A Heteroscedastic Robust Bayesian Optimization Method for Solving Simulation-Based Transportation Problems

Jinbiao Huo[a†], Ziyuan Gu[a†], Zhiyuan Liu[a], Shuaian Wang[b], Gilbert Laporte[c,d]*

[a] School of Transportation, Southeast University, China

[b] Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[c] Department of Decision Sciences, HEC Montréal, Montréal, Canada

[d] School of Management, University of Bath, Bath, United Kingdom

**Corresponding author. Email: zhiyuanl@seu.edu.cn*; †*Co-first author*;

Abstract: This study focuses on simulation-based optimization (SBO) in transportation systems considering the pervasive and influential heteroscedastic noise. Existing studies rarely consider the effects of such heteroscedasticity on the solution robustness, giving rise to suboptimal solutions that could compromise the reliability and resilience of the system in real-world applications. To address this concern, a simulation-based robust optimization problem is investigated in this study, which focuses on minimizing the expectation of simulation outputs while maintaining the stochasticity of transportation systems within predefined limits. To solve the problem and identify a robust solution under varying levels of stochasticity, a heteroscedastic robust Bayesian optimization (HRBO) method is proposed by fusing key SBO concepts and techniques with the widely used Bayesian optimization (BO) algorithm. The formulation of surrogate models, strategies for sampling new points, and evaluation issues of samples are systematically designed. Specifically, surrogate models for the stochastic objective and constraint functions are separately formulated using the Gaussian process (GP) model. To accommodate simulation noise, Bayesian posterior inference is employed to estimate objective function values and constraint function values, which is incorporated into the GP models. To locate promising feasible solutions, a constrained expected improvement (EI) function is constructed and optimized using a tailored two-stage method, which can effectively tackle the inherent issue of "flat" areas of EI functions. Considering the usually high computational cost of simulators, an adaptive simulation resource allocation scheme is designed, by incorporating ranking and selection techniques into the BO framework, to efficiently allocate computational resources. The proposed methods are validated on a test function and two representative simulation-based transportation problems, a variant of the M/M/1 queueing problem and a continuous network design problem. Experimental results demonstrate the superior performance of HRBO in addressing heteroscedastic noise and identifying robust solutions.

Key words*:* Simulation-based robust optimization, Heteroscedastic noise, Bayesian optimization

1 Introduction

The increasing complexity of urban transportation systems, including the large size of networks, multiple management and control schemes, intricate logistics of transporting people and goods, high-dimensionality of system attributes, etc., calls for high-resolution simulators to model the complicated, dynamic, and stochastic system. Consequently, simulation models for transportation systems have garnered increasing attention over the years, and have been extensively applied in areas such as network design and analysis, daily operational management, and long-term planning (Fu 2002, Osorio and Chong 2015, Sauré et al. 2015, He et al. 2017, Juan et al. 2019, Bortolomiol et al. 2021).

The problem of employing simulators to inform traffic planning or management decisions is a typical simulation-based optimization (SBO) problem. The SBO refers to the optimization of objective functions subject to constraints, both of which can be evaluated through a stochastic simulation (Amaran et al. 2016). Due to the involvement of complex simulations, SBO problems are often expensive to evaluate, non-convex, and non-differentiable, with many local minima (Osorio 2019). Substantial efforts, therefore, have been devoted to developing effective and computationally efficient SBO algorithms (Osorio and Bierlaire 2013, Gosavi 2015, Hu et al. 2022, Li et al. 2023).

Despite the rapid and stunning progress in SBO, one pervasive but frequently disregarded issue in this field is the heteroscedasticity of stochastic simulators. Heteroscedasticity, also known as heterogeneity of variance, refers to the phenomenon that the variance of simulation output varies across the design space (Ankenman et al. 2010). Figure 1 illustrates the heteroscedasticity using an example of congestion pricing, in which the mean and variance of total travel cost are estimated based on replications of simulation under different toll rates. In traffic simulators, the heteroscedasticity can stem from various sources of uncertainty, and the uncertainties will be discussed in Section 1.1 and Section 2. In this study, we specifically focus on the heteroscedasticity arising from random objects within simulators that are used to model transportation systems, such as stochastic driving behaviors and desired speeds of drivers (Allen et al. 2006). The stochasticity of the random objects and their complex interactions reveal the inherent randomness of real-world transportation systems, and the heteroscedasticity is embedded in the fabric of various traffic simulators with different levels of fidelity.

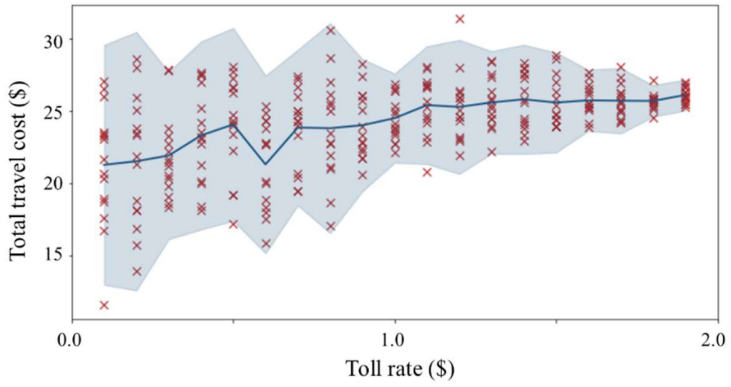


Figure 1. Heteroscedasticity of traffic simulation. The red crosses are simulation outputs, representing the simulated total travel costs. The solid blue line represents the mean value of total travel cost w.r.t. toll rates. The blue area represents the 2σ confidence interval, the width of which implies the level of stochasticity.

While heteroscedasticity is a prevalent feature of traffic simulators, most existing SBO algorithms, especially surrogate-based optimization algorithms, implicitly assume the homoscedastic distribution of simulation noise (Chen et al. 2019), affecting the efficiency of SBO algorithms in practical applications. Some efforts have been devoted to designing SBO algorithms considering heteroscedasticity (Peng et al. 2016, Chen et al. 2019, Guzman et al. 2020, Gu et al. 2023). Their primary objective is to enhance the efficiency of SBO algorithms under noisy conditions, such as efficiently allocating simulation resources. However, a crucial concern that has yet to receive enough attention is the *robustness* of solutions under heteroscedastic noise. In this study, the *robustness* quantifies the capacity of a solution to sustain its performance amidst inherent uncertainties within the system. This concept, which will be discussed in Section 2, aligns with Taguchian robustness, as delineated by Dellino et al. (2012).

Neglecting solution robustness under heteroscedastic noise can diminish the effectiveness of planning and management operations, thereby compromising the applicability and practicality essential for real-world applications. For instance, while using simulators to design traffic networks, the typical objective is to determine the optimal scheme to minimize the expectation of total travel cost. However, within the setting, a potential scheme can exhibit a low degree of robustness, characterized by a high level of stochasticity of the total travel cost. This suggests lower travel time reliability within the system (i.e., travel times vary greatly over days, see experiments in Section 4.2), resulting in poor travel experiences for commuters and inefficiency of the system. To the best of our knowledge, SBO algorithms tailored for transportation problems have yet to systematically deal with the heteroscedasticity of simulators and to identify robust solutions. This study, therefore, aims to fill this gap by designing a novel simulation-based robust optimization method.

1.1 Literature review

The past few decades have seen a growing interest in simulation-based optimization, owing to the possibility of using computer programs to analyze complex and random systems (Stummer 2012). Roughly speaking, SBO can be categorized into two classes: static optimization and dynamic optimization (Gosavi 2015). The former focuses on searching for optimal decision variables to optimize some performance measures such as cost or reward functions. The latter, also known as control optimization, aims to find a set of actions to take in different states that a system visits (Gosavi 2015), such that performance measures can be optimized. In this study, we specifically focus on static optimization, in which stochastic simulators

are used to inform traffic planning or management decisions (Osorio and Bierlaire 2013).

In the field of transportation, SBO problems have been extensively studied on a variety of topics, dealing with congestion pricing problems (Chen et al. 2016, Gu et al. 2019, Gu and Saberi 2021, Zheng et al. 2022b, Song et al. 2023), traffic signal control problems (Osorio and Selvam 2017), network design problems (Chen et al. 2006, Li et al. 2022, Yin et al. 2022), parking management problems (Dandl et al. 2021, Gu et al. 2023), parameter calibration problems (Antoniou et al. 2015, Cheng et al. 2019, Patwary et al. 2021, Qurashi et al. 2022), and so forth. In addition, several successful attempts have been made to design SBO algorithms considering characteristics of transportation systems, e.g., the large scale of networks and the high-dimensionality of decision variables (Osorio and Atasoy 2021, Tay and Osorio 2022), the expensive evaluation cost (Patwary et al. 2021, Fakhrmoosavi et al. 2022), the spatio-temporal correlations (Antoniou et al. 2015, Qurashi et al. 2022), and the multiple objectives in management and control (Avci and Selim 2018, Zheng et al. 2022a).

Recent research has begun to explore the heteroscedastic characteristics of traffic simulation noise. In the transportation discipline, Chen et al. (2019) developed a novel kriging model to approximate the relationship between simulation inputs and outputs. This model accounts for heteroscedastic simulation noise, and it has been applied to optimize toll rates and dynamic message signs on freeways. Similarly, Gu et al. (2023) proposed the NoisyDIRECT method, an extension of the DIRECT method proposed by Jones (1993). NoisyDIRECT incorporates simulation noise into the partitioning of the search space. Zheng et al. (2022b) examined the allocation of computational resources under heteroscedastic noise. In the optimization domain, several algorithms have been designed to address simulation noise. These methods can be broadly classified into continuous SBO methods and discrete SBO methods. Continuous SBO methods are typically developed within frameworks of surrogate-based optimization (Ankenman et al. 2010, Jalali et al. 2017) or stochastic gradient estimation (Hu and Fu 2024). Discrete SBO methods, on the other hand, often rely on random search techniques (Hong and Nelson 2006, Xu et al. 2013) and partitioning strategies (Xu and Nelson 2013, Lu et al. 2021). A key focus of random search techniques is the development of sampling strategies that allocate computational resources to solutions while explicitly accounting for heteroscedastic noise. The partitioning strategies focus on dividing the solution space and adaptively allocating resources based on simulation noise, with the goal of balancing exploration and exploitation. The genetic algorithm (GA) has also been incorporated into discrete SBO methods (Tsai and Fu 2014). For a comprehensive review of existing SBO methods and their approaches to handling simulation noise, readers can refer to Fu (2015). It is worth noting that most of the aforementioned SBO methods aim to optimize the expected system performance, with limited attention given to the robustness of the solutions. This study investigates the robustness of solutions considering simulation noise and concentrates on generating robust solutions to simulation-based transportation problems, which are

simulation-based robust optimization (SBRO) problems (Bertsimas et al. 2010).

SBRO aims to tackle uncertainties in simulation and to provide solutions that are insensitive to perturbations of factors involved in simulation models, e.g., environmental factors and measurement errors (Dellino et al. 2015). The origins of SBRO can be traced back to the late 1970s. Genichi Taguchi, a Japanese engineer, proposed that the quality of products could be improved by accounting for variations of objectives (Parnianifard et al. 2018). Taguchi's method provided initial insights into SBRO, and has been extensively studied and improved during the past few decades (Tsui 1996, Dellino et al. 2012, Kleijnen 2017). After approximately 50 years of development, the research of SBRO can be broadly categorized into three classes based on the sources of uncertainties (Allen et al. 2006). The first class of SBRO research (Type-I SBRO) deals with uncertainties in noise factors, also known as environmental factors, which are exogenous parameters of simulation and cannot be controlled by practitioners (Dellino et al. 2015). Taguchi's method and its variants are widely used to address such problems. The basic idea of Taguchi's method is to strategically design simulation experiments that explore combinations of noise factors and decision variables. The solutions that demonstrate the least sensitivity to variations in noise factors are subsequently selected based on simulation evaluation results (Karna and Sahai 2012). One limitation of Taguchi's method lies in the high computational cost required by the exploration of various combinations of noise factors and decision variables. To deal with the problem, Taguchi's method has been improved by combining with surrogate models (Trosset 1996, Dellino et al. 2012). In the transportation discipline, one pervasive noise factor (or environmental factor) is the random travel demand (Ma and Qian 2018). Bernardo et al. (2021) analyzed the effects of uncertain demands in the stochastic capacitated vehicle routing problem, in which an SBRO method is developed. Zheng et al. (2022b) investigated the robust design of congestion pricing schemes while considering the uncertainties in origin-destination (OD) matrices, proposing a distributionally robust simulation-based optimization model to solve the problem. In addition, Zhang (2017) considered uncertainties of travel times in the robust design of single-direction transit routes. The author solved the robust design problem by combining Taguchi's method and kriging metamodels.

The second class of SBRO (Type-II SBRO) focuses on uncertainties stemming from imperfect implementations of decision variables (Bertsimas et al. 2010, Beland and Nair 2017, Bogunovic et al. 2018, Z. Hu and Hong 2022, Zheng et al. 2022a). While implementation errors have been widely studied in fields such as structural design (Gu et al. 2000) and chemical engineering (Prett and García 2013), relatively little research has been carried out on implementation errors in transportation. Zheng et al. (2022a) developed a bi-objective SBRO method to find robust solutions to unconstrained SBO problems involving implementation errors and parameter perturbations, in which a traffic signal design problem is taken as an example to validate the proposed methods.

The third class of SBRO research (Type-III SBRO) deals with the inherent uncertainties within

simulators, i.e., the stochasticity of the system embedded in simulators (Allen et al. 2006, Apley et al. 2006, Zhu et al. 2013). In transportation systems, such uncertainties include the randomness in drivers' reaction times and route choice behaviors. The variation in simulation outputs caused by these inherent uncertainties reflects the reliability and resilience of transportation networks (Zang et al. 2022). This study investigates the heteroscedasticity resulting from these inherent uncertainties. Compared to Type-I and Type-II SBRO research, Type-III SBRO research is still emerging and has received limited attention. Type-III SBRO problems are characterized by uncertainties arising from various random factors with unknown distributions. Consequently, the solution robustness is typically measured by the variance of the system performance (i.e., simulation outputs). To identify robust solutions, the Type-III SBRO problems are formulated as constrained simulation-based optimization problems that aim to minimize the expected system performance, subject to implicit robustness constraints on its variance. Even though some studies investigate Type-III SBRO problems with known distributions of random objects in simulation, which is similar to the Type-I SBRO (Beland and Nair 2017, Iwazaki et al. 2021). However, this is seldom the case in transportation systems, because identifying distributions and interactions of thousands of random objects in complicated and large-scale simulations is impractical. Effectively solving Type-III SBRO problems requires addressing the implicit robustness constraints. Constrained simulation-based optimization methods can be applied to tackle this challenge (Luo and Lim 2013, Gramacy et al. 2016, Le Digabel and Wild 2024). The most widely used approach is the surrogate-based optimization algorithms (Shaibu and Cho 2009, Yanıkoğlu et al. 2016, Wu et al. 2019, Griffiths et al. 2021). This study focuses on the Type-III SBRO problems and aims to solve the problems in surrogate-based optimization frameworks.

The basic idea of surrogate-based optimization is to fit surrogate models (cheap-to-evaluate approximate functions of the underlying complex objective functions) by evaluating the objective functions at a few points (termed sample points), which is then used to estimate the location of the optimum (Jones et al. 1998). Specifically, surrogate-based optimization algorithms consist of three iterative steps: *i*) constructing surrogate models based on sample points, *ii*) suggesting new samples based on the surrogate models, and *iii*) evaluating the new samples for the refinement of surrogate models (Forrester and Keane 2009). To solve Type-III SBRO problems, surrogate models are typically fitted for both the expectation and variance of simulation outputs and are then used to inform the sampling of robust solutions (Yanıkoğlu et al. 2016, Wu et al. 2019, Griffiths et al. 2021). One successful application of the technique in the field of transportation is reported by Wu et al. (2019). The authors explored the robust design of bus services, considering the presence of stochastic travel times. They approached the problem by first iteratively generating infill samples to develop precise polynomial surrogate models for the expectation and variance of simulation outputs. Subsequently, robust designs were obtained by analyzing the surrogate models in a constrained optimization setting, using the surrogate models of the expectation and variance as objective

and constraint functions, respectively.

Despite the potential of surrogate-based optimization, current methods largely follow a conventional framework: they approximate both the objective function and constraints, solving an estimated optimization problem. This conventional framework overlooks key characteristics of SBRO problems, such as heteroscedastic noise and implicit robustness constraints, leaving several critical challenges unaddressed. Several issues remain largely unexplored. The first issue lies in accounting for estimation errors within surrogate models. System performance (e.g., the expectation of simulation outputs) is usually estimated based on simulation replications, which serves as inputs of surrogate models (Wu et al. 2019). However, estimation errors are rarely discussed in existing methods for Type-III SBRO problems, compromising the accuracy of surrogate models. Secondly, when generating new samples under implicit robustness constraints, existing methods generally replace actual constraints with surrogate functions (Dellino et al. 2015). Nonetheless, uncertainties of surrogate models should be considered in sampling strategies. Thirdly, traffic simulation typically incurs high execution costs. While the need for intelligent allocation of simulation replications to conserve limited computational resources is recognized (Jalali et al. 2017, Gu et al. 2024), this allocation strategy has never been discussed in the SBRO literature. Effectively managing computational resources under constrained settings presents an ongoing challenge in surrogate-based optimization.

Among various surrogate-based optimization techniques, Bayesian optimization (BO) algorithms stand out by utilizing surrogate models constructed via Bayesian statistics (Frazier 2018). The Gaussian process (GP) is a prevalent surrogate model in BO. GP predictions are also well known in the geostatistics field, where they are known as kriging models (Williams and Rasmussen 2006). GP differs from models such as linear regression and the radial basis function in that it does not assume a predefined functional form from inputs to outputs. This attribute enables GP to adeptly model complex nonlinear functions (Williams and Rasmussen 2006). Additionally, compared to conventional surrogate models such as the polynomial functions, GP naturally provides a measure of uncertainty in its predictions, enabling an efficient trade-off between exploration and exploitation. GP and its extensions constitute the most popular class of surrogate models (Hong and Zhang 2021). The remarkable data efficiency of Bayesian Optimization (BO) has made it increasingly popular in transportation research, as demonstrated by recent studies including Huo et al. (2023), who proposed a particularly effective BO-based method for solving congestion pricing problems (Tay and Osorio 2022, Yin et al. 2022, Huo et al. 2023). This study attempts to explore solution algorithms within the framework of BO. To the best of our knowledge, BO has never been applied to solve Type-III SBRO problems in the existing literature.

It should be noted that even though some studies have explored surrogate-based optimization techniques for addressing issues such as heteroscedasticity (Ankenman et al. 2010) or implicitly defined constraints

(Gardner et al. 2014), these techniques have never been integrated into an algorithm for the Type-III SBRO problem. These studies inspired this research. The literature review and discussions on these techniques are presented in Section 3.

1.2 Objectives and contributions

Motivated by these considerations, this study focuses on SBO problems in the transportation discipline. Taking account of the pervasive heteroscedastic simulation noise, the aim is to achieve robust solutions under a specific level of stochasticity. To this end, a heteroscedastic robust Bayesian optimization (HRBO) method is built upon the BO framework (Frazier 2018), which features several customized and integrated sub-algorithms for addressing heteroscedastic noise, computational efficiency, and solution robustness.

The primary contribution of this study is the HRBO algorithm, which innovatively addresses heteroscedasticity and robustness constraint within the surrogate model optimization framework. It is one of the first to comprehensively tackle heteroscedastic noise, offering an efficient solution to Type-III SBRO problems through advanced surrogate formulation, sampling strategies, and evaluation techniques. Detailed technical contributions are as follows:

i. The Gaussian process, an advanced Bayesian regression model, is used to construct surrogate models. Different from existing GP formulations, to accommodate simulation noise, Bayesian posterior inference is employed to estimate distributions of simulation outputs and is then incorporated into GP models. Particularly, a novel objective prior distribution, the GP-based prior, is introduced in this study to derive posterior distributions of simulation outputs. Numerical experiments confirm the accuracy of the GP models as effective surrogate models.

ii. A constrained expected improvement (EI) function is formulated to inform the sampling of points. Existing EI functions have individually examined constraints or simulation noise. In this study, the two issues are jointly investigated in the proposed EI formulation. The pervasive "flat" area issue of the EI function (i.e., the value of the EI function is close to zero in many areas of the feasible set, see Section 3.2) is investigated in this study. The causes of the "flat" area issue are analyzed, based on which a two-stage method is proposed to tackle the problem. To the best of our knowledge, this is the first study that designs a tailored method to tackle the "flat" area issue while considering constraints. Experiments show that the two-stage method can identify local optima in most cases.

iii. The allocation of simulation resources has seldom been addressed in the SBRO literature. While BO can implicitly manage simulation evaluations in unconstrained optimization scenarios, the allocation of these resources in constrained settings remains largely unexplored. In this study, a customized ranking and selection (R&S) procedure, namely the adaptive simulation resource allocation scheme, is developed to deal with the evaluation issues in HRBO. In each iteration of

HRBO, the scheme informs the allocation of simulation replications and the selection of robust solutions. Experiments validate the effectiveness of the simulation resource allocation scheme in managing limited computational budgets.

The remainder of this paper is structured as follows. Section 2 formulates the simulation-based robust optimization problem, and challenges in solving the problem are discussed. Section 3 presents the HRBO method for solving the problem. The efficiency of the proposed methods is verified by case studies in Section 4. Section 5 concludes the paper with remarks on the proposed improvements and recommendations for further study.

2 Problem Statement

We consider simulation-based robust optimization problems in the following form:

$$\min_{\boldsymbol{x}} \mathbb{E}[\![f(\boldsymbol{x},\boldsymbol{\xi})]\!] \tag{1}$$

subject to

$$\mathbb{VAR}[\![f(\boldsymbol{x},\boldsymbol{\xi})]\!] \le \sigma_c^2, \tag{2}$$

$$x_{l,i} \le x_i \le x_{u,i},\ i = 1,...,d. \tag{3}$$

In the SBRO problem, $\boldsymbol{x}$ is a d-dimensional continuous decision vector, $\boldsymbol{x} = (x_1,...,x_d)$, $d \in \mathbb{Z}^+$, where $\mathbb{Z}^+$ is the set of nonnegative integers. $\mathbb{E}[\![\cdot]\!]$ ($\mathbb{VAR}[\![\cdot]\!]$) is the expectation (variance) operator. The function $f(\cdot)$ is a stochastic system performance measured by a traffic simulator. The value of $f(\cdot)$ depends on two elements: the decision vector $\boldsymbol{x}$ and the set $\boldsymbol{\xi}$ of random variables representing uncertainties of the simulation model. $f(\cdot)$ typically represents the average performance of all agents (e.g., the average travel time of all travelers) in simulation. Thus, the central limit theorem holds, and given $\boldsymbol{x}$, $f(\boldsymbol{x},\boldsymbol{\xi})$ is assumed to be normally distributed (Zhang et al. 2023). The parameter σ_c^2 in constraint (2) is a threshold controlling the stochasticity of $f(\cdot)$. In constraint (3), $x_{l,i}$ ($x_{u,i}$) defines the lower (upper) bounds of x_i, $i = 1,...,d$. For instance, a simulation-based congestion pricing problem may take $f(\cdot)$ as the average travel time of vehicles (to achieve system optimum), and $\boldsymbol{x}$ as the vector of toll values. The objective of the optimization problem is to minimize the expectation of $f(\cdot)$ within the predetermined level σ_c^2 of stochasticity. For brevity, we denote $f(\boldsymbol{x},\boldsymbol{\xi})$ as $f(\boldsymbol{x})$ hereafter and use $y(\cdot)$ and $r(\cdot)$ to represent $\mathbb{E}[\![f(\cdot)]\!]$ and $\mathbb{VAR}[\![f(\cdot)]\!]$, respectively.

