first formulate the intersection of some subset of $\mathscr{M}$, $\mathscr{A}(x)$, and $\mathscr{C}$ and then ask questions about the existence of objective functions in this intersection or the values such functions can take at certain settings of interest. In practice, these methods carry out analogous operations using projections of these sets of functions and ultimately entail solving a handful of mathematical programs. The "cooking" metaphor is an apt way to describe the process, because just as recipes combine assorted ingredients in specific ways to create unique dishes, methods of PI combine functional properties, acceptability, and simulation outputs *via optimization* to perform screening and uncertainty quantification.

5.1 Screening

One application of the framework is to assess whether a given setting is plausibly acceptable, that is, whether it plausibly has an objective vector that is acceptable to the decision maker. When this assessment is performed at all settings in $\mathbb{X}$, we refer to this operation as *plausible screening*, or simply screening. Screening is more practical when $\mathbb{X}$ is finite, but even when $\mathbb{X}$ is continuous, screening can be carried out approximately by discretizing the feasible region. Alternatively, assessing whether a given setting is plausibly acceptable can be a useful subroutine within an optimization algorithm that searches over $\mathbb{X}$ for acceptable settings.

A setting $x_0 \in \mathbb{X}$ is deemed plausibly acceptable if there exists a function (i) that possesses the specified functional properties, (ii) for which x_0 is acceptable, and (iii) that belongs to the confidence set of functions, i.e., if $\mathscr{M} \cap \mathscr{A}(x_0) \cap \mathscr{C} \neq \emptyset$. Rather than directly trying to determine the non-emptiness of $\mathscr{M} \cap \mathscr{A}(x_0) \cap \mathscr{C}$, we find it convenient to work with a projected version of this problem that assesses whether $\mathbb{M}(x_0) \cap \mathbb{C} \neq \emptyset$. This feasibility check is tantamount to solving the optimization problem

$$\min_{\mathsf{M}} 0 \text{ such that } \mathsf{M} \in \mathbb{M}(x_0) \cap \mathbb{C}. \tag{1}$$

If the optimal value to (1) is 0, then x_0 is plausibly acceptable and would be retained if screening out unacceptable settings; otherwise, if (1) is infeasible, then x_0 is screened out. We note that for the functional properties and the basic constructions of $\mathbb{C}$ we have investigated, solving (1) is equivalent to checking $\mathscr{M} \cap \mathscr{A}(x_0) \cap \mathscr{C} \neq \emptyset$ because it can be shown that $\mathbb{M}(x_0) \cap \mathbb{C} = \mathrm{proj}_{\mathsf{X}}(\mathscr{M} \cap \mathscr{A}(x_0) \cap \mathscr{C})$, i.e., the confidence region introduces no additional complications when taking projections. For many of the functional properties, definitions of acceptability, and confidence regions we have mentioned, (1) is either a linear program, a quadratic program, or a mixed-integer program—the latter arising in, for instance, cases where acceptability refers to weak efficiency (Pareto optimality) due to the presence of disjunctive constraints.

Screening all feasible settings results in the set of plausibly acceptable ones being returned, namely

$$\mathbb{S}_{\mathsf{n}} = \{x_0 \in \mathbb{X} \colon \mathbb{M}(x_0) \cap \mathbb{C} \neq \emptyset\},$$

where the subscript n signifies the dependence of this set on the simulation effort. The set $\mathcal{S}_{\mathsf{n}}$ automatically controls the probability of incorrectly screening out acceptable settings to be below α by virtue of basing the inference on only a subset of those functions belonging to the $100(1-\alpha)\%$ confidence set $\mathscr{C}$. Under mild conditions on $\mathbb{C}$, the method also asymptotically screens out certain unacceptable settings as the simulation effort at the settings in X increases, and the screening power depends on $\mathbb{M}(x_0)$, $\mathbb{C}$, and X. These statistical guarantees are discussed more in Section 5.3.

Plausible screening is related to subset selection, a classical problem in ranking and selection, wherein one seeks to return a subset of settings believed to contain the optimal (Zhao et al. 2023). More specifically, when the functional properties ingredient is left out of the recipe, $\mathbb{X}$ is finite, and the experimental set is exhaustive (i.e., $\mathsf{X} = \mathbb{X}$), plausible screening amounts to subset selection, but with a more inclusive definition of acceptability. This setting is addressed by the One-Shot Screening for Acceptability (OSSA) framework of Zhao and Eckman (2025), which similarly leverages optimization to screen out unacceptable settings. Other connections to subset selection are explored in Eckman et al. (2020), but through the lens of discrepancy functions instead of general confidence regions.

