# Gaussian Process Based Search for Continuous Optimization via Simulation

**Preprint** · May 2021

CITATIONS
0

READS
224

**4 authors**, including:

Xiuxian Wang
Shanghai Jiao Tong University
**10** PUBLICATIONS **49** CITATIONS
SEE PROFILE

L. Jeff Hong
Fudan University
**106** PUBLICATIONS **2,162** CITATIONS
SEE PROFILE

Haihui Shen
City University of Hong Kong
**8** PUBLICATIONS **27** CITATIONS
SEE PROFILE

# Gaussian Process Based Search for Continuous Optimization via Simulation

Xiuxian Wang[a], L. Jeff Hong[b,*], Zhibin Jiang[a, c], Haihui Shen[a]

[a]Sino-US Global Logistics Institute, Shanghai Jiao Tong University, Shanghai, China

[b]School of Management and School of Data Science, Fudan University, Shanghai, China

[c]Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China

**Abstract**

Random search is an important category of algorithms to solve continuous optimization via simulation (COvS) problems. To design an efficient random search algorithm, the handling of the triple "E", i.e., exploration, exploitation and estimation, is critical. The first two E's refer to the design of sampling distribution to balance explorative and exploitative searches. The third E refers to the estimation of objective function values based on noisy simulation observations. In this paper we propose a Gaussian process based random search algorithm for COvS problems, which is called the GPS-C algorithm. By utilizing the properties of Gaussian processes, the GPS-C algorithm achieves a seamless integration of the single-observation estimation scheme and the adaptive sampling distributions. Under the assumption of homoscedastic and unknown simulation noises, we prove the global convergence of the GPS-C algorithm. Moreover, for a specific class of Gaussian processes, we show that the GPS-C algorithm has a rate of convergence close to $O_p(n^{-1/(d+2)})$. Numerical experiments show that the GPS-C algorithm has excellent performances, even for problems with heteroscedastic simulation noises.

KEYWORDS: continuous optimization via simulation; random search algorithms; Gaussian process regression; convergence; rate of convergence.

## 1 Introduction

Stochastic simulation is an important modeling tool for complex systems. It is widely used in the area of operations research and management science to model and to optimize the performances of supply chain networks, healthcare systems, transportation systems etc. This approach of stochastic optimization is often called optimization via simulation (OvS), where decision variables are typically the design parameters of the simulation models. When the decision variables are continuous, the problem is known as a continuous OvS (COvS) problem.

Examples of COvS problems include inventory-level optimization to minimize the total expected production cost, appointment-time optimization to minimize the total expected patient waiting

*Corresponding author

time, traffic-signal optimization to optimize the throughput of a transportation hub, and many others. Readers may refer to Amaran et al. (2016) for a comprehensive introduction to COvS and the related algorithms. Recently, parameter tuning is gaining a lot of research interests, especially in the area of machine learning where complicated stochastic black-box models need to be tuned. It is interesting to note that many of these problems may be viewed as COvS problems as well, where the stochastic black-box model and a call to the model may be treated as a stochastic simulation model and an experiment of the model, respectively. Readers may refer to Yu and Zhu (2020) for a comprehensive introduction to parameter tuning and the related algorithms.

Many types of algorithms have been proposed to solve COvS problems, including stochastic approximation algorithms (Robbins and Monro 1951, Kiefer et al. 1952, Spall 1992, Harold et al. 1997), response surface methodologies (Box and Wilson 1951, Kleijnen 1998, Chang et al. 2013) and random search algorithms (Andradóttir 2006, 2015). Different types of algorithms offer different types of convergence guarantees and are applicable to different settings of COvS problems. In this paper we focus on random search algorithms, which typically do not require gradient information, have global convergence and work for a wide range of COvS problems.

The key to designing efficient random search algorithms is the handling of the "triple E", i.e., exploration, exploitation and estimation (Andradóttir and Prudius 2009). The first two E's focus on the designing of sampling distributions used in the algorithm iterations to place the search effort so that it can balance global and local searches, also known as explorative and exploitative searches. The third E focuses on the estimation of objective values using noisy simulation outputs. Next we briefly review the literature on random search COvS algorithms along these two lines (i.e., designing of sampling distributions and estimation of objective values) and position our work relative to the literature.

In terms of sampling distributions, Sun et al. (2014) divide random search discrete OvS (DOvS) algorithms into four classes, exploration-based, exploitation-based, combined and integrated, based on their approaches to handle the exploration and exploitation tradeoff. Their classification is also applicable to random search COvS algorithms. *Exploration-based algorithms* include the simple random search algorithm of Chia and Glynn (2013) and the grid search algorithms of Ensor and Glynn (1997) and Yakowitz et al. (2000), which represent the feasible region by a set of either randomly generated solutions or equally spaced grid points and evaluate all of them. *Exploitation-based algorithms* include the surrogate-based promising area search algorithm of Fan and Hu (2018), which samples only from the most promising area in each iteration. Based on Sun et al. (2014), "*combined algorithms* typically focus on exploitative search while either adding a fixed amount of effort in each iteration or assigning a fixed sequence of iterations to conduct explorative search," and "*integrated algorithms* typically have an integrated sampling distribution governing the search effort in each iteration instead of separating the exploitation and exploration as in the combined algorithms." The adaptive sampling and resampling (ASR) algorithm of Andradóttir and Prudius (2010) is an example of the combined algorithms. It samples from the feasible region in each iteration from a predetermined sampling distribution and add resampling from some of the previously

visited solutions. The MARS algorithm of Hu et al. (2007) and the Gaussian-mixture model-based random search of Sun et al. (2018) are both examples of integrated algorithms. In each iteration, they both build surrogate models based on the simulation observations collected through the iteration and construct a sampling distribution to guide the random search. Indeed, the use of surrogate models in guiding random search is very common in DOvS problems and deterministic continuous black-box optimization. Among different surrogate models, the Gaussian process regression (also known as kriging) is the most popular one, because its mean and variance surfaces naturally provide information on exploitation and exploration. It is used by the Gaussian process-based search (GPS) algorithm of Sun et al. (2014) and the efficient global optimization (EGO) algorithm of Jones et al. (1998) and many other Bayesian optimization algorithms (Frazier 2018).

In terms of the estimation of objective values, there are in general two different approaches, the multi-observation approach and the single-observation approach. The multi-observation approach estimates the objective value based on repeatedly sampling the same solution and builds the convergence based on the strong law of large numbers. For instance, the simple random search algorithms of Chia and Glynn (2013) and the grid search algorithms of Ensor and Glynn (1997) and Yakowitz et al. (2000) all use the multi-observation approach. The single-observation approach samples each solution only once, but relies on the samples from other solutions to ensure the convergence to the true objective value. To the best of our knowledge, this approach dated back to Devroye (1978), who uses a $k$-nearest neighbor (KNN) scheme to estimate the objective value of any feasible solution. In recent years, the single-observation approach becomes more popular due to its superior empirical performance, and it is typically implemented through a shrinking-ball mechanism that is very similar to the KNN scheme (Baumert and Smith 2002). For instance, the algorithms of Andradóttir and Prudius (2010) and Fan and Hu (2018) all use the shrinking-ball mechanism. Kiatsupaibul et al. (2018) propose a general framework of random search algorithms with shrinking-ball mechanism and provide conditions under which the algorithms are convergent.

In this paper our goal is to propose a *single-observation integrated COvS algorithm* that uses the Gaussian process regression to construct a surrogate model in each iteration. We achieve this by extending the GPS algorithm of Sun et al. (2014) from DOvS to COvS, and therefore, we call it the GPS-C algorithm. *Notice that the extension is straightforward and is hardly a contribution of this paper.* The difficulty lies in the convergence analysis, where there is a discrepancy that one has to overcome. The Gaussian-process surrogate model of the GPS-C algorithm provides an estimate of the objective value for every feasible solution, which may be different from the one estimated through the shrinking ball. One way to resolve this discrepancy is to use only the surrogate model to construct the sampling distribution and use only the shrinking ball to estimate the objective value. However, we are not content with this approach, and we aim to build an integrated algorithm that uses the surrogate model both for constructing the sampling distribution and for estimating the objective value and, therefore, seamlessly integrates exploration, exploitation and estimation. This presents a theoretical challenge, i.e., how to develop global convergence for the GPS-C algorithm without using explicitly the shrinking-ball mechanism. In this paper we solve this problem by

exploring the properties of Gaussian process regression, and prove the surrogate model converges uniformly to the true objective function. This result is important in its own, and it has never been established in the kriging literature where function values are observed with noise, e.g., through stochastic simulation experiments.

While there are many globally convergent random search algorithms for COvS problems, very few of them have rate of convergence results. Indeed, only exploration-based algorithms, such as simple random search algorithm of Chia and Glynn (2013) and grid search algorithms of Ensor and Glynn (1997) and Yakowitz et al. (2000) have known rate of convergence. This is because these algorithms have very simple structure, e.g., determining the set of candidate solutions at the beginning and allocating the same number of observations for all candidate solutions, and are in general easy to analyze. When the sampling distributions are more complicated, rate of convergence results are in general very difficult to establish. In this paper we are able to prove, under some technical conditions, the rate of convergence of the GPS-C algorithm is $\tilde{O}_p\left(n^{-1/(d+2)}\right)$, where $d$ is the dimension of the decision variables and $\tilde{O}_p(\cdot)$ is a big-O notation ignoring logarithmic factors. To the best of our knowledge, such rate of convergence result has not been established for adaptive random search COvS algorithms before. This result also depends critically on the use of Gaussian process. Recall that the EGO algorithm of Jones et al. (1998) for deterministic continuous black-box optimization problems has a rate of convergence of $O_p\left(n^{-1/d}\right)$ (Bull 2011) and the Kiefer-Wolfowitiz stochastic approximation has a rate of convergence of $O_p\left(n^{-1/3}\right)$ for convex optimization problems for any $d$ including $d = 1$ (Spall 1992). These results suggest that the rate of convergence of the GPS-C algorithm is quite tight.