In this study, $\boldsymbol{\xi}$ involves only the uncertainties inherent in traffic simulators. The inherent uncertainties

refer to the stochastic characteristics embedded in traffic simulators, arising from the random behavior of system components such as driver responses, route choice decisions, and interactions among traffic entities. These uncertainties are independent of external environmental factors or implementation errors and are intrinsic to the simulation process itself. In the investigated SBRO, we assume that there are no uncertainties of environmental factors or implementation errors. This justification is based on the following reasons. First, environmental factors—such as network topology and traffic demand—are predetermined within the simulator settings, meaning there are no uncertainties in these variables. This setting aligns with common practices in both practical applications and existing studies. Second, in traffic simulators, implementation errors are generally minimal. Given inputs, computers precisely compute interactions among simulation entities. While some rounding errors may occur (with a magnitude of approximately 1E-16 in floating-point calculations), their impact on simulation outputs is negligible. Admittedly, certain implementation errors can occur in real-world traffic applications (e.g., the actual width of a constructed road may deviate slightly from the planned width). These errors fall outside the scope of this study.

It is important to clarify that the traffic demand of the environmental factors refers to the OD-level demand. Given the fixed OD demand, the variability in link-level traffic volumes arises from travelers' stochastic route choice behaviors in response to network conditions. This source of randomness is inherent to the traffic simulator. Accordingly, in our formulation, fluctuations in link volumes under fixed OD demand are captured through the random parameter $\boldsymbol{\xi}$ in the SBRO model.

Proposition 1. The SBRO problem (1)–(3) is mathematically equivalent to the following robust optimization problem

$$\min_{x} \max_{\theta} \mathbb{E}\left[\!\left[f(\boldsymbol{x},\boldsymbol{\xi}) \right]\!\right] + \theta\left(\mathbb{VAR}\left[\!\left[f(\boldsymbol{x},\boldsymbol{\xi}) \right]\!\right] - \sigma_c^2 \right) \tag{4}$$

subject to

$$\theta \geq 0\,,\ x_{l,i} \leq x_i \leq x_{u,i}\,,\ i = 1,...,d\,. \tag{5}$$

A formal proof of **Proposition 1** can be found in Zhang (2017). **Proposition 1** demonstrates that the SBRO problem investigated in this study can be transformed into a minmax robust optimization problem, confirming the concept of *robustness* used in this research.

Let us first illustrate the intricacy of the function $f(\cdot)$. Detailed descriptions of elaborate behaviors and the interactions between individual travelers are embedded in traffic simulators. Specifically, the car-following behaviors, lane-changing behaviors, and other travel decisions such as route choice are delicately defined for each vehicle considering its surrounding traffic conditions. In addition, randomness is incorporated into traffic simulators in processes such as transportation mode choice and network loading, which is indirectly affected by $\boldsymbol{x}$ in its realization. As a result, given $\boldsymbol{x}$, $f(\boldsymbol{x})$ is the consequence of

interactions of all travelers in the stochastic transportation system, and the function $f(\cdot)$ is highly nonlinear, non-differentiable, and with heteroscedastic noise, usually without a closed-form expression.

Challenges of solving the simulation-based robust optimization problem are threefold. First, due to the complexity of $f(\cdot)$, both $y(\cdot)$ and $r(\cdot)$ are highly nonlinear and without closed-form expressions or derivatives with respect to the decision vector $\boldsymbol{x}$. For a realization of $\boldsymbol{x}$, $y(\boldsymbol{x})$ and $r(\boldsymbol{x})$ can only be estimated through multiple replications of evaluating $f(\boldsymbol{x})$. Thus, identifying solutions that minimize the intricate objective function while satisfying the implicitly defined constraint is a challenging task. Second, heteroscedastic noise exists in $f(\cdot)$. The optimization method should, therefore, be able to incorporate different levels of stochasticity. More specifically, given $\boldsymbol{x}$, the method needs to identify the level of stochasticity through evaluations of $f(\boldsymbol{x})$ and construct reliable estimates of $y(\boldsymbol{x})$ ($r(\boldsymbol{x})$) considering the existence of heteroscedastic noise. Third, traffic simulators are usually time-consuming for each simulation run. Optimization techniques that rely on extensive domain exploration are deemed unsuitable. Instead, the optimization method should be computationally efficient to save on simulation runs. This paper aims to solve these challenges by designing an efficient Bayesian optimization method.

3 HRBO for Solving Simulation-Based Robust Optimization Problems

BO is a widely used method to solve expensive-to-evaluate black-box optimization problems (Frazier 2018) and consists of three iterative steps. First, it constructs a surrogate model using sample points. Second, it suggests a new sample point based on the surrogate model. Third, it evaluates the new sample to refine the surrogate model. These steps are typically executed by constructing Gaussian process models, optimizing acquisition functions (i.e., functions that measure sampling decisions), and executing simulation, respectively. Conventional BO methods are primarily designed for unconstrained optimization problems. To address the SBRO problem considered in this study, constrained Bayesian optimization (CBO) techniques provide valuable insights into managing the implicit variance constraint (2). These methods typically adapt acquisition functions to accommodate constraints, enabling the generation of feasible sample points. For example, a common approach integrates acquisition functions with feasibility indicators, such as the probability of feasibility. However, existing CBO methods are seldom tailored to handle noisy observations, rendering their direct application to SBRO problems nontrivial. The proposed HBRO can be regarded as an extension of CBO techniques, in which we simultaneously investigate the robustness constraint and the heteroscedastic noise.

For clarity, Figure 2 depicts the sketch of HRBO. Similar to BO, HRBO consists of three main steps: the construction of surrogate models, the optimization of acquisition functions, and the evaluation of samples. Specifically, in each iteration of HRBO, two surrogate models are formulated using GP,

approximating the objective and the constraint functions, respectively (Section 3.1). Based on the surrogate models, a customized constrained EI function is formulated, and a two-stage method is proposed to solve the constrained EI function, deriving a new sample point (Section 3.2). Given a new sample point, HRBO intelligently allocates computational resources to the sample. The objective function value and the constraint function value associated with the sample are estimated based on simulation evaluation results, which update the dataset (Section 3.3 and Section 3.4). HRBO iteratively conducts the three steps until meeting predetermined stopping criteria.

Notice that some of the SBO techniques integral to HRBO were previously investigated in limited studies. For example, Zhang (2017) used two Kriging models to approximate the mean and the standard deviation. Nonetheless, all the SBO techniques in this study are designed or modified considering the characteristics of heteroscedasticity and the solution robustness, contributing to a novel method suitable for the SBRO problem investigated in this study. These technical details and their novelty will be elaborated upon in the following subsections.

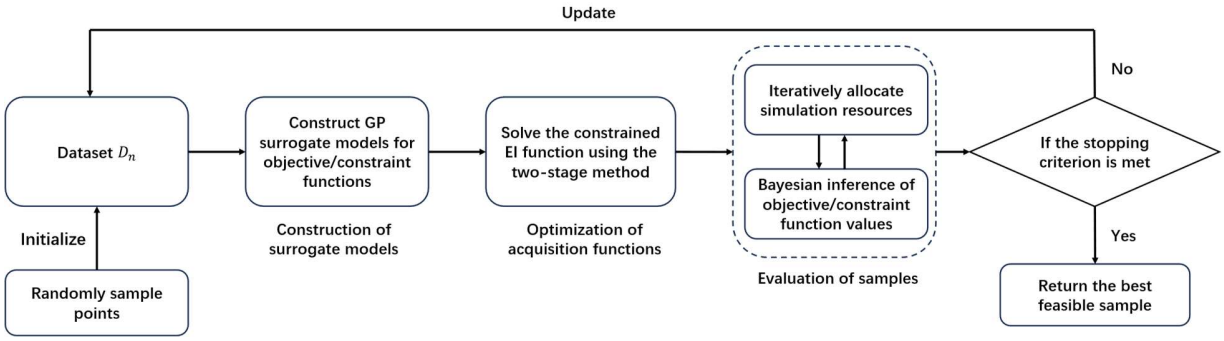


Figure 2. The flow chart of HRBO

3.1 Dual response formulated by Gaussian process

Inspired by dual-response methods in robust design (Yanikoglu et al. 2016), this section constructs surrogate models for the functions $y(\cdot)$ and $r(\cdot)$. Given that $r(\cdot)$ is non-negative, we do not construct a surrogate model for $r(\cdot)$ directly. Instead, we model its natural logarithm, a widely adopted approach in the literature for constructing surrogate models of variance (Goldberg et al. 1997, Lázaro-Gredilla and Titsias 2011).

Accounting for heteroscedastic noise in GP models is challenging when $r(\cdot)$ is an unknown function. To effectively learn GP models, two primary approaches are employed: *joint modeling* and *separate modeling*. The joint modeling approach jointly models both expectation and variance within a single probabilistic framework. This typically involves treating the variance as a latent function in a hierarchical GP framework (Lázaro-Gredilla and Titsias 2011). However, this approach requires customized training

methods (e.g., Monte Carlo sampling and Laplace approximations), which are typically computationally expensive due to the complexity of joint probabilistic modeling. In contrast, the separate modeling approach treats $y(\cdot)$ and $r(\cdot)$ as distinct functions and approximates them with two independent Gaussian processes. A typical workflow involves first estimating the expectation and variance values from simulation replications, then using these estimates as inputs to formulate GP models. This approach simplifies implementation, allows for specialized modeling of noise variance, and facilitates the analysis of solution robustness, making it a widely adopted method in the SBRO literature (Ankenman et al. 2010, Dellino et al. 2015, Chen et al. 2019). Accordingly, this study adopts the separate modeling approach to construct GP surrogate models.

One limitation of the separate modeling approach is that it overlooks the inherent relationship between the expectation and variance of simulation outputs. For example, when estimating $y(\boldsymbol{x})$, methods such as stochastic kriging use a point estimate of $r(\boldsymbol{x})$ to assess uncertainty, without accounting for the estimation error of $r(\boldsymbol{x})$ itself (Ankenman et al. 2010). To address this, we treat both the expectation and variance as random variables and jointly model them within a Bayesian framework. Given a solution $\boldsymbol{x}$, the posterior distributions of $y(\boldsymbol{x})$ and $r(\boldsymbol{x})$ are estimated simultaneously and incorporated into the GP models.

Definition 1. A Gaussian process is a set of random variables, any finite number of which have a joint Gaussian distribution.

Let y_{GP} denote a Gaussian process, and it is completely specified by two components: the mean function $h(\boldsymbol{x}):\mathbb{R}^d \mapsto \mathbb{R}$ and the covariance function $\kappa(\boldsymbol{x},\boldsymbol{x}'):\mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$. The two components are defined as

$$h(\boldsymbol{x}) := \mathbb{E}\left[\!\left[y_{GP}(\boldsymbol{x}) \right]\!\right], \tag{6}$$

$$\kappa(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}\left[\!\left[\left(y_{GP}(\boldsymbol{x}) - h(\boldsymbol{x})\right)\left(y_{GP}(\boldsymbol{x}') - h(\boldsymbol{x}')\right) \right]\!\right]. \tag{7}$$

When constructing surrogate models, it is assumed that the function $y(\cdot)$ is a sample path of a GP. A GP prior is initially applied to $y(\cdot)$, denoted by

$$y_{GP}(\boldsymbol{x}) \sim \mathcal{GP}\left(h_0(\boldsymbol{x}), \kappa_0(\boldsymbol{x},\boldsymbol{x}')\right) \tag{8}$$

This study assumes that the prior mean function $h_0(\boldsymbol{x})$ and the prior covariance function $\kappa_0(\boldsymbol{x},\boldsymbol{x}')$ satisfy the following conditions:

Assumption 1. The mean function $h_0(\boldsymbol{x})$ is continuous, and the covariance function has the form of $\kappa_0(\boldsymbol{x},\boldsymbol{x}') = \sigma^2\rho(\|\boldsymbol{x}-\boldsymbol{x}'\|)$, where $\sigma > 0$ is a constant. $\|\cdot\|$ denotes the Euclidean norm. The function $\rho(\|\boldsymbol{x}-\boldsymbol{x}'\|)$ satisfies the following conditions:

i) $\rho(\|\boldsymbol{x}-\boldsymbol{x}'\|)$ is decreasing in $\|\boldsymbol{x}-\boldsymbol{x}'\|$

ii) $\rho(\|\boldsymbol{x}-\boldsymbol{x}'\|)\to 0$ as $\|\boldsymbol{x}-\boldsymbol{x}'\|\to\infty$, and $\rho(0)=1$.

Most widely used covariance functions satisfy the assumption. The reader can refer to Williams and Rasmussen (2006) for more details. This study adopts the radial basis function $\kappa_0(\boldsymbol{x},\boldsymbol{x}')=\sigma^2\exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/2l^2)$ with $\sigma>0$ and $l>0$ being the hyperparameters. The prior distribution (8) reflects prior beliefs about the function $y(\cdot)$, and it can be chosen by users. Typically, in the absence of any information about $y(\cdot)$, $h_0(\boldsymbol{x})$ is set to a constant, $h_0(\boldsymbol{x})=0$.

Given simulation outputs, the distribution of y_{GP} can be updated. Let $X_n=\{\boldsymbol{x}_1,...,\boldsymbol{x}_n\}$ denote the set of sample points, where $n\in\mathbb{Z}^+$ is the number of sample points. For all $i\in\{1,...,n\}$, the sample point $\boldsymbol{x}_i$ denotes a realization of the decision vector, which is also referred to as a solution or a sample. Multiple simulation replications are conducted at $\boldsymbol{x}_i$ to obtain realizations of $f(\boldsymbol{x}_i)$, denoted as $\boldsymbol{f}_i=\{f_i^{(1)},...,f_i^{(m_i)}\}$, where $m_i\in\mathbb{Z}^+$ is the number of simulation replications at $\boldsymbol{x}_i$. Let $\boldsymbol{\mathcal{F}}_n=\{\boldsymbol{f}_1,...,\boldsymbol{f}_n\}$ denote the set of simulation outputs. The dataset is defined as $D_n:=(X_n,\boldsymbol{\mathcal{F}}_n)$. For all $\boldsymbol{x}_i\in X_n$, $y(\boldsymbol{x}_i)$ and $\log r(\boldsymbol{x}_i)$ are deemed to be random variables from the Bayesian perspective. Posterior distributions of $y(\boldsymbol{x}_i)$ and $\log r(\boldsymbol{x}_i)$ are estimated based on D_n (the estimation method is discussed in Section 3.3). Let $\tilde{\mu}_{y,i}$ and $\tilde{\tau}_{y,i}^2$ denote the expectation and variance of the posterior distribution of $y(\boldsymbol{x}_i)$, respectively. Let $\tilde{\mu}_{lr,i}$ and $\tilde{\tau}_{lr,i}^2$ denote the expectation and variance of the posterior distribution of $\log r(\boldsymbol{x}_i)$, respectively. The posterior distribution of y_{GP} is updated as

$$y_{GP}(\boldsymbol{x})\mid D_n\sim\mathcal{GP}\left(h_n(\boldsymbol{x}),\kappa_n(\boldsymbol{x},\boldsymbol{x}')\right) \tag{9}$$

specifically,

$$h_n(\boldsymbol{x})=h_0(\boldsymbol{x})+\kappa_0(\boldsymbol{x},X_n)[\kappa_0(X_n,X_n)+\Lambda_n]^{-1}[\mathbf{Y}_n-h_0(X_n)], \tag{10}$$

$$\kappa_n(\boldsymbol{x},\boldsymbol{x}')=\kappa_0(\boldsymbol{x},\boldsymbol{x}')-\kappa_0(\boldsymbol{x},X_n)[\kappa_0(X_n,X_n)+\Lambda_n]^{-1}\kappa_0(X_n,\boldsymbol{x}'), \tag{11}$$

where $\kappa_0(X_n,X_n)=[\kappa_0(\boldsymbol{x}_i,\boldsymbol{x}_j)]_{1\le i,j\le n}\in\mathbb{R}^{n\times n}$, $\kappa_0(\boldsymbol{x},X_n)=(\kappa_0(\boldsymbol{x},\boldsymbol{x}_1),...,\kappa_0(\boldsymbol{x},\boldsymbol{x}_n))^T\in\mathbb{R}^{1\times n}$, $\kappa_0(X_n,\boldsymbol{x}')=(\kappa_0(\boldsymbol{x}_1,\boldsymbol{x}'),...,\kappa_0(\boldsymbol{x}_n,\boldsymbol{x}'))\in\mathbb{R}^n$, $h_0(X_n)=(h_0(\boldsymbol{x}_1),...,h_0(\boldsymbol{x}_n))\in\mathbb{R}^n$, $\mathbf{Y}_n=(\tilde{\mu}_{y,1},...,\tilde{\mu}_{y,n})$, and Λ_n is an n-dimensional diagonal matrix with $\tilde{\tau}_{y,i}^2$ being its diagonal elements, $i=1,...,n$. Given the dataset D_n and an arbitrary solution $\boldsymbol{x}\in\mathbb{R}^d$, the value of $y(\boldsymbol{x})$ can be predicted based on the posterior distribution (9)

$$y(\boldsymbol{x})\mid\boldsymbol{x},D_n\sim\mathcal{N}(\mu_y(\boldsymbol{x}),\ \sigma_y^2(\boldsymbol{x})), \tag{12}$$

where $\mu_y(\boldsymbol{x})=h_n(\boldsymbol{x})$ is defined in Eq. (10), and $\sigma_y^2(\boldsymbol{x})=\kappa_n(\boldsymbol{x},\boldsymbol{x})$ is defined in Eq. (11). In the predictive

distribution (12), $\sigma_y^2(\boldsymbol{x})$ can be interpreted as the variance of prediction, providing information about the uncertainty of the estimation.

The surrogate model of $\log r(\cdot)$ can be formulated in the same way. A GP prior is first assigned to the function. To distinguish between the surrogate models for $y(\cdot)$ and $r(\cdot)$, let $\mathcal{GP}(\tilde{h}_0(\boldsymbol{x}), \tilde{\kappa}_0(\boldsymbol{x}, \boldsymbol{x}'))$ denote the GP prior, where $\tilde{h}_0(\boldsymbol{x})$ and $\tilde{\kappa}_0(\boldsymbol{x}, \boldsymbol{x}')$ are the prior mean function and prior covariance function, respectively. They can be defined by users and follow **Assumption 1**. The GP prior is then updated based on observations. Given D_n and a solution $\boldsymbol{x}$, we directly give the predictive distribution of $\log r(\boldsymbol{x})$,

$$\log r(\boldsymbol{x}) \mid \boldsymbol{x}, D_n \sim \mathcal{N}(\mu_{lr}(\boldsymbol{x}),\ \sigma_{lr}^2(\boldsymbol{x})), \tag{13}$$

specifically,

$$\mu_{lr}(\boldsymbol{x}) = \tilde{h}_0(\boldsymbol{x}) + \tilde{\kappa}_0(\boldsymbol{x}, X_n)[\tilde{\kappa}_0(X_n, X_n) + \tilde{\Lambda}_n]^{-1}[\mathbf{r}_n - \tilde{h}_0(X_n)], \tag{14}$$

$$\sigma_{lr}^2(\boldsymbol{x}) = \tilde{h}_0(\boldsymbol{x}, \boldsymbol{x}) - \tilde{\kappa}_0(\boldsymbol{x}, X_n)[\tilde{\kappa}_0(X_n, X_n) + \tilde{\Lambda}_n]^{-1}\tilde{\kappa}_0(X_n, \boldsymbol{x}), \tag{15}$$

where $\tilde{\kappa}_0(X_n, X_n) = [\tilde{\kappa}_0(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$, $\tilde{\kappa}_0(\boldsymbol{x}, X_n) = (\tilde{\kappa}_0(\boldsymbol{x}, \boldsymbol{x}_1), ..., \tilde{\kappa}_0(\boldsymbol{x}, \boldsymbol{x}_n))^T \in \mathbb{R}^{1\times n}$, $\tilde{\kappa}_0(X_n, \boldsymbol{x}') = (\tilde{\kappa}_0(\boldsymbol{x}_1, \boldsymbol{x}'), ..., \tilde{\kappa}_0(\boldsymbol{x}_n, \boldsymbol{x}')) \in \mathbb{R}^n$, $\tilde{h}_0(X_n) = (\tilde{h}_0(\boldsymbol{x}_1), ..., \tilde{h}_0(\boldsymbol{x}_n)) \in \mathbb{R}^n$, $\mathbf{r}_n = (\tilde{\mu}_{lr,1}, ..., \tilde{\mu}_{lr,n})$, and $\tilde{\Lambda}_n$ is an n-dimensional diagonal matrix with $\tilde{\tau}_{lr,i}^2$ being its diagonal elements, $i = 1, ..., n$.

3.2 Constrained EI function for sampling decisions

This section presents the method for sample selection. In BO algorithms, samples are selected sequentially by optimizing acquisition functions derived from surrogate models. Commonly used acquisition functions include the probability of improvement (PI), expected improvement (EI), and upper confidence bound (UCB). This study employs the EI function, which is well-regarded for effectively balancing exploration and exploitation in sampling decisions (Zhan and Xing 2020). The EI function is originally designed for deterministic and unconstrained optimization problems. To tackle the SBRO problem, the EI function is adapted here, drawing inspiration from studies on noisy and constrained BO. We propose a variant of the EI function and discuss its optimization method.

3.2.1 Constrained EI function

Let y^* denote the minimal objective value of all samples. The basic idea of EI is to sample a new solution $\boldsymbol{x}$ to maximize the expected improvement over the current best sample (Kleijnen et al. 2012), that is,

$$\max_{\boldsymbol{x}} \mathbb{E}\left[\max\left(y^* - y(\boldsymbol{x}), 0\right)\right]. \tag{16}$$

Given the predictive distribution $y(\boldsymbol{x})\,|\,\boldsymbol{x}, D_n$ (i.e., the distribution (12)), the optimization problem (16) is equivalent to the following problem:

$$\max_{\boldsymbol{x}} \left(y^* - \mu_y(\boldsymbol{x})\right)\Phi\left(\frac{y^* - \mu_y(\boldsymbol{x})}{\sigma_y(\boldsymbol{x})}\right) + \sigma_y(\boldsymbol{x})\phi\left(\frac{y^* - \mu_y(\boldsymbol{x})}{\sigma_y(\boldsymbol{x})}\right), \tag{17}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution, respectively.