5.2 Uncertainty Quantification

The PI framework can also be used to quantify uncertainty about possible objectives of an arbitrary setting or of the (unknown) set of acceptable settings.

To motivate the former case, consider a single-objective problem. A decision maker may wish to know a range of plausible values for the objective of an arbitrary setting x_0. The PI framework can leverage the outputs of simulated settings and the functional properties ascribed to θ to return such an interval, *even if x_0 has not been simulated.* This can be achieved by considering all functions m that satisfy the function properties (i.e., are in $\mathscr{M}$) and coincide with the observed simulation outputs (i.e., are in $\mathscr{C}$) and considering the range of values these functions can take at x_0. As before, we can approach this by working with projections of these sets of functions. More specifically, define

$$\begin{aligned} \mathcal{I}_{\mathsf{n}}^{-}(x_0) &\equiv \min_{\mathsf{m}_0,\ \mathbf{m}} \mathsf{m}_0 \text{ s.t. } (\mathsf{m}_0, \mathbf{m}) \in \mathbb{M}^{+}(x_0) \text{ and } \mathbf{m} \in \mathbb{C} \text{ and} \\ \mathcal{I}_{\mathsf{n}}^{+}(x_0) &\equiv \max_{\mathsf{m}_0,\ \mathbf{m}} \mathsf{m}_0 \text{ s.t. } (\mathsf{m}_0, \mathbf{m}) \in \mathbb{M}^{+}(x_0) \text{ and } \mathbf{m} \in \mathbb{C}, \end{aligned} \tag{2}$$

where the vector $\mathsf{m} \in \mathbb{R}^k$ contains candidate objectives at $x_1, x_2, \ldots, x_k$ and the scalar $\mathsf{m}_0 \in \mathbb{R}$ represents a candidate objective at x_0. Therefore, $\mathcal{I}_{\mathsf{n}}^{-}(x_0)$ and $\mathcal{I}_{\mathsf{n}}^{+}(x_0)$ are the smallest and largest values, respectively, that any function that passes through the confidence region $\mathbb{C}$ and satisfies the functional properties—as dictated by $\mathbb{M}^{+}(x_0)$—can take. The interval $[\mathcal{I}_{\mathsf{n}}^{-}(x_0), \mathcal{I}_{\mathsf{n}}^{+}(x_0)]$ is a $100(1-\alpha)\%$ confidence interval for $\theta(x_0)$ and is called a *plausible interval* (Qiao et al. 2025). For the convex and Lipschitz continuous cases, if $\mathbb{C}$ is a hyperrectangle, the optimization problems in (2) are linear programs; moreover, those for the Lipschitz continuous case admit closed-form solutions. The approach outlined above for generating plausible intervals can be extended to the multi-objective setting to produce, for instance, confidence hyperrectangles for $\theta(x_0)$.

The acceptability ingredient is notably left out of the recipe above. Nevertheless, a decision maker might be interested in a confidence region for the *image* of the set of acceptable settings, denoted by $\theta(\mathbb{A})$, where $\mathbb{A} = \{x_0 \in \mathbb{X} \colon \theta \in \mathscr{A}(x_0)\}$ is the set of acceptable settings. Here, too, the PI framework provides a means to construct such a confidence region in the form of

$$\mathcal{P}_{\mathsf{n}}^{\mathsf{A}} \equiv \left\{ \mathbf{m}_0 \in \mathbb{R}^d \colon \text{there exists } x_0 \in \mathbb{X} \text{ and } \mathbf{M} \in \mathbb{C} \text{ such that } (\mathbf{m}_0, \mathbf{M}) \in \mathbb{M}_A^{+}(x_0) \right\}.$$

In words, $\mathcal{P}_{\mathsf{n}}^{\mathsf{A}}$ is the set of objective vectors for which we can find an associated setting that is plausibly acceptable. As an example, when acceptability is defined as weak efficiency, $\mathcal{P}_{\mathsf{n}}^{\mathsf{A}}$ is the set of plausibly Pareto optimal objective vectors and serves as a confidence region for the Pareto front. For this definition of acceptability, and for many of the functional properties and confidence region geometries mentioned in this tutorial, assessing whether $\mathsf{m}_0 \in \mathcal{P}_{\mathsf{n}}^{\mathsf{A}}$ entails solving a convex program or a mixed-integer program.

