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.

NEURAL NETWORK-BASED METHODS FOR CONTINUOUS SIMULATION OPTIMIZATION PROBLEMS WITH COVARIATES

Yize Hao[1], and Guangxin Jiang[1]

[1]School of Management, Harbin Institute of Technology, Harbin, Heilongjiang, CHINA

ABSTRACT

Simulation-based methods for real-time decision-making have attracted increasing research attention, and such problems are usually formulated as the simulation optimization problem with covariates. There are generally two methods to address this problem. The first builds a relationship between the objective function and the covariates, allowing a solver to quickly find the solution. The second directly builds a relationship between the optimal solution and the covariates. In this paper, we focus on continuous simulation optimization with covariates and investigate neural network-based implementations of both methods. We demonstrate that even when the objective function is continuous, the optimal solution may exhibit discontinuities with respect to the covariates, limiting the applicability of the optimal solution method. In contrast, the objective function method remains effective and broadly applicable. We further establish stability conditions under which the optimal solution method is valid. Numerical experiments are conducted to support our theoretical findings.

1 INTRODUCTION

Simulation optimization is a powerful tool for analyzing and optimizing complex systems, relying on simulation experiments to evaluate performance (Fu 2015; Fan et al. 2024). Since these experiments often produce noisy outputs and are computationally expensive, traditional methods are usually designed to address offline decision-making problem, where the input parameters of the simulation models are known in advance and sufficient time is available to conduct extensive experiments before making decisions. However, in practice, many problems require real-time decisions, where actions must be taken almost immediately. For instance, in healthcare, personalized treatment plans must adapt instantly to patient-specific factors such as age, sex, and baseline disease severity (Bertsimas et al. 2019; Shen et al. 2021). In finance, portfolio risk must be assessed and classified within seconds as risk factors fluctuate (Jiang et al. 2020; Jiang et al. 2024). These scenarios highlight the growing need for fast, high-quality decision-making, motivating the development of online simulation optimization techniques.

Recently, Hong and Jiang (2019) proposed the offline-simulation-online-application (OSOA) framework to address the challenge of using time-intensive simulations for real-time decisions. The core idea is to leverage extensive offline simulations to train predictive models, which are then deployed online once the input parameters of the simulation models, also called covariates, are observed. This OSOA framework has successful applications in, for example, real-time risk monitoring (Jiang et al. 2020), real-time derivative pricing and hedging (Jiang et al. 2024), and online ranking and selection problem (Shen et al. 2021; Keslin et al. 2024).

In addition to online ranking and selection problems, other types of online simulation optimization, also known as simulation optimization with covariates, have also attracted increasing attention recently (e.g., Zhang et al. 2021; Luo et al. 2024). Within the OSOA framework, two main types of methods have been developed to address this problem. The first approach, known as the objective function method, employs machine learning to model the objective function as a function of covariates. Once the covariates are observed, this model provides an analytical approximation of the objective function, and standard

optimization solvers can be applied to obtain the optimal solution. The development of predictive modeling methods has been extensively studied in both the optimization and simulation literature. For example, Hannah et al. (2010) proposed a method that constructs predictive models by weighting the stochastic objective function using observed data. They assumed that the objective function is convex, which allows for fast optimization using solvers. Bertsimas and McCord (2019) used methods such as k-nearest neighbors, CART, and random forests to construct weight functions and estimate the conditional expectation of the cost function given covariates through weighted averaging. Kannan et al. (2025) studied a sample average approximation framework with covariate information to approximate the objective function. The framework is flexible and supports parametric, nonparametric, and semiparametric regression techniques.

In contrast to constructing a predictive objective function model, the second approach to address simulation optimization with covariates, known as the optimal solution method, directly learns a mapping from covariates to optimal solutions, allowing immediate prediction of the solution for any given covariate. For example, Ban and Rudin (2019) investigated the newsvendor problem with stochastic demand, where the demand distribution is assumed to be influenced by observable features. As the optimization objective depends on the given observed features, machine learning techniques could be employed to construct a mapping relationship from covariates to the optimal order quantity. Bertsimas and Kallus (2020) and Bertsimas and Koduri (2022) examined both the objective function method and the optimal solution method, with the primary difference being that the former employs local machine learning methods, while the latter utilizes global methods to construct predictive models. These studies mainly originate from the optimization literature, focusing on problems where the cost function depends on uncertain parameters influenced by covariates. The goal is to optimize the conditional expectation of the cost function given the covariates. In practice, however, the joint distribution of the uncertainties and covariates is unknown, and only limited observational data are available for estimating the optimal decision. These methods are fully data-driven.

In this paper, we focus on real-time decision-making problems in the simulation context. Unlike data-driven settings, simulation-based decisions rely on stochastic models whose parameters are designed to reflect the physical characteristics of real systems. By selecting different design parameters, simulation samples associated with varying covariates can be generated to evaluate performance metrics of interest. These methods are thus model-driven. Moreover, the covariates in our setting can be deliberately designed to align with specific optimization objectives, further distinguishing this approach from data-driven methods. Nevertheless, the predictive modeling techniques developed in previous studies are general and contribute valuable tools for tackling simulation-based real-time decision problems.

We investigate two neural network-based methods for constructing predictive models. A key observation is that the optimal solution with respect to covariates can be discontinuous or unstable, making solution-based methods unreliable or inapplicable in certain cases. This phenomenon is reflected in real-world examples—for instance, an aircraft's control strategy may shift in response to environmental changes, and such transitions often involve instability that requires specialized handling (Kirst and Wang 2024). Similar challenges arise in autonomous driving (Song et al. 2024). Therefore, the applicability of optimal solution method depends on specific stability conditions of the performance function of the stochastic system. We propose a stability condition that helps ensure the validity of the neural network-based optimal solution method to some extent. In contrast, when the optimal solution varies abruptly with covariates, we recommend the neural network-based objective function method. Given sufficient simulation runs, training a neural network to map covariates to objective values can better exploit structural patterns in the data, leading to improved performance.