However, the EI function (17) cannot be directly used in the SBRO problem due to two key limitations: *i*) it does not account for the problem constraints (i.e., (2) and (3)), making it ineffective in generating feasible samples, and *ii*) it fails to consider simulation noise, potentially leading to a biased estimation of the improvement $\max(y^* - y(\boldsymbol{x}), 0)$. To tackle these issues, two modifications are made to the original EI formulation (16). First, y^* is replaced by the observed objective value $\tilde{y}^*$ of the current best *feasible* solution, (the method for checking feasibility and estimating the current best feasible solution are discussed in Section 3.4). Second, a chance constraint on the solution robustness is incorporated into the EI function. New points are sampled by optimizing the following constrained EI function:

P1:

$$\max_{\boldsymbol{x}} \left(\tilde{y}^* - \mu_y(\boldsymbol{x})\right)\Phi\left(\frac{\tilde{y}^* - \mu_y(\boldsymbol{x})}{\sigma_y(\boldsymbol{x})}\right) + \sigma_y(\boldsymbol{x})\phi\left(\frac{\tilde{y}^* - \mu_y(\boldsymbol{x})}{\sigma_y(\boldsymbol{x})}\right) \tag{18}$$

subject to

$$P\left(r(\boldsymbol{x}) \le \sigma_c^2 \,|\, \boldsymbol{x}, D_n\right) > 1 - \varepsilon_{EI}\,, \tag{19}$$

$$x_{l,i} \le x_i \le x_{u,i}\,,\ i = 1, \ldots, d\,, \tag{20}$$

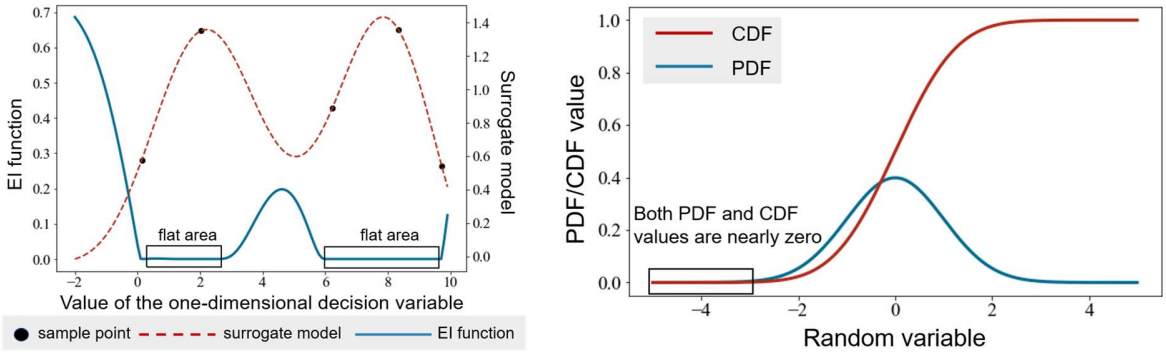
where $1 - \varepsilon_{EI}$ is a user-defined confidence level. A chance constraint (19) is introduced for two reasons. First, the feasibility of a solution $\boldsymbol{x}$ can only be definitively determined through an infinite number of simulation replications. With finite observations, feasibility can only be estimated, introducing uncertainty, particularly when the number of sample points is limited. The chance constraint can well accommodate uncertainties in the estimation. Second, the chance constraint can strike a balance between sampling in feasible and infeasible regions. Infeasible samples are also valuable (Gardner et al. 2014) as they can provide information about the search space. The chance constraint (19) is equivalent to

$$\Phi\left(\frac{\log \sigma_c^2 - \mu_{lr}(\boldsymbol{x})}{\sigma_{lr}(\boldsymbol{x})}\right) > 1 - \varepsilon_{EI}\,, \tag{21}$$

where the functions $\mu_{lr}(\cdot)$ and $\sigma_{lr}(\cdot)$ are defined in the distribution (13).

3.2.2 A two-stage method to solve the constrained EI function

Optimizing EI functions is challenging due to the presence of many "flat" areas where the EI function values are nearly zero. In Figure 3(a), we illustrate the "flat" areas using a one-dimensional (1D) test function. The x-axis represents the decision variable, while the y-axis shows the EI function (on the left) and the surrogate model (on the right). Five sample points are used to construct the surrogate model (the red dashed line). Figure 3(a) reveals the existence of two "flat" areas (the solid blue line).



(a) (b)

Figure 3. (a) The "flat" area issue in a one-dimensional test function example, (b) PDF and CDF of the standard normal distribution

This "flat" area issue is prevalent in BO algorithms when EI functions are applied (Jones et al. 1998, Tay and Osorio 2022). Due to the "flat" areas, gradient-based methods are prone to converging in the search of these areas, leading to suboptimal sampling decisions. Jones et al. (1998) addressed this issue by developing a branch-and-bound algorithm to optimize unconstrained EI functions. However, the method cannot handle constraints and has received little attention. In the literature, one of the most commonly used methods for solving EI functions is the genetic algorithm (He et al. 2017, Zhang et al. 2018). Inspired by the principles of natural selection and genetics, this approach evolves populations of candidate solutions through selection, crossover, and mutation to identify satisfactory solutions. However, local or global optimality of GA cannot be guaranteed. In this study, we develop a two-stage method to optimize the proposed constrained EI functions. The method can tackle the issue of the "flat" areas.

The basic idea of the two-stage method is to first search for feasible solutions outside of the "flat" areas, and then take these solutions as starting points to optimize the constrained EI function using gradient-based methods. A critical issue of the method lies in the first stage, that is, identifying feasible solutions outside of the "flat" areas. We have analyzed the components of EI functions in various "flat" areas and observed that, in most cases, the "flat" areas are caused where both PDF and CDF values in (18) are nearly zero (see Figure 3(b) where the x-axis is the realization of the random variable that follows the standard normal distribution, and the y-axis displays the PDF and CDF values). In the "flat" areas, the value of

$(\tilde{y}^* - \mu_y(\boldsymbol{x}))/\sigma_y(\boldsymbol{x})$ is typically small, causing PDF and CDF values close to zero. Therefore, to identify feasible solutions outside of the "flat" areas, we just need to find $\boldsymbol{x}$ such that $(\tilde{y}^* - \mu_y(\boldsymbol{x}))/\sigma_y(\boldsymbol{x})$ is larger than a predetermined threshold, e.g., -3. The first stage is to find $\boldsymbol{x}_{start}$ that satisfies the following three conditions:

$$\frac{\tilde{y}^* - \mu_y(\boldsymbol{x}_{start})}{\sigma_y(\boldsymbol{x}_{start})} > c, \tag{22}$$

$$\Phi\left(\frac{\log\sigma_c^2 - \mu_{lr}(\boldsymbol{x}_{start})}{\sigma_{lr}(\boldsymbol{x}_{start})}\right) > 1-\varepsilon_{EI}, \tag{23}$$

$$x_{l,i} \le x_i \le x_{u,i},\ i = 1,\ldots,d, \tag{24}$$

where c is the predetermined threshold that controls PDF and CDF values. A satisfactory value of $\boldsymbol{x}_{start}$ can be obtained using penalty function methods (Yeniay 2005) by introducing two auxiliary variables, a_1 and a_2.

P2:

$$\min_{\boldsymbol{x}_{start},a_1,a_2}\left(\frac{\tilde{y}^* - \mu_y(\boldsymbol{x}_{start})}{\sigma_y(\boldsymbol{x}_{start})} - c - a_1^2\right)^2 + \left(\frac{\log\sigma_c^2 - \mu_{lr}(\boldsymbol{x}_{start})}{\sigma_{lr}(\boldsymbol{x}_{start})} - z_{1-\varepsilon_{EI}} - a_2^2\right)^2 \tag{25}$$

subject to

$$x_{l,i} \le x_i \le x_{u,i},\ i = 1,\ldots,d. \tag{26}$$

Here $z_{1-\varepsilon_{EI}}$ is the quantile function value such that $\Phi(z_{1-\varepsilon_{EI}}) = 1-\varepsilon_{EI}$. The second term in the objective function (25) is corresponding to constraint (23). It is formulated according to the fact that $\Phi(\cdot)$ is monotonically increasing. Functions $\mu_y(\cdot)$, $\sigma_y(\cdot)$, $\mu_{lr}(\cdot)$, $\sigma_{lr}(\cdot)$ are constructed based on the dataset D_n, and they are defined in distributions (12) and (13). Problem **P2** can be solved using the methods of feasible directions (Bazaraa et al. 2013). Since **P2** is non-convex, multiple starting points are employed to solve the problem. Once a solution, e.g., $(\boldsymbol{x}_{start}^*, a_1^*, a_2^*)$, is obtained, $\boldsymbol{x}_{start}^*$ is examined to determine whether it complies with constraints (22)–(24). $\boldsymbol{x}_{start}^*$ is considered a feasible solution if it satisfies constraints (22)–(24). By solving **P2** with multiple starting points, it is possible to obtain several feasible values of $\boldsymbol{x}_{start}$. Let Υ_{start} denote the set of the feasible $\boldsymbol{x}_{start}$. As **P2** is non-convex, a potential but infrequent case is that solving **P2** does not yield a feasible $\boldsymbol{x}_{start}$, that is, Υ_{start} is empty. The second stage is designed to determine the value of $\boldsymbol{x}_{n+1}$ according to whether Υ_{start} is empty.

We first discuss the case that Υ_{start} is not empty. For each $\boldsymbol{x}_{start} \in \Upsilon_{start}$, satisfying constraints (22)–(24) implies that $\boldsymbol{x}_{start}$ is a feasible solution and it is located outside of the "flat" areas. Thus, in the second stage, gradient-based methods can be applied to solve problem **P1** by taking $\boldsymbol{x}_{start}$ as a starting point. Taking each $\boldsymbol{x}_{start} \in \Upsilon_{start}$ as a starting point, this study employs the Zoutendijk method to search for local minima of **P1** (Bazaraa et al. 2013).

Whenever Υ_{start} is empty, instead of optimizing the constrained EI function, a new sample is randomly selected from the region defined by the inequality (26). Incorporating random sampling into the BO framework, such as substituting random search for acquisition function optimization in some iterations, has been shown in prior literature to enhance algorithm performance (Shahriari et al. 2015). **Algorithm 1**, available in **Appendix A.1**, summarizes the two-stage method to solve the constrained EI function.

3.3 Bayesian inference of distributions

In each iteration of HRBO, a new sample is generated by optimizing the constrained EI function. Suppose HRBO has already sampled n points, the new sample is denoted as $\boldsymbol{x}_{n+1}$. Given $\boldsymbol{x}_{n+1}$ and its associated simulation evaluations, this section presents methods to estimate distributions of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$. Note that the methods discussed in this section differ from those in Section 3.1. Section 3.1 focuses on methods for "predicting" the function values of $y(\cdot)$ and $r(\cdot)$. The input is D_n. The objective is to construct surrogate models that approximate the objective and constraint functions. In contrast, this section addresses, given $\boldsymbol{x}_{n+1}$ and realizations of $f(\boldsymbol{x}_{n+1})$, how to "estimate" the values (distributions) of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$. The input in this section includes simulation evaluations at $\boldsymbol{x}_{n+1}$ and the dataset D_n. The primary goal here is to generate accurate estimates of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$ for the enrichment of the dataset and enhancement of surrogate models.

Existing studies have developed point estimates for $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$. However, the uncertainty inherent in these estimates—critical for constructing GP models and effectively allocating simulation resources (see Section 3.4)—remains largely unaddressed. For instance, consider the case where $m_{n+1} > 1$ simulation replications are performed at $\boldsymbol{x}_{n+1}$, yielding $\hat{y}(\boldsymbol{x}_{n+1})$ as the sample mean and $\hat{r}(\boldsymbol{x}_{n+1})$ as the sample variance. Most existing approaches use $\hat{r}(\boldsymbol{x}_{n+1})/m_{n+1}$ to quantify the uncertainty in $\hat{y}(\boldsymbol{x}_{n+1})$ (Ankenman et al. 2010, Dellino et al. 2012, Chen et al. 2019). This practice, however, neglects the uncertainty in estimating $r(\boldsymbol{x}_{n+1})$, leading to biased estimates, particularly when only a few observations are available. To tackle this issue, this study models $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$ as random variables and employs Bayesian inference to estimate their posterior distributions. In a fully Bayesian treatment, this study

provides two posterior distributions, termed *non-informative posterior* and *GP-based posterior*. The former is developed purely based on simulation evaluations at $\boldsymbol{x}_{n+1}$, while the latter is developed considering both evaluations at $\boldsymbol{x}_{n+1}$ and at other solutions. In practical applications, the posterior distributions can be self-adaptively selected, which will be discussed at the end of this section.

3.3.1 Non-informative posterior

Let $\boldsymbol{f}_{n+1} = \{f_{n+1}^{(1)}, ..., f_{n+1}^{(m_{n+1})}\}$ denote realizations of $f(\boldsymbol{x}_{n+1})$ within m_{n+1} simulation replications. Note that evaluation issues (e.g., determining the value of m_{n+1}) will be discussed in Section 3.4. According to Bayes' theorem, the joint posterior distribution of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$ can be derived as

$$p\left(y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1}) \middle| \boldsymbol{f}_{n+1}\right) \propto p\left(y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right) p\left(\boldsymbol{f}_{n+1} \middle| y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right), \tag{27}$$

where $p\left(y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right)$ is the joint prior distribution, and $p\left(\boldsymbol{f}_{n+1} \middle| y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right)$ represents the likelihood function of the normal distribution. In the case of *non-informative posterior*, $p\left(y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right)$ takes the widely used non-informative prior $p\left(y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right) = r(\boldsymbol{x}_{n+1})^{-1}$. This indicates that the posterior distributions of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$ are purely estimated based on $\boldsymbol{f}_{n+1}$.

Given (27) and the non-informative prior, the *non-informative posteriors* of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$ are

$$y(\boldsymbol{x}_{n+1}) \middle| \boldsymbol{f}_{n+1} \sim \mathcal{N}\left(\tilde{\mu}_{y,n+1}(m_{n+1}), \tilde{\tau}_{y,n+1}^2(m_{n+1})\right), \tag{28}$$

$$r(\boldsymbol{x}_{n+1}) \middle| \boldsymbol{f}_{n+1} \sim InverseGamma\left((m_{n+1}-1)/2, S_m/2\right), \tag{29}$$

where $\tilde{\mu}_{y,n+1}(m_{n+1}) = \overline{f}_{n+1} = \sum_{i=1}^{m_{n+1}} f_{n+1}^{(i)} / m_{n+1}$, $\tilde{\tau}_{y,n+1}^2 = r(\boldsymbol{x}_{n+1})/m_{n+1}$, $S_m = \sum_{i=1}^{m_{n+1}} (f_{n+1}^{(i)} - \overline{f}_{n+1})^2$. $\tilde{\tau}_{y,n+1}^2$ is affected by the posterior distribution of $r(\boldsymbol{x}_{n+1})$. To derive point estimates of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$, the Monte Carlo sampling technique can be applied. The reader can refer to Hoff (2009) for more details.

3.3.2 GP-based posterior

Typically, the prior distribution in Bayesian inference is difficult to determine, because it is open to subjective determination. However, the GP model provides a new way to construct objective prior distributions. The *GP-based posterior* takes the predictive distributions of GP (i.e., (12) and (13)) as prior distributions. Compared with the non-informative prior, the prior distribution constructed using GP is formulated based on evaluations of other solutions. As a result, the prior distribution can incorporate more information about SBO problems. Thus, it is expected to achieve more accurate estimates while taking fewer simulation replications. According to Bayes' theorem,

$$p\left(y(\boldsymbol{x}_{n+1}) \middle| \boldsymbol{f}_{n+1}, \boldsymbol{\mathcal{F}}_n, r(\boldsymbol{x}_{n+1})\right) \propto p\left(y(\boldsymbol{x}_{n+1}) \middle| r(\boldsymbol{x}_{n+1}), \boldsymbol{\mathcal{F}}_n\right) p\left(\boldsymbol{f}_{n+1} \middle| y(\boldsymbol{x}_{n+1}), r(\boldsymbol{x}_{n+1})\right). \tag{30}$$

$p\left(y(\boldsymbol{x}_{n+1})|r(\boldsymbol{x}_{n+1}),\mathcal{F}_n\right)$ takes the predictive distribution (12) of GP. According to (30), the *GP-based posterior* of $y(\boldsymbol{x}_{n+1})$ is

$$y(\boldsymbol{x}_{n+1})\left|\mathcal{F}_n,\boldsymbol{f}_{n+1},r(\boldsymbol{x}_{n+1})\right.\sim\mathcal{N}\left(\tilde{\mu}_{y,n+1}(m_{n+1}),\tilde{\tau}^2_{y,n+1}(m_{n+1})\right), \tag{31}$$

where $\tilde{\mu}_{y,n+1}(m_{n+1})=(\sigma_y^2(\boldsymbol{x}_{n+1})\bar{f}_{n+1}+\mu_y(\boldsymbol{x}_{n+1})r(\boldsymbol{x}_{n+1})/m_{n+1})\big/(\sigma_y^2(\boldsymbol{x}_{n+1})+r(\boldsymbol{x}_{n+1})/m_{n+1})$,

$\tilde{\tau}^2_{y,n+1}(m_{n+1})=1\big/\left(1/\sigma_y^2(\boldsymbol{x}_{n+1})+m_{n+1}/r(\boldsymbol{x}_{n+1})\right)$. Similarly, the distribution of $y(\boldsymbol{x}_{n+1})$ is affected by the posterior distribution of $r(\boldsymbol{x}_{n+1})$. To derive point estimates of $y(\boldsymbol{x}_{n+1})$, the Monte Carlo sampling technique can be applied by sampling from the posterior distribution of $r(\boldsymbol{x}_{n+1})$. The *GP-based posterior* of $r(\boldsymbol{x}_{n+1})$ is also obtained according to Bayes' theorem

$$\begin{aligned}p\left(r(\boldsymbol{x}_{n+1})\left|\mathcal{F}_n,\boldsymbol{f}_{n+1}\right.\right)&\propto p\left(r(\boldsymbol{x}_{n+1})\left|\mathcal{F}_n\right.\right)p\left(\boldsymbol{f}_{n+1}\left|r(\boldsymbol{x}_{n+1}),\mathcal{F}_n\right.\right)\\&=p\left(r(\boldsymbol{x}_{n+1})\left|\mathcal{F}_n\right.\right)\int p\left(\boldsymbol{f}_{n+1}\left|y(\boldsymbol{x}_{n+1}),r(\boldsymbol{x}_{n+1})\right.\right)\mathrm{d}y(\boldsymbol{x}_{n+1}),\end{aligned} \tag{32}$$

where $p\left(r(\boldsymbol{x}_{n+1})\left|\mathcal{F}_n\right.\right)$ is the predictive distribution of GP. Closed-form expressions on *GP-based posterior* of $r(\boldsymbol{x}_{n+1})$ are not available. To estimate its corresponding statistics, the Metropolis-Hastings (M-H) algorithm (Hoff 2009) is used.

Remark 1. Given the dataset $D_n=(X_n,\mathcal{F}_n)$, the *GP-based posterior* can be used to update the estimates of $y(\boldsymbol{x}_i)$ and $r(\boldsymbol{x}_i)$, for all $\boldsymbol{x}_i\in X_n$. Suppose we have $m_i^{ad}\in\mathbb{Z}^+$ additional realizations of $f(\boldsymbol{x}_i)$, denoted as $\boldsymbol{f}_i^{ad}$. The distributions of $y(\boldsymbol{x}_i)$ and $r(\boldsymbol{x}_i)$ are updated using Bayes' theorem:

$$p\left(y(\boldsymbol{x}_i)\left|\boldsymbol{f}_i^{ad},\mathcal{F}_n,r(\boldsymbol{x}_i)\right.\right)\propto p\left(y(\boldsymbol{x}_i)|r(\boldsymbol{x}_i),\mathcal{F}_n\right)p\left(\boldsymbol{f}_i^{ad}\left|y(\boldsymbol{x}_i),r(\boldsymbol{x}_i)\right.\right), \tag{33}$$

$$p\left(r(\boldsymbol{x}_i)\left|\mathcal{F}_n,\boldsymbol{f}_i^{ad}\right.\right)\propto p\left(r(\boldsymbol{x}_i)\left|\mathcal{F}_n\right.\right)\int p\left(\boldsymbol{f}_i^{ad}\left|y(\boldsymbol{x}_i),r(\boldsymbol{x}_i)\right.\right)\mathrm{d}y(\boldsymbol{x}_i). \tag{34}$$

In HRBO, simulation replications are iteratively allocated to samples. For $\boldsymbol{x}_i\in X_n$, once additional realizations of $f(\boldsymbol{x}_i)$ are obtained, we use (33) and (34) to update distributions of $y(\boldsymbol{x}_i)$ and $r(\boldsymbol{x}_i)$. The allocation of simulation replications will be discussed in Section 3.4.

3.3.3 Self-adaptive selection of posterior distributions

An appropriate prior distribution is crucial for deriving accurate posterior estimates. In Bayesian statistics, *prior predictive selection* techniques are usually used to evaluate the compatibility of prior distributions. The basic idea is to check if the prior distribution aligns with the observed patterns in the actual data. Motivated by these approaches, we develop a self-adaptive method to determine the posterior for estimating $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$. Specifically, the posterior distributions are selected by assessing whether GP surrogate

models can accurately model the distribution of $f(\boldsymbol{x}_{n+1})$. Given $\boldsymbol{x}_{n+1}$, the self-adaptive selection process is described as follows.

Step 1: Perform m_{n+1}^{init} initial simulation replications and obtain realizations $\{f_{n+1}^{(1)},...,f_{n+1}^{(m_{n+1}^{init})}\}$ of $f(\boldsymbol{x}_{n+1})$.

Step 2: Generate $N \geq 1$ samples of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$ from the predictive distributions of GP (i.e., the distributions (12) and (13)). Let $(\hat{y}_1,\hat{r}_1),...,(\hat{y}_N,\hat{r}_N)$ denote the samples.

Step 3: For $i=1,...,N$, randomly generate one observation from the normal distribution $\mathcal{N}(\hat{y}_i,\hat{r}_i)$. Let $\{f_{GP}^{(1)},...,f_{GP}^{(N)}\}$ denote the set of observations.

Step 4: Apply hypothesis test to examine whether $\{f_{n+1}^{(1)},...,f_{n+1}^{(m_{n+1}^{init})}\}$ and $\{f_{GP}^{(1)},...,f_{GP}^{(N)}\}$ are generated by the same distribution.

Step 5: If $\{f_{n+1}^{(1)},...,f_{n+1}^{(m_{n+1}^{init})}\}$ and $\{f_{GP}^{(1)},...,f_{GP}^{(N)}\}$ are generated by the same distribution, the *GP-based posterior* is chosen to estimate or update the distributions of $y(\boldsymbol{x}_{n+1})$ and $r(\boldsymbol{x}_{n+1})$. Otherwise, the *non-informative posterior* is taken.

Several statistical methods can be used to perform the hypothesis test in Step 4, for instance, the likelihood ratio test and the Kolmogorov-Smirnov (KS) test. In this study, the KS test is used. For details of the method, the reader can refer to Hollander et al. (2013)

3.4 Evaluation of samples

This section addresses sample evaluation. After obtaining a new sample in each iteration, multiple simulation evaluations are allocated to both the new and existing samples to estimate or update their performance measures, thereby improving the GP surrogate models. More importantly, as noted by Jalali et al. (2017), existing SBO methods suffer from the identification issue in handling heteroscedastic noise—methods frequently sample promising solutions but fail to confirm them due to insufficient replications. Thus, intelligently allocating simulation replications is crucial for accurately identifying the best solutions among all samples. Furthermore, identifying the best solution in each iteration aids sampling decisions by supporting the construction of the constrained EI function. Existing studies typically assign identical simulation replications to all samples. However, the equal allocation suffers from inefficiency, especially under budgeted computational resources. Regarding the heteroscedasticity of the simulation, this section investigates efficient allocation of simulation replications. Particularly, the aim is to design simulation resource allocation schemes such that the best feasible sample can be identified in each iteration while minimizing the number of simulation replications.