5.3 Statistical Guarantees

Whether applied for screening or uncertainty quantification, methods of PI are subject to making two kinds of errors. The first is incorrectly eliminating a setting or objective vector that should be returned, and the second is incorrectly returning a setting or objective vector that should be eliminated.

The first kind of error can be described in terms of *confidence*, of which we consider two forms:

Definition 1 (Finite-sample uniform confidence) For $\alpha \in [0,1]$, a random region $\mathbb{S}_\mathsf{n}$ achieves $1-\alpha$ finite-sample uniform confidence for a fixed region $\mathbb{T}$ if for any $\theta \in \mathscr{M}$, $\mathrm{P}(\mathbb{T} \subseteq \mathbb{S}_\mathsf{n}) \geq 1-\alpha$.

Definition 2 (Asymptotic uniform confidence) For $\alpha \in [0,1]$, a random region $\mathbb{S}_\mathsf{n}$ achieves $1-\alpha$ asymptotic uniform confidence for a fixed region $\mathbb{T}$ if for any $\theta \in \mathscr{M}$, $\mathrm{P}(\mathbb{T} \subseteq \mathbb{S}_\mathsf{n}) \gtrsim 1-\alpha$ as $\min_{i=1,2\dots,k} n_i \to \infty$.

In Definitions 1 and 2, the term *uniform* refers to the fact that the confidence region, $\mathbb{S}_\mathsf{n}$, contains the entire target region, $\mathbb{T}$, with high probability. If $\mathbb{C}$ is a $1-\alpha$ confidence region for $\theta(\mathsf{X})$, then the results of using PI for screening or uncertainty quantification achieve uniform confidence with the same confidence level and the same distinction (finite-sample or asymptotic) as that of $\mathbb{C}$. For screening, $\mathbb{S}_\mathsf{n}$ refers to the returned subset $\mathcal{S}_\mathsf{n}$ and $\mathbb{T}$ refers to the true set of acceptable settings $\mathbb{A}$. Similarly, for uncertainty quantification, $\mathbb{S}_\mathsf{n}$ refers to the interval $[\mathcal{I}^-_\mathsf{n}(x_0), \mathcal{I}^+_\mathsf{n}(x_0)]$ (for plausible intervals) or $\mathcal{P}^\mathsf{A}_\mathsf{n}$ (for confidence regions for the image of the acceptable settings) and $\mathbb{T}$ refers to the true objective function value $\theta(x_0)$ or the true image $\theta(\mathbb{A})$, respectively.

Remark 1 When methods of PI are modified to use different confidence regions when making inference at different settings or objective vectors, as opposed to using a common $\mathbb{C}$, they deliver a weaker, point-wise confidence, i.e., for all $t_0 \in \mathbb{T}$, $\mathrm{P}(t_0 \in \mathbb{S}_\mathsf{n}) \geq 1-\alpha$ or $\mathrm{P}(t_0 \in \mathbb{S}_\mathsf{n}) \gtrsim 1-\alpha$.

The second kind of error refers to *consistency*, which describes a method's ability to correctly screen out settings that are not plausibly acceptable or to eliminate objective vectors that are not plausible or not plausibly acceptable as the number of replications taken at settings in X approaches infinity. Except in special cases, such as when all settings are simulated, i.e., $\mathsf{X} = \mathbb{X}$, or when θ is a polynomial, there will exist some unacceptable settings or objective vectors that cannot be correctly eliminated with certainty even if we were to exactly observe $\theta(\mathsf{X})$. Because this gap cannot be overcome with greater simulation effort directed at X, methods of PI achieve a weaker form of consistency that depends on X.

Definition 3 ($\mathbb{T}(\mathsf{X})$ Consistency) A random set $\mathbb{S}_\mathsf{n}$ achieves $\mathbb{T}(\mathsf{X})$ consistency if for any $\theta \in \mathscr{M}$, $\mathrm{P}(t_0 \in \mathbb{S}_\mathsf{n}) \to 0$ as $\min_{i=1,2,\dots,k} n_i \to \infty$ for all $t_0 \notin \mathbb{T}(\mathsf{X})$.