Our work is most closely related to Zhang et al. (2021) and Luo et al. (2024). Zhang et al. (2021) explored using neural networks in both methods: approximating the mapping from covariates to optimal solutions, and enhancing stochastic kriging models for fast evaluation of the objective function. However, their method does not guarantee the conditions under which the approximated optimal solutions are valid. Luo et al. (2024) focused on the optimal solution method, employing KNN and kernel methods to construct predictive models that map covariates to optimal decisions. Assuming convexity and smoothness of the

objective function, they ensure that the optimal solution is a smooth function of the covariates, which provides theoretical guarantees for their method. In contrast, our work addresses the problem under different assumptions. Notice that this paper treats the simulation model and neural network separately. The simulation model is responsible for generating training data, while the neural network uses this data for learning. There is no parameter or structural coupling between the two modules; their only connection lies in the transfer of data. Related but distinct work includes the study by Zheng et al. (2025) on arrival process modeling. In their proposed framework, the simulation model and neural network are coupled, which not only increases the difficulty of theoretical analysis but also adds complexity to practical computation. To address these issues, Zheng et al. (2025) proposed targeted solutions and provided corresponding theoretical guarantees.

The remainder of the paper is organized as follows. Section 2 presents the problem formulation and notation. Section 3 discusses the applicability conditions for the optimal solution method and summarizes practical procedures for implementing the neural network-based objective function and optimal solution methods. Section 4 presents numerical experiments, and Section 5 concludes with potential directions for future research.

2 PROBLEM SETTINGS

We focus on a stochastic optimization problem of the following form

$$\min_{\theta\in\Theta} f(\theta,x) = \mathbb{E}[F(\theta,x,\xi)], \tag{1}$$

where $\theta \in \Theta$ denotes the decision variable with $\Theta \subset \mathbb{R}^{d_\theta}$, $x \in X$ denotes the covariate with $X \subset \mathbb{R}^{d_x}$, and ξ represents the randomness, e.g., a stream of underlying random numbers in a stochastic simulation. Let $F(\theta,x,\xi)$ denote the performance function of the stochastic system, given x. In general, the covariate x may influence the random source ξ, which makes the analysis more complex. We leave such cases for future work. In this paper, we assume that the covariates do not affect the random source. For convenience, we refer to the optimization problem in (1) with covariate x as P_x.

Generally, we assume that the simulation outputs $F(\theta,x,\xi)$ can be obtained for specific values of x and θ. However, the analytical form of $f(\theta,x)$ is unknown. Our goal is to solve this problem in real time once the covariate x is observed. In this section, we present the formulations of two commonly used methods within the OSOA framework and provide a detailed analysis of both methods in the next section.

2.1 Objective Function Method

Suppose that $\hat{f}(\theta,x)$ is a predictive model used to approximate $f(\theta,x)$. To obtain $\hat{f}(\theta,x)$, we solve the following optimization problem

$$\min_{\hat{f}\in\mathscr{F}} \mathbb{E}\left[(\hat{f}(\theta,x) - f(\theta,x))^2\right], \tag{2}$$

where $\mathscr{F}$ denotes the function space over which $\hat{f}$ is optimized, and the mean squared error is used to measure the approximation quality. In practice, alternative metrics may also be considered to enhance performance. Note that the expectation in (2) is taken over both θ and x, meaning that they are sampled from their respective domains according to appropriate probability distributions. Let p_θ and p_x denote the sampling distributions of θ and x, respectively. The optimization problem (2) is a theoretical formulation. In practice, it is implemented by performing a large number of randomized simulation experiments. The OSOA procedure for this method is described below.

Offline phase: Select appropriate probability distributions p_θ and p_x, sample N design points of θ and x, respectively, resulting in N sample pairs $(\theta_i,x_i), i = 1,2,\ldots,N$. For each sample pair, run M_i simulation

experiments to obtain M_i samples of $F(\theta, x, \xi)$, then we can solve the empirical version of (2)

$$w^* = \arg\min_{w \in \mathscr{W}} \frac{1}{N} \sum_{i=1}^{N} \left(\hat{f}(\theta_i, x_i; w) - \frac{1}{M_i} \sum_{n=1}^{M_i} F(\theta_i, x_i, \xi_n) \right)^2. \tag{3}$$

Note that we use $\hat{f}(\cdot, \cdot; w)$ to represent a parameterized function implemented by a neural network, where $\mathscr{W}$ denotes the parameter space of the network. By updating w using standard machine learning training procedures, we aim to find the optimal $\hat{f}(\cdot, \cdot; w)$ that best approximates the target function in $\mathscr{F}$.