The problem of assigning simulation replications to each sample and of selecting the best one is referred to as ranking and selection (R&S) in the literature. R&S has been extensively studied over the years and

has found wide-ranging practical applications (Chen et al. 2000, Hong et al. 2015, Peng et al. 2016). For detailed discussions on R&S, readers can refer to Kim and Nelson (2006) and Hong et al. (2021). Despite the long academic pedigree along with R&S, it has been rarely discussed in the context of surrogate-based optimization frameworks. Some existing surrogate-based optimization algorithms have adopted the *optimal computing budget allocation* (OCBA) proposed by Chen et al. (2000) to inform the allocation of computational resources (Zheng et al. 2022b, Zhou et al. 2023). In OCBA, the authors focus on maximizing the probability of correct selection (PCS)—a metric that quantifies the likelihood of correctly identifying the best solution among all alternatives—through the strategic allocation of computational resources within a given budget. Since the PCS is complex to analyze directly, an approximate PCS (APCS) is formulated based on the Bonferroni inequality (Rom 1990). The APCS serves as a lower bound of PCS and is optimized under budget constraints. The allocation rules of OCBA are derived using the Lagrange method and Karush-Kuhn-Tucker conditions. However, the effectiveness of OCBA is not assured when stochastic constraints are present. Moreover, traditional R&S research primarily aims to identify the best solution from a fixed set of alternatives. This differs from the BO framework, where samples are selected sequentially, and the set of sample points expands iteratively. This study, for the first time, develops a tailored R&S scheme, namely the *adaptive simulation resource allocation scheme*, and incorporates the scheme to solve SBRO problems. The proposed scheme deals with simulation resource allocation problems within the BO framework and accounts for robustness constraints. Particularly, the proposed scheme first replicates simulation to filter out infeasible solutions, and then intelligently allocates simulation resources to feasible solutions.

Let $D_n = (X_n, \boldsymbol{\mathcal{F}}_n)$, $n \in \mathbb{Z}^+$, denote the dataset at the beginning of an iteration, where $\boldsymbol{x}^* \in X_n$ is the best solution identified by the algorithm. We focus only on cases where a best solution exists. If no such solution is found (e.g., all samples are infeasible), resource allocation simplifies, as outlined in **Algorithm 2** (**Appendix A.2**). Once sampling a new solution (sample point) $\boldsymbol{x}_{n+1}$ in the iteration, the first step is to replicate simulation evaluations at $\boldsymbol{x}_{n+1}$ to check its feasibility, i.e., whether the solution satisfies the constraint (2). Since the variance of simulation outputs is regarded as a random variable in this study, we check whether the constraint is satisfied over a predetermined confidence level $1-\varepsilon_r$. Given ε_r, the simulation is replicated at $\boldsymbol{x}_{n+1}$ until either of the following two conditions is satisfied:

$$P\left(r(\boldsymbol{x}_{n+1}) < \sigma_c^2 \,|\, \boldsymbol{f}_{n+1}, \boldsymbol{\mathcal{F}}_n\right) > 1-\varepsilon_r, \tag{35}$$

$$P\left(r(\boldsymbol{x}_{n+1}) > \sigma_c^2 \,|\, \boldsymbol{f}_{n+1}, \boldsymbol{\mathcal{F}}_n\right) > 1-\varepsilon_r, \tag{36}$$

where $\boldsymbol{f}_{n+1}$ is the set of realizations of $f(\boldsymbol{x}_{n+1})$ in the first step. The distribution of $r(\boldsymbol{x}_{n+1})|\boldsymbol{f}_{n+1}, \boldsymbol{\mathcal{F}}_n$ is

obtained from either (29) or (32).

The second step is to update the dataset D_n based on the simulation evaluations in the first step, that is, $X_{n+1} = X_n \cup \{\boldsymbol{x}_{n+1}\}$, $\boldsymbol{\mathcal{F}}_{n+1} = \boldsymbol{\mathcal{F}}_n \cup \{\boldsymbol{f}_{n+1}\}$, $D_{n+1} = (X_{n+1}, \boldsymbol{\mathcal{F}}_{n+1})$. If the inequality (36) is met in the first step, then $\boldsymbol{x}_{n+1}$ is deemed an infeasible solution, the adaptive simulation resource allocation scheme is terminated and $\boldsymbol{x}^*$ is still the best feasible solution. If the inequality (35) is finally met, $\boldsymbol{x}_{n+1}$ is regarded as feasible. Under the circumstances, the third step is implemented to allocate computational resources and identify the best solution among all samples. The third step of the adaptive simulation resource allocation scheme consists of two policies: *i*) the allocation policy that allocates computational resources to sample points, and *ii*) the selection policy that selects the best solution among all feasible solutions. In this study, we argue that computational resources can be only allocated to $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$, and that the best solution can be selected from the two solutions.

Proposition 2. If $\boldsymbol{x}_{n+1}$ is feasible, then for an arbitrary allocation policy, selecting the best solution from $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$ achieves a higher lower bound of PCS than OCBA.

The proof of **Proposition 2** is available in **Appendix B**. **Proposition 2** implies that to identify the best solution among all samples in each iteration, we just need to evaluate whether the new sample, if it is feasible, is better than the historical best one. Therefore, given a predetermined significance level ε_y, the adaptive simulation resource allocation scheme allocates additional simulations to $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$, if necessary, until either of the following two conditions is satisfied (Peng et al. 2016)

$$P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*) \middle| \boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right) > 1-\varepsilon_y, \tag{37}$$

$$P\left(y(\boldsymbol{x}_{n+1}) > y(\boldsymbol{x}^*) \middle| \boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right) > 1-\varepsilon_y, \tag{38}$$

where $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$ are the new and historical best solutions, respectively. $\boldsymbol{f}_{n+1}^{ad}$ ($\boldsymbol{f}_*^{ad}$) is the set of additional simulation evaluations at $\boldsymbol{x}_{n+1}$ ($\boldsymbol{x}^*$). The posterior distributions of $y(\boldsymbol{x}_{n+1})$ and $y(\boldsymbol{x}^*)$ are derived from (31). If the inequality (37) is met, then $\boldsymbol{x}_{n+1}$ is the best solution. If the inequality (38) is met, $\boldsymbol{x}^*$ is the best solution.

A critical issue in the third step is to determine the allocation policy. We aim to execute the least number of simulation replications such that either the inequality (37) or (38) is satisfied. Let m_1 and m_2 denote the simulation replication numbers at $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$, respectively. The optimal values of m_1 and m_2 can be obtained by solving the following optimization problem:

P3:

$$\min_{m_1,m_2,u} m_1+m_2 \tag{39}$$

subject to

$$\begin{aligned} &u\left(1-\varepsilon_y-P\left(y(\boldsymbol{x}_{n+1})<y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)\right)\\ &+(1-u)\left(1-\varepsilon_y-P\left(y(\boldsymbol{x}_{n+1})>y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)\right)\le 0, \end{aligned} \tag{40}$$

$$m_1,m_2\in\mathbb{Z}^+,u\in\{0,1\}, \tag{41}$$

According to the law of large numbers, $P\left(y(\boldsymbol{x}_{n+1})<y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)$ will finally converge to 0 or 1 as the number of simulation replications increases. Thus, **P3** has solutions. To solve **P3**, we relax the integer constraints on m_1 and m_2.

Proposition 3. Suppose that m_1 and m_2 are continuous variables, **P3** achieves its local optimal solutions at the boundary of its feasible set.

The proof of **Proposition 3** is available in **Appendix C**. **Proposition 3** demonstrates that **P3** can be reformulated as the following **P4** by regarding m_1 and m_2 as continuous variables,

P4:

$$\min_{m_1,m_2,u} m_1+m_2 \tag{42}$$

subject to

$$\begin{aligned} &u\left(1-\varepsilon_y-P\left(y(\boldsymbol{x}_{n+1})<y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)\right)\\ &+(1-u)\left(1-\varepsilon_y-P\left(y(\boldsymbol{x}_{n+1})>y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)\right)=0, \end{aligned} \tag{43}$$

$$u\in\{0,1\},m_1\ge 0,m_2\ge 0. \tag{44}$$

P4 can be solved using the Lagrange multiplier method (Bertsekas 2014) while taking u as 0 and 1 respectively. While implementing the Lagrange multiplier method, an intricate issue is that the values of $P\left(y(\boldsymbol{x}_{n+1})>y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)$ and $P\left(y(\boldsymbol{x}_{n+1})<y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)$ are obtained only after m_1+m_2 simulations are conducted (i.e., after obtaining $\boldsymbol{f}_{n+1}^{ad}$ and $\boldsymbol{f}_*^{ad}$). One way to tackle this problem is to perform Monte Carlo simulations. However, the Monte Carlo simulation can be time-consuming. To efficiently solve **P4**, this study approximates $P\left(y(\boldsymbol{x}_{n+1})>y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)$ and $P\left(y(\boldsymbol{x}_{n+1})<y(\boldsymbol{x}^*)\middle|\boldsymbol{\mathcal{F}}_{n+1},\boldsymbol{f}_{n+1}^{ad},\boldsymbol{f}_*^{ad}\right)$ using GP surrogate models. Since simulation evaluations in the first step have been incorporated into the dataset (in the second step of the proposed scheme), it would be reasonable to expect that the GP surrogate models could provide relatively accurate estimates.

Proposition 4. Let $\bar{f}_{n+1}$ and $\bar{f}_*$ denote the mean of elements in $\boldsymbol{f}_{n+1}^{ad}$ and $\boldsymbol{f}_*^{ad}$, respectively. Assume that $\bar{f}_{n+1} = \mu_y(\boldsymbol{x}_{n+1})$, $\bar{f}_* = \mu_y(\boldsymbol{x}^*)$ in (31). Let m_1^* and m_2^* denote the optimal simulation replication numbers obtained by solving **P4**. m_1^* and m_2^* have the following relationship,

$$\frac{\hat{r}(\boldsymbol{x}_{n+1})^{1/2}}{m_1^* + m_1^p} = \frac{\hat{r}(\boldsymbol{x}^*)^{1/2}}{m_2^* + m_2^p}, \tag{45}$$

where $m_1^p = \hat{r}(\boldsymbol{x}_{n+1}) / \sigma_y^2(\boldsymbol{x}_{n+1})$, $m_2^p = \hat{r}(\boldsymbol{x}^*) / \sigma_y^2(\boldsymbol{x}^*)$, and $\hat{r}(\cdot)$ is the expectation of $r(\cdot)|\mathcal{F}_{n+1}$.

The proof of **Proposition 4** is available in **Appendix D**. **Proposition 4** gives an allocation rule for the SBRO problem. It is exactly derived by using $P\left(\cdot|\mathcal{F}_{n+1}\right)$ to approximate $P\left(\cdot|\mathcal{F}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right)$. In the third step of the adaptive simulation resource allocation scheme, we iteratively allocate resources based on Eq. (45). In each iteration, m_1 is set to a constant value, and m_2 can be obtained by substituting m_1 into Eq. (45). Then, $\lfloor m_1 \rfloor$ and $\lfloor m_2 \rfloor$ additional simulation replications are conducted at $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$ respectively, where $\lfloor \cdot \rfloor$ denotes the floor function. Simulation replications are iteratively conducted until either (37) or (38) is satisfied. Note that we only use Eq. (45) to inform the allocation of simulation resources, but do not adopt exact solutions derived from **P4**. This is because when $y(\boldsymbol{x}_{n+1})$ and $y(\boldsymbol{x}^*)$ are close, m_1^* and m_2^* become relatively large, indicating that substantial simulation resources are required to determine the superior solution. However, from a practical perspective, distinguishing between such similar solutions is often unnecessary, as their real-world impacts are likely indistinguishable. To address this, we preset m_1 and determine m_2 proportionally, effectively controlling the computational resources dedicated to each sample. This approach can limit the maximum number of simulations, preventing excessive resource consumption. Furthermore, m_1^* and m_2^* are estimated based on current observations, and the optimal allocation may change as more observations are available. Consequently, we allocate only a small portion of simulation resources in each iteration, adopting a sequential resource allocation strategy.

Algorithm 2, available in **Appendix A.2**, depicts detailed processes of the adaptive simulation resource allocation scheme. Details of the HRBO method are summarized in **Algorithm 3** (see **Appendix A.3**).

4 Numerical Experiments

In this section, we conduct numerical experiments to validate the proposed methods. Experiments on a test function are first performed to verify the superior performance of the proposed techniques. The HRBO method is then evaluated using a simulation-based continuous network design problem (CNDP) and a variant of the M/M/1 queuing model. This section provides experimental results for the test function and the CNDP, while experiments on the M/M/1 queuing model are presented in **Appendix E**.

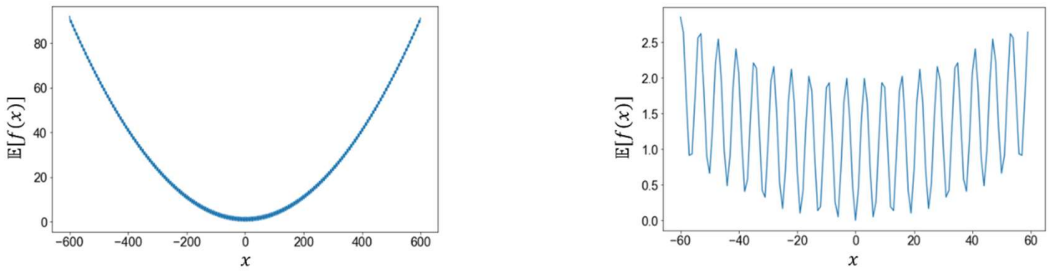
There are two sources of randomness in SBRO algorithms: *i*) the selection of initial sample points and multiple starting locations for optimization procedures (e.g., the first stage of the two-stage method) and *ii*) the stochasticity of simulation outputs. In the following experiments, to ensure consistency and comparability among different methods, both sources of randomness are controlled using random seeds. Initial sample points and starting points are generated via Latin Hypercube Sampling (LHS) with a fixed random seed applied in each run to maintain consistency across optimization algorithms. For the stochasticity of simulation, we adopt a two-level random seed generation mechanism, where an upper-level random number generator (RNG) is initialized with a fixed seed at the start of each run, generating a consistent sequence of lower-level seeds. Each time a simulation is invoked, a new lower-level seed is drawn and passed to the simulation model. To ensure fair comparisons, all algorithms share the same upper-level RNG seeds, ensuring identical sequences of lower-level seeds. These settings allow algorithms to be evaluated under identical stochastic conditions, enhancing comparability and reproducibility.

4.1 Numerical experiments on a test function

In this study, we propose and integrate three SBO techniques within the HRBO framework: *i*) the Gaussian process incorporating Bayesian posterior inference, *ii*) the constrained EI formulation and the two-stage algorithm, and *iii*) the adaptive simulation resource allocation scheme. This section aims to validate the performance of the proposed techniques through numerical experiments on the Griewank function, a notably non-linear, non-convex function characterized by numerous widely dispersed local minima and maxima throughout the design space. Gaussian noise is added to the function (Gu et al. 2024).

$$f(\boldsymbol{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + \varepsilon(\boldsymbol{x}), \tag{46}$$

where d is the number of dimensions of the input $\boldsymbol{x}$. $\varepsilon(\boldsymbol{x})$ is the Gaussian noise, $\varepsilon(\boldsymbol{x}) \sim \mathcal{N}(0, \tau^2(\boldsymbol{x}))$. We will discuss settings of $\tau(\boldsymbol{x})$ later. The function is defined over the hypercube, $\boldsymbol{x} \in [-600, 600]^d$. Experiments are performed on the one-dimensional noisy Griewank function. The noise-free Griewank function is depicted in Figure 4, which illustrates its numerous local minima and maxima.



(a) $x \in [-600, 600]$ (b) $x \in [-60, 60]$

Figure 4. One-dimensional noise-free Griewank function

This study follows the noise settings in Jalali et al. (2017), where the maximum and minimum values of $\tau(\boldsymbol{x})$ are linked to the range of the objective function and varies linearly across the design space. Let R_f denote the range. This study considers two levels of heteroscedastic noise: the *light level* and the *heavy level*. For the light level, $\tau(\boldsymbol{x})$ varies between 15% and 60% of R_f. With heavy level, $\tau(\boldsymbol{x})$ varies between 150% and 600% of R_f. Additionally, we consider two cases of noise structures, the *best case* and the *worst case*. In the best case, $\tau(\boldsymbol{x})$ linearly increases with the objective function (i.e., the optimal solution has the lowest variance). In the worst case, $\tau(\boldsymbol{x})$ linearly decreases with the objective function. In this study, the SBRO problem is set as follows

$$\min_{x} \mathbb{E}[\![f(\boldsymbol{x})]\!] \tag{47}$$

subject to

$$\mathbb{VAR}[\![f(\boldsymbol{x})]\!] \le \sigma_c^2, \tag{48}$$

For the light level of noise, σ_c is set to 35% of R_f, and with heavy level, the value increases to 350% of R_f. Settings on $\tau(\boldsymbol{x})$ and σ_c are summarized in Table 1. In Scenarios 1 and 3, the global optimal objective value is 0.0, and the value is 28.3 in Scenarios 2 and 4.

Table 1. Settings on the heteroscedasticity and robustness constraints

Scenario (noise level/structure)	Noise level*	Noise structure**	σ_c
1 (Light-Best)	Light	Best case	$0.35R_f$
2 (Light-Worst)	Light	Worst case	$0.35R_f$
3 (Heavy-Best)	Heavy	Best case	$3.5R_f$
4 (Heavy-Worst)	Heavy	Worst case	$3.5R_f$

* Light level: $\min \tau(\boldsymbol{x}) = 0.15R_f$, $\max \tau(\boldsymbol{x}) = 0.6R_f$; Heavy level: $\min \tau(\boldsymbol{x}) = 1.5R_f$, $\max \tau(\boldsymbol{x}) = 6R_f$;

** the Best case: $\min \tau(\boldsymbol{x})$ at $\min \mathbb{E}[\![f(\boldsymbol{x})]\!]$, and $\max \tau(\boldsymbol{x})$ at $\max \mathbb{E}[\![f(\boldsymbol{x})]\!]$; the Worst case: $\min \tau(\boldsymbol{x})$ at $\max \mathbb{E}[\![f(\boldsymbol{x})]\!]$, and $\max \tau(\boldsymbol{x})$ at $\min \mathbb{E}[\![f(\boldsymbol{x})]\!]$;

4.1.1 Performance of Gaussian process with Bayesian inference

In this section, we aim to explore whether the proposed GP models achieve higher accuracy by incorporating the Bayesian inference. Consequently, the accuracy of the proposed surrogate models is evaluated and compared with benchmark surrogate models. To further investigate the impact of posterior selection, three GP surrogate models using different posterior approaches are considered: non-informative posteriors, GP-based posteriors, and self-adaptively selected posteriors (denoted as GP-N, GP-G, and GP-A, respectively).

For the surrogate model of the objective function, the stochastic kriging model is taken as the benchmark model (Ankenman et al. 2010). The accuracy is evaluated using the mean square error (MSE),

which is calculated as

$$MSE = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (\mathbb{E}[\![f(\boldsymbol{x}_i)]\!] - \hat{y}_{ij})^2 , \tag{49}$$

where N_1 is the number of test inputs $\{\boldsymbol{x}_i\}_{i=1}^{N_1}$ that are generated at equidistant intervals within the feasible domain. For each test input $\boldsymbol{x}_i$, we randomly generate N_2 estimates of $\mathbb{E}[\![f(\boldsymbol{x}_i)]\!]$, denoted by $\{y_{ij}\}_{j=1}^{N_2}$, from the predictive distribution $\mathcal{N}(\mu_y(\boldsymbol{x}_i),\ \sigma_y^2(\boldsymbol{x}_i))$ that is generated by the proposed surrogate model. In this experiment, N_1 and N_2 are set to 1000 and 100, respectively.

For the surrogate model of the constraint function, its accuracy is evaluated by comparing it to that of the logarithmic surrogate function (Lázaro-Gredilla and Titsias 2011), with no Bayesian inference incorporated. The accuracy is evaluated using the 0-1 loss function.

$$L = \frac{1}{N_1} \sum_{i=1}^{N_1} \delta(\hat{\varsigma}(\boldsymbol{x}_i), \varsigma(\boldsymbol{x}_i)) , \tag{50}$$

where $\hat{\varsigma}(\boldsymbol{x}_i)$ denotes the estimate regarding the feasibility of $\boldsymbol{x}_i$. $\hat{\varsigma}(\boldsymbol{x}_i)$ equals one if $\boldsymbol{x}_i$ is deemed feasible and zero if $\boldsymbol{x}_i$ is deemed infeasible. Specifically, the feasibility of $\boldsymbol{x}_i$ is estimated using Eq. (35) and Eq. (36) with ε_r set to 0.1. If neither condition is satisfied, $\hat{\varsigma}(\boldsymbol{x}_i)$ is assigned a null value, resulting in losses. $\varsigma(\boldsymbol{x}_i)$ indicates the real feasibility of $\boldsymbol{x}_i$, and it equals one if $\boldsymbol{x}_i$ is feasible and zero otherwise.

For each scenario in Table 1, we solve the SBRO problem 10 times. Specifically, to mitigate the influence of other techniques, in each run, we manually generate 20 sample points in a random sequence. Each point is assigned the same number of simulation replications, specifically 50. Based on these points, we construct surrogate models for the objective and constraint functions. Note that these sample points do not guarantee that the surrogate model achieves satisfactory fitting accuracy for solving optimization problems. This experiment only compares the accuracy of surrogate models with identical sample points. The average MSEs and 0-1 losses are reported in Table 2.

The results show that among the four surrogate models for the objective function, GP-A achieves the lowest MSE in Scenarios 1-3. For the variance constraint, GP-A demonstrates significantly lower 0-1 loss compared to the benchmark model and performs comparably to GP-G, which exhibits optimal performance across all four scenarios. These findings underscore the effectiveness of incorporating Bayesian inference into GP surrogate models for improving both objective function approximation and solution feasibility assessment.

Our analysis reveals that posterior selection substantially influences surrogate model performance. In Scenarios 1 and 2, neither non-informative posteriors nor GP-based posteriors yield improvements over stochastic kriging in expectation modeling. However, the GP-A model, utilizing appropriately selected

posterior estimates, significantly outperforms stochastic kriging, highlighting the efficacy of our self-adaptive posterior selection approach. Under heavy-noise conditions (Scenarios 3 and 4), all proposed GP surrogate models demonstrate superior precision compared to stochastic kriging, with the performance gap widening as noise levels increase. This finding indicates that our posterior estimation approach is particularly advantageous for expectation and variance estimation in scenarios characterized by high noise levels and computational constraints.

Compared to the non-informative posterior, the GP-based posterior is more suitable for dealing with heavy noise, which aligns with theoretical expectations given the substantial simulation resources required in heavy-noise environments. When simulation resources are limited, the GP-based posterior leverages information from other samples to enhance estimation accuracy and consequently improves surrogate model precision. Furthermore, the GP-based posterior proves particularly effective in modeling constraint functions.