The set $\mathbb{S}_\mathsf{n}$ is as described above for the two applications of PI, while $\mathbb{T}(\mathsf{X})$ is the smallest set guaranteed to contain the target region $\mathbb{T}$ if $\theta(\mathsf{X})$ were observed exactly. For example, in screening, $\mathbb{T}(\mathsf{X})$ are those settings that would be possibly acceptable if $\theta(\mathsf{X})$ were known, i.e., $\mathbb{T}(\mathsf{X}) = \{x_0 \in \mathbb{X} \colon \theta(\mathsf{X}) \in \mathbb{M}(x_0)\}$.

6 SELECT RECIPES

We present concrete examples of PI as applied to the queueing network introduced in Section 2. For each inference task, we will specify the relevant ingredients and how they are prepped and assembled into optimization problems to produce the results displayed in Figures 4a, 4b, and 5. All inferences are based on the confidence region shown in Figure 3, which is constructed from crossing confidence intervals $[L_{ir}, U_{ir}]$ for $\theta_r(x_i)$ for all $i = 1,2,\dots,32$ and $r = 1,2$.

6.1 Confidence Band with Lipschitz Continuity

We exploited the λ-Lipschitz continuity of the expected average waiting time to construct the confidence band shown in Figure 4a. The first ingredient is the set of all λ-Lipschitz continuous functions

$$\mathscr{M} = \{m \in \mathscr{F} \colon |m(x) - m(x')| \leq \lambda \|x - x'\| \text{ for all } x, x' \in \mathbb{X}\},$$

the projection of which onto $\mathsf{X} \cup \{x_0\}$ is

$$\mathbb{M}^+(x_0) = \{(\mathsf{m}_0, \mathbf{m}^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{k+1} : |\mathsf{m}_i - \mathsf{m}_j| \le \lambda \|x_i - x_j\| \text{ for all } i, j = 0, 1, 2, \dots, k\},$$

where $\mathbf{m} = (\mathsf{m}_1, \mathsf{m}_2, \dots, \mathsf{m}_k)^\mathsf{T} \in \mathbb{R}^k$ is a generic vector.

The upper plausible bound on the expected average waiting time of a given setting x_0, denoted by $\mathcal{I}^+_\mathsf{n}(x_0)$, is the optimal value of the following optimization problem:

$$\begin{aligned} \max_{\mathsf{m}_0,\ \mathbf{m}} \quad & \mathsf{m}_0 \\ \text{s.t.} \quad & |\mathsf{m}_i - \mathsf{m}_j| \le \lambda \|x_i - x_j\| \text{ for all } i, j = 0, 1, 2, \dots, k \\ & L_{i2} \le \mathsf{m}_i \le U_{i2} \text{ for all } i = 1, 2, \dots, k. \end{aligned} \tag{3}$$

The corresponding lower plausible bound, denoted by $\mathcal{I}^-_\mathsf{n}(x_0)$, is the optimal value when the objective in (3) is changed from maximizing m_0 to minimizing m_0. Optimization problem (3) and its minimization counterpart are linear programs whose optimal values have closed forms. Specifically, for any setting x_0,

$$\mathcal{I}^+_\mathsf{n}(x_0) = \min_{i=1,2,\dots,k} \{U_{i2} + \lambda \|x_0 - x_i\|\} \text{ and } \mathcal{I}^-_\mathsf{n}(x_0) = \max_{i=1,2,\dots,k} \{L_{i2} - \lambda \|x_0 - x_i\|\}.$$

6.2 Screening for Pareto Optimality with Convexity and Directional Lipschitz Continuity

We mentioned in Section 2 the decision maker's interest in settings having low expected total cost and low expected average waiting time. We can therefore regard a setting as being acceptable if it is *weakly efficient*, meaning that it is not strictly dominated by any other setting, where a setting x_0 is said to be strictly dominated by another setting x if $\theta(x_0) > \theta(x)$, where the inequality applies component-wise. The set of functions for which x_0 is weakly efficient is $\mathscr{A}(x_0) = \{m \in \mathscr{F}^d : m(x_0) \not> m(x) \text{ for all } x \in \mathbb{X}\}$, and the set of acceptable settings is $\mathbb{A} = \{x_0 \in \mathbb{X} : \theta(x_0) \not> \theta(x) \text{ for all } x \in \mathbb{X}\}$. Plausible inference was applied to construct a confidence region for $\mathbb{A}$, as shown in Figure 4b.