Online phase: When the observed value $\tilde{x}$ is obtained, since $\hat{f}(\cdot, \cdot, w^*)$ is in an analytical form, we can invoke an optimization solver, e.g., Adam (Kingma and Ba 2014) and SciPy (Virtanen et al. 2020), to solve the optimization problem and obtain the optimal decision $\hat{\theta}^*$, i.e.,

$$\hat{\theta}^* = \arg\min_{\theta \in \Theta} \hat{f}(\theta, \tilde{x}; w^*).$$

2.2 Optimal Solution Method

Compared to the objective function method, the optimal solution method has a simpler form. Suppose that we can obtain the optimal decision by solving the simulation optimization problem with sufficient accuracy using some standard methods such as stochastic approximation (SA) or sample average approximation (SAA), see Fu (2015). It is evident that the optimal solution depends on x, and under certain assumptions, the optimal solution can be regarded as a function of x (Luo et al. 2024). Let $\theta^*(x)$ represent the true mapping from x to θ^* that we aim to obtain. Our goal is to achieve the best possible approximation of this mapping, denoted by $h(x)$. We aim to solve an optimization problem of the following form

$$\min_{h \in \mathscr{H}} \mathbb{E}_{p_x} \left[(h(x) - \theta^*(x))^2 \right], \tag{4}$$

where $\mathscr{H}$ denotes the function space in which we optimize h, p_x denotes the sampling distribution of the covariates, then the process of the OSOA for this method is given below.

Offline phase: Select appropriate probability distribution p_x, sample N design points of x. Then running a number of simulation experiments and using standard method to solve the problems $\min_{\theta \in \Theta} \mathbb{E}[F(\theta, x_k, \xi)]$, $k = 1, 2, \ldots, N$, so we obtain N optimal solutions θ_k^*, $k = 1, 2, \ldots, N$. Then we can solve the empirical version of (4)

$$w^* = \arg\min_{w \in \mathscr{W}} \frac{1}{N} \sum_{k=1}^{N} \left(h(x_k; w) - \theta_k^* \right)^2. \tag{5}$$

Here, we also employ a parameterized predictive model to approximate the target function in $\mathscr{H}$.

Online phase: When the observed value $\tilde{x}$ is obtained, we can directly compute the optimal decision through the following equation

$$\hat{\theta}^* = h(\tilde{x}; w^*).$$

3 ANALYSIS OF OBJECTIVE FUNCTION AND OPTIMAL SOLUTION METHODS

In this section, we first present an illustrative example that represents a class of continuous simulation optimization problems with covariates. In these problems, the objective function is continuous with respect to the covariates, whereas the optimal solution may exhibit discontinuities. We argue that, although neural networks possess universal function approximation capabilities and are effective for constructing predictive models, the optimal solution method is not well suited for such problems. Moreover, we highlight that applying the optimal solution method requires the performance function to satisfy certain assumptions. In contrast, we contend that the objective function method, implemented with neural networks, offers a more effective approach for addressing this class of problems. We also propose a simple and practical implementation of this method.

3.1 An Illustrative Example

Consider a performance function of the following form

$$F(\theta,x,\xi) = \min\left\{\sum_{i=1}^{d}(\theta_i - x_i)^2 + x_{d+1}, \sum_{i=1}^{d}(\theta_i + x_i)^2\right\} + \xi, \tag{6}$$

where the model consists of a deterministic component and an additive stochastic noise term. This implies that the stochastic noise is independent of the covariates. We assume that the stochastic noise has a zero expectation, which allows us to eliminate the randomness by taking the expectation. It is easy to obtain

$$f(\theta,x) = \mathbb{E}[F(\theta,x,\xi)] = \min\left\{\sum_{i=1}^{d}(\theta_i - x_i)^2 + x_{d+1}, \sum_{i=1}^{d}(\theta_i + x_i)^2\right\}.$$

Then we solve the following optimization problem

$$\min_{\theta}\left\{\min\left\{\sum_{i=1}^{d}(\theta_i - x_i)^2 + x_{d+1}, \sum_{i=1}^{d}(\theta_i + x_i)^2\right\}\right\}. \tag{7}$$

Notice that the optimal solution is influenced by the $(d+1)$th component of the covariate and undergoes a sudden change near $x_{d+1} = 0$. When $x_{d+1} > 0$, $\theta_i^* = -x_i$ whereas when $x_{d+1} < 0$, $\theta_i^* = x_i$. If the first d-dimensional components of x are large, the variation of the covariate near $x_{d+1} = 0$ leads to a significant abrupt change in the optimal solution.

The study most relevant to our problem is Zhang et al. (2021), in which neural networks were used to approximate the mapping from covariates to optimal solutions. However, they did not verify the applicability conditions of the optimal solution method. Although neural networks possess powerful function approximation capabilities, accurately approximating in regions with abrupt changes may require a large number of sample points. For complex systems in real-world applications, it is often difficult to detect the existence of such discontinuous regions, making it infeasible to design effective covariate sampling strategies tailored to these cases. Under such circumstances, employing the optimal solution method becomes impractical. This limitation is demonstrated in the numerical experiments presented in Section 4. Furthermore, we provide a set of assumptions under which the mapping from covariates to optimal solutions is guaranteed to be stable and free of abrupt changes. These conditions, to some extent, establish the applicability of the optimal solution method.

3.2 Stability Conditions for Optimal Solution with Respect to Covariates

We derive the applicability of the optimal solution method based on the perturbation analysis of optimal solutions (Bonnans and Shapiro 2000). We first introduce some notations and assumptions for the problem (1). Let $v^* = \min_{\theta\in\Theta}\mathbb{E}[F(\theta,x,\xi)]$, $\bar{\theta}$ is said to be an ε-optimal solution of (1) if $\mathbb{E}[F(\bar{\theta},x,\xi)] \le v^* + \varepsilon$. Note that both v^* and $\bar{\theta}$ may depend on the covariate x. Let $dist(\theta,S) = \inf_{s\in S}||\theta - s||$. Recall that we refer to the optimization problem in (1) with covariate x as P_x. Suppose that problem P_{x_0} has a non-empty set S_0 of optimal solutions for given x_0. In addition, it is said that the second-order growth condition holds at S_0 if there exist a neighborhood $\mathcal{N}$ of S_0 and a constant $C > 0$ such that $\mathbb{E}[F(\theta,x_0,\xi)] \ge \mathbb{E}[F(S_0,x_0,\xi)] + C[dist(\theta,S_0)]^2$. Then we provide the following assumptions for $F(\theta,x,\xi)$.