Table 2. The accuracy of surrogate models

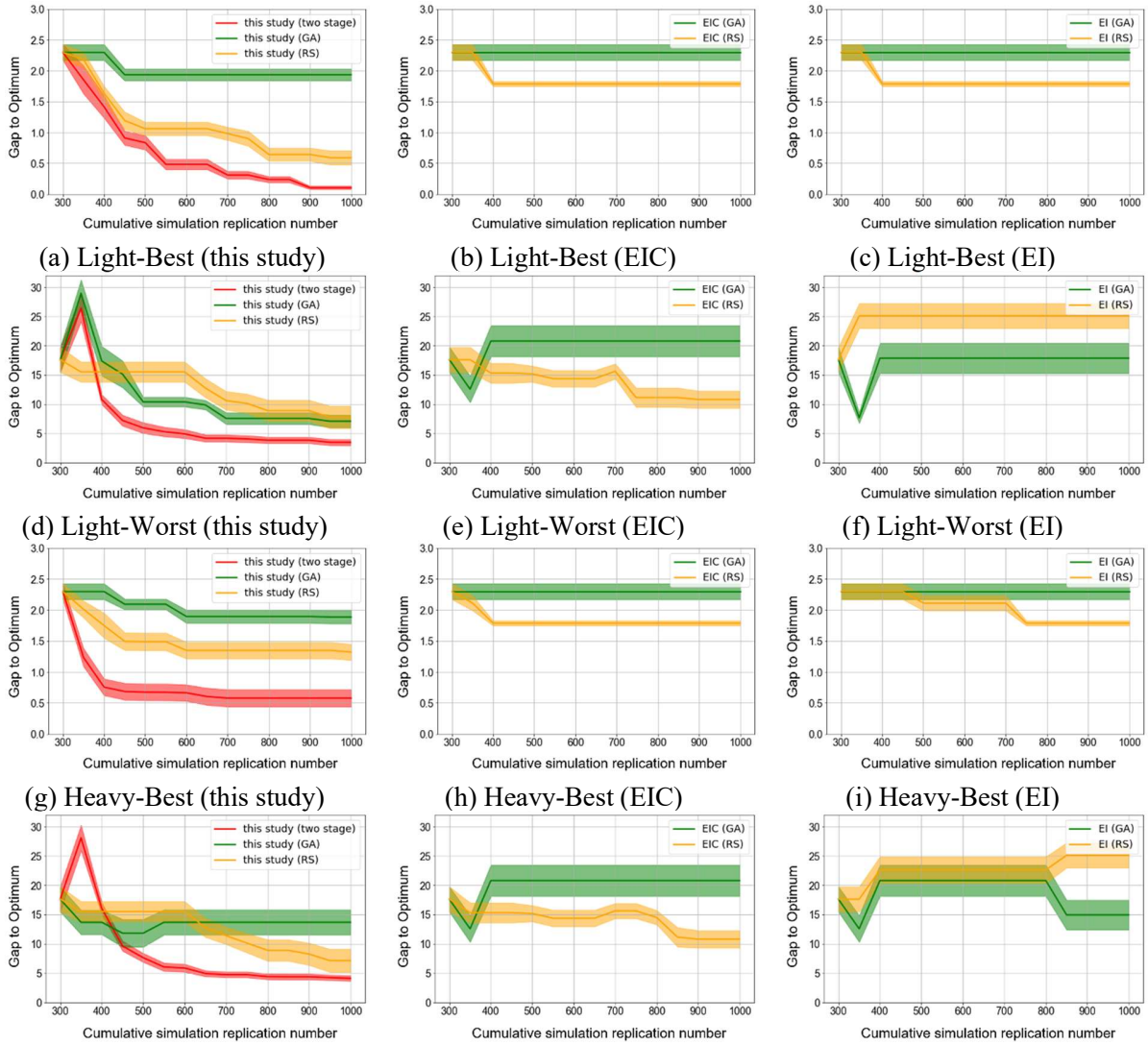
Average MSE of surrogate models for the objective function				
Scenarios (noise level/structure)	GP-A	GP-N	GP-G	The stochastic kriging
Scenario 1 (Light-Best)	25.83	28.31	38.94	27.99
Scenario 2 (Light-Worst)	23.52	26.33	28.44	24.87
Scenario 3 (Heavy-Best)	1066.85	1132.87	1197.28	1167.66
Scenario 4 (Heavy-Worst)	887.54	1092.42	868.33	1191.33
Average 0-1 loss of surrogate models for the variance constraint				
Scenarios (noise level/structure)	GP-A	GP-N	GP-G	The benchmark model
Scenario 1 (Light-Best)	0.15	0.17	0.13	0.61
Scenario 2 (Light-Worst)	0.14	0.16	0.13	0.64
Scenario 3 (Heavy-Best)	0.17	0.18	0.15	0.58
Scenario 4 (Heavy-Worst)	0.15	0.17	0.15	0.63

4.1.2 Validation of the proposed EI formulation and the optimization method

This section validates the effectiveness of the proposed EI formulation and the two-stage method. Two benchmark EI formulations are considered for comparison. The first, a conventional EI function, does not consider noise or constraints. The second, the EI with constraints (EIC) function, modifies the conventional EI by incorporating the probability of feasibility (Bagheri et al. 2017). In our proposed EI formulation, the parameter ε_{EI} in the chance constraint (19) is set to 0.1. The parameter c in the constraint (22) is set to -3.

We optimize all three EI formulations using two benchmark optimization methods: the GA and the random search (RS) method. The GA is configured with a population size of 20 and set to run for 5 iterations. In the RS method, we uniformly and randomly sample 100 solutions from the search space and

designate the feasible solution with the maximal EI value as the sample point (if no feasible solutions are found, the solution with the maximal EI value is selected). Additionally, the two-stage method is specifically applied to our proposed EI formulation for enhanced comparison, with the number of starting points set to 10. With different EI formulations and optimization methods, we compare HRBO methods using the same initial samples and computational resources: six initial samples, 50 simulation replications per sample, and a total of 1,000 replications for each method (i.e., a total of 20 samples). Each method is tested 10 times on the SBRO problem.



(a) Light-Best (this study) (b) Light-Best (EIC) (c) Light-Best (EI)

(d) Light-Worst (this study) (e) Light-Worst (EIC) (f) Light-Worst (EI)

(g) Heavy-Best (this study) (h) Heavy-Best (EIC) (i) Heavy-Best (EI)

(j) Heavy-Worst (this study) (k) Heavy-Worst (EIC) (l) Heavy-Worst (EI)

Figure 5. Convergence curves of different EI formulations and optimization algorithms

Figure 5 presents the convergence curves of various EI formulations and optimization methods across different test scenarios. Each row in the figure corresponds to a test scenario, and each panel within a row represents a distinct EI formulation. The convergence curves are color-coded as follows: red for the two-

stage method proposed in this study, green for the GA, and orange for the RS method. For each convergence curve, the x-axis represents the cumulative number of simulation replications, and the y-axis indicates the *gap to optimum* (also known as *simple regret* in the BO literature). The gap to optimum is measured as the distance between the *current best objective value* and the global optimal objective value. Notably, the current best objective value refers to the minimal objective function value among all feasible samples, rather than the objective value of the best identified sample. This distinction allows us to evaluate the effectiveness of sampling by excluding identification issues, such as cases where the best identified sample is not the true optimal solution. Solid lines represent the average values from 10 runs. The figure also illustrates the variance among these runs.

Some convergence curves in Figure 5 show fluctuations due to changes in the number of feasible solutions. For example, the red curve in Figure 5(d)—representing the proposed EI formulation solved by the two-stage method—demonstrates an increase in feasible solutions from six to eight between 300 and 350 simulations, causing variability in mean accuracy. Figure 5 illustrates that, for each optimization method (GA and RS), the proposed EI formulation consistently achieves higher accuracy than the other two EI formulations across all four test scenarios, highlighting the effectiveness of the proposed EI formulation. In terms of optimization methods, the two-stage method excels, achieving higher accuracy than both the GA and RS methods, as shown in Figures 5(a), 5(d), 5(g), and 5(j). For each combination of EI formulation and optimization method, 560 samples were generated across the four test scenarios over 10 runs (i.e., $14\times4\times10$). Table 3 displays the percentage of each combination's success in identifying local optima, demonstrating the two-stage method's efficiency in most cases.

Table 3. Percentage of identifying local optima within all samples

EI formulation	**EI**	**EI**	**EIC**	**EIC**	**this study**	**this study**	**this study**
Optimization method	GA	RS	GA	RS	GA	RS	Two-stage
Percentage	76%	81%	73%	68%	73%	65%	**83%**

4.1.3 Validation of the simulation resource allocation method

This section validates the efficiency of the proposed adaptive simulation resource allocation scheme. We aim to verify that the proposed scheme can utilize computational resources more efficiently, thereby achieving better solutions with limited computational resources. To this end, we compare the performance of the HRBO algorithm with and without the allocation scheme. For brevity, these two methods are referred to as HRBO-A (with allocation scheme) and HRBO-NA (without allocation scheme). For each test scenario, both methods are evaluated under the same computational budget: six initial samples are assigned to each method, with each initial sample undergoing 50 simulation replications. The total computational budget is set to 1,800 simulation replications. Consequently, after evaluating the initial samples, both methods have 1,500 simulation replications remaining ($1800-6\times50$). HRBO-NA assigns a

fixed 50 simulation replications per sample, whereas HRBO-A employs the adaptive simulation resource allocation scheme. In this scheme, the parameters ε_r and ε_y are set to 0.05 and 0.1, respectively. In HRBO-A, the initial and maximum number of simulation replications per sample are set to 20 and 150, respectively. In each test scenario, both methods are evaluated for five times.

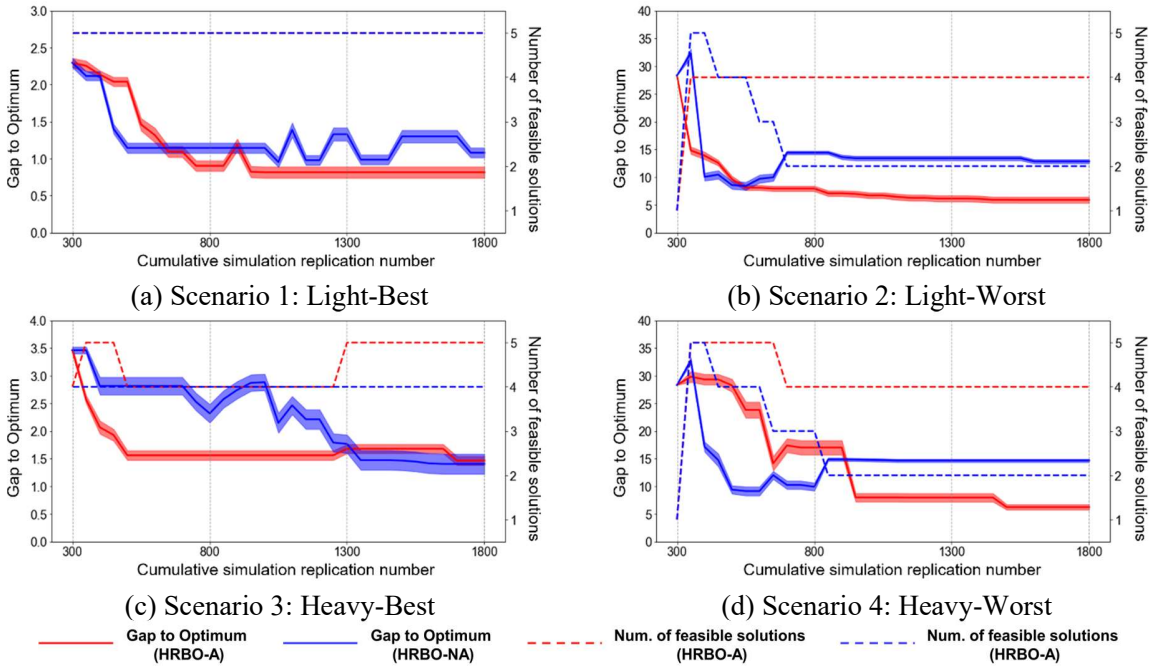


Figure 6. Performance comparison of HRBO with and without the adaptive simulation resource allocation scheme

Figure 6 illustrates the performance of each method under different computational budgets across four test scenarios. Red lines represent HRBO-A, and blue lines represent HRBO-NA. In each panel, the x-axis represents the cumulative number of simulation replications, while the y-axis shows the performance of the two methods. The performance is evaluated using two metrics: *i*) the deviation of the best identified solution's objective value from the optimal objective value, i.e., the gap to optimum, and *ii*) the number of feasible solutions among all best identified solutions. Note that only feasible solutions are considered when calculating the first metric, resulting in fluctuations in the gap to optimum curves. These two metrics are displayed on the left and right y-axis of each panel, respectively. As we have discussed, simulation noise can lead to identification issues where the best identified solution may not represent the true optimal one. In practical applications, enumerating all samples to determine the true optimal solution is often infeasible, making the best identified solution more valuable for real-world practice. Additionally, the proposed simulation resource allocation scheme is designed to identify the best solution in each iteration. Measuring the best identified solutions effectively assesses the performance of the proposed allocation scheme. For analytical purposes, the computational budget is categorized into three levels: low (300–800),

medium (800–1300), and high (1300–1800), as shown on the x-axis.

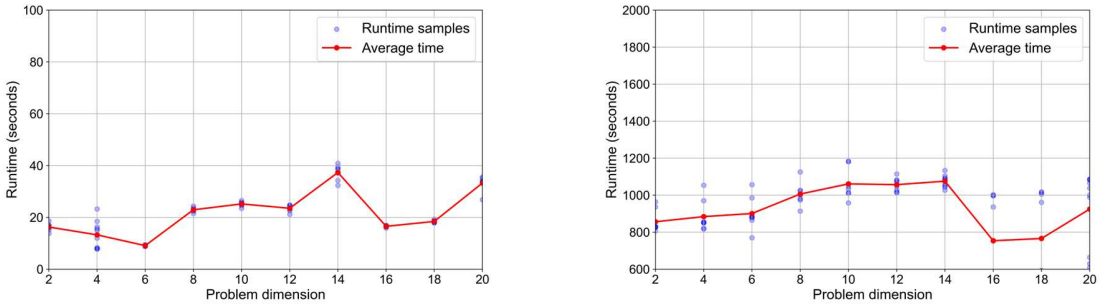
The results in Figure 6 illustrate the significant advantages of the adaptive simulation resource allocation scheme (HRBO-A) over the baseline method (HRBO-NA) under varying computational budgets. From the perspective of solution accuracy, HRBO-A and HRBO-NA perform similarly under low-budget conditions. However, in medium- and high-budget cases, HRBO-A consistently outperforms HRBO-NA across almost all test cases, demonstrating its ability to leverage additional computational resources to achieve superior solution accuracy. Furthermore, under varying computational budgets, HRBO-A identifies a greater number of feasible solutions in nearly all test scenarios, reflecting its capability to effectively allocate resources to explore the feasible region more thoroughly.

4.1.4 Runtime compared with traditional BO

This experiment assesses the runtime performance of HRBO. Our motivation stems from the fact that this study builds upon BO with a series of enhancements. While these improvements are designed to enhance accuracy, feasibility, and robustness, it is equally important to evaluate their impact on computational cost. Specifically, we aim to compare the runtime of the proposed method with that of traditional BO under identical simulation resources. Moreover, we examine how this comparison varies across different problem dimensions. To this end, we conducted 10 independent runs across varying problem dimensions to measure the computation time required by both HRBO and traditional BO, using the same simulation budget: six initial samples, 50 simulation replications per initial sample, and a total of 1,000 replications for each method. For dimension *d*, the search spaces for both methods are $[-600, 600]^d$. The traditional BO method considered in this study ignores simulation noise and does not account for variance constraints. Each sample in the traditional BO method is evaluated using 50 simulation replications. In this experiment, we focus solely on the algorithm's runtime rather than solution accuracy. The runtime measurement includes the execution time of the algorithm itself, excluding the simulation time (i.e., test function evaluations). For brevity, we present results only for the heavy-best test scenario, as experiments on other scenarios yield similar findings. Figure 7 represents the runtime of both methods under different problem dimensions. The blue dots represent the runtime samples in different algorithm executions and the red dots represent the average runtime under different problem dimensions.

As shown in Figure 7, HRBO requires more computation time than BO. However, as the problem dimension increases, the execution time of both algorithms remains stable. This indicates that, in terms of runtime, HRBO maintains strong scalability across different problem dimensions. Across all experiments, the average execution time of HRBO is around 1,000 seconds, while that of BO is 21 seconds. Given that in each algorithm run, 700 simulation replications are allocated ($1000 - 6 \times 50$), HRBO incurs an additional runtime of 1.4 (i.e., $1000/700 - 21/700$) seconds per simulation replication compared to BO. This

additional time is negligible relative to the typical runtime of urban traffic simulations, which can take several hours per run. With this slight increase in runtime, HRBO significantly improves the robustness and feasibility of the solutions, resulting in substantial gains (see Section 4.2 for details on its practical impact on solution quality).



(a) Traditional BO method (b) HRBO

Figure 7. Runtime of BO and HRBO under different problem dimensions (Scenario 3: Heavy-Best)

4.2 Simulation-based robust optimization of a continuous network design problem

In this section, experiments are performed to validate the effectiveness of the HRBO method on a real-world transportation problem, a continuous network design problem. We first briefly introduce the CNDP and the parameter settings. Then, experimental results are presented and discussed.

4.2.1 Construction of the SBRO problem and parameter settings

The continuous network design problem aims to maximize the operational efficiency of transportation systems by modifying design variables such as link capacities, tolls, or free-flow speeds (FFSs) (Li et al. 2012, Wang et al. 2022). The CNDP is usually formulated as a bilevel programming model (Meng et al. 2001), in which the upper level determines the network design scheme aiming at maximizing social welfare, while the lower level, which is a traffic assignment model, accounts for the resulting travel behaviors of travelers under the given network design scheme. Social welfare is generally measured using the total cost, which is a weighted sum of the travel costs and the investment costs. In this study, the traffic assignment model is implemented by stochastic traffic simulation, i.e., performing simulation-based dynamic traffic assignment (DTA). In conventional CNDP research, where the lower-level model relies on analytical traffic assignment, link capacity is typically used as the decision variable. However, most microscopic simulators do not allow direct configuration of link capacity as an input parameter. To tackle this issue, this study adopts the free-flow speed as the decision variable. Regarding heteroscedastic simulation noise, the objective is to determine the optimal FFSs while maintaining oscillations of total costs within a predetermined level. The CNDP investigated in this study can be mathematically formulated

as:

Upper level:

$$\min_{\boldsymbol{x}} C(\boldsymbol{x}) = \min_{\boldsymbol{x}} \left(\mathbb{E}\left[w_1 \sum_{i=1}^{N} t_i\left(\boldsymbol{x}, \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\xi})\right)/N + w_2 g(\boldsymbol{x}) \right] \right) \tag{51}$$

subject to

$$\mathbb{VAR}\left[w_1 \sum_{i=1}^{N} t_i\left(\boldsymbol{x}, \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\xi})\right)/N + w_2 g(\boldsymbol{x}) \right] \le \sigma_c^2, \tag{52}$$

$$x_{l,i} \le x_i \le x_{u,i},\ i = 1, ..., d, \tag{53}$$

Lower level:

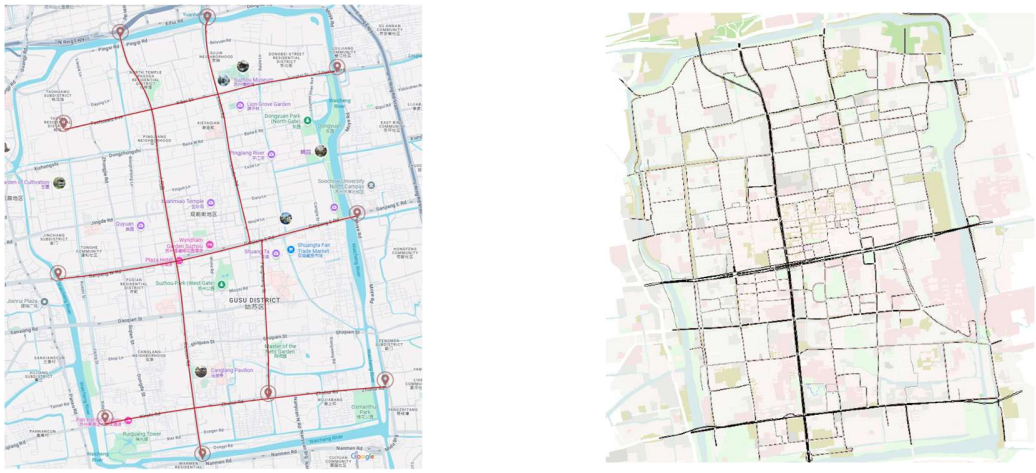
$$\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\xi}) = \arg\min_{\boldsymbol{v}} T\left(\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{\xi}\right) \tag{54}$$

where $\boldsymbol{x}$ is a d-dimensional decision vector, $\boldsymbol{x} = (x_1, ..., x_d)$, $d \in \mathbb{Z}^+$, which defines FFSs on predetermined d links. $t_i(\cdot)$ is the travel time of traveler $i \in \{1, 2, ..., N\}$, where N denotes the total number of travelers in the system. The value of $t_i(\cdot)$ is determined by two elements: the decision vector $\boldsymbol{x}$, and the (approximate) equilibrium traffic flow $\boldsymbol{v}$ that is solved from the lower-level simulation-based DTA model. The function $g(\cdot)$ quantifies the investment costs associated with increasing FFSs. w_1 and w_2 denote weights of the travel cost and investment cost, respectively. $x_{u,i}$ ($x_{l,i}$) defines the upper (lower) bounds of x_i, $i = 1, ..., d$.

Given a network design scheme $\boldsymbol{x}$, the lower-level problem is to solve a simulation-based DTA model to obtain the equilibrium traffic flow $\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\xi})$ that satisfies the dynamic user equilibrium condition. The dynamic user equilibrium condition is a state in which, for each origin-destination pair and departure time, all utilized paths have equal and minimal experienced travel costs, and no traveler can reduce their cost by unilaterally changing their route (Levin et al., 2015). Specifically, travelers are modeled with high resolution in a simulation, making travel decisions (e.g., route choice and car-following) based on the FFSs determined at the upper level, aiming to minimize their travel costs. For brevity, we frame the simulation-based DTA problem as an optimization model, which abstracts away from the details of simulation models. The optimization problem is formulated with the objective function $T(\cdot)$. We assume that the simulation network— including the intersections, lane geometries, and signal control settings— as well as the travel demand are predetermined and these factors have been incorporated into the formulation of the function $T(\cdot)$. We use $\boldsymbol{\xi}$ to denote the set of random variables representing the source of uncertainty (e.g., the random desired speed for each traveler) in the simulation model. Due to the stochasticity of simulation, given $\boldsymbol{x}$, different realizations of $\boldsymbol{\xi}$ lead to different values of $\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{\xi})$. Consequently, stochasticity exists in the total cost. To account for stochasticity, we aim to minimize the expected total cost, $C(\cdot)$, while ensuring that its variance remains below a predetermined threshold σ_c^2.

Due to the complexity of the simulation model, the lower-level DTA can be regarded as a black box and it can be embedded in the upper level. In this case, the CNDP can be abstracted into the SBRO problem described in Section 2.

We consider the CNDP within the inner city of Suzhou. Figure 8 shows the traffic network layout, encompassing 8,176 nodes and 22,778 directed links. OD demands are generated based on the automated license plate recognition data from intersections and junctions, recorded on Oct. 11, 2021. Five main roads within the network are chosen for FFS enhancement (highlighted in Figure 8(a)). Thus, the dimension of the decision vector is five in this experiment. In this study, the popular microscopic simulator SUMO (Krajzewicz 2010) is used in the CNDP and the simulation network is shown in Figure 8(b). Due to the high computational cost of simulations and the need to perform thousands of simulations in this experiment, we set the study period to a 30-minute peak-hour interval (8:30–9:00). The first five minutes of this period are designated as the simulation warm-up time. For brevity, we assume that the cost of increasing the FFS by 1 m/s per meter per lane is \$10, and the function $g(\cdot)$ is assumed to be a linear function of the product of the capacity expansion and the road length. In this study, w_1 and w_2 are set to 1 and 1/300,000, respectively, where 300,000 approximates the total number of car trips within the study area on a typical day. Under this setting, the term $w_2 g(\boldsymbol{x})$ can be regarded as the average investment cost per traveler. σ_c^2 is set to100. For a given $\boldsymbol{x}$, $w_2 g(\boldsymbol{x})$ is a constant. Thus, constraint (52) regulates variations in average travel time.