To summarize, this paper contributes to existing literature on COvS in a few aspects. First, it establishes the global convergence of the single-observation GPS-C algorithm without the explicit use of the shrinking-ball mechanism, so that the surrogate models may be used both for constructing sampling distribution and estimating the objective values. Second, it establishes the rate of convergence of the GPS-C algorithm, which appears to be the first of such results for adaptive random search algorithms. We want to emphasize that the techniques and intermediate results used in establishing the convergence and rate of convergence may be extendable to other surrogate-based OvS algorithms and other applications of Gaussian process regression.

There are also some drawbacks in the methods used in establishing the convergence and rate of convergence results of the GPS-C algorithm. The results depend critically on two assumptions, i.e., the objective function is a sample path from a Gaussian process and the simulation noises follow a normal distribution with unknown but equal variances. Even though the results may be extended (straightforwardly) to simulation noises with unequal but known variances, we admit these assumptions are restrictive. However, they are needed because we rely heavily on the properties of Gaussian process regression, which only work under these assumptions at this moment. In the numerical study, we relax these assumptions and find that the GPS-C algorithm is robust and has good numerical performance even when the assumptions are not satisfied.

The rest of this paper is organized as follows. In Section 2, we describe the problem setting

and introduce the algorithm in detail. In Section 3, we analyze the algorithm and show its global convergence. In Section 4, the rate of convergence is established for Gaussian process with Gaussian correlation function. Illustrative numerical experiments are presented in Section 5. We conclude in Section 6 and include some technical proofs in the Appendices.

## 2 The Problem and the GPS-C Algorithm

We are interested in solving the COvS problems with the following form

$$\max_{\boldsymbol{x}\in\mathcal{X}} \ \mathbb{E}[G(\boldsymbol{x};\omega)], \tag{1}$$

where $\boldsymbol{x}$ is a vector of continuous decision variables and $\omega$ represents the randomness of simulation experiments, and the expectation is taken with respect to $\omega$. Let $g(\boldsymbol{x}) = \mathbb{E}[G(\boldsymbol{x};\omega)]$. The functional form of $g(\boldsymbol{x})$ is unknown to us and can only be evaluated via noisy simulation observation $G(\boldsymbol{x};\omega)$, which is denoted as $G(\boldsymbol{x})$ in the sequel for short. We make the following Assumption 1 on the objective function $g(\boldsymbol{x})$ and the feasible region $\mathcal{X}$.

**Assumption 1.** The objective function $g(\boldsymbol{x})$ is continuous on $\mathcal{X}$, and the feasible region $\mathcal{X}$ is a compact set in $\mathbb{R}^d$ that satisfies $\text{cl}(\text{int}(\mathcal{X})) = \mathcal{X}$, where $\text{cl}(\mathcal{A})$ and $\text{int}(\mathcal{A})$ denote the closure and interior of a set $\mathcal{A}$, respectively.

The condition that $\text{cl}(\text{int}(\mathcal{X})) = \mathcal{X}$ in Assumption 1 is a condition on constraint qualifications. It implies that for any boundary point $\bar{\boldsymbol{x}} \in \mathcal{X}$, there exists a sequence of interior points $\{\boldsymbol{x}_i\}$ such that $\boldsymbol{x}_i \to \bar{\boldsymbol{x}}$. This is similar to the Slater's condition for convex constrained optimization problems (Boyd et al. 2004), which ensures strict feasibility, though the feasible region $\mathcal{X}$ needs not to be convex. With Assumption 1, for any $\boldsymbol{x} \in \mathcal{X}$ and any $d$-dimensional ball centered at $\boldsymbol{x}$ with positive radius, say $\mathcal{S}(\boldsymbol{x})$, the volume of $\mathcal{X} \cap \mathcal{S}(\boldsymbol{x})$ is larger than zero. Based on that we can ensure that $\mathcal{X} \cap \mathcal{S}(\boldsymbol{x})$ has positive probability to be sampled for any $\boldsymbol{x} \in \mathcal{X}$, if the sampling distribution has positive density on $\mathcal{X}$.

Let $\varepsilon(\boldsymbol{x}) = G(\boldsymbol{x}) - g(\boldsymbol{x})$ be the simulation noise at each point $\boldsymbol{x} \in \mathcal{X}$. We make the following Assumption 2 on $\varepsilon(\boldsymbol{x})$.

**Assumption 2.** For all $\boldsymbol{x} \in \mathcal{X}$, $\varepsilon(\boldsymbol{x})$ follows a normal distribution with mean 0 and variance $\lambda^2$, i.e., $\varepsilon(\boldsymbol{x}) \sim \mathcal{N}(0, \lambda^2)$, where $\lambda^2$ is a constant but *unknown* variance of simulation noise.

The normality and homoscedasticity assumptions of the simulation noise $\varepsilon(\boldsymbol{x})$ are made because the convergence analysis of the GPS-C algorithm is based on the properties of the kriging surfaces. In the literature of Gaussian process regression, results are typically derived under the assumptions of known (and possibly unequal) variances (e.g., the stochastic kriging approach of Ankenman et al. (2010)) or unknown but equal variances (e.g., the co-kriging regression of Forrester et al. (2007)). In the COvS setting, variances are typically unknown, and therefore, we have to make the assumption that the variances are equal. However, we also want to note that the convergence results developed

in this paper may be extended straightforwardly to the problems where the variances are known but unequal. If the assumption is not satisfied, one may carefully adjust the batch size at each design point so that the sample variances of the batch means at all design points are approximately normal and equal.

### 2.1 Gaussian Process Regression

The GPS-C algorithm uses Gaussian process regression to build a surrogate model of the objective function $g(\boldsymbol{x})$ in each iteration of the algorithm. It takes a Bayesian viewpoint and assumes that the unknown objective function $g(\boldsymbol{x})$ is a (random) sample path of a Gaussian process $f_{\mathcal{GP}}$ on $\mathcal{X}$, with the *mean function* $\mu_0 : \mathcal{X} \to \mathbb{R}$, defined by $\mu_0(\boldsymbol{x}) = \mathbb{E}[f_{\mathcal{GP}}(\boldsymbol{x})]$, and the *covariance function* $k_0 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined by $k_0(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f_{\mathcal{GP}}(\boldsymbol{x}) - \mu_0(\boldsymbol{x}))(f_{\mathcal{GP}}(\boldsymbol{x}') - \mu_0(\boldsymbol{x}'))]$. In the GPS-C algorithm, we require the mean and covariance functions to satisfy the following assumption.

**Assumption 3.** The mean function $\mu_0(\boldsymbol{x})$ is continuous on $\mathcal{X}$, and the covariance function $k_0(\boldsymbol{x}, \boldsymbol{x}') = \tau^2\rho(\boldsymbol{x} - \boldsymbol{x}')$ for some $\tau > 0$ and some continuous function $\rho : \mathbb{R}^d \to \mathbb{R}$, which further satisfies the following:

(i) $\rho(|\boldsymbol{\delta}|) = \rho(\boldsymbol{\delta})$, where $|\cdot|$ means taking absolute value component-wise;

(ii) $\rho(\boldsymbol{\delta})$ is decreasing in $\boldsymbol{\delta}$ component-wise for $\boldsymbol{\delta} \geq \mathbf{0}$;

(iii) $\rho(\mathbf{0}) = 1$, $\rho(\boldsymbol{\delta}) \to 0$ as $\|\boldsymbol{\delta}\| \to \infty$, and for some $0 < C < \infty$ and some $\epsilon, \delta > 0$, $1 - \rho(\boldsymbol{\delta}) \leq C|\log(\|\boldsymbol{\delta}\|)|^{-1-\epsilon}$ for all $\|\boldsymbol{\delta}\| < \delta$, where $\|\cdot\|$ denotes the Euclidean norm.

Assumption 3 is in general a weak assumption and most covariance functions used in practice satisfy it. Notable examples include the power exponential covariance function $k_0(\boldsymbol{x}, \boldsymbol{x}') = \tau^2 \exp\left\{-\sum_{j=1}^d \theta_j |x_j - x_j'|^\kappa\right\}$ with $\theta_j > 0$ and $0 < \kappa \leq 2$, and the Matérn covariance function; see Rasmussen and Williams (2006, Chapter 4) for more types of covariance functions. The mean and covariance functions reflect one's prior belief about the unknown function $g(\boldsymbol{x})$, and is subject to user's choice. When no structural information for $g(\boldsymbol{x})$ is available, it is a convention to set $\mu_0 \equiv 0$. Shen et al. (2018) demonstrate that it is beneficial to embed some stylized models into $\mu_0$ if they are capable of capturing the structure information of $g(\boldsymbol{x})$. Furthermore, Assumption 3 also implies that the correlation function $\frac{1}{\tau^2}k_0$ is *stationary*, i.e., it depends on $\boldsymbol{x}$ and $\boldsymbol{x}'$ only through the difference $\boldsymbol{x} - \boldsymbol{x}'$, and the sample paths of $f_{\mathcal{GP}}$ are continuous with probability one (Adler and Taylor 2007, Theorem 1.4.1).