Assumption 1 For all $\theta \in \Theta$, $F(\theta,x,\xi)$ is a.s. continuously differentiable in θ, and for all sufficiently small $\|\Delta\theta\|$, it holds that $|F(\theta+\Delta\theta,x,\xi) - F(\theta,x,\xi)| \le g(\xi)$ with $\mathbb{E}[g(\xi)] < \infty$. The family of random variables

$$\left\{\frac{F(\theta+h,x,\xi) - F(\theta,x,\xi)}{||h||} : 0 < ||h|| < \delta\right\}$$

is uniformly integrable for some $\delta > 0$.

Assumption 2 $\forall \theta \in \Theta \cap \mathscr{N}$, $F(\theta, x_0, \xi) \geq F(S_0, x_0, \xi) + c(\xi)[dist(\theta, S_0)]^2$ *w.p.*1 for $0 < \mathbb{E}[c(\xi)] < \infty$.

Assumption 3 $\nabla_\theta F(\theta, x, \xi)$ is Lipschitz continuous *w.p.*1 in a neighborhood of x_0 with the corresponding Lipschitz constant $L(\xi)$ independent of $\theta \in \mathscr{N}$ and $\mathbb{E}[L(\xi)] < \infty$.

Assumption 4 Let $\bar{\theta}$ be the ε-optimal solution of P_x, assume that $\varepsilon = O(||x - x_0||^2)$.

Next, we need the following lemma given by Bonnans and Shapiro (2000).

Lemma 1 (Bonnans and Shapiro 2000, Proposition 4.32) Suppose that the second-order growth condition holds at S_0 for P_{x_0} and $\mathbb{E}[F(\theta, x, \xi)] - \mathbb{E}[F(\theta, x_0, \xi)]$ is Lipschitz continuous modulus κ on N. Let $\bar{\theta}$ be an ε-optimal solution of P_x, then

$$dist(\bar{\theta}, S_0) \leq C^{-1}\kappa + C^{-\frac{1}{2}}\varepsilon^{\frac{1}{2}}. \tag{8}$$

According to above lemma, we can obtain the following result for the stability conditions for the optimal solution method.

Proposition 1 Let $F(\theta, x, \xi)$ be the performance function of the stochastic system. If Assumptions 1-4 holds, then $dist(\bar{\theta}, S_0) = O(||x - x_0||)$.

Proof. According to Assumption 1, using uniform integrability and the dominated convergence theorem, $\mathbb{E}[F(\theta, x, \xi)]$ is continuously differentiable in Θ and $\mathbb{E}[\nabla_\theta F(\theta, x, \xi)] = \nabla_\theta \mathbb{E}[F(\theta, x, \xi)]$. For problem (1) and given $x = x_0$, P_{x_0} has a nonempty set S_0 of optimal solutions. Consider a neighborhood $\mathscr{N}$ of S_0, continuously differentiable implies $\mathbb{E}[F(\theta, x, \xi)] - \mathbb{E}[F(\theta, x_0, \xi)]$ is Lipschitz continuous modulus κ on $\mathscr{N}$. By taking the expectation of the inequality in Assumption 2, we have $\mathbb{E}[F(\theta, x_0, \xi)] \geq \mathbb{E}[F(S_0, x_0, \xi)] + C[dist(\theta, S_0)]^2$ for some $C > 0$. Then by lemma 1, we have (8). By mean value theorem, Lipschitz modulus in (8) can be estimated as

$$\begin{aligned}
\kappa &\leq \sup_{\theta \in N} ||\nabla_\theta \mathbb{E}[F(\theta, x, \xi)] - \nabla_\theta \mathbb{E}[F(\theta, x_0, \xi)]|| \\
&= \sup_{\theta \in N} ||\mathbb{E}[\nabla_\theta F(\theta, x, \xi) - \nabla_\theta F(\theta, x_0, \xi)]|| \\
&\leq \mathbb{E}[L(\xi)||x - x_0||] \qquad (9) \\
&= O(||x - x_0||). \qquad (10)
\end{aligned}$$

The inequality (9) holds by using the Lipschitz property of the assumption on $\nabla_\theta F(\theta, x, \xi)$ and the integrable Lipschitz constant $L(\xi)$ independent of $\theta \in N$ under Assumption 3. Finally, according to Assumption 4, (8) and (10), we have $dist(\bar{\theta}, S_0) = O(||x - x_0||)$, which completes the proof. □

Proposition 1 indicates that when the performance function $F(\theta, x, \xi)$ possesses the properties described in Assumptions 1-4, the optimal solution of problem (1) is Lipschitz stable with respect to variations in the covariates, with the deviation between optimal solutions being of liner order. Furthermore, Proposition 1 is a sufficiently general result that, in theory, allows for the existence of multiple optimal solutions. In fact, when the optimal solution is unique, Proposition 1 implies that the optimal solution is a Lipschitz continuous function of the covariates, which ensures the feasibility of the neural network-based optimal solution method. In the more general case where multiple optimal solutions may exist for a given covariate, one can simply select a single optimal solution to construct a dataset mapping covariates to optimal solutions, which can then be used to train a neural network predictive model.