(a) Layout of the study area with highlighted links to be expanded

(b) The simulation network

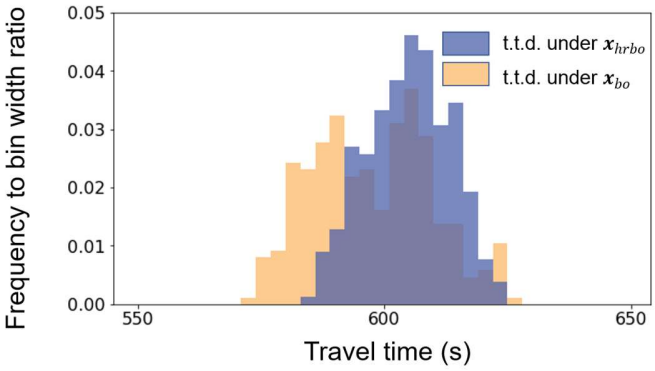
Figure 8. Layout of the study area

4.2.2 Experimental results and discussions

We first validate the necessity of implementing robust network design schemes. HRBO is employed to solve

the above-mentioned CNDP. For comparison, with the same initial samples, BO is adopted to solve the CNDP while relaxing the constraint (52) on variance. The two methods are terminated once the total number of simulation replications reaches 450. The HRBO method finds a robust solution $\boldsymbol{x}_{hrbo}$, whose objective function is approximately 622, and the variance of average travel time is approximately 71. The BO method identifies a non-robust solution, $\boldsymbol{x}_{bo}$, whose objective function value and variance of average travel time are approximately 615 and 146, respectively.

We first evaluate the two solutions from the perspective of the travel time reliability. Travel time reliability measures the consistency or dependability of travel times. It can reflect the robustness of transportation systems (de Jong and Bliemer 2015) and affect the travel experiences of travelers. Typically, the travel time reliability is assessed by analyzing characteristics of travel time distributions (TTD) throughout different times of the day, days of the week, or months of the year, under uncertainties like weather conditions and road works (van Lint and van Zuylen 2005). In this study, we adopt three commonly utilized metrics to assess the travel time reliability, which are the range, variance, and skew of travel time distributions. The two network design schemes (i.e., $\boldsymbol{x}_{hrbo}$ and $\boldsymbol{x}_{bo}$) are implemented in simulation. For each scheme, 250 independent simulations are carried out to evaluate the travel time reliability. Figure 9 represents the distributions of the average travel times. For each distribution, the range, variance, and skew are calculated. The skew is measured by the difference between the mean of all average travel time records and the mean of the top 20% of the records (van Lint and van Zuylen 2005).



Solution	Variance	Range	Skew
$\boldsymbol{x}_{hrbo}$	70.8	40.3	11.5
$\boldsymbol{x}_{bo}$	145.5	54.6	16.8

Figure 9. Travel time distributions of different solutions.

Figure 9 clearly demonstrates that the robust solution $\boldsymbol{x}_{hrbo}$ exhibits higher travel time reliability than the non-robust solution $\boldsymbol{x}_{bo}$. Intuitively, by implementing the network design scheme $\boldsymbol{x}_{hrbo}$, the performance of the transportation system is less likely to be affected by its inherent uncertainties (e.g., stochastic travel behaviors). This robustness of the solution would be desirable for authorities for the ease

of subsequent management, despite slightly higher objective function value.

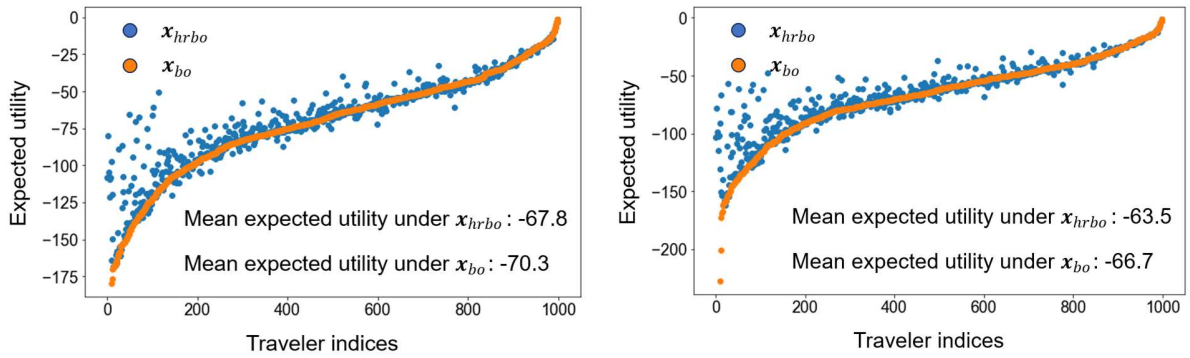
To further evaluate the practical impacts of these two solutions, the concept of the *value of reliability* (VOR) can be introduced. Developed in microeconomics, VOR connects the monetary values travelers place on improving the predictability (i.e., reducing the variability) of their travel times (Carrion and Levinson 2012). The VOR, together with the *value of travel time* (VOT), have been jointly investigated in measuring the costs generated by travel time and modeling route choice behaviors. Typically, there are two methods to evaluate the VOR, depending on the assumption made about the reasons why it is valued. The first approach (*the mean-variance approach*) assumes that a traveler has preferences against travel time uncertainty per second. The second approach (*the scheduling approach*) assumes that a traveler holds preferences for activity timing, and the VOR is defined from the outcomes, i.e., being early or late. For more details of the two approaches, the reader can refer to Fosgerau and Karlstrom (2010). Given a travel time distribution of a traveler, the expected utility of the traveler can be derived based on VOR and VOT. For the mean-variance approach, the expected utility is

$$\mathbb{E}[\![U]\!] = k_1\mu + k_2\sigma, \tag{55}$$

where U denotes the utility of travel time, μ is the expected travel time, and σ is the standard deviation of travel time, while k_1 and k_2 are preference parameters. For the scheduling approach, the expected utility can be expressed as

$$\mathbb{E}[\![U]\!] = k_1\mu + k_2\mathbb{E}[\![(T-D)^-]\!] + k_3\mathbb{E}[\![(T-D)^+]\!] + k_4 p_L, \tag{56}$$

where T is the actual travel time, D is the desired travel time, and p_L is the probability of being late. In Eq. (56), $(T-D)^- = \min(T-D,0)$ and $(T-D)^+ = \max(T-D,0)$. k_1, k_2, k_3 and k_4 are preference parameters. Given Eq. (55) and Eq. (56), we can measure, under different network design schemes, the expected utility for each traveler. For each traveler, the value of μ, σ, $\mathbb{E}[\![(T-D)^-]\!]$, $\mathbb{E}[\![(T-D)^+]\!]$, and p_L can be estimated based on 250 simulations. In the scheduling approach, the value of D is set as the mean value of all travel times. We assume that travelers are homoscedastic and the preference parameters are determined according to the empirical study conducted by Asensio and Matas (2008). For all travelers in the system, their expected utilities under $\boldsymbol{x}_{hrbo}$ and $\boldsymbol{x}_{bo}$ are calculated according to both Eq. (55) and Eq. (56). Figure 10 represents the utilities of all travelers. The x-axis displays the traveler indices sorted in ascending order of their expected utility under $\boldsymbol{x}_{bo}$ and the y-axis represents the expected utility value.



(a) Evaluation using the mean-variance approach (b) Evaluation using the scheduling approach

Figure 10. The utility of travel time under two solutions

As shown in Figure 10, compared to $\boldsymbol{x}_{bo}$, the robust solution $\boldsymbol{x}_{hrbo}$ can increase the utility of travelers using both evaluation approaches. Notably, travelers with low utilities under $\boldsymbol{x}_{bo}$ —often corresponding to those with longer trip distances—experience more significant utility gains under $\boldsymbol{x}_{hrbo}$. This suggests that incorporating simulation noise into the network design process can lead to more equitable improvements, especially benefiting long-distance travelers in the network. In both evaluation methods, even though the average travel time in the robust solution (604 sec.) was longer than that in the non-robust solution (597 sec.), the robust solution resulted in at least a 3.5% increase in utility values compared to the non-robust solution. These results highlight the necessity of accounting for simulation noise and implementing robust network design strategies in real-world applications.

We then benchmark the performance of the proposed HRBO method to that of three benchmark methods: a Taguchi and kriging combined method (Dellino et al. 2012), termed TKM, a response surface model-based method (Wu et al. 2019), termed RSM, and a constrained Bayesian optimization method, termed CBO (Gardner et al. 2014). We adopt the four methods to solve the SBRO problem 10 times. In each run, all four methods utilize the same computational resources (450 simulation replications) and run until these resources are exhausted. HRBO, CBO, and RSM sequentially select sample points, with an initial set of five identical points. Each sample point in the benchmark methods is evaluated using 15 simulation replications. In TKM, 30 sample points (i.e., 450/15) are randomly generated from the search space. In HRBO, the sample points are evaluated using a self-adaptive simulation resource allocation scheme, with the initial and maximum number of simulation replications set to 10 and 50, respectively. The parameters ε_r, ε_y, and ε_{EI} are set to 0.05, 0.1 and 0.1, respectively.

Table 4 summarizes the performance of the four methods. In this experiment, we use the *final solution* to denote the best identified solution when the algorithm is terminated. The performance of a method is evaluated using two metrics: *i*) the number of feasible solutions among all 10 *final solutions* (i.e., robustness), and *ii*) the average objective value of the feasible final solutions (i.e., solution accuracy). As

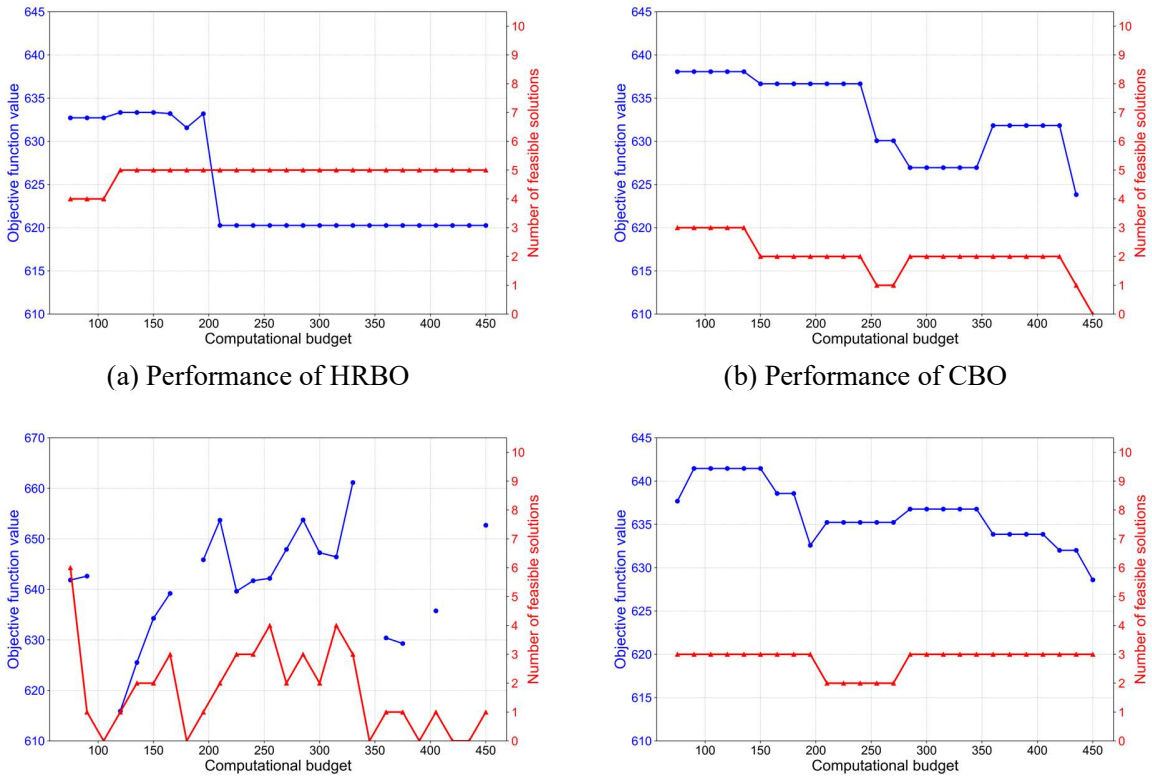
we have discussed, simulation noise can lead to identification issues where the best identified solution may not represent the true optimal one. In practical applications, enumerating all samples to determine the true optimal solution is often infeasible, making the best identified solution more valuable for real-world practice. The objective function value as well as the constraint function value of a solution is estimated using 100 simulation replications.

As shown in Table 4, HRBO achieves the highest number of feasible solutions compared with the benchmark methods, indicating its ability of identifying robust solutions. In terms of the accuracy of feasible solutions, HRBO achieves superior performance against benchmark methods. The results in Table 4 imply that HRBO effectively balances solution accuracy and robustness, selecting solutions with higher accuracy while ensuring robustness. The comparison between CBO and HRBO also highlights that merely incorporating robustness constraints into the acquisition function into BO is not sufficient to effectively generate robust solutions. Handling of variance estimation, feasibility assessment, and computational resource allocation are crucial for achieving satisfactory solutions.

Table 4. Summary of feasible solutions and average accuracy for each method

Method	**HRBO**	**CBO**	**RSM**	**TKM**
Number of feasible solutions	**5**	0	1	3
Average objective value of feasible solutions	**620.3**	/	635.2	628.6

Figure 11 evaluates the performance of the methods as a function of computational budgets. Each panel displays the performance of the proposed method, CBO, RSM, and TKM. The x-axis displays the computational budget (i.e., the available simulation replications). The y-axis displays the performance of the method if it is terminated at different computational budgets. As mentioned, the performance of a method is evaluated using two metrics: *i*) the number of feasible solutions among all 10 final solutions (i.e., robustness), and *ii*) the average objective value of the feasible final solutions (i.e., solution accuracy). These two metrics are displayed on the right and left y-axis of each panel, respectively. Each solution is evaluated using 100 simulation replications to estimate its feasibility and objective value. The average objective value of final solutions is represented using blue solid lines. The blue line in Figure 11(c) exhibits discontinuities because, in some iterations, none of the 10 runs produced a feasible solution. The number of feasible solutions is displayed using red lines. Figure 11 demonstrates that HRBO consistently finds robust solutions across different computational resource levels. As computational budgets increase, the solution accuracy of HRBO steadily improves. Compared to the benchmark methods, HRBO achieves superior performance across all computational resource levels in terms of both the number of feasible solutions and their average accuracy.



(a) Performance of HRBO (b) Performance of CBO

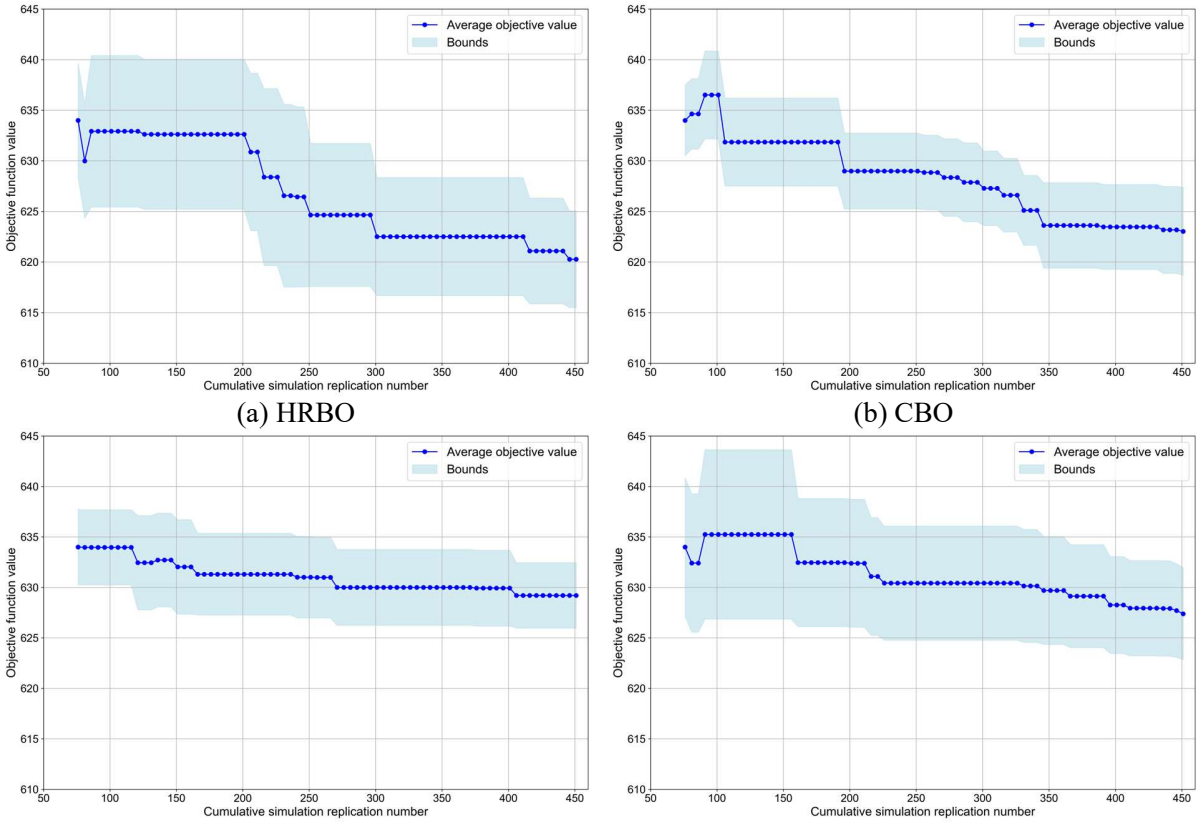
(c) Performance of RSM (d) Performance of TKM

Figure 11. Performance of each method as a function of computational budgets

Figure 11 illustrates the performance of each algorithm under different computational budgets, reflecting the combined effects of the *solution sampling* (i.e., the ability of identifying satisfactory solutions from search spaces) and the *best solution identification* (i.e., the ability of identifying the best solution among all samples). To further assess the effectiveness of HRBO in sampling, we track the best solution at each iteration, defined as the feasible solution with the smallest objective value among all sampled points. The results are presented in Figure 12, where the x-axis represents the cumulative number of simulation replications, and the y-axis shows the objective values of the best solutions.

As shown in Figure 12, HRBO converges faster than RSM and TKM. Compared to CBO, HRBO exhibits a similar convergence rate in the early stages (when the number of simulation replications is below 200). However, as computational resources increase, HRBO demonstrates both faster convergence and higher solution accuracy. This suggests that HRBO achieves greater efficiency in sampling promising solutions than the benchmark methods, validating the effectiveness of the proposed EI formulation and the two-stage method. A comparison between Figure 12(a) and Figure 11(a) shows that, across different computational budgets, the accuracy of the best solutions identified by HRBO closely matches the average accuracy of the true best solutions. This confirms HRBO's ability to effectively identify the best solution within the sampled points. In contrast, benchmark methods such as TKM and CBO struggle to accurately

identify the true best solutions, even with increased computational resources. This further underscores the importance of strategic resource allocation in improving optimization efficiency. A potential area for enhancing HRBO is improving the identification of feasible solutions, as indicated by Figures 11 and 12.



(a) HRBO (b) CBO

(c) RSM (d) TKM

Figure 12. Performance of the best solutions over iterations. The convergence curves in Figure 12(a), (b), and (d) show a slight increase at the beginning. This occurs because, among the 10 runs, only eight have best solutions, while the other two do not contain feasible solutions. The initial increase results from the inclusion of new best solutions, which also leads to a decrease in average accuracy. Aside from the initial phase, the convergence curves predominantly represent scenarios where all 10 runs contain best solutions. Therefore, the number of feasible solutions is not explicitly marked in the figure.

We then evaluate the efficiency of simulation resource allocation. Each method is employed to solve the CNDP 10 times, each time with 450 simulation replications. The efficiency of allocation is evaluated using two metrics: *i*) the percentage of simulation resources that are allocated to feasible solutions, and *ii*) the percentage of simulation resources that are allocated to the *set of satisfactory solutions*. In this experiment, the set of satisfactory solutions is defined as $\{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{X}_f, |C(\boldsymbol{x}) - C(\boldsymbol{x}^*)| / C(\boldsymbol{x}^*) \leq 1\%\}$, where $\boldsymbol{X}_f$ is the feasible set defined by the constraints (52) and (53), and $C(\boldsymbol{x}^*)$ is the optimal objective value. Since the optimal objective value is unknown, we take the minimal objective value of all samples in this

experiment as an estimate of $C(\boldsymbol{x}^*)$, which is approximately 604. As shown in the Table 5, compared to the benchmark methods, HRBO allocates more computational resources to feasible and satisfactory solutions. This demonstrates the efficiency of the proposed method in computational resource allocation.

Table 5. The efficiency of simulation resource allocation

Method	**HRBO**	**CBO**	**RSM**	**TKM**
Percentage of simulations on feasible solutions	**53%**	46%	19%	51%
Percentage of simulations on satisfactory solutions	**8‰**	3‰	0‰	5‰

The performance of HRBO is also confirmed in a variant M/M/1 queueing problem. The experimental results are consistent with those obtained in the simulation-based CNDP. Detailed experimental information can be found in **Appendix E**.

5 Conclusion

This study focused on SBRO problems in analyzing transportation systems. Regarding the pervasive heteroscedasticity of traffic simulators, the HRBO method was proposed by incorporating several SBO techniques in the BO framework. Particularly, GP was employed to construct surrogate models of objective and constraint functions. A constrained EI function was developed and solved using a two-stage method. To accurately estimate distributions of the objective function values and constraint function values, a posterior estimation method was developed in the Bayesian manner, in which two objective priors were introduced. An adaptive simulation resource allocation scheme was developed tailored for allocating simulation replications, based on which an allocation rule was derived.

The proposed methods were assessed using a test function, a CNDP, and a variant of the M/M/1 queuing model. Experiments on the test function validated the effectiveness of the proposed techniques across different levels of heteroscedasticity and noise structures. Results from the CNDP demonstrated that incorporating robustness constraints significantly enhances travel time reliability in the transportation network. From a utility perspective, the robust solution led to at least a 3.5% increase in average utility values compared to the non-robust solution, highlighting the importance of identifying robust solutions in the presence of heteroscedastic noise. HRBO's performance was benchmarked against three existing methods. The results show that, across different computational budgets, HRBO effectively balances solution accuracy and robustness by selecting high-accuracy solutions while maintaining robustness. Moreover, HRBO exhibits greater sampling efficiency, demonstrating its ability to identify promising solutions within the search space. In terms of computational efficiency, HRBO addresses SBRO problems through strategic resource allocation, prioritizing feasible and satisfactory solutions.

The proposed HRBO algorithm is best suited for simulation-based optimization problems with low to moderate-dimensional decision spaces (typically ranging from 1 to 20 variables), where simulation noise

is non-negligible and evaluation budgets are constrained. It remains effective for large-scale or computationally intensive simulations, as its computational overhead scales approximately linearly with simulation time under a fixed budget of simulation replications. These characteristics make HRBO particularly applicable to real-world transportation problems where simulation costs are high and precision in solution evaluation is critical. For high-dimensional problems (e.g., with more than 30 decision variables), alternative approaches such as dimension reduction or scalable simulation-based optimization methods may offer better performance.

This research is motivated by experiences in traffic simulation and the associated SBO problems. The characteristics explored in this study, such as problem stochasticity, noise heteroscedasticity, and high computational costs, are likely to be shared by SBO problems in other fields. Hence, the proposed methods are expected to be applicable beyond transportation simulation. Future work could incorporate advanced ranking-and-selection procedures into the BO framework, and explore the extension of HRBO to handle multiple implicit constraints. Readers may refer to SimOpt, a library of simulation-optimization problems (Pasupathy and Henderson 2011), for more constraint structures. This study focuses on uncertainties inherent in traffic simulation. An important direction for future work is to extend the SBRO framework to account for multiple sources of uncertainty, including environmental variability and implementation-related errors.