Suppose that the GPS-C algorithm has simulated a set of solutions, denoted by $\boldsymbol{X}^n = \{\boldsymbol{x}_i\}_{i=1}^n$ with the corresponding simulation observations $\boldsymbol{G}^n = (G(\boldsymbol{x}_1), \ldots, G(\boldsymbol{x}_n))^\intercal \in \mathbb{R}^n$. Then, conditioned on these observations, the conditional Gaussian process is still a Gaussian process whose mean and covariance functions are given by $\mu_n(\boldsymbol{x}) = \mathbb{E}[f_{\mathcal{GP}}(\boldsymbol{x})|\{\boldsymbol{X}^n, \boldsymbol{G}^n\}]$ and $k_n(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f_{\mathcal{GP}}(\boldsymbol{x}) - \mu_n(\boldsymbol{x}))(f_{\mathcal{GP}}(\boldsymbol{x}') - \mu_n(\boldsymbol{x}'))|\{\boldsymbol{X}^n, \boldsymbol{G}^n\}]$, respectively. By Assumption 3, they can be expressed as

$$\mu_n(\boldsymbol{x}) = \mu_0(\boldsymbol{x}) + \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\lambda^2/\tau^2)\boldsymbol{I}^n]^{-1}[\boldsymbol{G}^n - \mu_0(\boldsymbol{X}^n)], \tag{2}$$

$$k_n(\boldsymbol{x}, \boldsymbol{x}') = \tau^2\big\{\rho(\boldsymbol{x}, \boldsymbol{x}') - \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\lambda^2/\tau^2)\boldsymbol{I}^n]^{-1}\rho(\boldsymbol{X}^n, \boldsymbol{x}')\big\}, \tag{3}$$

where $\boldsymbol{I}^n$ is $n$-dimensional identity matrix, $\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) = [\rho(\boldsymbol{x}_i - \boldsymbol{x}_j)]_{1\le i,j\le n} \in \mathbb{R}^{n\times n}$, $\rho(\boldsymbol{x}, \boldsymbol{X}^n) = (\rho(\boldsymbol{x} - \boldsymbol{x}_1), \ldots, \rho(\boldsymbol{x} - \boldsymbol{x}_n)) \in \mathbb{R}^{1\times n}$, and $\rho(\boldsymbol{X}^n, \boldsymbol{x}') = (\rho(\boldsymbol{x}_1 - \boldsymbol{x}'), \ldots, \rho(\boldsymbol{x}_n - \boldsymbol{x}'))^\intercal \in \mathbb{R}^n$.

Notice that $\mu_n(\boldsymbol{x})$ may be viewed as our prediction of $g(\boldsymbol{x})$ given the observations $\{\boldsymbol{X}^n, \boldsymbol{G}^n\}$. It may be served as the basis for exploitative search. Unlike many other surrogate-building approaches, the Gaussian process regression also provides information on the uncertainty of the prediction, measured by the variance of the conditional Gaussian process, i.e., $k_n(\boldsymbol{x}, \boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. As noted by Sun et al. (2014), it serves as the basis for explorative search. In the following Section 2.2, we show how to combine $\mu_n(\boldsymbol{x})$ and $k_n(\boldsymbol{x}, \boldsymbol{x})$ to construct a sampling distribution that balances both the exploitative and explorative searches.

Before moving to the introduction of the detailed GPS-C algorithm, there is a final technical issue that we have to resolve. To use Equations (2) and (3), one needs to know the variance $\lambda^2$ of the simulation noises (defined in Assumption 2). However, it is typically unknown. We suggest to substitute it by a reasonable estimate of $\lambda^2$, denoted by $\tilde{\lambda}^2 > 0$, and use

$$\tilde{\mu}_n(\boldsymbol{x}) = \mu_0(\boldsymbol{x}) + \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\tilde{\lambda}^2/\tau^2)\boldsymbol{I}^n]^{-1}[\boldsymbol{G}^n - \mu_0(\boldsymbol{X}^n)], \tag{4}$$

$$\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x}') = \tau^2\big\{\rho(\boldsymbol{x}, \boldsymbol{x}') - \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\tilde{\lambda}^2/\tau^2)\boldsymbol{I}^n]^{-1}\rho(\boldsymbol{X}^n, \boldsymbol{x}')\big\}. \tag{5}$$

At the first sight, if the assigned $\tilde{\lambda}^2$ is not equal to $\lambda^2$, the practically used conditional mean function $\tilde{\mu}_n(\boldsymbol{x})$ is not equal to $\mu_n(\boldsymbol{x})$. However, notice that $\tau^2$ may be set arbitrarily, by replacing $\tau^2$ with $\lambda^2\tau^2/\tilde{\lambda}^2$ in Equation (2), we have $\tilde{\mu}_n(\boldsymbol{x}) = \mu_n(\boldsymbol{x})$. At this moment, the conditional variance function $k_n(\boldsymbol{x}, \boldsymbol{x}')$ differs from $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x}')$ with a constant ratio $\lambda^2/\tilde{\lambda}^2$, i.e., $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x}') = (\lambda/\tilde{\lambda})^2 \cdot \tilde{k}_n(\boldsymbol{x}, \boldsymbol{x}')$. We show that in Sections 3 and 4, the choice of $\tilde{\lambda}^2$ and this constant ratio do not affect the convergence and rate of convergence of the GPS-C algorithm.

## 2.2 The GPS-C Algorithm

### 2.2.1 Sampling Distribution

Sampling distribution is a key element of a random search algorithm, which is constructed in each iteration to determine where to allocate the simulation effort. As shown by Sun et al. (2014), their sampling distribution constructed based on Gaussian process can adaptively balance the trade-off between exploration and exploitation when solving DOvS problems. For COvS problems, we can construct the sampling distributions in a similar way.

From Equations (2) and (3), for any $\boldsymbol{x} \in \mathcal{X}$, we have

$$f_{\mathcal{GP}}(\boldsymbol{x})|\{\boldsymbol{X}^n, \boldsymbol{G}^n\} \sim \mathcal{N}(\mu_n(\boldsymbol{x}), k_n(\boldsymbol{x}, \boldsymbol{x})). \tag{6}$$

Then, we may define a probability density function $f_n(\boldsymbol{x})$ as follows:

$$f_n(\boldsymbol{x}) = \frac{\mathbb{P}\{Z(\boldsymbol{x}) > c\}}{\int_{\mathcal{X}} \mathbb{P}\{Z(\boldsymbol{z}) > c\} \mathrm{d}\boldsymbol{z}}, \quad \boldsymbol{x} \in \mathcal{X}, \tag{7}$$

where $Z(\boldsymbol{x}) \sim \mathcal{N}(\mu_n(\boldsymbol{x}), k_n(\boldsymbol{x}, \boldsymbol{x}))$, $c = \max_{\boldsymbol{x} \in \mathcal{X}} \mu_n(\boldsymbol{x})$, and $\boldsymbol{z} \in \mathbb{R}^d$. Notice that $c$ is well defined under Assumptions 1 and 3, which imply that $\mathcal{X}$ is compact and $\mu_n(\boldsymbol{x})$ is continuous. This sampling distribution maintains the desirable properties of the sampling distribution of the GPS algorithm of Sun et al. (2014): (1) It assigns higher probabilities to regions which contain good solutions (due to higher conditional mean); (2) it assigns higher probabilities to less explored regions (due to higher conditional variance). Therefore, it can balance the trade-off between exploration and exploitation adaptively.

As $\lambda^2$ is unknown, Equations (4) and (5) are actually used to construct the sampling distribution. Notice that, after replacing $\tau^2$ with $\lambda^2\tau^2/\tilde{\lambda}^2$ in Equations (2) and (3), though $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x})$ differs from $k_n(\boldsymbol{x}, \boldsymbol{x})$, it keeps the relative values of the conditional variance function $k_n(\boldsymbol{x}, \boldsymbol{x})$. Therefore, the sampling distribution constructed with $\tilde{\lambda}^2$ is still capable of balancing exploration and exploitation. Furthermore, as suggested by Sun et al. (2014), it is beneficial to set a lower bound for the density to help in the convergence analysis. Specifically, the user may specify a proper lower bound and upper bound for $\tilde{\mu}_n(\boldsymbol{x})$, say $\underline{M}$ and $\overline{M}$, and a lower bound for $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x})$, say $\underline{\tau}^2$ with $\underline{\tau} > 0$. Then, we can define $\tilde{k}_n^{\mathrm{cap}}(\boldsymbol{x}, \boldsymbol{x}) = \max\{\underline{\tau}^2, \tilde{k}_n(\boldsymbol{x}, \boldsymbol{x})\}$, and

$$\tilde{\mu}_n^{\mathrm{cap}}(\boldsymbol{x}) = \begin{cases} \underline{M}, & \text{if } \tilde{\mu}_n(\boldsymbol{x}) < \underline{M}, \\ \overline{M}, & \text{if } \tilde{\mu}_n(\boldsymbol{x}) > \overline{M}, \\ \tilde{\mu}_n(\boldsymbol{x}), & \text{otherwise.} \end{cases}$$

Then, the density of the sampling distribution used in the algorithm is given by

$$\tilde{f}_n(\boldsymbol{x}) = \frac{\mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{c}\}}{\int_{\mathcal{X}} \mathbb{P}\{\tilde{Z}(\boldsymbol{z}) > \tilde{c}\} \, \mathrm{d}\boldsymbol{z}}, \quad \boldsymbol{x} \in \mathcal{X}, \tag{8}$$

where $\tilde{Z}(\boldsymbol{x}) \sim \mathcal{N}(\tilde{\mu}_n^{\mathrm{cap}}(\boldsymbol{x}), \tilde{k}_n^{\mathrm{cap}}(\boldsymbol{x}, \boldsymbol{x}))$, and $\tilde{c} = \max_{\boldsymbol{x} \in \mathcal{X}} \tilde{\mu}_n^{\mathrm{cap}}(\boldsymbol{x})$. It is not difficult to see that $\tilde{f}_n(\boldsymbol{x})$ has a lower bound on $\mathcal{X}$, which is explicitly stated in the following Lemma 1, whose proof is provided in Appendix B.