We know that the first objective (expected total cost) is convex while the second (expected average waiting time) is (λ_1, λ_2) directional Lipschitz continuous. (Here, we choose to work with directional Lipschitz continuity instead of Lipschitz continuity because it provides more powerful information and affords greater tractability in the resulting optimization problems.) The set of all paired objective functions for which the first is convex, the second is directional Lipschitz continuous, and for which the setting x_0 is weakly efficient is

$$\begin{aligned} \mathscr{M} \cap \mathscr{A}(x_0) = \{m \equiv (m_1, m_2) \in \mathscr{F}^2 : \text{ there exists } g \in \mathscr{F}^2 \text{ such that} & \\ m_1(x) - m_1(x') \le g(x)^\mathsf{T}(x - x') \text{ for all } x, x' \in \mathbb{X}, & \\ |m_2(x) - m_2(x')| \le \sum_{\ell=1}^{2} \lambda_\ell |x_\ell - x'_\ell| \text{ for all } x, x' \in \mathbb{X}, & \\ m(x_0) \not> m(x) \text{ for all } x \in \mathbb{X}\}, & \end{aligned}$$

where $g(x)$ represents a subgradient of m_1 at x. We do not have an exact form for $\mathbb{M}(x_0) = \text{proj}_\mathsf{X}(\mathscr{M} \cap \mathscr{A}(x_0))$, but a relaxation is easily derivable:

$$\begin{aligned}\mathbb{M}'(x_0) &= \mathrm{proj}_{\mathsf{X}}(\mathscr{M}) \cap \mathrm{proj}_{\mathsf{X}}(\mathscr{A}(x_0)) \\ &= \{\mathbf{M} \in \mathbb{R}^{k\times 2} \colon \text{there exist } \mathsf{m}_0 \in \mathbb{R}^2 \text{ and } \mathbf{G} = (\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_k)^\intercal \in \mathbb{R}^{(k+1)\times 2} \text{ such that} \\ &\qquad \mathsf{m}_{i1} - \mathsf{m}_{j1} \le \mathbf{g}_i^\intercal (x_i - x_j) \text{ for all } i,j = 0,1,\ldots,k, \\ &\qquad |\mathsf{m}_{i2} - \mathsf{m}_{j2}| \le \sum_{\ell=1}^{2} \lambda_\ell |x_{i\ell} - x_{j\ell}| \text{ for all } i,j = 0,1,\ldots,k, \\ &\qquad \mathbf{m}_0 \not> \mathbf{m}_i \text{ for all } i = 1,2,\ldots,k\}.\end{aligned}$$

Replacing $\mathbb{M}(x_0)$ by its relaxation $\mathbb{M}'(x_0)$ preserves the confidence guarantees mentioned in Section 5.3. Inference about the plausible acceptability of a given setting x_0 is then made by checking whether $\mathbb{M}'(x_0) \cap \mathbb{C} = \emptyset$ and returning x_0 if the intersection is non-empty, which entails solving a mixed-integer linear program.

6.3 Uncertainty Quantification for Pareto Front with Convexity and Lipschitz Continuity

We continue to work with the definition of acceptability and functional properties assumed in Section 6.2. Plausible inference was applied to construct a confidence region for the Pareto front (shown in Figure 5), which is the image of the set of acceptable settings, $\theta(\mathbb{A})$.

Similar to the previous case, we derive a relaxation of $\mathbb{M}_A^{+}(x_0) = \mathrm{proj}_{\mathsf{X}\cup\{x_0\}}(\mathscr{M} \cap \mathscr{A}(x_0))$ by switching the order of projection and intersection:

$$\begin{aligned}\mathbb{M}_A'^{+}(x_0) &= \mathrm{proj}_{\mathsf{X}\cup\{x_0\}}(\mathscr{M}) \cap \mathrm{proj}_{\mathsf{X}\cup\{x_0\}}(\mathscr{A}(x_0)) \\ &= \{(\mathbf{m}_0, \mathbf{M}) \in \mathbb{R}^{(k+1)\times 2} \colon \text{there exists } \mathbf{G} \in \mathbb{R}^{(k+1)\times 2} \text{ such that} \\ &\qquad \mathsf{m}_{i1} - \mathsf{m}_{j1} \le \mathbf{g}_i{}^\intercal (x_i - x_j) \text{ for all } i,j = 0,1,\ldots,k, \\ &\qquad |\mathsf{m}_{i2} - \mathsf{m}_{j2}| \le \sum_{\ell=1}^{2} \lambda_\ell |x_{i\ell} - x_{j\ell}| \text{ for all } i,j = 0,1,\ldots,k, \\ &\qquad \mathbf{m}_0 \not> \mathbf{m}_i \text{ for all } i = 1,2,\ldots,k\}.\end{aligned}$$