In this study, we focus on allocating simulation resources across sample points. This problem shares similarities with the multi-fidelity SBO, which addresses black-box optimization problems where the objective function can be evaluated at multiple fidelity levels with varying costs. Multi-fidelity SBO aims to accelerate optimization and reduce costs by efficiently allocating resources across models of different fidelities. Both approaches prioritize maximizing information gain through strategic resource allocation to support optimization decisions. Building on this shared principle, we propose that multi-fidelity SBO techniques can be used to enhance the simulation resource allocation scheme in this study. For example, simulation resource allocation among sample points can be interpreted as implicitly leveraging different fidelity levels, where higher noise levels correspond to lower fidelity, even though all evaluations incur the same computational cost.

In addition, several directions may further strengthen the robustness and generality of the proposed framework. Increasing the number of repeated runs would improve the statistical reliability of performance evaluation. The numerical challenges of optimizing EI functions can also be mitigated by reformulating the EI function itself. For example, Ament et al. (2023) proposed *LogEI*, a family of acquisition functions—including *LogConstrainedEI*—that retain the (exact or approximate) optima of their standard counterparts while offering significantly improved numerical stability and ease of optimization. Finally, our adaptive resource allocation mechanism shares conceptual similarities with

cost-aware Bayesian optimization. Future extensions could consider explicitly integrating cost-sensitive acquisition strategies to improve efficiency in heterogeneous simulation environments.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China (72471057, 52131203), the Natural Science Foundation of Jiangsu Province (BK20232019), and the Jiangsu Provincial Scientific Research Center of Applied Mathematics (BK20233002).

REFERENCES

Allen JK, Seepersad C, Choi H, Mistree F (2006) Robust design for multiscale and multidisciplinary applications. *Journal of Mechanical Design* 128(4):832–843.

Amaran S, Sahinidis NV, Sharda B, Bury SJ (2016) Simulation optimization: A review of algorithms and applications. *Annals of Operations Research* 240(1):351–380.

Ament, S., Daulton, S., Eriksson, D., Balandat, M. & Bakshy, E. (2023) Unexpected improvements to expected improvement for Bayesian optimization. *Advances in Neural Information Processing Systems* 36:20577-20612.

Ankenman B, Nelson BL, Staum J (2010) Stochastic kriging for simulation metamodeling. *Operations Research* (2):371–382.

Antoniou C, Azevedo CL, Lu L, Pereira F, Ben-Akiva M (2015) W-SPSA in practice: Approximation of weight matrices and calibration of traffic simulation models. *Transportation Research Part C: Emerging Technologies* 59:129–146.

Apley DW, Liu J, Chen W (2006) Understanding the effects of model uncertainty in robust design with computer experiments. *Journal of Mechanical Design* 128(4):945–958.

Asensio J, Matas A (2008) Commuters' valuation of travel time variability. *Transportation Research Part E: Logistics and Transportation Review* 44(6):1074–1085.

Avci MG, Selim H (2018) A multi-objective simulation-based optimization approach for inventory replenishment problem with premium freights in convergent supply chains. *Omega* 80:153–165.

Bagheri S, Konen W, Allmendinger R, Branke J, Deb K, Fieldsend J, Quagliarella D, Sindhya K (2017) Constraint handling in efficient global optimization. *Proceedings of the Genetic and Evolutionary Computation Conference*, 673–680.

Bazaraa MS, Sherali HD, Shetty CM (2013) *Nonlinear Programming: Theory and Algorithms*, 3rd ed. (John Wiley & Sons, Hoboken).

Beland JJ, Nair PB (2017) Bayesian optimization under uncertainty. *NIPS BayesOpt 2017 workshop*.

Bernardo M, Du B, Pannek J (2021) A simulation-based solution approach for the robust capacitated vehicle routing problem with uncertain demands. *Transportation Letters* 13(9):664–673.

Bertsekas DP (2014) *Constrained Optimization and Lagrange Multiplier Methods* (Academic Press, New York).

Bertsimas D, Nohadani O, Teo KM (2010) Robust optimization for unconstrained simulation-based problems. *Operations Research* 58(1):161–178.

Bogunovic I, Scarlett J, Jegelka S, Cevher V (2018) Adversarially robust optimization with Gaussian processes. *Advances in Neural Information Processing Systems* 31:5760–5770.

Bortolomiol S, Lurkin V, Bierlaire M (2021) A simulation-based heuristic to find approximate equilibria with disaggregate demand models. *Transportation Science* 55(5):1025–1045.

Carrion C, Levinson D (2012) Value of travel time reliability: A review of current evidence. *Transportation Research Part A: Policy and Practice* 46(4):720–741.

Chen A, Subprasom K, Ji Z (2006) A simulation-based multi-objective genetic algorithm (SMOGA) procedure for BOT network design problem. *Optimization and Engineering* 7(3):225–247.

Chen C-H, Lin J, Yücesan E, Chick SE (2000) Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems* 10:251–270.

Chen X, He X, Xiong C, Zhu Z, Zhang L (2019) A Bayesian stochastic kriging optimization model dealing

with heteroscedastic simulation noise for freeway traffic management. *Transportation Science* 53(2):545–565.
Chen XM, Xiong C, He X, Zhu Z, Zhang L (2016) Time-of-day vehicle mileage fees for congestion mitigation and revenue generation: A simulation-based optimization method and its real-world application. *Transportation Research Part C: Emerging Technologies* 63:71–95.
Cheng Q, Wang S, Liu Z, Yuan Y (2019) Surrogate-based simulation optimization approach for day-to-day dynamics model calibration with real data. *Transportation Research Part C: Emerging Technologies* 105:422–438.
Dandl F, Engelhardt R, Hyland M, Tilg G, Bogenberger K, Mahmassani HS (2021) Regulating mobility-on-demand services: Tri-level model and Bayesian optimization solution approach. *Transportation Research Part C: Emerging Technologies* 125:103075.
de Jong GC, Bliemer MC (2015) On including travel time reliability of road traffic in appraisal. *Transportation Research Part A: Policy and Practice* 73:80–95.
Dellino G, Kleijnen JP, Meloni C (2012) Robust optimization in simulation: Taguchi and Krige combined. *INFORMS Journal on Computing* 24(3):471–484.
Dellino G, Kleijnen JP, Meloni C (2015) *Uncertainty Management in Simulation-Optimization of Complex Systems* (Springer, Boston)
Fakhrmoosavi F, Kamjoo E, Kavianipour M, Zockaie A, Talebpour A, Mittal A (2022) A stochastic framework using Bayesian optimization algorithm to assess the network-level societal impacts of connected and autonomous vehicles. *Transportation Research Part C: Emerging Technologies* 139:103663.
Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences* 45(1-3):50–79.
Fosgerau M, Karlström A (2010) The value of reliability. *Transportation Research Part B: Methodological* 44(1):38–49.
Frazier PI (2018) A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.
Fu MC (2002) Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14(3):192–215.
Fu MC (2015) *Handbook of Simulation Optimization* (Springer, New York).
Gardner JR, Kusner MJ, Xu ZE, Weinberger KQ, Cunningham JP (2014) Bayesian optimization with inequality constraints. *The 31st International Conference on Machine Learning,* (ICML, Beijing) 937–945.
Goldberg P, Williams C, Bishop C (1997) Regression with input-dependent noise: A Gaussian process treatment. *Advances in Neural Information Processing Systems* 10:493–499.
Gosavi A (2015) *Simulation-Based Optimization* (Springer, New York).
Gramacy RB, Gray GA, Le Digabel S, Lee HKH, Ranjan P, Wells G, Wild SM (2016) Modeling an augmented Lagrangian for blackbox constrained optimization. *Technometrics* 58(1):1-11.
Griffiths R-R, Aldrick AA, Garcia-Ortegon M, Lalchand V (2021) Achieving robustness to aleatoric uncertainty with heteroscedastic Bayesian optimisation. *Machine Learning: Science and Technology* 3(1):015004.
Gu X, Renaud JE, Batill SM, Brach RM, Budhiraja AS (2000) Worst case propagated uncertainty of multidisciplinary systems in robust design optimization. *Structural and Multidisciplinary Optimization* 20:190–213.
Gu Z, Saberi M (2021) Simulation-based optimization of toll pricing in large-scale urban networks using the network fundamental diagram: A cross-comparison of methods. *Transportation Research Part C: Emerging Technologies* 122:102894.
Gu Z, Waller ST, Saberi M (2019) Surrogate-based toll optimization in a large-scale heterogeneously congested network. *Computer-Aided Civil and Infrastructure Engineering* 34(8):638–653.
Gu Z, Li Y, Saberi M, Liu Z (2024) Simulation-based robust and adaptive optimization method for heteroscedastic transportation problems. *Transportation Science* 58(4):685–918.
Gu Z, Li Y, Saberi M, Rashidi TH, Liu Z (2023) Macroscopic parking dynamics and equitable pricing: Integrating trip-based modeling with simulation-based robust optimization. *Transportation Research Part B: Methodological* 173:354–381.

Guzman R, Oliveira R, Ramos F (2020) Heteroscedastic Bayesian optimisation for stochastic model predictive control. *IEEE Robotics and Automation Letters* 6(1):56–63.
He X, Chen X, Xiong C, Zhu Z, Zhang L (2017) Optimal time-varying pricing for toll roads under multiple objectives: A simulation-based optimization approach. *Transportation Science* 51(2):412–426.
Hoff PD (2009) *A First Course in Bayesian Statistical Methods* (Springer, Dordrecht).
Hollander M, Wolfe DA, Chicken E (2013) *Nonparametric Statistical Methods* (John Wiley & Sons Inc).
Hong LJ, Nelson BL (2006) Discrete optimization via simulation using COMPASS. *Operations Research* 54(1):115–129.
Hong LJ, Zhang X (2021) Surrogate-based simulation optimization. *Tutorials in Operations Research: Emerging Optimization Methods and Modeling Techniques with Applications* (INFORMS), 287–311.
Hong LJ, Fan W, Luo J (2021) Review on ranking and selection: A new perspective. *Frontiers of Engineering Management* 8(3):321–343.
Hong LJ, Luo J, Nelson BL (2015) Chance constrained selection of the best. *INFORMS Journal on Computing* 27(2):317–334.
Hu J, Fu MC (2024) On the convergence rate of stochastic approximation for gradient-based stochastic optimization. *Operations Research.* https://doi.org/10.1287/opre.2023.0055.
Hu J, Peng Y, Zhang G, Zhang Q (2022) A stochastic approximation method for simulation-based quantile optimization. *INFORMS Journal on Computing* 34(6):2889–2907.
Hu Z, Hong LJ (2022) Robust simulation with likelihood-ratio constrained input uncertainty. *INFORMS Journal on Computing* 34(4):2350–2367.
Huo J, Liu Z, Chen J, Cheng Q, Meng Q (2023) Bayesian optimization for congestion pricing problems: A general framework and its instability. *Transportation Research Part B: Methodological* 169:1–28.
Iwazaki S, Inatsu Y, Takeuchi I (2021) Mean-variance analysis in Bayesian optimization under uncertainty. *International Conference on Artificial Intelligence and Statistics* (PMLR), 973–981.
Jalali H, Van Nieuwenhuyse I, Picheny V (2017) Comparison of kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research* 261(1):279–301.
Jones DR, Perttunen CD, Stuckman BE (1993) Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications* 79:157–181.
Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* 13(4):455–492.
Juan AA, Panadero J, Reyes-Rubiano L, Faulin J, de la Torre R, Latorre I (2019) Simulation-based optimization in transportation and logistics: Comparing sample average approximation with simheuristics. *2019 Winter Simulation Conference* (IEEE, Maryland), 1906–1917.
Karna SK, Sahai R (2012) An overview on Taguchi method. *International Journal of Engineering and Mathematical Sciences* 1(1):1–7.
Kim S-H, Nelson BL (2006) Selecting the best system. *Handbooks in Operations Research and Management Science* 13:501-534.
Kleijnen JP (2017) Regression and kriging metamodels with their experimental designs in simulation: A review. *European Journal of Operational Research* 256(1):1–16.
Kleijnen JP, Van Beers W, Van Nieuwenhuyse I (2012) Expected improvement in efficient global optimization through bootstrapped kriging. *Journal of Global Optimization* 54(1):59–73.
Krajzewicz D (2010) Traffic simulation with SUMO–simulation of urban mobility. Barcelo J ed. *Fundamentals of Traffic Simulation*, (Springer New York), 269–293.
Lazaro-Gredilla M, Titsias MK (2011) Variational heteroscedastic Gaussian process regression. *The 28th International Conference on Machine Learning* (ICML, Bellevue) 841–848.
Le Digabel, S. & Wild, S. M. (2024) A taxonomy of constraints in black-box simulation-based optimization. *Optimization and Engineering* 25(2):1125-1143.
Li C, Gao S, Du J (2023) Convergence analysis of stochastic kriging-assisted simulation with random covariates. *INFORMS Journal on Computing* 35(2):386–402.
Li C, Yang H, Zhu D, Meng Q (2012) A global optimization method for continuous network design problems. *Transportation Research Part B: Methodological* 46(9):1144–1158.
Li Z, Tian Y, Sun J, Lu X, Kan Y (2022) Simulation-based optimization of large-scale dedicated bus lanes allocation: Using efficient machine learning models as surrogates. *Transportation Research Part C:*

Emerging Technologies 143:103827.
Lu J, Zhou T, Osorio C (2021) Adaptive partitioning strategy for high-simensional discrete simulation-based optimization problems. *arXiv preprint arXiv:2104.14119*.
Luo, Y. & Lim, E. (2013) Simulation-based optimization over discrete sets with noisy constraints. *IIE Transactions* 45(7):699-715.
Ma W, Qian ZS (2018) Statistical inference of probabilistic origin-destination demand using day-to-day traffic data. *Transportation Research Part C: Emerging Technologies* 88:227–256.
Meng Q, Yang H, Bell MG (2001) An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research Part B: Methodological* 35(1):83–105.
Neelamkavil F (1987) *Computer Simulation and Modelling* (John Wiley & Sons, Inc.).
Osorio C (2019) High-dimensional offline origin-destination (OD) demand calibration for stochastic traffic simulators of large-scale road networks. *Transportation Research Part B: Methodological* 124:18–43.
Osorio C, Atasoy B (2021) Efficient simulation-based toll optimization for large-scale networks. *Transportation Science* 55(5):1010–1024.
Osorio C, Bierlaire M (2013) A simulation-based optimization framework for urban transportation problems. *Operations Research* 61(6):1333–1345.
Osorio C, Chong L (2015) A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science* 49(3):623–636.
Osorio C, Selvam KK (2017) Simulation-based optimization: achieving computational efficiency through the use of multiple simulators. *Transportation Science* 51(2):395–411.
Parnianifard A, Azfanizam A, Ariffin MKAM, Ismail MIS (2018) An overview on robust design hybrid metamodeling: Advanced methodology in process optimization under uncertainty. *International Journal of Industrial Engineering Computations* 9:1–32.
Pasupathy R, Henderson SG (2011) SimOpt: A library of simulation optimization problems. *Proceedings of the 2011 Winter Simulation Conference (WSC)* (IEEE), 4075–4085.
Patwary AUZ, Huang W, Lo HK (2021) Metamodel-based calibration of large-scale multimodal microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies* 124:102859.
Peng Y, Chen C-H, Fu MC, Hu J-Q (2016) Dynamic sampling allocation and design selection. *INFORMS Journal on Computing* 28(2):195–208.
Prett DM, García CE (2013) *Fundamental Process Control* (Butterworth-Heinemann).
Qurashi M, Lu Q-L, Cantelmo G, Antoniou C (2022) Dynamic demand estimation on large scale networks using Principal Component Analysis: The case of non-existent or irrelevant historical estimates. *Transportation Research Part C: Emerging Technologies* 136:103504.
Rom DM (1990) A sequentially rejective test procedure based on a modified Bonferroni inequality. *Biometrika* 77(3):663–665.
Sauré A, Patrick J, Puterman ML (2015) Simulation-based approximate policy iteration with generalized logistic functions. *INFORMS Journal on Computing* 27(3):579–595.
Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE* 104(1):148–175.
Shaibu A, Cho BR (2009) Another view of dual response surface modeling and optimization in robust parameter design. *The International Journal of Advanced Manufacturing Technology* 41(7):631–641.
Song, Z., Zhao, K., Ji, S. & Jin, J. G. (2023) Simulation and optimization of transfer system for ore terminal with complex waterways. *Multimodal Transportation* 2(4):100107.
Stummer C, Zsifkovits M, Doerner KF (2021) Preface: The modeling and simulation of complex systems. *Annals of Operations Research* 305(1):423–424.
Tay T, Osorio C (2022) Bayesian optimization techniques for high-dimensional simulation-based transportation problems. *Transportation Research Part B: Methodological* 164:210–243.
Trosset MW (1996) Taguchi and robust optimization. Report, Department of Computational & Applied Mathematics, Rice University, Houston.
Tsai SC, Fu SY (2014) Genetic-algorithm-based simulation optimization considering a single stochastic constraint. *European Journal of Operational Research* 236(1):113–125.
Tsui K-L (1996) A critical look at Taguchi's modelling approach for robust design. *Journal of Applied*

Statistics 23(1):81–96.
van Lint J, van Zuylen HJ (2005) Monitoring and predicting freeway travel time reliability: Using width and skew of day-to-day travel time distribution. *Transportation Research Record* 1917(1):54–62.
Wang J, He X, Peeta S, Wang W (2022) Globally convergent line search algorithm with Euler-based step size-determination method for continuous network design problem. *Transportation Research Part B: Methodological* 163:119-144.
Williams CK, Rasmussen CE (2006) *Gaussian Processes for Machine Learning* (MIT Press Cambridge, MA).
Wu W, Liu R, Jin W, Ma C (2019) Simulation-based robust optimization of limited-stop bus service with vehicle overtaking and dynamics: A response surface methodology. *Transportation Research Part E: Logistics and Transportation Review* 130:61–81.
Xu J, Nelson BL, Hong LJ (2013) An adaptive hyperbox algorithm for high-dimensional discrete optimization via simulation problems. *INFORMS Journal on Computing* 25(1):133–146.
Xu WL, Nelson BL (2013) Empirical stochastic branch-and-bound for optimization via simulation. *IIE Transactions* 45(7):685–698.
Yanıkoğlu İ, den Hertog D, Kleijnen JP (2016) Robust dual-response optimization. *IIE Transactions* 48(3):298–312.
Yeniay Ö (2005) Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications* 10(1):45–56.
Yin R, Liu X, Zheng N, Liu Z (2022b) Simulation-based analysis of second-best multimodal network capacity. *Transportation Research Part C: Emerging Technologies* 145:103925.
Zang Z, Xu X, Qu K, Chen R, Chen A (2022) Travel time reliability in transportation networks: A review of methodological developments. *Transportation Research Part C: Emerging Technologies* 143:103866.
Zhan D, Xing H (2020) Expected improvement for expensive optimization: A review. *Journal of Global Optimization* 78(3):507–544.
Zhang G, Peng Y, Zhang J, Zhou E (2023) Asymptotically optimal sampling policy for selecting top-m alternatives. *INFORMS Journal on Computing* 35(6): 1261–1285.
Zhang W (2017) Simulation-based robust optimization for the schedule of single-direction bus transit route: The design of experiment. *Transportation Research Part E: Logistics and Transportation Review* 106:203–230.
Zhang Y, Han Z-H, Zhang K-S (2018) Variable-fidelity expected improvement method for efficient global optimization of expensive functions. *Structural and Multidisciplinary Optimization* 58(4):1431–1451.
Zheng L, Bao J, Xu C, Tan Z (2022a) Biobjective robust simulation-based optimization for unconstrained problems. *European Journal of Operational Research* 299(1):249–262.
Zheng L, Liu P, Huang H, Ran B, He Z (2022b) Time-of-day pricing for toll roads under traffic demand uncertainties: A distributionally robust simulation-based optimization method. *Transportation Research Part C: Emerging Technologies* 144:103894.
Zhou Y, Ong GP, Meng Q, Cui H (2023) Electric bus charging facility planning with uncertainties: Model formulation and algorithm design. *Transportation Research Part C: Emerging Technologies* 150:104108.
Zhu L, Ma Y, Zhang L (2013) The robust design of EOQ considering the compound effect of both noise uncertainty and meta-modeling uncertainty. *2013 Sixth International Symposium on Computational Intelligence and Design* (IEEE, Hangzhou), 173–177.

Supplemental Material

Appendix A: Algorithms

A.1 Algorithm 1: Two-stage method to optimize the constrained EI function

Algorithm 1 Two-stage method to optimize the constrained EI function

Input: dataset D_n; GP surrogate models of $y(\cdot)$ and $\log r(\cdot)$; the minimal objective function value $\tilde{y}^*$ of all feasible samples*; the predetermined significance level ε_{EI}; the number of multiple starting points while solving **P2**, n_{EI}^{start};

Output: the new sample $\boldsymbol{x}_{n+1}$

randomly generate n_{EI}^{start} initial starting points for **P2** from the region constrained by the interval constraint (26)**. Denote the starting points as $\mathcal{X} = \{(\boldsymbol{x}_{start}^{init,1}, a_1^{init,1}, a_2^{init,1}), ..., (\boldsymbol{x}_{start}^{init,n_{EI}^{start}}, a_1^{init,n_{EI}^{start}}, a_2^{init,n_{EI}^{start}})\}$. Initialize $\Upsilon_{start} = \{\}$.

for each initial starting point $(\boldsymbol{x}_{start}^{init,i}, a_1^{init,i}, a_2^{init,i})$ in $\mathcal{X}$ **do**

  taking $(\boldsymbol{x}_{start}^{init,i}, a_1^{init,i}, a_2^{init,i})$ as the starting point, solve **P2** using the methods of feasible directions. Denote the solution as $(\boldsymbol{x}_{start}^{i}, a_1^{i}, a_2^{i})$.

  if $\boldsymbol{x}_{start}^{i}$ satisfies the constraints (22)–(24) **then**

    $\Upsilon_{start} = \Upsilon_{start} \cup \{\boldsymbol{x}_{start}^{i}\}$

  end if

end for

if Υ_{start} is not empty **then**

  for each $\boldsymbol{x}_{start}^{i}$ in Υ_{start} **do**

    solve **P1** using the Zoutendijk method while taking $\boldsymbol{x}_{start}^{i}$ as the starting point. Denote the solution and its associated EI function value as $\boldsymbol{x}^i$ and EI^i, respectively.

  end for

  let $k = \arg\max_i EI^i$ and $\boldsymbol{x}_{n+1} = \boldsymbol{x}^k$. **Return** $\boldsymbol{x}_{n+1}$.

else

  randomly sample $\boldsymbol{x}_{n+1}$ from the region constrained by inequality (26). **Return** $\boldsymbol{x}_{n+1}$.

end if

*A relatively extreme and uncommon scenario is when all sample points are infeasible solutions. In such cases, this paper suggests selecting the next sample point using random search approaches. **The starting points can be generated using random sampling techniques, e.g., the Latin hypercube sampling methods.