**Lemma 1.** *Let $\alpha = 2[1 - \Phi((\overline{M} - \underline{M})/\underline{\tau})]/\nu(\mathcal{X}) > 0$, where $\Phi$ is the cumulative distribution function of the standard normal random variable, and $\nu(\mathcal{X}) = \int_{\mathcal{X}} \mathrm{d}\boldsymbol{z}$ for $\boldsymbol{z} \in \mathbb{R}^d$ is the volume of $\mathcal{X}$ . Then, $\tilde{f}_n(\boldsymbol{x}) \geq \alpha$ for all $\boldsymbol{x} \in \mathcal{X}$.*

Once we have the sampling distribution, we need an algorithm to sample from it. Notice that the explicit form of $\tilde{f}_n(\boldsymbol{x})$ is typically not applicable, because the denominator involves an integration that is computationally expensive to calculate. Here we consider two sampling schemes, which both are the direct extensions of the sampling schemes of Sun et al. (2014) to the continuous context.

Notice that $\mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{c}\} \leq \mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{\mu}_n^{\text{cap}}(\boldsymbol{x})\} = 1/2$, and

$$\tilde{f}_n(\boldsymbol{x}) = \frac{\mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{c}\}}{\int_{\mathcal{X}} \mathbb{P}\{\tilde{Z}(\boldsymbol{z}) > \tilde{c}\}\, \mathrm{d}\boldsymbol{z}} \leq \frac{(1/2)\nu(\mathcal{X})}{\int_{\mathcal{X}} \mathbb{P}\{\tilde{Z}(\boldsymbol{z}) > \tilde{c}\}\, \mathrm{d}\boldsymbol{z}} \cdot \frac{1}{\nu(\mathcal{X})}.$$

Since $\nu(\mathcal{X})/\int_{\mathcal{X}} \mathbb{P}\{\tilde{Z}(\boldsymbol{z}) > \tilde{c}\}\, \mathrm{d}\boldsymbol{z}$ is a constant and $\frac{1}{\nu(\mathcal{X})}$ is the probability density of the uniform distribution on $\mathcal{X}$, it is easy to see that the following acceptance-rejection sampling (ARS) scheme can output a sample following the density $\tilde{f}_n(\boldsymbol{x})$.

*ARS Scheme*

*Step 1.* Generate a sample $\boldsymbol{y}$ uniformly in $\mathcal{X}$ and $u$ uniformly in $(0, 1)$.

*Step 2.* If $u \leq 2\,\mathbb{P}\{\tilde{Z}(\boldsymbol{y}) > \tilde{c}\}$, return $\boldsymbol{y}$; otherwise, go to Step 1.

The ARS algorithm avoids the calculation of the integration in the denominator of Equation (8). However, when the sampling probabilities are concentrated on small regions, the acceptance rate may be very small, and thus impacts the efficiency of the ARS scheme. Therefore, a Markov chain coordinate sampling (MCCS) scheme is proposed.

*MCCS Scheme*

*Step 0.* Let $t = 0$, $\boldsymbol{y} = \boldsymbol{y}_0$. Specify the iteration number $T$.

*Step 1.* Let $t = t + 1$. Sample an integer $j$ from 1 to $d$ uniformly. Let $l(\boldsymbol{y}, j)$ be the line that passes through $\boldsymbol{y}$ and parallel to the $y_j$ coordinate axis. Then $l(\boldsymbol{y}, j) \cap \mathcal{X}$ is the line segment that is contained in $\mathcal{X}$. Sample a point on $l(\boldsymbol{y}, j) \cap \mathcal{X}$ uniformly, whose $j$-th coordinate is denoted as $b$. Set $\boldsymbol{z} = \boldsymbol{y}$ and $z_j = b$.

*Step 2.* Sample an $u$ uniformly in $(0, 1)$. If $u \leq \tilde{f}_n(\boldsymbol{z})/\tilde{f}_n(\boldsymbol{y}) = \mathbb{P}\{\tilde{Z}(\boldsymbol{z}) > \tilde{c}\}/\,\mathbb{P}\{\tilde{Z}(\boldsymbol{y}) > \tilde{c}\}$, set $\boldsymbol{y} = \boldsymbol{z}$.

*Step 3.* If $t = T$, return $\boldsymbol{y}$; otherwise go to Step 1.

Similar to the proof in Baumert et al. (2009), it can be show that, as $T \to \infty$, the probability density function of the random output $\boldsymbol{y}$ converges to $\tilde{f}_n(\boldsymbol{x})$, regardless of $\boldsymbol{y}_0$. In practice, the starting point $\boldsymbol{y}_0$ is usually the current optimal solution. The MCCS scheme guarantees to sample (approximately) a point every $T$ steps. Therefore, it may become more efficient than the ARS scheme when the acceptance rate in the ARS scheme becomes very low (i.e., lower than $1/T$).

### 2.2.2 The GPS-C Algorithm

We now present the full GPS-C algorithm in detail.

**Step 0 (Initialization).** Impose a Gaussian process with $\mu_0$ and $k_0$ that satisfy Assumption 3. Specify a $\tilde{\lambda} > 0$, $\underline{\tau} > 0$, $\underline{M}$ and $\overline{M}$. Let $r$ denote the number of solutions sampled in each iteration and set $r$ to be a positive integer. Let $s$ and $n$ denote the counters of the iterations and the sampled solutions, respectively, and let $\boldsymbol{X}^n$ and $\boldsymbol{G}^n$ denote the set of solutions and the vector of observations, respectively, as defined in Section 2.1. Set $s = 0$, $n = 0$, $\boldsymbol{X}^0 = \emptyset$ and $\boldsymbol{G}^0 = \emptyset$. Furthermore, set $\tilde{f}_0(\boldsymbol{x})$ as the uniform distribution over $\mathcal{X}$.

**Step 1 (Sampling).** Set $s = s + 1$. Sample $\boldsymbol{x}_{r(s-1)+1}, \ldots, \boldsymbol{x}_{rs}$ independently from $\tilde{f}_n(\boldsymbol{x})$ using either the ARS Sampling Scheme or the MCCS Sampling Scheme discussed in Section 2.2.1, and obtain corresponding simulation observations $G(\boldsymbol{x}_{r(s-1)+1}), \ldots, G(\boldsymbol{x}_{rs})$.

**Step 2 (Calculation).** Set $n = rs$. Let $\boldsymbol{X}^n = \boldsymbol{X}^{r(s-1)} \cup \{\boldsymbol{x}_{r(s-1)+1}, \ldots, \boldsymbol{x}_{rs}\}$ and $\boldsymbol{G}^n = \big([\boldsymbol{G}^{r(s-1)}]^\intercal, G(\boldsymbol{x}_{r(s-1)+1}), \ldots, G(\boldsymbol{x}_{rs})\big)^\intercal$. Calculate $\tilde{\mu}_n(\boldsymbol{x})$ and $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x}')$ according to Equations (4) and (5). Let $\tilde{\boldsymbol{x}}_n^* = \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} \tilde{\mu}_n(\boldsymbol{x})$ and break the tie arbitrarily if it exists. Then, construct the sampling distribution $\tilde{f}_n(\boldsymbol{x})$ according to Equation (8).

**Step 3 (Stopping).** If the stopping condition is not met, go to Step 1; otherwise, stop and output $\tilde{\boldsymbol{x}}_n^*$ and $\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*)$ as the estimated optimal solution and the estimated optimal objective value.

We summarize some features of the GPS-C algorithm for COvS problems. First, the GPS-C algorithm is a natural extension of the GPS algorithm of Sun et al. (2014) to COvS problems. The sampling distributions constructed based on the Gaussian process regression enable the GPS-C algorithm to balance the exploration and exploitation in an adaptive manner, just as the GPS algorithm does for DOvS problems. Second, the GPS-C algorithm is a single-observation random search algorithm for COvS problems. Compared with those random search algorithms with multiple observations, requiring only a single observation at each solution allows the algorithm to better explore the feasible region given the same simulation budget, which is an appealing property for COvS problems. Third, rather than averaging observations in a shrinking ball centered at a solution, which is commonly used in single-observation random search algorithms to estimate the objective function (Kiatsupaibul et al. 2018), the GPS-C algorithm uses the conditional mean function of the Gaussian process. As a result, the analysis of the asymptotic behavior of the GPS-C algorithm is different from those of shrinking ball algorithms, and it relies on the property of the Gaussian process regression.

It is worth mentioning an implementation issue of the GPS-C algorithm. We need to repeatedly calculate $\tilde{\boldsymbol{x}}_n^* = \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} \tilde{\mu}_n(\boldsymbol{x})$ in Step 2, but it is computationally difficult because $\tilde{\mu}_n(\boldsymbol{x})$ is generally non-convex. This issue is common for surrogate model-based optimization algorithms (see, for instance, the EGO algorithm of Jones et al. (1998) and the metamodel-based optimization algorithm of Osorio and Bierlaire (2013)). When the dimension is low (e.g., $d$ is small), one may simply evaluate $\tilde{\mu}_n(\boldsymbol{x})$ on a dense grid of $\boldsymbol{x}$ within $\mathcal{X}$ to approximate $\tilde{\boldsymbol{x}}_n^*$. When the dimension is high, one may set $\tilde{\boldsymbol{x}}_n^\dagger = \operatorname{argmax}_{\boldsymbol{x} \in \boldsymbol{X}^n} \tilde{\mu}_n(\boldsymbol{x})$ as the initial solution and use a nonlinear optimization solvers to find an approximate optimal solution. Our numerical experiments show that these methods work quite well.