The objective space typically is a continuous region, which is $\mathbb{R}^2$ in the queueing network example. Rather than checking whether each individual objective vector belongs to $\mathcal{P}_{\mathsf{n}}^{\mathrm{A}}$, one can instead check whether an arbitrary region $\mathcal{P}_q \subseteq \mathbb{R}^d$ (so-called pixel) overlaps with the confidence region for the Pareto front, $\mathcal{P}_n^{\mathrm{A}}$, by solving the optimization problem

$$\begin{aligned}\min_{x_0,\, \mathbf{m}_0,\, \mathbf{M},\, \mathbf{G}} \quad & 0 \\ \text{s.t.} \quad & \mathsf{m}_{i1} - \mathsf{m}_{j1} \le \mathbf{g}_i{}^\intercal (x_i - x_j) \text{ for all } i,j = 0,1,\ldots,k, \\ & |\mathsf{m}_{i2} - \mathsf{m}_{j2}| \le \sum_{\ell=1}^{2} \lambda_\ell |x_{i\ell} - x_{j\ell}| \text{ for all } i,j = 0,1,\ldots,k, \\ & \mathbf{m}_0 \not> \mathbf{m}_i \text{ for all } i = 1,2,\ldots,k, \\ & L_{ir} \le \mathsf{m}_i \le U_{ir} \text{ for all } i = 1,2,\ldots,k \text{ and } r = 1,2, \\ & x_0 \in \mathbb{X},\ \mathbf{m}_0 \in \mathcal{P}_q.\end{aligned} \tag{4}$$

By covering the objective space with a finite number of pixels, we construct an outer approximation of $\mathcal{P}_n^{\mathrm{A}}$ by collecting all pixels that intersect it:

$$\hat{\mathcal{P}}_n^{\mathrm{A}} \equiv \bigcup_{q\colon \mathcal{P}_n^{\mathrm{A}} \cap \mathcal{P}_q \neq \emptyset} \mathcal{P}_q.$$

By controlling the size of each pixel, one can approximate $\mathcal{P}_n^{\mathrm{A}}$ by $\hat{\mathcal{P}}_n^{\mathrm{A}}$ at any resolution.

7 ADVANCED RECIPES

This section covers topics currently on the cutting edge of plausible inference.

7.1 Detecting Functional Properties

Plausible inference exploits functional properties of the objectives to extend inference beyond the particular settings of x that were simulated, which begs the question, "From where do the structural properties come?" Usually the answer will be "from an intelligent analyst," but in the case of Lipschitz continuity PI reasoning can provide an empirical response.

If the setting space $\mathbb{X}$ is bounded, then it is hard to imagine a realistic simulation whose performance functions are not Lipschitz; the central problem is knowing the Lipschitz constant $\lambda^\star$. A naïve method to estimate $\lambda^\star$ is to use $\max_{i \neq j}(\widehat{\theta}(x_i) - \widehat{\theta}(x_j))/\|x_i - x_j\|_2$ as the estimator, where $\widehat{\theta}(x_i)$ is the sample estimate of $\theta(x_i)$. However, as more design points in a continuous space $\mathbb{X}$ are chosen, $\lim_{k \to \infty} \min_{i \neq j} \|x_i - x_j\|_2 \to 0$; this leads to a badly behaved and inconsistent estimator whose variance increases to infinity as $k \to \infty$.

For a given value of $\overline{\lambda}$, we consider the set of all $\overline{\lambda}$-Lipschitz continuous functions,

$$\mathscr{M}_{\overline{\lambda}} = \{m \in \mathscr{F} : |m(x) - m(x')| \leq \overline{\lambda}\|x - x'\|_2 \text{ for all } x, x' \in \mathbb{X}\},$$

and its corresponding projection onto X, $\mathbb{M}_{\overline{\lambda}} = \mathrm{proj}_{\mathsf{X}}(\mathscr{M}_{\overline{\lambda}})$.