Compared to the assumptions made in Luo et al. (2024), Assumption 1 imposes a weaker smoothness assumption on the performance function. Furthermore, in contrast to the strong convexity assumption, the second-order growth condition does not require explicit gradient information. These differences make our assumptions more realistic and easier to satisfy in practice, thereby rendering the neural network-based optimal solution method more broadly applicable.

3.3 Summarized Implementation Procedures

Due to the complexity of stochastic systems and the lack of an explicit analytical form for the objective function, establishing a stable relationship between the optimal solution and covariates is challenging. Directly applying the optimal solution method thus carries high risk and may lead to decision errors. In contrast, the objective function method offers greater robustness and generality. Neural networks, with lower risks of model misspecification in complex predictive tasks, provide an effective means to implement this method. In the offline phase, a predictive model is trained using simulation input-output data; in the online phase, the trained network is used to optimize decision variables in real time with algorithms such as Adam (Kingma and Ba 2014). The procedure is summarized in Algorithm 1.

Algorithm 1 Objective Function Method Using Neural Networks Under OSOA Framework.

Input: Domains Θ, $\mathscr{X}$; sampling strategies p_θ, p_x over Θ and $\mathscr{X}$; number of design points N; number of simulation samples on each design point M_i; learning rate η; epochs of the optimization algorithm T.
Offline Phase:
Sample design points $\{(\theta_i, x_i)\}_{i=1}^N$ from $\Theta \times \mathscr{X}$ based on some sampling rules p_θ and p_x.
for $i = 1$ to N **do**
  Run M_i simulations to obtain $\{F(\theta_i, x_i, \xi_n)\}_{n=1}^{M_i}$.
end for
Train model $\hat{f}(\theta, x; w)$ by solving (3) using a standard method (e.g., Adam).
Online Phase:
Given an observation $\tilde{x} \in \mathscr{X}$, optimize $\theta \in \Theta$ by minimizing $\hat{f}(\theta, \tilde{x}; w^*)$.
  Initialize θ (e.g., randomly in Θ).
for $t = 1$ to T **do**
  Compute prediction $y_t = \hat{f}(\theta_t, \tilde{x}; w^*)$.
  Compute gradient $\nabla_\theta \hat{f}(\theta_t, \tilde{x}; w^*)$.
  Update θ_{t+1} using Adam optimizer with learning rate η.
  Project θ_{t+1} onto the feasible region Θ.
end for
Output: final $\hat{\theta}^* = \theta_{T+1}$ as the optimal decision.

In practice, to enhance accuracy and maintain real-time performance, multiple initial solutions and parallel computing can be employed. Compared to the neural network-assisted stochastic kriging model of Zhang et al. (2021), this implementation imposes fewer structural assumptions and offers greater flexibility, making it more practical for real-world applications.

On the other hand, when sufficient knowledge about the problem structure is available and an accurate mapping from covariates to the optimal solution can be established, the optimal solution method becomes a more direct and efficient approach. In this case, a predictive model mapping covariates to the optimal solution can be trained offline based on simulation data; during the online phase, the optimal decision for any given covariates can be computed almost instantaneously, making this approach suitable for applications requiring high real-time performance under limited computational resources. The detailed procedure for the optimal solution method is summarized in Algorithm 2.

4 NUMERICAL EXPERIMENTS

In this section, we use the performance function (6) from Section 3.1 to compare the two methods. The goal is to solve problem (7) via stochastic simulation. Both methods are tested under identical settings to evaluate their performance in predicting the optimal solution.

Algorithm 2 Optimal Solution Method Using Neural Networks Under OSOA Framework.

Input: Domain $\mathscr{X}$; sampling strategies p_x over $\mathscr{X}$; number of design points N; number of simulation samples on each design point M_k.
Offline Phase:
Sample design points $\{x_k\}_{k=1}^N$ from $\mathscr{X}$ using p_x.
for $k = 1$ to N **do**
  Run M_k simulations to obtain optimal solution θ_k^*.
end for
Train model $h(x;w)$ by solving (5) using a standard method (e.g., Adam).
Online Phase:
Given an observation $\tilde{x} \in \mathscr{X}$, compute $\hat{\theta}^* = h(\tilde{x};w^*)$.
Output: $\hat{\theta}^*$ as the optimal decision.

Specifically, let $d_\theta = d$ and $d_x = d+1$, then we conduct experiments for $d = 1,2,4,8,16,32$. We set the value range of each dimension to $[-1,1]$, which means $\Theta \times \mathscr{X} = [-1,1]^{2d+1}$. Let the random source ξ be a standard normal random variable (with mean 0 and variance 1). It should be noted that in our experiments, $F(\theta,x,\xi)$ is used only to generate simulation samples and obtain the true optimal solution, and no other information can be used when building the predictive model.

Offline Phase of Objective Function Method

As our goal is to build a predictive model from the covariates to the objective function, we first obtain training and validation data for training our predictive model. We adopt the Latin Hypercube Sampling (LHS) strategy to sample N design points $\{(\theta_i,x_i)\}_{i=1}^N$ from $\Theta \times \mathscr{X}$ as the input of predictive model. For each design point, we sample ξ for M_i times and we set $M_i = 10,000$. Then we compute $(1/M_i)\sum_{n=1}^{M_i} F(\theta_i,x_i,\xi_n)$ for each i as the output. Validation data are generated using the same strategy, and we adopt an early stopping mechanism to prevent overfitting using validation data. The number of validation samples is set to $N/5$.

Then we use a fully connected neural network with three hidden layers, each containing 512 neurons. The input layer has $2d+1$ dimensions, and the output layer has 1 dimension. Based on the above dataset and model architecture, we train our model by minimizing the MSE loss to obtain our predictive model $\hat{f}(\theta,x;w_{obj}^*)$, w_{obj}^* denotes the parameters of the neural network model of objective function method.