# 3 Global Convergence

Let $g^* = \max_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$ be the optimal objective function value, and let $\mathcal{X}^* = \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} g(\boldsymbol{x})$ be the set of optimal solutions. In this section, we establish the almost sure global convergence of the

GPS-C algorithm, i.e., we prove

$$\mathbb{P}\left\{\lim_{n\to\infty}\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*)=g^*\right\}=1 \quad \text{and} \quad \mathbb{P}\left\{\lim_{n\to\infty} d(\tilde{\boldsymbol{x}}_n^*,\mathcal{X}^*)=0\right\}=1, \tag{9}$$

where for any set $\mathcal{A}\subset\mathbb{R}^d$ and a point $\boldsymbol{x}\in\mathbb{R}^d$, we define the distance from $\boldsymbol{x}$ to $\mathcal{A}$ by $d(\boldsymbol{x},\mathcal{A})=\inf_{\boldsymbol{x}'\in\mathcal{A}}\|\boldsymbol{x}-\boldsymbol{x}'\|$. Notice that Equation (9) implies that the estimated optimal value converges to the true optimal value with probability 1 and the estimated optimal solution converges to the set of true optimal solutions with probability 1.

Even though the probability statements in Equation (9) are common goals for convergence analysis for COvS algorithms, there is a subtle difference between ours and the ones in the literature in terms of the randomness considered in these statements. For most algorithms in the literature (see, for instance, the ASR algorithm of Andradóttir and Prudius (2010) and the shrinking ball algorithm of Kiatsupaibul et al. (2018)), the objective function $g(\boldsymbol{x})$ is considered deterministic (but unknown) and the randomness comes from the sampling and the simulation experiments. In our case, however, the objective function $g(\boldsymbol{x})$ is assumed to be a (random) sample path from the Gaussian process $f_{\mathcal{GP}}$. Therefore, the randomness not only comes from the sampling and the simulation experiments, but also from the Gaussian process. This treatment of the objective function is consistent with the Bayesian viewpoint of the Gaussian process regression, and it has also been used to analyze the convergence properties of the EGO algorithm (Bull 2011).

The convergence analysis of the GPS-C algorithm contains three major steps. In the first step, we establish the convergence of the conditional variance function $k_n(\boldsymbol{x},\boldsymbol{x})$. Based on that, in the second step, we prove the uniform convergence of the conditional mean function $\mu_n(\boldsymbol{x})$. However, this uniform convergence is for $\mu_n(\boldsymbol{x})$, which requires knowing the *unknown* variance $\lambda^2$. Therefore, in the third step, we substitute the unknown $\lambda^2$ with $\tilde{\lambda}^2$, show that $\tilde{\mu}_n(\boldsymbol{x})$ converges uniformly as well, and conclude the almost sure convergence of the GPS-C algorithm. In the following subsections, we elaborate these three steps.

### 3.1 The Convergence of the Conditional Variance

In this subsection, our goal is to show that the conditional variance $k_n(\boldsymbol{x},\boldsymbol{x})$ converges to zero as $n\to\infty$ for any $\boldsymbol{x}\in\mathcal{X}$. For any $\boldsymbol{x}\in\mathcal{X}$, let $\mathcal{S}(\boldsymbol{x},\epsilon)$ denote the closed $d$-dimensional ball centered at $\boldsymbol{x}$ with radius $\epsilon>0$, and let $s_n(\boldsymbol{x},\epsilon)$ denote the number of solutions in $\mathcal{S}(\boldsymbol{x},\epsilon)$ among all $n$ sampled solutions. We first establish the following lemma regarding to the asymptotic behavior of $s_n(\boldsymbol{x},\epsilon)$, whose proof is provided in Appendix B.

**Lemma 2.** *Suppose that Assumption 1 holds, and that distributions with density functions $\psi_i$ satisfying $\psi_i\geq\alpha>0$ on $\mathcal{X}$ are used to generate solutions $\boldsymbol{x}_i\in\boldsymbol{X}^n$, for $i=1,\ldots,n$. Then, for any fixed $\epsilon>0$ and any $\boldsymbol{x}\in\mathcal{X}$, $s_n(\boldsymbol{x},\epsilon)\to\infty$ almost surely as $n\to\infty$.*

Lemma 2 shows that for any small ball centered at $\boldsymbol{x}\in\mathcal{X}$ with radius $\epsilon$, the number of sampled solutions in that ball goes to infinity as $n\to\infty$. This implies that the sampled solution will eventually be dense on $\mathcal{X}$, and it provides a preliminary result for our convergence analysis.

Another preliminary result is that the conditional variance is upper bounded, which is a direct result of Lemma 4 of Zhang et al. (2019), and is provided in the following Lemma 3.

**Lemma 3** (Zhang et al. (2019), Lemma 4)**.** *Fix a compact set $\mathcal{A} \subset \mathcal{X}$. Suppose $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{A}$. If Assumptions 2 and 3 hold, then for any $\boldsymbol{x} \in \mathcal{A}$,*

$$k_n(\boldsymbol{x}, \boldsymbol{x}) \leq \tau^2 - \frac{n \min_{\boldsymbol{x}' \in \mathcal{A}} [k_0(\boldsymbol{x}, \boldsymbol{x}')]^2}{n\tau^2 + \lambda^2},$$

*where $k_n(\cdot, \cdot)$ is defined in Equation (3).*

Based on Lemmas 2 and 3, we have the following lemma on the convergence of the conditional variance function. This is an important result and its proof is provided in Appendix A.

**Lemma 4.** *Suppose that Assumptions 1–3 hold, and that distributions with density functions $\psi_i$ satisfying $\psi_i \geq \alpha > 0$ on $\mathcal{X}$ are used to generate solutions $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$. Then, for any $\boldsymbol{x} \in \mathcal{X}$, $k_n(\boldsymbol{x}, \boldsymbol{x}) \to 0$ almost surely as $n \to \infty$.*

Lemma 4 provides the almost-sure pointwise convergence of the conditional variance $k_n(\boldsymbol{x}, \boldsymbol{x})$ to zero as $n \to \infty$. By Chebyshev's inequality, Lemma 4 further implies that the conditional mean function $\mu_n(\boldsymbol{x})$ converges in probability for any $\boldsymbol{x} \in \mathcal{X}$. However, this pointwise convergence is not enough. In the following subsection, we show that the conditional mean function converges uniformly.

### 3.2 The Uniform Convergence of the Conditional Mean Function

Notice that Lemma 4 only implies the pointwise convergence of $\mu_n(\boldsymbol{x})$ to $\mathbb{E}[\mu_n(\boldsymbol{x})]$. However, this is not enough. In this subsection, our goal is to show that the conditional mean function $\mu_n(\boldsymbol{x})$ converges to $g(\boldsymbol{x})$ uniformly as $n \to \infty$. The following lemma of Bect et al. (2019) is critical to fill the gap. It shows that $\mu_n(\boldsymbol{x})$ converges uniformly to a limiting function.

**Lemma 5** (Bect et al. (2019), Proposition 2.9)**.** *Suppose that Assumptions 1–3 hold. Then, $\mu_n(\boldsymbol{x})$ converges uniformly on $\mathcal{X}$ to a function, denoted by $\mu_\infty(\boldsymbol{x})$, almost surely as $n \to \infty$.*

Notice that Lemma 5 only ensures that $\mu_n(\cdot)$ converges uniformly to a certain function, which is not necessarily $g(\boldsymbol{x})$. By combining Lemmas 4 and 5, we have the following proposition, which shows that the limit is indeed $g(\boldsymbol{x})$.

**Proposition 1.** *Suppose that Assumptions 1–3 hold, and that distributions with density functions $\psi_i$ satisfying $\psi_i \geq \alpha > 0$ on $\mathcal{X}$ are used to generate solutions $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$. Then, $\mu_n(\boldsymbol{x}) \to g(\boldsymbol{x})$ uniformly on $\mathcal{X}$ almost surely as $n \to \infty$.*

*Proof.* Fix any $\boldsymbol{x} \in \mathcal{X}$. First notice that

$$\mathbb{E}[k_n(\boldsymbol{x}, \boldsymbol{x})] = \mathbb{E}[\mathbb{E}[(g(\boldsymbol{x}) - \mu_n(\boldsymbol{x}))^2 | \{\boldsymbol{X}^n, \boldsymbol{G}^n\}]] = \mathbb{E}[(g(\boldsymbol{x}) - \mu_n(\boldsymbol{x}))^2]. \tag{10}$$

By Lemma 4, $k_n(\boldsymbol{x}, \boldsymbol{x}) \to 0$ almost surely as $n \to \infty$. Then, together with the fact that $0 \leq k_n(\boldsymbol{x}, \boldsymbol{x}) \leq k_0(\boldsymbol{x}, \boldsymbol{x}) = \tau^2$ from Equation (22) in the proof of Lemma 4, we can conclude that $\mathbb{E}[k_n(\boldsymbol{x}, \boldsymbol{x})] \to \mathbb{E}[0] = 0$, by the dominated convergence theorem (Durrett 2010, Theorem 1.5.6). It implies that $\mathbb{E}[(g(\boldsymbol{x}) - \mu_n(\boldsymbol{x}))^2] \to 0$, i.e., $\mu_n(\boldsymbol{x}) \to g(\boldsymbol{x})$ in $\mathsf{L}^2$, for any $\boldsymbol{x} \in \mathcal{X}$. Lemma 5 implies that $\mu_n(\boldsymbol{x}) \to \mu_\infty(\boldsymbol{x})$ almost surely on $\mathcal{X}$. Due to the almost sure uniqueness of convergence in probability (Gut 2013, Theorem 2.1 of Chapter 5), it can be obtained that $\mathbb{P}\{\mu_\infty(\boldsymbol{x}) = g(\boldsymbol{x})\} = 1$, for any $\boldsymbol{x} \in \mathcal{X}$.