A.2 Algorithm 2: Adaptive simulation resource allocation scheme

Algorithm 2 Adaptive simulation resource allocation scheme

Input: $D_n = (X_n, \mathcal{F}_n)$; surrogate models of $y(\cdot)$ and $\log r(\cdot)$; the best feasible solution* among X_n, $\boldsymbol{x}^*$; the new sample, $\boldsymbol{x}_{n+1}$; significance levels ε_r and ε_y; initial simulation replication number m_{init}^{new} for $\boldsymbol{x}_{n+1}$; additional simulation replication number m_{add}; the maximal simulation replication number $m_{\max}$; the threshold σ_c^2 of variance;

Output: updated dataset $D_{n+1} = (X_{n+1}, \mathcal{F}_{n+1})$ and the best solution among X_{n+1}

conduct m_{init}^{new} simulation replications at $\boldsymbol{x}_{n+1}$, and denote the set of results as $\boldsymbol{f}_{n+1}$.

while inequalities (35) and (36) are unsatisfied and simulation replications at $\boldsymbol{x}_{n+1} < m_{\max}$ **do**

  conduct m_{add} simulation replications at $\boldsymbol{x}_{n+1}$, and update $\boldsymbol{f}_{n+1}$.

end while

update $X_{n+1} = X_n \cup \{\boldsymbol{x}_{n+1}\}$, $\mathcal{F}_{n+1} = \mathcal{F}_n \cup \{\boldsymbol{f}_{n+1}\}$, $D_{n+1} = (X_{n+1}, \mathcal{F}_{n+1})$. Let $\hat{r}(\boldsymbol{x}_{n+1})$ denote the expectation of the posterior distribution of $r(\boldsymbol{x}_{n+1})$

if the inequality (36) is satisfied or $\hat{r}(\boldsymbol{x}_{n+1}) > \sigma_c^2$ **then**

  return dataset D_{n+1} and the best solution $\boldsymbol{x}^*$.

else

  if $\boldsymbol{x}^*$ does not exist **then**

    $\boldsymbol{x}^* = \boldsymbol{x}_{n+1}$. **Return** dataset D_{n+1} and the best solution $\boldsymbol{x}^*$.

  else

    while (37) and (38) are unsatisfied and simulation replications at $\boldsymbol{x}_{n+1}, \boldsymbol{x}^* < m_{\max}$ **do**

      set $m_1 = m_{add}$, substitute m_1 into Eq. (45), and obtain the value of m_2.

      conduct $\lfloor m_1 \rfloor$ and $\lfloor m_2 \rfloor$ simulation replications at $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}^*$, respectively, and update $\boldsymbol{f}_{n+1}^{ad}$ and $\boldsymbol{f}_*^{ad}$.

    end while

    update dataset D_{n+1}. Let $\hat{y}(\boldsymbol{x}_{n+1})$ ($\hat{y}(\boldsymbol{x}^*)$) denote the expectation of the posterior distribution of $y(\boldsymbol{x}_{n+1})$ ($y(\boldsymbol{x}^*)$).

  end if

end if

if the inequality (37) is satisfied or $\hat{y}(\boldsymbol{x}^*) < \hat{y}(\boldsymbol{x}_{n+1})$ **then**

  $\boldsymbol{x}^* = \boldsymbol{x}_{n+1}$. **Return** dataset D_{n+1} and the best solution $\boldsymbol{x}^*$.

else

  return dataset D_{n+1} and the best solution $\boldsymbol{x}^*$.

end if

*Note that $\boldsymbol{x}^*$ could not exist if all solutions in X_n are infeasible.

A.3 Algorithm 3: HRBO for solving simulation-based robust optimization problems

Algorithm 3 HRBO for solving simulation-based robust optimization problems

Input: the stochastic performance, $f(\cdot)$; the threshold σ_c^2 of variance; the lower (upper) bounds of the decision vector $x_{l,i}$ ($x_{u,i}$), $i = 1,...,d$;

Output: $\boldsymbol{x}^*$ that minimizes $\mathbb{E}[\![f(\cdot)]\!]$ while satisfying the constraint on $\mathbb{VAR}[\![f(\cdot)]\!]$.

initialize the number of initial sample points, n_{init}; the initial simulation replication number for initial sample points, m_{init}^{init}; additional simulation replication number for initial samples points, m_{add}^{init}; the predetermined stopping criteria Π *; initialize m_{init}^{new}, m_{add}, $m_{\max}$, ε_{EI}, ε_y, ε_r; set $n = 0$; initialize $X_n = \{\ \}$, $\boldsymbol{\mathcal{F}}_n = \{\ \}$, $D_n = \{X_n, \boldsymbol{\mathcal{F}}_n\}$.

for $i = (1,...,n_{init})$ **do**

  randomly generate one sample point from the region defined by (3), and denote the point as $\boldsymbol{x}_i$.

  conduct m_{init}^{init} simulation replications at $\boldsymbol{x}_i$, and denote the set of results as $\boldsymbol{f}_i$.

  while neither (35) nor (36) is satisfied and simulation replications at $\boldsymbol{x}_i < m_{\max}$ **do**

    conduct m_{add}^{init} simulation replications at $\boldsymbol{x}_i$, update $\boldsymbol{f}_i$.

  end while

  update $X_{n+1} = X_n \cup \{\boldsymbol{x}_i\}$, $\boldsymbol{\mathcal{F}}_{n+1} = \boldsymbol{\mathcal{F}}_n \cup \{\boldsymbol{f}_i\}$, $D_{n+1} = (X_{n+1}, \boldsymbol{\mathcal{F}}_{n+1})$, $n = n+1$. Let $\hat{r}(\boldsymbol{x}_i)$ denote the expectation of the posterior distribution of $r(\boldsymbol{x}_i)$

  estimate the distribution of $y(\boldsymbol{x}_i)$ and $\log r(\boldsymbol{x}_i)$ using either the non-informative posterior or GP-based posterior

  if (35) is met or $\hat{r}(\boldsymbol{x}_i) < \sigma_c^2$ **then**

    $\boldsymbol{x}_i$ is deemed a feasible solution.

  end if

end for

let $\boldsymbol{x}^*$ be the solution with the minimal expectation of $y(\cdot)$ among all feasible solutions.

while the stopping criteria Π are not met **do**

  construct GP surrogate models of $y(\cdot)$ and $\log r(\cdot)$ based on D_n.

  use **Algorithm 1** to solve **P1**, and obtain a new sample $\boldsymbol{x}_{n+1}$.

  use **Algorithm 2** to allocate simulation resources, update the dataset D_n, and update the current best solution $\boldsymbol{x}^*$.

  update $n = n+1$

end while

return $\boldsymbol{x}^*$

*The following conditions can be set as stopping criteria of sampling in **Algorithm 3**: *i*) the sampling reaches computational budgets, *ii*) the minimal objective function value of samples reaches a predetermined value (not necessarily the global optimal value), and *iii*) the best-found solution remains unchanged for multiple iterations.

Appendix B: Proof of Proposition 2

Suppose that we have k ($k>1$) feasible solutions before sampling $\boldsymbol{x}_{n+1}$, denoted by $\boldsymbol{x}_{[1]},...,\boldsymbol{x}_{[k]}$. Let $\boldsymbol{x}_{[b]}$ denote the best solution among the k feasible solutions. Let $\widehat{\boldsymbol{\mathcal{F}}}_{n+1}$ denote the collection of all simulation evaluation results until conducting the selection. We first define the PCS in $\{\boldsymbol{x}_{[1]},...,\boldsymbol{x}_{[k]}\}$

$$\begin{aligned} PCS &= P\left(\boldsymbol{x}_{[b]} \text{ is selected} \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right) \\ &= P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right), i=1,...,k, i\neq b \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}. \end{aligned} \tag{A1}$$

According to the Bonferroni inequality

$$\begin{aligned} PCS &= P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right), i=1,...,k, i\neq b \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \\ &\geq 1-\sum_{i=1,i\neq b}^{k}\left[1-P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\right] \\ &= 1-\sum_{i=1,i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \\ &= APCS. \end{aligned} \tag{A2}$$

As can be seen in (A2), the APCS is a lower bound of PCS. OCBA optimizes the APCS to derive allocation rules.

Suppose that $\boldsymbol{x}_{n+1}$ is the best solution among $\boldsymbol{x}_{[1]},...,\boldsymbol{x}_{[k]},\boldsymbol{x}_{n+1}$, the APCS in OCBA is

$$APCS_{OCBA} = 1-\sum_{i=1}^{k} P\left\{y\left(\boldsymbol{x}_{n+1}\right) > y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}. \tag{A3}$$

In the proposed scheme, PCS is calculated as

$$PCS_{proposed} = P\left\{y\left(\boldsymbol{x}_{n+1}\right) < y\left(\boldsymbol{x}_{*}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} = 1-P\left\{y\left(\boldsymbol{x}_{n+1}\right) > y\left(\boldsymbol{x}_{*}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}. \tag{A4}$$

Since $\boldsymbol{x}_{*}\in\left\{\boldsymbol{x}_{[1]},...,\boldsymbol{x}_{[k]}\right\}$, we have

$$PCS_{proposed} > APCS_{OCBA}. \tag{A5}$$

Since $PCS_{proposed}$ can serve as a lower bound of itself, **Proposition 2** holds while $\boldsymbol{x}_{n+1}$ is the best solution among $\boldsymbol{x}_{[1]},...,\boldsymbol{x}_{[k]},\boldsymbol{x}_{n+1}$.

When $\boldsymbol{x}_{[b]}$ is the best solution among $\boldsymbol{x}_{[1]},...,\boldsymbol{x}_{[k]},\boldsymbol{x}_{n+1}$, the APCS in OCBA is

$$APCS_{OCBA} = 1-\sum_{i=1,i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} - P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y\left(\boldsymbol{x}_{n+1}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}. \tag{A6}$$

In the proposed scheme, PCS is derived as

$$
\begin{aligned}
PCS_{proposed} &= P\left\{y(\boldsymbol{x}_*) < y(\boldsymbol{x}_{n+1}), \boldsymbol{x}_* = \boldsymbol{x}_{[b]} \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \\
&= P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right), i = 1,\ldots,k, i \neq b \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \\
&\geq P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\left(1 - \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\right) \\
&= APCS_{proposed}.
\end{aligned}
\tag{A7}
$$

The $APCS_{proposed}$ can be reformulated as

$$
\begin{aligned}
APCS_{proposed} &= P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\left(1 - \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\right) \\
&= \left(1 - P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\right)\left(1 - \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}\right) \\
&= 1 - \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} - P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \\
&\quad + P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \\
&= APCS_{OCBA} + P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\}.
\end{aligned}
\tag{A8}
$$

Since $P\left\{y\left(\boldsymbol{x}_{[b]}\right) > y(\boldsymbol{x}_{n+1}) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} \sum_{i=1, i\neq b}^{k} P\left\{y\left(\boldsymbol{x}_{[b]}\right) < y\left(\boldsymbol{x}_{[i]}\right) \middle| \widehat{\boldsymbol{\mathcal{F}}}_{n+1}\right\} > 0$, we have

$$
APCS_{proposed} > APCS_{OCBA}. \tag{A9}
$$

Appendix C: Proof of Proposition 3

P3 achieves its local optimal solution when *u* equals 0 or 1. Let $g(m_1, m_2) = m_1 + m_2$ denote the objective function with respect to m_1 and m_2. Since $\partial g(m_1, m_2)/\partial m_1 = \partial g(m_1, m_2)/\partial m_2 = 1$, **P3** does not have any stationary points in the interior of its feasible set. Therefore, P3 achieves its local optimal solutions at the boundary of its feasible set.

Appendix D: Proof of Proposition 4

To simplify matters, we will demonstrate the scenario where *u*=1, with the understanding that the case where *u*=0 can be proven using the same approach. Firstly, formulate the Lagrange function

$$
L(m_1, m_2, \lambda) = m_1 + m_2 + \lambda\left(1 - \varepsilon_y - P\left\{y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*) \middle| \boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right\}\right). \tag{A10}
$$

Since $\bar{f}_{n+1} = \mu_y(\boldsymbol{x}_{n+1})$, $\bar{f}_* = \mu_y(\boldsymbol{x}^*)$, distributions of $y(\boldsymbol{x}_{n+1})$ and $y(\boldsymbol{x}^*)$ can be rewritten as

$$y(\boldsymbol{x}_{n+1})\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad} \sim \mathcal{N}\left(\mu_y(\boldsymbol{x}_{n+1}), \tilde{\tau}_{y,n+1}^2(m_1)\right)\right., \tag{A11}$$

$$y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad} \sim \mathcal{N}\left(\mu_y(\boldsymbol{x}^*), \tilde{\tau}_{y,*}^2(m_2)\right)\right., \tag{A12}$$

where $\tilde{\tau}_{y,n+1}^2(m_1) = 1\Big/\left(1\big/\sigma_y^2(\boldsymbol{x}_{n+1}) + m_1\big/\hat{r}(\boldsymbol{x}_{n+1})\right)$, $\tilde{\tau}_{y,*}^2(m_2) = 1\Big/\left(1\big/\sigma_y^2(\boldsymbol{x}^*) + m_2\big/\hat{r}(\boldsymbol{x}^*)\right)$. The probability of $P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right)$ can be derived as

$$P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right) = \Phi\left(\frac{\mu_y(\boldsymbol{x}^*) - \mu_y(\boldsymbol{x}_{n+1})}{\sqrt{\tilde{\tau}_{y,n+1}^2(m_1) + \tilde{\tau}_{y,*}^2(m_2)}}\right). \tag{A13}$$

P4 achieves its optimal solution when the following equalities hold:

$$\frac{\partial L(m_1, m_2, \lambda)}{\partial m_1} = 1 - \lambda \frac{\partial P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right)}{\partial m_1} = 0, \tag{A14}$$

$$\frac{\partial L(m_1, m_2, \lambda)}{\partial m_2} = 1 - \lambda \frac{\partial P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right)}{\partial m_2} = 0, \tag{A15}$$

$$\frac{\partial L(m_1, m_2, \lambda)}{\partial \lambda} = 1 - \varepsilon_y - P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right) = 0. \tag{A16}$$

It can be easily found that when **P4** achieves its optimal solution, the following relationship holds

$$\frac{\partial P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right)}{\partial m_1} = \frac{\partial P\left(y(\boldsymbol{x}_{n+1}) < y(\boldsymbol{x}^*)\left|\boldsymbol{\mathcal{F}}_{n+1}, \boldsymbol{f}_{n+1}^{ad}, \boldsymbol{f}_*^{ad}\right.\right)}{\partial m_2}. \tag{A17}$$

Substitute Eq. (A13) into Eq. (A17), and the relationship between m_1^* and m_2^* (i.e., Eq. (45)) can be derived.

Appendix E: Experiments on a variant M/M/1 queueing problem

In the variant of M/M/1 queueing model, the customer arrival process is modeled by a Poisson distribution with a rate of λ, while the service time distribution for each customer is exponential with a mean duration of $1/\mu$. It is assumed that the queuing system serves n_0 customers every day. The average time of the n_0 customers spend in the system can be represented as $f(\mu, \lambda, n_0, \boldsymbol{\xi})$, where n_0 and λ are predetermined environmental factors that control the arrival of customers, and $\boldsymbol{\xi}$ represents the collection of service times and arrival times of the n_0 customers in the M/M/1 queuing simulation. $\boldsymbol{\xi}$ is affected by the values of μ and λ. In these experiments, we aim to find the optimal service rate μ^* such that the expectation of the total cost, denoted as $C(\mu^*)$, is minimized subject to maintaining the stochasticity of the queuing system at a predetermined level. That is,

$$\min_{\mu} C(\mu) = \min_{\mu} \left(\mathbb{E}[\![f(\mu, \lambda, n_0, \boldsymbol{\xi})]\!] + c\mu \right), \tag{A18}$$

subject to

$$\mathbb{VAR}[\![f(\mu, \lambda, n_0, \boldsymbol{\xi})]\!] \le \sigma_c^2, \tag{A19}$$

$$\lambda/\mu < 1, \tag{A20}$$

where $C(\mu)$ denotes the expectation of the total cost, which consists of two components: *i*) the average time that customers spend in the queueing system $f(\mu, \lambda, n_0, \boldsymbol{\xi})$, and *ii*) the cost $c\mu$ of keeping the service rate μ. The parameter c denotes the cost per unit increase in the service rate. In this experiment, σ_c^2, λ, c and n_0 are set as 0.1, 1, 4, 250, respectively.

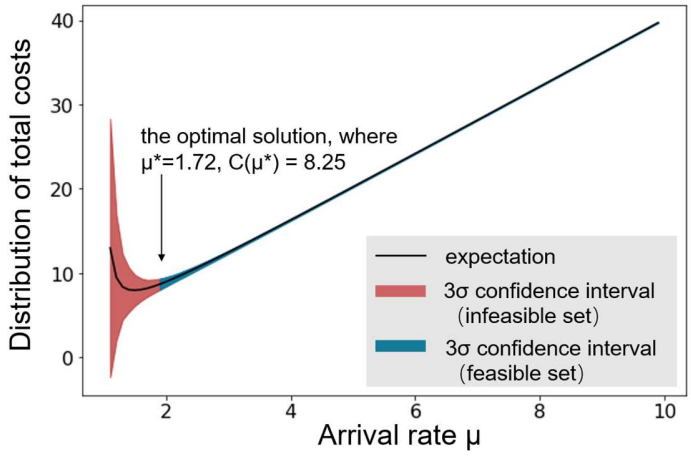
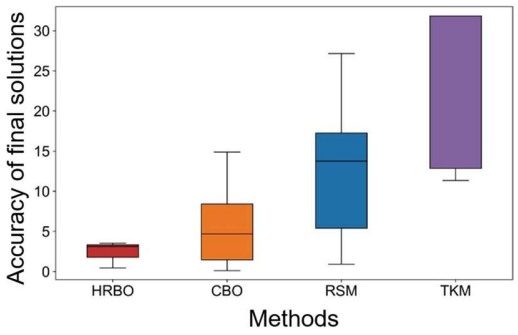


Figure E-1. The distribution of total costs with respect to μ

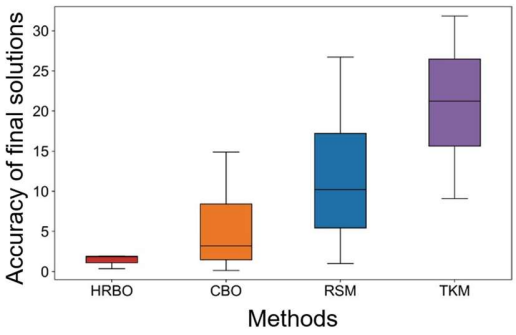
The expectation and variance of total costs are enumerated when μ varies from 1 to 10, with an interval of 0.01, i.e., $\mu = 1.01, 1.02, ..., 10$. For each realization of μ, 5000 simulation replications are

conducted to estimate the expectation and variance of the associated total cost. According to enumeration results, $\{\mu \mid \mu \geq 1.72\}$ defines the feasible set. Figure E-1 represents the distributions of total costs. The black line displays the expectation of total costs. The red and blue regions in Figure E-1 represent 3σ confidence intervals of total costs for feasible and infeasible sets, respectively. As shown in Figure E-1, the optimal solution is $\mu^* = 1.72$, with the associated objective function value is $C(\mu^*) = 8.25$.

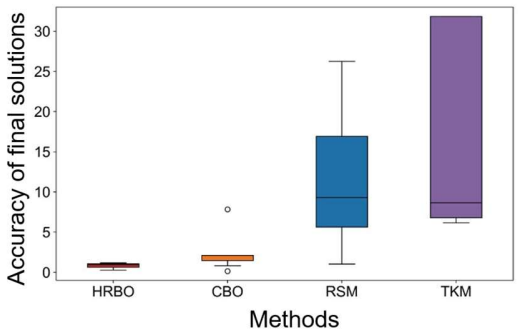
HRBO is employed to solve the SBRO problem. The performance of HRBO is compared with that of three benchmark methods: CBO, RSM, and TKM. We adopt the four methods to solve the SBRO problem 10 times. In each experiment, the four methods take the same initial sample points and the same computational resources. We consider three levels of computational budgets: low (200 simulation in total), medium (300 simulation in total), and high (400 simulation in total). In each experiment, the four methods are executed until the computational budget is exhausted. Figure E-2 records the best solutions identified by each method and their accuracy, measured as the deviation from the minimal objective function value (8.25). It notes that only feasible solutions are considered, as some final solutions may be infeasible. Figure E-2 shows that HRBO consistently achieves high-accuracy solutions with limited computational resources, whereas CBO requires significantly more resources to achieve similar precision in solving the SBRO problem. Additionally, doubling computational resources only yields marginal improvements in solution accuracy for RSM and TKM, which still underperform compared to HRBO.



(a) Low budget (200 simulation replications)



(a) Medium budget (300 simulation replications)



(c) High budget (400 simulation replications)

Figure E-2. Accuracy of final solutions for each method under different computational budgets

In practical applications, the termination conditions for the SBO algorithm are often diverse. We then evaluate algorithms under multiple termination criteria. The stopping criteria of HRBO, RSM, and CBO are set as either the following three conditions is met: *i*) the total number of simulation calls reaches 1000, *ii*) the minimal objective function value of samples reaches the global optimum (i.e., 8.25), and *iii*) the best-found solution remains unchanged for 10 iterations. Since TKM does not sequentially select samples, sample points are randomly drawn and used to identify robust solutions. Table E-1 represents, for each method, the average accuracy of final solutions and the average number of simulation replications of the 10 experiments. Among 10 experiments, RSM and TKM failed to identify robust solutions for five and two experiments, respectively. By contrast, CBO and HRBO can identify robust solutions for all 10 experiments. Table 5 implies that HRBO takes only 41.5% of the simulation replications of CBO, and achieves solutions of higher accuracy than CBO. Therefore, in comparison with benchmark methods, the proposed method achieves satisfactory solutions with higher efficiency.

Table E-1. Average accuracy of final solutions and average simulation replication numbers of the four methods within 10 experiments

Method	**HRBO**	**CBO**	**RSM**	**TKM**
Number of feasible solutions	10	10	5	8
Average accuracy of final solutions*	0.13	1.53	10.69	2.82
Average simulation replication number	335	810	200	1000

*Only feasible solutions are considered in calculating the average accuracy of final solutions

We then counted the distribution of simulation replications within the 10 experiments. Figure E-3 shows a frequency histogram that displays, for each method, the frequency of simulation replications in different bins. The x-axis represents the value of the decision variable, while the y-axis represents the *frequency to bin width ratio*. The bin width is fixed at 0.2 for all four panels. Since the bin width is fixed, a large *frequency to bin width ratio* means a large number of samples or simulation replications in the bin.

As shown in Figure E-3, CBO and TKM tend to uniformly select samples. For RSM, most samples are concentrated around an infeasible set, $1 < \mu \leq 1.2$. In contrast, the simulation replications of HRBO are concentrated around the optimal solution. This demonstrates the capability of HRBO to identify promising regions and allocate simulation resources accordingly.

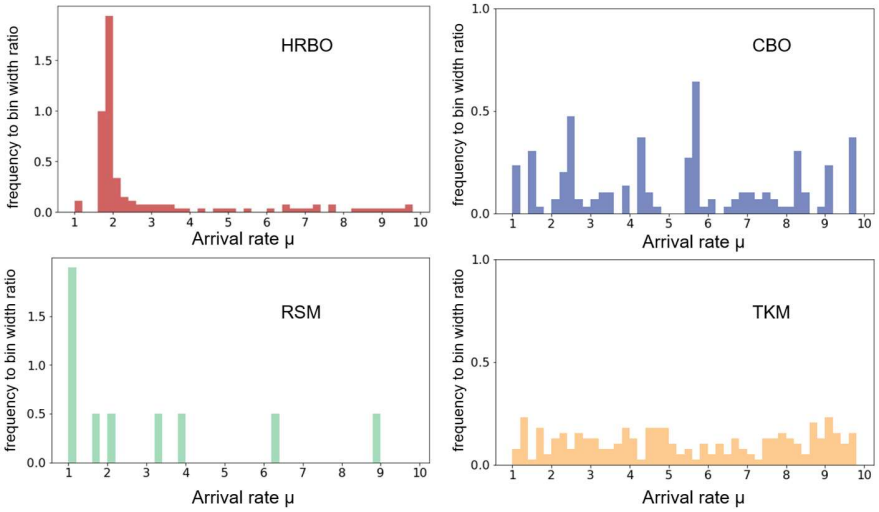


Figure E-3. Distribution of simulation replications within the 10 experiments