Offline Phase of Optimal Solution Method

We adopt the same sampling strategy as objective function method to generate N covariate design points $\{x_i\}_{i=1}^N$. For each x_i, we sample ξ for M_i times and then solve the corresponding optimal solution using the SAA method, thereby obtaining N optimal solutions $\{\theta_i^*\}_{i=1}^N$ for model training. We generate $N/5$ data pairs in a same way as validation dataset for overfitting prevention. We also use a fully connected neural network as our model architecture, with three hidden layers, each containing 512 neurons. The input layer has $d+1$ dimensions, and the output layer has d dimensions. Based on the above dataset and model architecture, we train our model by minimizing the MSE loss to obtain our predictive model $h(x;w_{sol}^*)$, w_{sol}^* denotes the parameters of the neural network model of optimal solution method.

To demonstrate the impact of both the number of samples and the problem dimensionality, we set $d = 1,2,4,8,16,32$ and $N = 1000,5000,10000,50000,100000$, resulting in 30 groups of experiments to compare the performance of the two methods in handling optimal solution discontinuities. The optimal solution of problem (7) with respect to the covariates is continuous or not, depending on the $(d+1)$-th component of the covariate. When the covariate x_{d+1} is close to 0, the optimal solution undergoes a sudden change. We have already obtained two predictive models, $\hat{f}(\theta,x;w_{obj}^*)$ and $h(x;w_{sol}^*)$. Therefore, we now need to test the performance of these two models near $x_{d+1} = 0$ in an online setting.

First, we need to generate online dataset. We still use LHS to generate N_{online} covariate samples and we set $N_{online} = 100$. To examine the performance in the discontinuous region, we set the sampling range of the $(d+1)$-th component x_{d+1} to [-0.001, 0.001], while keeping the other dimensions within their original domain [-1,1] to simulate the discontinuity of the optimal solution with respect to the covariates. As a result, we obtain N_{online} covariate observations $\tilde{x}_i$. Then, we can analytically compute the true optimal solutions corresponding to these N_{online} covariates to obtain the test dataset $\{(\tilde{x}_i, \theta_i^*)\}_{i=1}^{N_{online}}$.

Next, we need to use the two predictive models to quickly estimate the optimal solutions under the N_{online} observed covariate values.

Online Phase of Objective Function Method:

For N_{online} observed covariates $\tilde{x}_i$ in the online setting, we use the method described in Section 3.3 to solve the following N_{online} optimization problems:

$$\hat{\theta}_i^* = \arg\min_{\theta \in \Theta} \hat{f}(\theta, \tilde{x}_i; w_{obj}^*).$$

Therefore, we obtain N_{online} estimated optimal solutions $\hat{\theta}_i^*, i = 1, 2, \ldots, N_{online}$.

Online Phase of Optimal Solution Method:

The estimates of the N_{online} optimal solutions are directly computed using the following formula:

$$\hat{\theta}_i^* = h(\tilde{x}_i; w_{sol}^*), i = 1, 2, \ldots, N_{online}.$$

Evaluation:

We compute the average optimality gap Δ_f over N_{online} covariate test samples for each of the two methods to compare their performance. The evaluation metric is calculated using the following formula:

$$\Delta_f = \frac{1}{N_{online}} \sum_{i=1}^{N_{online}} f(\hat{\theta}_i^*, \tilde{x}_i) - f(\theta_i^*, \tilde{x}_i).$$

Our experimental results are presented in Figures 1–4, which show the variation of the optimality gap with respect to problem dimensions and sample sizes. Figure 1 illustrates the performance of the objective function method. When the dimensionality of the decision variables is low, the method can achieve good performance even with a small number of covariate sampling points. However, as the dimensionality increases, more sampling points are required to construct an accurate predictive model. Figure 2 shows the performance of the optimal solution method. The results indicate that all models exhibit significant decision errors, and model performance does not improve with an increased number of sampling points; instead, it demonstrates a certain degree of randomness. This suggests that although fully-connected neural networks are generally chosen when functional relationships are unknown, they fail when confronted with unforeseen abrupt changes. Therefore, forcibly using neural networks to fit unknown functional relationships without sufficient understanding of the problem's structure is an unstable approach.

Figure 3 compares the performance of the two methods when the number of covariate samples is set to $N = 100,000$. The results clearly demonstrate that the optimal solution method performs poorly when dealing with problems characterized by abrupt changes, while the objective function method continues to maintain good performance.

We also conduct a preliminary verification of the conditions proposed in Section 3.2 under the same experimental setup. In this experiment, we compare the performance of the optimal solution method under two different settings: one where the conditions are satisfied and one where they are unsatisfied. Specifically, in the satisfied setting, we deliberately avoid sampling test points in regions with abrupt changes by setting x_{d+1} in range [0.5,1], ensuring that the optimal solution is Lipschitz continuous with respect to the covariates. In the unsatisfied setting, test points are sampled in regions with abrupt changes, as in the previous setup. The comparison between these two settings is presented in Figure 4.

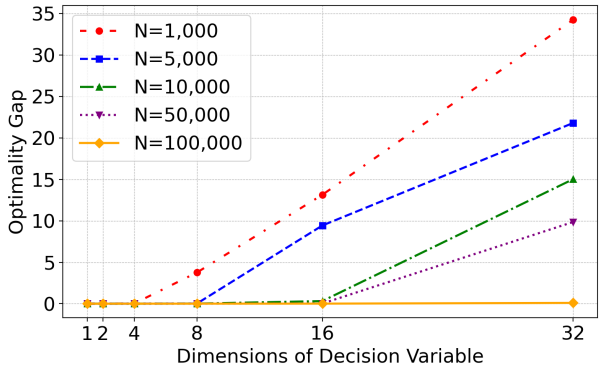

Figure 1: Objective function method performance.