Consider a dense but countable subset $\overline{\mathcal{X}}$ of $\mathcal{X}$, e.g., the set of all $\boldsymbol{x} \in \mathcal{X}$ such that all elements of $\boldsymbol{x}$ are rational numbers. Then, we have $\mathbb{P}\{\mu_\infty(\boldsymbol{x}) = g(\boldsymbol{x}), \text{ for all } \boldsymbol{x} \in \overline{\mathcal{X}}\} = 1$, since the probability measure is a nonnegative countably additive set function (Durrett 2010, p. 1). We now focus on one generic sample path such that $\mu_\infty(\boldsymbol{x}) = g(\boldsymbol{x})$ for all $\boldsymbol{x} \in \overline{\mathcal{X}}$ and $\mu_n(\boldsymbol{x}) \to \mu_\infty(\boldsymbol{x})$ uniformly on $\mathcal{X}$ (by Lemma 5). Recall that $\mu_n(\boldsymbol{x})$ is continuous on $\mathcal{X}$ for each $n$, then $\mu_\infty(\boldsymbol{x})$ is also continuous on $\mathcal{X}$, as the uniform convergence maintains continuity (Tao 2009, Corollary 14.3.2). Besides, $g(\boldsymbol{x})$ is continuous on $\mathcal{X}$ by Assumption 1. For any $\boldsymbol{x} \in \mathcal{X}$, since $\overline{\mathcal{X}}$ is a dense subset, we can find a sequence $\boldsymbol{x}_i \in \overline{\mathcal{X}}$, $i = 1, 2, \ldots$, such that $\boldsymbol{x}_i \to \boldsymbol{x}$ as $i \to \infty$. Hence, $\mu_\infty(\boldsymbol{x}_i) \to \mu_\infty(\boldsymbol{x})$ and $g(\boldsymbol{x}_i) \to g(\boldsymbol{x})$, as $i \to \infty$. Since $\mu_\infty(\boldsymbol{x}_i) = g(\boldsymbol{x}_i)$, for $i = 1, 2, \ldots$, it can be concluded that $\mu_\infty(\boldsymbol{x}) = g(\boldsymbol{x})$, due to the uniqueness of limit (Tao 2009, Proposition 6.1.7). Therefore, we conclude that $\mathbb{P}\{\mu_\infty(\boldsymbol{x}) = g(\boldsymbol{x}), \text{ for all } \boldsymbol{x} \in \mathcal{X}\} = 1$. The proof is then completed by combining this result with Lemma 5. □

Proposition 1 shows that $\mu_n(\boldsymbol{x})$ defined in Equation (2) converges uniformly to $g(\boldsymbol{x})$ almost surely as the number of observations goes to infinity. It plays a crucial role in establishing the convergence of the GPS-C algorithm. However, it is also worthwhile noting that Proposition 1 is itself an important result, which may be useful in other applications as well.

### 3.3 The Global Convergence of the GPS-C algorithm

Notice that the uniform convergence of $\mu_n(\boldsymbol{x})$ established in proposition 1 requires knowing $\lambda^2$. In the following theorem we show that, by replacing the unknown $\lambda^2$ by a known value $\tilde{\lambda}^2$, the uniform convergence holds for $\tilde{\mu}_n(\boldsymbol{x})$ as well. This result is established based on two observations: (1) $\mu_n(\boldsymbol{x})$ converges to $g(\boldsymbol{x})$ uniformly with any $\tau^2 > 0$ assigned for the Gaussian process, and (2) $\lambda^2$ matters to the value of $\mu_n(\boldsymbol{x})$ only through the ratio $\lambda^2/\tau^2$ according to Equation (2).

**Theorem 1.** *Suppose Assumptions 1–3 hold and the GPS-C algorithm is used to solve Problem 1. Then, $\tilde{\mu}_n(\boldsymbol{x}) \to g(\boldsymbol{x})$ uniformly on $\mathcal{X}$ almost surely as $n \to \infty$.*

*Proof.* Suppose that in the GPS-C algorithm, a specific mean function $\mu_0$, a specific correlation function $\rho$ and a specific unconditional variance $\tau_1^2$ are assigned for the Gaussian process, which satisfy Assumption 3. A specific $\tilde{\lambda}^2 > 0$ is also chosen. Then, by Equation (4),

$$\tilde{\mu}_n(\boldsymbol{x}) = \tilde{\mu}_n(\boldsymbol{x}; \mu_0, \rho, \tau_1^2) = \mu_0(\boldsymbol{x}) + \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\tilde{\lambda}^2/\tau_1^2)\boldsymbol{I}^n]^{-1}[\boldsymbol{G}^n - \mu_0(\boldsymbol{X}^n)],$$

where the design points in $\boldsymbol{X}^n$ are generated from the densities $\{\tilde{f}_i(\cdot) : i = 1, \ldots, \lceil n/r \rceil\}$, and $\boldsymbol{G}^n$ are their associated observations.

In order to see the convergence of $\tilde{\mu}_n(\boldsymbol{x}; \mu_0, \rho, \tau_1^2)$, let us consider another imaginary Gaussian process together with the conditional mean function $\mu_n(\cdot)$. We assign the same $\mu_0$ and $\rho$ for the Gaussian process, but let the unconditional variance be $\tau_2^2 = \tau_1^2\lambda^2/\tilde{\lambda}^2$. Suppose we use the same design points $\boldsymbol{X}^n$ and their associated observations $\boldsymbol{G}^n$, which are generated by the GPS-C algorithm, in this imaginary Gaussian process. Since the design points in $\boldsymbol{X}^n$ are generated from one of $\{\tilde{f}_i(\cdot) : i = 1, \ldots, \lceil n/r \rceil\}$, by Lemma 1, each design point $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$, is generated with density no smaller than $\alpha$, where $\alpha$ is defined in Lemma 1. Therefore, by Proposition 1, $\mu_n(\boldsymbol{x}) \to g(\boldsymbol{x})$ uniformly on $\mathcal{X}$ almost surely as $n \to \infty$.

Furthermore, notice that by Equation (2),

$$\begin{aligned}
\tilde{\mu}_n(\boldsymbol{x}; \mu_0, \rho, \tau_1^2) &= \mu_0(\boldsymbol{x}) + \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\tilde{\lambda}^2/\tau_1^2)\boldsymbol{I}^n]^{-1}[\boldsymbol{G}^n - \mu_0(\boldsymbol{X}^n)] \\
&= \mu_0(\boldsymbol{x}) + \rho(\boldsymbol{x}, \boldsymbol{X}^n)[\rho(\boldsymbol{X}^n, \boldsymbol{X}^n) + (\lambda^2/\tau_2^2)\boldsymbol{I}^n]^{-1}[\boldsymbol{G}^n - \mu_0(\boldsymbol{X}^n)] \\
&= \mu_n(\boldsymbol{x}; \mu_0, \rho, \tau_2^2) = \mu_n(\boldsymbol{x}). \qquad (11)
\end{aligned}$$

Then, $\tilde{\mu}_n(\boldsymbol{x}) \to g(\boldsymbol{x})$ uniformly on $\mathcal{X}$ almost surely as $n \to \infty$. □

Theorem 1 shows that the conditional mean function $\tilde{\mu}_n(\boldsymbol{x})$ used in GPS-C algorithm converges uniformly to $g(\boldsymbol{x})$. With this result, by Theorem 5.3 of Shapiro et al. (2009), it is straightforward to establish the global convergence of the GPS-C algorithm, which is formally stated in the following Theorem 2.

**Theorem 2.** *If Assumptions 1–3 hold and the GPS-C algorithm is used to solve Problem (1), then* $\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*) \to g^*$ *and* $d(\tilde{\boldsymbol{x}}_n^*, \mathcal{X}^*) \to 0$ *almost surely as* $n \to \infty$.

Theorem 2 shows that the GPS-C algorithm is globally convergent when solving COvS problems. It is a desirable property for random search algorithms. As discussed at the beginning of this section, the convergence result in Theorem 2 is different from (and weaker than) the typical ones in the literature, which treat the objective function as a deterministic function. To compare with the ones in the literature, we may interpret our result as follows: the GPS-C algorithm has the global convergence (in the typical sense) for almost all objective functions that are sampled randomly from the Gaussian process that satisfies Assumption 3. In other words, for those objective functions for which the GPS-C algorithm fails to converge, their combined probability under the Gaussian process assumption is zero.

## 4 Rate of Convergence

While almost all random search COvS algorithms in the literature have some sort of convergence guarantees, very few of them have results on the rate of convergence, which provides valuable information on the efficiency and scalability of the algorithm. An important reason for the lack

of rate-of-convergence results is the difficulty in analyzing them. In this section we show that, by leveraging on the properties of Gaussian process regression, we are able to establish the rate of convergence of the GPS-C algorithm, and the rate of convergence does provide valuable insights on the performance of the algorithm.

Let $\{\epsilon_n\}_{n\geq 1}$ be a deterministic sequence such that $\epsilon_n > 0$ and $\epsilon_n \to 0$ as $n \to \infty$. In this section we want to find $\{\epsilon_n\}_{n\geq 1}$ so that we can show that the rate of convergence of GPS-C algorithm is $O_p(\epsilon_n)$, i.e., for every $\delta > 0$ there exist constants $C_\delta \in (0, \infty)$ and $N_\delta \in \mathbb{N}$ such that

$$\mathbb{P}\{|\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*) - g^*| > C_\delta \epsilon_n\} < \delta, \tag{12}$$

for all $n > N_\delta$. Similar to that in Equation (9), the probability in Equation (12) also includes the randomness in the Gaussian process, as well as the randomness in the sampling and the simulation experiments.

To simplify the presentation we also adopt the $\widetilde{O}_p$ notation. For any constant $k$, if the rate of convergence of an algorithm is $O_p(\epsilon_n \log^k \epsilon_n)$, it is denoted as $\widetilde{O}_p(\epsilon_n)$. The $\widetilde{O}_p$ notation is a variant of the $O_p$ notation, which ignores the logarithmic factors in the rate and captures the main effect. Its deterministic version is popular in the area of theoretical computer science.