To empirically estimate the true Lipschitz constant, $\lambda^\star$, we consider all values of $\overline{\lambda}$ whose corresponding projection lies within a confidence region. However, because the empirical estimate will then later be used as a functional property for the Mise en Place step, it may not be advisable to use the same confidence region $\mathbb{C}$ when estimating $\lambda^\star$ and when conducting inference from the PI cookbook. Instead, depending on the problem and user, a less, or possibly more, conservative confidence region may be desired to estimate $\lambda^\star$. Therefore, a second confidence region, $\mathbb{C}_{1-\beta}$, is separately chosen for the estimation of $\lambda^\star$. Much like how $\mathbb{C}$ is assumed to achieve either finite-sample or asymptotic $100(1-\alpha)\%$ coverage, $\mathbb{C}_{1-\beta}$ is assumed to achieve either finite-sample or asymptotic $100(1-\beta)\%$ coverage. The same simulation outputs used to construct $\mathbb{C}$ can be used to construct $\mathbb{C}_{1-\beta}$.

Let $\beta \in [0,1]$ and define $\widehat{\lambda}_{1-\beta}$, the $1-\beta$ plausible Lipschitz estimator, as

$$\widehat{\lambda}_{1-\beta} = \inf\{\overline{\lambda} : \mathbb{M}_{\overline{\lambda}} \cap \mathbb{C}_{1-\beta} \neq \emptyset\}.$$

In words, $\widehat{\lambda}_{1-\beta}$, is the smallest $\overline{\lambda}$ for which there exists a $\overline{\lambda}$-Lipschitz continuous function lying within the confidence region.

Theorem 1 proves that, under some mild assumptions, $\widehat{\lambda}_{1-\beta}$ is a $100(1-\beta)\%$ lower confidence bound on the true Lipschitz constant of $\theta(\cdot)$. Because $\lambda^\star$ is unknown, for $x \in \mathbb{X}$ that are not in X, the values of $\theta(x)$ at these unobserved xs can be arbitrarily large and small. Since these values are not considered by the projection set $M_{\overline{\lambda}}$, we can only obtain a *lower* confidence bound.

Theorem 1 Suppose $\theta(\cdot)$ is a Lipschitz continuous function with constant $\lambda^\star$ and $\mathbb{C}_{1-\beta}$ achieves finite-sample $100(1-\beta)\%$ coverage. Then $\mathrm{P}\left(\widehat{\lambda}_{1-\beta} > \lambda^\star\right) \leq \beta$.

Keslin et al. (2024) describe using $\widehat{\lambda}_{1-\beta}$ as a plug-in Lipschitz parameter for sequential experiment design and plausible inference as described below.

7.2 Adaptive Design

To invoke any form of PI, simulation of an initial set of design points, $\mathsf{X}_k = \{x_1, x_2, \ldots, x_k\}$, is required. The placement of these design points, how many replications to simulate at each, and the best choice of additional design points and replications to adapt to the observed results are open research questions.

The flexibility of PI makes it naturally compatible with adaptive sampling strategies. Because the inference relies solely on the construction of a valid confidence region, the user may sequentially expand the experimental set by simulating additional settings, or increase the number of replications at existing ones. There are tradeoffs between simulating fewer replications at more settings and simulating more replications at fewer settings, and the framework accommodates both without requiring changes to its inferential machinery. Despite this flexibility, adaptive design remains one of the least studied aspects of PI.

For the initial design, we expect that space-filling designs, such as Latin Hypercube Samples, will typically be superior to designs that are tailored for, or even "optimal," under very strong performance-model assumptions; e.g., Central Composite Designs for fitting quadratic regression functions. This is because having simulated settings in close proximity to unsimulated settings about which PI is desired typically leads to sharper inference.

Keslin et al. (2024) report early work on sequential, adaptive design for feasibility checking and optimization. They consider acquisition functions that choose the next setting, or batch of settings, to simulate based on plausible intervals on unsimulated settings' objectives, and illustrate the effectiveness of this approach on one-dimensional newsvendor and two-dimensional (s,S) inventory problems using the plausible Lipschitz constant estimate $\widehat{\lambda}_{1-\beta}$ described in the previous section.