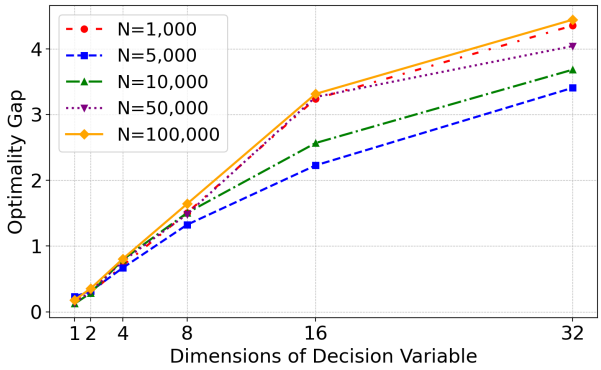

Figure 2: Optimal solution method performance.

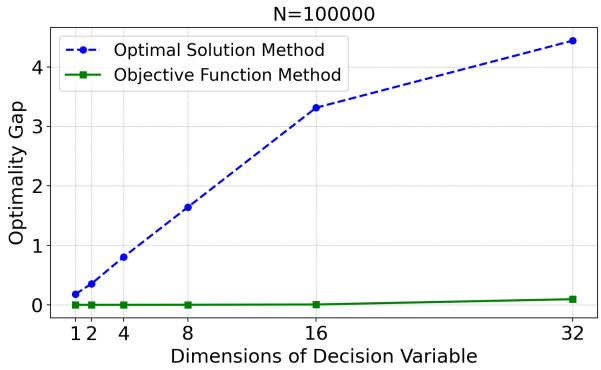

Figure 3: Comparison of two methods.

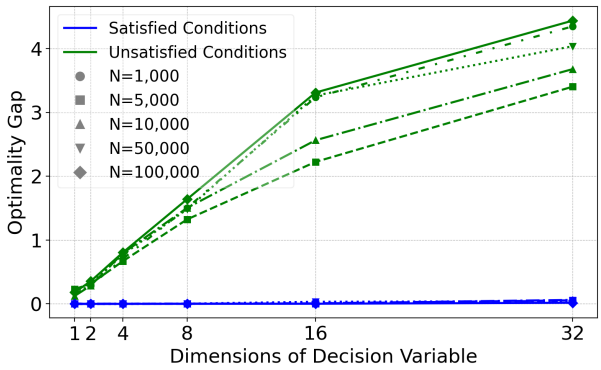

Figure 4: Satisfied vs. unsatisfied conditions.

The results demonstrate that when the optimal solution is a smooth function of the covariates, the optimal solution method performs remarkably well. It is worth noting that in this experiment, the optimal solution is in fact a linear function of the covariates, which satisfies a condition stronger than that assumed in Proposition 1. Therefore, this experiment only verifies the necessity of these conditions. A rigorous theoretical guarantee for neural network-based optimal solution method remains to be developed, which will be the focus of our future work.

To further demonstrate the generality of the objective function method, we also plot the predictive model approximated by the objective function method, along with the true objective function, in the one-dimensional case, as shown in Figure 5. The figure shows that the objective function-based method is capable of capturing the complex structural features of the true objective function based on simulation data. This ability is a key reason why the method remains effective in the presence of discontinuities. In practical applications, when the regularity of the underlying system (e.g., smoothness or continuity) cannot be guaranteed, the objective function method offers a more robust alternative.

5 CONCLUSION AND FUTURE WORK

In this paper, we study two commonly used methods for simulation optimization with covariates under the setting of continuous problems, with a primary focus on neural network-based implementations. We design numerical experiments to illustrate that, for continuous simulation optimization problems with covariates, even if the objective function itself is continuous, the optimal solution with respect to the covariates may still exhibit instability or even discontinuity. In such cases, directly predicting the optimal solution becomes

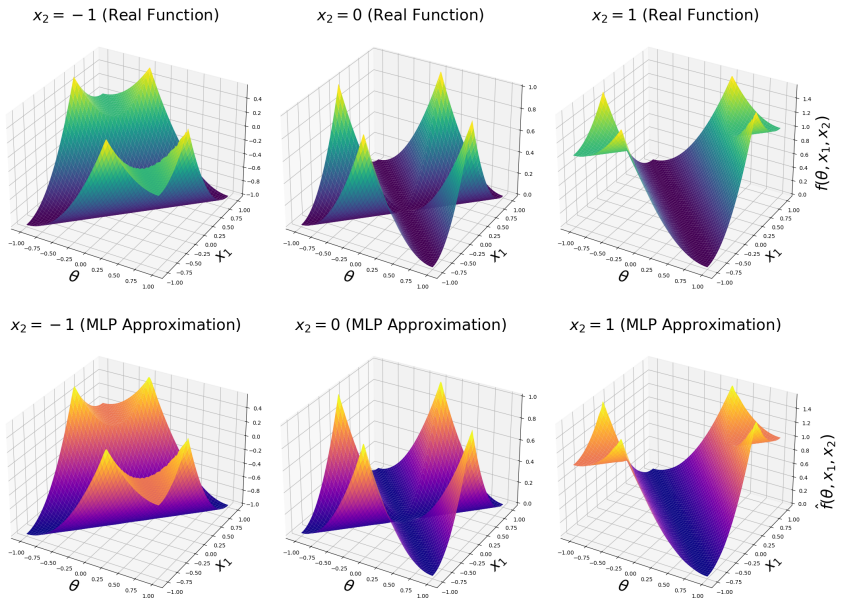

Figure 5: Approximation performance of the objective function method.

infeasible. We further point out that this method requires the performance function of the stochastic system to satisfy specific assumptions, for which we provide a set of applicability assumptions based on existing theories. In contrast, the objective function method is more general and can effectively handle such instability.