The rate-of-convergence analysis of the GPS-C algorithm also has three steps. In the first step, we establish the rate of convergence of the conditional variance function $k_n(\boldsymbol{x}, \boldsymbol{x})$ to zero. Based on that, in the second step, we prove the rate of convergence of the maximum value of conditional mean function $\mu_n(\boldsymbol{x}_n^*)$ to the global optimal value $g^*$. In the third step, we substitute the unknown $\lambda^2$ with $\tilde{\lambda}^2$, and obtain the rate of convergence of the GPS-C algorithm. In the following subsections, we elaborate these three steps.

### 4.1 The Rate of Convergence of the Conditional Variance

In this subsection, our goal is to establish the rate of convergence of the conditional variance $k_n(\boldsymbol{x}, \boldsymbol{x})$ to zero as $n \to \infty$ for any $\boldsymbol{x} \in \mathcal{X}$. To fulfill this goal, we need stronger assumptions on the Gaussian process and the feasible region. We first state the following assumption on the Gaussian process.

**Assumption 4.** The mean function $\mu_0(\boldsymbol{x})$ is continuously differentiable on $\mathcal{X}$, and the covariance function $k_0(\boldsymbol{x}, \boldsymbol{x}') = \tau^2 \exp\left\{-\sum_{j=1}^d \theta_j (x_j - x_j')^2\right\}$ with $\theta_j > 0$ for $j = 1, 2, \ldots, d$.

Compared with Assumption 3, Assumption 4 imposes stronger conditions on both $\mu_0(\boldsymbol{x})$ and $k_0(\boldsymbol{x}, \boldsymbol{x}')$. The mean function $\mu_0(\boldsymbol{x})$ is required not only to be continuous but also to be continuously differentiable. The correlation function $\frac{1}{\tau^2} k_0(\boldsymbol{x}, \boldsymbol{x}')$ needs to follow a Gaussian correlation function, which is also widely used in the Gaussian process regression. Given these conditions, it is easy to show that the sample path $g$ is continuously differentiable with probability one and its partial derivatives may be derived. We summarize the results in the following lemma, whose proof is provided in Appendix A.

**Lemma 6.** *Under Assumption 4, the sample paths of the Gaussian process are continuously differentiable with probability one. Moreover, for each $j = 1, \ldots, d$, the partial derivative, $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j = \frac{\partial f_{\mathcal{GP}}(\boldsymbol{x})}{\partial x_j}$ is still a Gaussian process, with mean function $\frac{\partial \mu_0(\boldsymbol{x})}{\partial x_j}$ and covariance function $2\tau^2\theta_j(1 - 2\theta_j\delta_j^2)\rho(\boldsymbol{\delta})$, where $\boldsymbol{\delta} = \boldsymbol{x} - \boldsymbol{x}'$.*

Lemma 6 states that Gaussian processes satisfying Assumption 4 have continuously differentiable sample paths, and its first-order derivative surfaces are still Gaussian processes. On a compact set $\mathcal{X}$, these derivative surfaces have continuous sample paths and are bounded almost surely (Adler and Taylor 2007, Theorem 1.5.4). As we will see later, these properties are critical in the analysis of the rate-of-convergence of the GPS-C algorithm.

Besides the stronger assumption on the Gaussian process, we also need a stronger assumption on the feasible region, which is stated as follows.

**Assumption 5.** The feasible region $\mathcal{X} \subset \mathbb{R}^d$ is a bounded convex set with nonempty interior.

Compared with Assumption 1, Assumption 5 is stronger by requiring the convexity of the feasible region $\mathcal{X}$. With this assumption, we can establish a lower bound for the volume of the intersection of $\mathcal{X}$ and a small ball $S(\boldsymbol{x}, \epsilon)$, which is proved by Baumert and Smith (2002, p. 14) and formally stated in the following Lemma 7.

**Lemma 7** (Baumert and Smith (2002) p. 14)**.** *If Assumption 5 holds, then for any $\boldsymbol{x} \in \mathcal{X}$ and sufficiently small $\epsilon > 0$, there exists some constant $C > 0$ such that*

$$\nu(S(\boldsymbol{x}, \epsilon) \cap \mathcal{X}) \geq C \cdot \nu(S(\boldsymbol{x}, \epsilon)). \tag{13}$$

For the interior points of $\mathcal{X}$, Equation (13) always holds even without Assumption 5 when $\epsilon$ is small enough. However, for $\boldsymbol{x}$ on the boundary of $\mathcal{X}$, it is not necessarily the case. Lemma 7 helps to rule out those situations. It ensures that for any $\boldsymbol{x} \in \mathcal{X}$, the sampling probability of $S(\boldsymbol{x}, \epsilon) \cap \mathcal{X}$ has a lower bound that is proportional to the volume of the ball $S(\boldsymbol{x}, \epsilon)$. This result is a foundation for the convergence analysis of the shrinking ball algorithms. Inspired by the shrinking ball idea, in the following analysis, we also construct balls that shrink with the number of sampled points, through which we can investigate the increasing rate of the sampled design points in local areas.

Recall that $s_n(\boldsymbol{x}, \epsilon)$ denotes the number of points sampled in the closed $d$-dimensional ball $\mathcal{S}(\boldsymbol{x}, \epsilon)$ with a deterministic radius $\epsilon$. We further let $s_n(\boldsymbol{x}, r_n)$ denote the number of points sampled in another closed $d$-dimensional ball $\mathcal{S}(\boldsymbol{x}, r_n)$, centered at $\boldsymbol{x}$ with radius $r_n$. We have the following lemma, whose proof is provided in Appendix B.

**Lemma 8.** *Suppose that Assumption 5 holds, and that density functions $\psi_i$ satisfying $\psi_i \geq \alpha > 0$ on $\mathcal{X}$ are used to generate design points $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$. Let $\varepsilon(n) = (\log\log n)/\log n$, $\gamma_n = \gamma - a\varepsilon(n)$, $p_n = \gamma - b\varepsilon(n)$ and $r_n = r_0 n^{-\frac{1-\gamma_n}{d}}$ with $r_0 > 0$. Then, for any $\gamma \in (0, 1)$ and $b - 1 > a$, $s_n(\boldsymbol{x}, r_n)$ is $\Omega(n^{p_n})$[1] almost surely, i.e., $\mathbb{P}\{s_n(\boldsymbol{x}, r_n)$ is $\Omega(n^{p_n})\} = 1$, for any $\boldsymbol{x} \in \mathcal{X}$.*

[1]Let $\{a_n\}_{n\geq 1}$ be a sequence such that $a_n > 0$ for all $n$. A function $h(n)$ of $n$ is called $\Omega(a_n)$, if there is a $c \in (0, \infty)$ such that for all $n \in \mathbb{N}$, $h(n) \geq ca_n$. A function $h(n)$ is called $O(a_n)$, if there is a $C \in (0, \infty)$ such that for all $n \in \mathbb{N}$, $0 < h(n) \leq Ca_n$. A function $h(n)$ is called $\Theta(a_n)$ if it is both $\Omega(a_n)$ and $O(a_n)$.

The result of Lemma 8 is stronger than that of Lemma 2. In particular, Lemma 2 shows that $s_n(\boldsymbol{x}, \epsilon)$ increases to $\infty$, while Lemma 8 shows the increasing rate of $s_n(\boldsymbol{x}, r_n)$ to $\infty$. Similar results have also been developed by shrinking ball algorithms (see, for instance, Baumert and Smith (2002), Andradóttir and Prudius (2010), Kiatsupaibul et al. (2018)).

Lemma 8 indicates that the increasing rate of $s_n(\boldsymbol{x}, r_n)$ is close to $\Omega(n^\gamma)$, which relies on the contracting rate of the radius. Similar to the proof of Lemma 4, by combining the increasing rate of $s_n(\boldsymbol{x}, r_n)$ and the upper bound of $k_n(\boldsymbol{x}, \boldsymbol{x})$ (Lemma 3), we can then establish the rate of convergence of the conditional variance $k_n(\boldsymbol{x}, \boldsymbol{x})$. This result is formally stated in the following Lemma 9, and its proof is provided in Appendix A.

**Lemma 9.** *Suppose that Assumptions 2, 4 and 5 hold, and that density functions $\psi_i$ satisfying $\psi_i \geq \alpha > 0$ on $\mathcal{X}$ are used to generate design points $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$. Then, for any $\boldsymbol{x} \in \mathcal{X}$, $k_n(\boldsymbol{x}, \boldsymbol{x})$ is $O(n^{-\kappa(n)})$ almost surely, i.e., $\mathbb{P}\{k_n(\boldsymbol{x}, \boldsymbol{x})$ is $O(n^{-\kappa(n)})\} = 1$, where*

$$\kappa(n) = \frac{2}{d+2} - b\varepsilon(n) \text{ and } \varepsilon(n) = \frac{\log \log n}{\log n}, \tag{14}$$

*for any $b > 2/(d+2)$.*

Lemma 9 is the main result of this subsection. It states that for any $\boldsymbol{x} \in \mathcal{X}$, the rate of convergence of its conditional variance $k_n(\boldsymbol{x}, \boldsymbol{x})$ is $\tilde{O}(n^{-2/(d+2)})$, which depends on the dimension $d$ of the problem. By Lemma 3, to make $k_n(\boldsymbol{x}, \boldsymbol{x})$ converge to zero, we need to let radii of the shrinking balls converge to zero while keeping the number of sampled points in these shrinking balls still going to infinity. Since the trends of the contracting rate of the radius and the increasing rate of the number of sampled points within shrinking balls are opposite, to obtain a good rate of convergence of $k_n(\boldsymbol{x}, \boldsymbol{x})$, it requires an adequate balance of these two rates. Under Assumption 4, by letting the radius $r_n$ contract at the rate $\tilde{O}(n^{-1/(d+2)})$, the proved rate of convergence of $k_n(\boldsymbol{x}, \boldsymbol{x})$ can be obtained, which is the optimal rate under the bound of Lemma 3. Notice that the dimension $d$ is included because the expected number of sampled points in the shrinking balls is proportional to the volumes of the shrinking balls, which are proportional to $r_n^d$.