7.3 Utilizing Gradients

For some stochastic simulation models, it is possible to obtain direct gradient estimators for the objective(s) (Fu 2015). In other words, when simulating a given setting x, the simulation model returns an estimate of $\theta(x)$ *and* an estimate of $\nabla_x\theta(x)$, where $\nabla_x\theta(x)$ is the gradient (in the multi-objective case, the Jacobian) of θ with respect to x. Methods of PI can be modified to exploit the supplementary information that stochastic gradient estimates provide about the local geometry of θ at simulated settings. For certain functional properties, such as convexity, gradient information can be especially powerful because it says something about the *global* behavior of θ. Gradient estimates can be incorporated into the PI framework by expanding the projected sets to also include the gradients at X and by working with a confidence region that captures the uncertainty about both $\theta(x)$ and $\nabla_x\theta(x)$. These ideas are more fully developed in Eckman et al. (2021), along with examples showing the additional screening power stochastic gradient estimates can provide when searching for near-optimal settings to problems with convex objectives.

8 CONCLUSION

Plausible inference offers a unifying and versatile framework for simulation-based inference capable of addressing a wide range of inference tasks. Its strength lies in the ability to mix and match ingredients—functional structure, acceptability, experiment results—to achieve screening, uncertainty quantification, and more. Methods of PI essentially trade simulation for optimization to make principled, on-demand statistical inference. PI is well suited for problems where simulation effort is precious, the setting space is vast, and the analyst can, through closer examination, gain deeper insight into the problem's structure that translates to sharper inference. The effectiveness of PI depends on the ability of the extracted functional information to restrict objective values in unexplored regions of the setting space and any relaxations used when deriving the necessary projections. Another limitation is the computational tractability of the resulting mathematical programs, though in our experience these problems are in many cases convex. We are developing a Julia package that implements the methods and supports tailoring them to users' specific needs.

9 ACKNOWLEDGMENTS

Zhao and Eckman's work was partially supported by NSF Grant No. CMMI-2206972. Nelson and Keslin's work was partially supported by NSF Grant No. CMMI-2206973. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Eckman, D. J., M. Plumlee, and B. L. Nelson. 2020. “Revisiting Subset Selection”. In *2020 Winter Simulation Conference (WSC)*, 2972–2983. IEEE.

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. IEEE.

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.

Fu, M. 2015. “Stochastic Gradient Estimation”. In *Handbook of Simulation Optimization*, edited by M. Fu, Chapter 5, 105–147. New York, New York: Springer.

Keslin, G., D. W. Apley, and B. L. Nelson. 2024. “Plausible Inference with a Plausible Lipschitz Constant”. In *2024 Winter Simulation Conference (WSC)*, 3554–3565. IEEE.

Patsis, N. T., C.-H. Chen, and M. E. Larson. 1997. “SIMD Parallel Discrete-Event Dynamic System Simulation”. *IEEE Transactions on Control Systems Technology* 5(1):30–41.

Plumlee, M. and B. L. Nelson. 2018. “Plausible Optima”. In *2018 Winter Simulation Conference (WSC)*, 1981–1992. IEEE.

Qiao, T., D. J. Eckman, and B. L. Nelson. 2025. “Plausible Intervals: Global Inference from Limited Simulation of Structured Problems”. Technical report, Department of Industrial and Systems Engineering, Texas A&M University, College Station, Texas.

Tobias, A. and M. K. Brunnermeier. 2016. “CoVaR”. *The American Economic Review* 106(7):1705–1741.

Zhao, J. and D. J. Eckman. 2025. “One-Shot Screening of Simulated Systems for Acceptability”. Technical report, Department of Industrial and Systems Engineering, Texas A&M University, College Station, Texas.

Zhao, J., J. Gatica, and D. J. Eckman. 2023. “Screening Simulated Systems for Optimization”. In *2023 Winter Simulation Conference (WSC)*, 1–15. IEEE.

AUTHOR BIOGRAPHIES

JINBO ZHAO is a Ph.D. student in the Wm Michael Barnes ’64 Department of Industrial and Systems Engineering at Texas A&M University. His research interest is simulation optimization featuring multiple responses. His e-mail address is jinbozhao@tamu.edu.

GREGORY KESLIN is an assistant professor in the Martin Tuchman School of Management at New Jersey Institute of Technology. His research interests are simulation optimization, plausible inference and statistics. His e-mail address is gnk@njit.edu.

DAVID J. ECKMAN is an Assistant Professor in the Wm Michael Barnes ’64 Department of Industrial and Systems Engineering at Texas A&M University. His research interests deal with optimization and output analysis for stochastic simulation models. He is a co-creator of SimOpt, a testbed of simulation optimization problems and solvers. His e-mail address is eckman@tamu.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.