It should be noted that this paper only considers the case where the random source is independent of the covariates. More complex scenarios where covariates influence the random source remain to be investigated in future work. When the structure of the optimization problem is unknown, using neural networks to construct predictive models is a common choice due to their low risk of model misspecification. Although neural network-based methods often perform well empirically, they typically lack rigorous theoretical guarantees, which significantly limits their applicability. Therefore, developing corresponding theoretical foundations remains an important research direction.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China [Grants72293562 and 72171060].

REFERENCES

Ban, G.-Y., and C. Rudin. 2019. "The Big Data Newsvendor: Practical Insights from Machine Learning". *Operations Research* 67(1):90–108.

Bertsimas, D., and N. Kallus. 2020. "From Predictive to Prescriptive Analytics". *Management Science* 66(3):1025–1044.

Bertsimas, D., and N. Koduri. 2022. "Data-Driven Optimization: A Reproducing Kernel Hilbert Space Approach". *Operations Research* 70(1):454–471.

Bertsimas, D., N. Korolko, and A. M. Weinstein. 2019. "Covariate-Adaptive Optimization in Online Clinical Trials". *Operations Research* 67(4):1150–1161.

Bertsimas, D., and C. McCord. 2019. "From Predictions to Prescriptions in Multistage Optimization Problems". *arXiv preprint arXiv:1904.11637*.

Bonnans, J. F., and A. Shapiro. 2000. *Perturbation Analysis of Optimization Problems*. New York: Springer.

Fan, W., L. J. Hong, G. Jiang, and J. Luo. 2024. "Review of Large-Scale Simulation Optimization". *arXiv preprint arXiv:2403.15669*.

Fu, M. C. 2015. *Handbook of Simulation Optimization*. New York: Springer.

Hannah, L., W. Powell, and D. Blei. 2010. "Nonparametric Density Estimation for Stochastic Optimization with an Observable State Variable". In *Advances in Neural Information Processing Systems 23*. December 6th-11th, Vancouver, Canada, 820–828.

Hong, L. J., and G. Jiang. 2019. "Offline Simulation Online Application: A New Framework of Simulation-Based Decision Making". *Asia-Pacific Journal of Operational Research* 36(06):1940015.

Jiang, G., L. J. Hong, and B. L. Nelson. 2020. "Online Risk Monitoring Using Offline Simulation". *INFORMS Journal on Computing* 32(2):356–375.

Jiang, G., L. J. Hong, and H. Shen. 2024. "Real-Time Derivative Pricing and Hedging with Consistent Metamodels". *INFORMS Journal on Computing* 36(5):1168–1189.

Kannan, R., G. Bayraksan, and J. R. Luedtke. 2025. "Technical Note: Data-Driven Sample Average Approximation with Covariate Information". *Operations Research*. Forthcoming.

Keslin, G., B. L. Nelson, B. Pagnoncelli, M. Plumlee, and H. Rahimian. 2024. "Ranking and Contextual Selection". *Operations Research*. Forthcoming.

Kingma, D. P., and J. Ba. 2014. "Adam: A Method for Stochastic Optimization". *arXiv preprint arXiv:1412.6980*.

Kirst, G., and X. Wang. 2024. "Mode Transition Control of Large-size Tiltrotor Aircraft". *arXiv preprint arXiv:2411.06753*.

Luo, H., Z. Liang, and L. J. Hong. 2024. "Reliable Online Decision Making with Covariates". In *2024 Winter Simulation Conference (WSC)*, 3265–3276 https://doi.org/10.1109/WSC63780.2024.10838740.

Shen, H., L. J. Hong, and X. Zhang. 2021. "Ranking and Selection with Covariates for Personalized Decision Making". *INFORMS Journal on Computing* 33(4):1500–1519.

Song, J., Y. Sun, Q. Dong, and X. Ji. 2024. "A Parameter Adaptive Trajectory Tracking and Motion Control Framework for Autonomous Vehicle". *arXiv preprint arXiv:2411.17745*.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, *et al*. 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* 17(3):261–272.

Zhang, H., J. He, D. Zhan, and Z. Zheng. 2021. "Neural Network-Assisted Simulation Optimization with Covariates". In *2021 Winter Simulation Conference (WSC)*, 1–12 https://doi.org/10.1109/WSC52266.2021.9715516.

Zheng, Y., Z. Zheng, and T. Zhu. 2025. "A Doubly Stochastic Simulator with Applications in Arrivals Modeling and Simulation". *Operations Research* 73(2):910–926.

AUTHOR BIOGRAPHIES

YIZE HAO is a Ph.D. student in the School of Management at Harbin Institute of Technology. He received his B.S. degree in Information Management and Information System from Harbin Institute of Technology. His research interests include simulation optimization and machine learning. His email address is 24B910001@stu.hit.edu.cn.

GUANGXIN JIANG is a professor in the School of Management at Harbin Institute of Technology. He earned his Ph.D. in Applied Mathematics from Tongji University. His research interests include stochastic models and simulation, machine learning, financial engineering and risk management, FinTech, etc. His email address is gxjiang@hit.edu.cn (Corresponding author).