## 4.2 The Rate of Convergence of the Maximum Value of the Conditional Mean Function

In this subsection, our goal is to establish the rate of convergence of the maximum value of the conditional mean function $\mu_n(\boldsymbol{x}_n^*)$ to the global optimal value $g^*$, assuming the unknown $\lambda^2$ is known. We will deal with the issue of unknown $\lambda^2$ in next subsection. To establish the rate, two preliminary results are needed. The first is stated in the following Lemma 10, which establishes an upper bound for the probability that the estimation error of $\mu_n(\boldsymbol{x})$ is beyond a certain threshold. This result is obtained by applying the Chernoff bound on the conditional mean function $\mu_n(\boldsymbol{x})$, and its proof is provided in Appendix A.

**Lemma 10.** *For any $n = 1, 2, \ldots$ and any $\boldsymbol{x} \in \mathcal{X}$, we have*

$$\mathbb{P}\{|\mu_n(\boldsymbol{x}) - g(\boldsymbol{x})| > \epsilon_n\} \leq 2\,\mathbb{E}\left[e^{-\epsilon_n^2/(2k_n(\boldsymbol{x},\boldsymbol{x}))}\right],$$

*for any deterministic sequence $(\epsilon_n)_{n\geq 1}$ such that $\epsilon_n > 0$ and $\epsilon_n \to 0$ as $n \to \infty$.*

Lemma 10 indicates that the probability that the estimation error of $\mu_n(\boldsymbol{x})$ is greater than $\epsilon_n$ is bounded by how fast the conditional variance function $k_n(\boldsymbol{x}, \boldsymbol{x})$ converges to 0. If it converges at a faster rate than $\epsilon_n^2$ for any $\boldsymbol{x} \in \mathcal{X}$, then the probability converges to zero for any feasible point.

The following Lemma 11 provides another preliminary result, which establishes an upper bound for the probability that no point is ever sampled near the global optima. Such a bound is critical to establishing the rate of convergence of the GPS-C algorithm. The proof of Lemma 11 is nontrivial, and it utilizes the properties of the derivative surfaces of the Gaussian process and the Borell-TIS inequality to construct a valid bound.

**Lemma 11.** *Suppose that Assumptions 4 and 5 hold, and that density functions $\psi_i$ satisfying $\psi_i \geq \alpha > 0$ on $\mathcal{X}$ are used to generate design points $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$. Then, for sufficiently large $n$, there exist some positive constants $C_j$, $a_j$ and $\sigma_j^2$, for $j = 1 \ldots, d$, such that*

$$\mathbb{P}\left(\cap_{i=1}^n\{g^* - g(\boldsymbol{x}_i) > \epsilon_n\}\right) \leq e^{-\frac{\alpha C \pi^{d/2} \epsilon_n^d n}{\Gamma(d/2+1)\log n}} + 2\sum_{j=1}^{d} e^{C_j\left(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\right) - \frac{\left(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\right)^2}{2\sigma_j^2}},$$

*where $C$ is as defined in Equation (13) and $\Gamma(\cdot)$ is the gamma function.*

*Proof.* Because the proof is long, we only provide a sketch here, and provide the detailed proof in Appendix A. The proof contains three major steps.

1. The first step is to bound the probability $\mathbb{P}\left(\cap_{i=1}^n\{g^* - g(\boldsymbol{x}_i) > \epsilon_n\}\right)$ with the probabilities of another two disjoint events.

   Suppose $\boldsymbol{x}^*$ is a solution in $\mathcal{X}^*$, by applying the mean value theorem and the Cauchy-Schwarz inequality, we can bound the gap $g^* - g(\boldsymbol{x}_i)$ with the product of the distance $\|\boldsymbol{x}^* - \boldsymbol{x}_i\|$ and the norm of the gradient $\nabla g(\boldsymbol{\xi})$, where and $\boldsymbol{\xi}$ is a point in $\mathcal{X}$. Since the derivative surfaces $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ is almost surely bounded on $\mathcal{X}$ for all $j = 1, 2, \ldots, d$, let $\dot{g}(\boldsymbol{x})_j$ be the (random) sample path of $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ and $\dot{g}^* = \max_{j=1,\ldots,d}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}}|\dot{g}(\boldsymbol{x})_j|\right\}$, we can further bound $\|\nabla g(\boldsymbol{\xi})\|$ with $\sqrt{d}\dot{g}^*$, and hence $\mathbb{P}\left(\cap_{i=1}^n\{g^* - g(\boldsymbol{x}_i) > \epsilon_n\}\right) \leq \mathbb{P}(\cap_{i=1}^n\{\sqrt{d}\dot{g}^*\|\boldsymbol{x}^* - \boldsymbol{x}_i\| > \epsilon_n\})$. For large enough $n$, by dividing $\sqrt{d}\dot{g}^*$ at $(\log n)^{1/d}$, we can bound the aforementioned probability with the summation of another two probabilities $\mathbb{P}\left(\cap_{i=1}^n\left\{\|\boldsymbol{x}^* - \boldsymbol{x}_i\| > \frac{\epsilon_n}{(\log n)^{1/d}}\right\}\right)$ and $\mathbb{P}\left(\sqrt{d}\dot{g}^* \geq (\log n)^{1/d}\right)$. The first probability corresponds to the probability that no point is ever sampled within a ball centered at one global optimal solution given a small enough radius. The second probability is the tail probability of the supremum of all $d$ derivative surfaces of the Gaussian process.

2. The second step is to bound the first probability by utilizing the fact that the constructed sampling distributions of the GPS-C algorithm are bounded below by a positive constant $\alpha$ (Lemma 1). Specifically, this is achieved by constructing a sequence of Bernoulli random variables $(B_i)_{i\geq 1}$ with parameter $\alpha C \cdot \nu\left(S\left(\boldsymbol{x}, \frac{\epsilon_n}{(\log n)^{1/d}}\right)\right)$, and calculating the probability $\mathbb{P}\{\sum_{i=1}^{n} B_i > 0\}$.

3. The third step is to bound the second probability by utilizing the fact that the derivative surfaces are almost surely bounded and then applying the Borell-TIS inequality.

By combining the bounds in the second and third steps, the proof is completed. □

In the proof of Lemma 11, we devise a novel approach to using the properties of the Gaussian process and its derivative surfaces. It is one of the major technical contributions of this paper, and it may be used in other applications of Gaussian process regression as well.

Now, with Lemmas 9–11, we are ready to establish the rate of convergence of $\mu_n(\boldsymbol{x}_n^*)$ to $g^*$, when the design points are generated properly. It is formally stated in the following Proposition 2.

**Proposition 2.** *Suppose that Assumptions 2, 4 and 5 hold, and that density functions $\psi_i$ satisfying $\psi_i \geq \alpha > 0$ on $\mathcal{X}$ are used to generate design points $\boldsymbol{x}_i \in \boldsymbol{X}^n$, for $i = 1, \ldots, n$. Then, there exists a constant $C_0 > 0$ such that*

$$\mathbb{P}\left\{|\mu_n(\boldsymbol{x}_n^*) - g^*| > \left(\frac{16 C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}\right\} \to 0$$

*as $n \to \infty$, where $\kappa(n)$ is defined in Equation (14).*

*Proof.* For any $n \geq 1$, define

$$\begin{aligned}
A_0 &= \{|\mu_n(\boldsymbol{x}_n^*) - g^*| > \epsilon_n\},\\
A_1 &= \{|\mu_n(\boldsymbol{x}_n^*) - g(\boldsymbol{x}_n^*)| > \epsilon_n/2\},\\
A_2 &= \big\{|\mu_n(\boldsymbol{x}_i) - g(\boldsymbol{x}_i)| > \epsilon_n/2 \text{ for some } \boldsymbol{x}_i \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}\big\},\\
A_3 &= \big\{g^* - g(\boldsymbol{x}_i) > \epsilon_n/2 \text{ for all } \boldsymbol{x}_i \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}\big\}.
\end{aligned}$$

First, it is easy to see that if $A_0$ happens, then $A_1 \cup A_2 \cup A_3$ must happen. Suppose none of $A_1$ $A_2$, $A_3$ happens, then $A_1^{\mathsf{c}} \cap A_2^{\mathsf{c}} \cap A_3^{\mathsf{c}}$ happens. Since both $A_0$ and $A_1^{\mathsf{c}}$ happen, we must have $g^* - \mu_n(\boldsymbol{x}_n^*) > \epsilon_n$, otherwise we conclude that $g^* - g(\boldsymbol{x}_n^*) < -\epsilon_n/2$, which contradicts the definition of $g^*$. Notice that $A_2^{\mathsf{c}}$ implies that $|\mu_n(\boldsymbol{x}_i) - g(\boldsymbol{x}_i)| \leq \epsilon_n/2$ for all $\boldsymbol{x}_i$, $i = 1, \ldots, n$, while $A_3^{\mathsf{c}}$ implies that $g^* - g(\boldsymbol{x}_i) \leq \epsilon_n/2$ for some $\boldsymbol{x}_i$, say $\boldsymbol{x}_1$. They together imply that $|g^* - \mu_n(\boldsymbol{x}_1)| \leq \epsilon_n$. Recall that $g^* - \mu_n(\boldsymbol{x}_n^*) > \epsilon_n$, so we have $\mu_n(\boldsymbol{x}_1) - \mu_n(\boldsymbol{x}_n^*) > 0$, but it contradicts to the definition of $\boldsymbol{x}_n^*$.

Based on the above observations, we then have

$$\begin{aligned}
&\mathbb{P}\{|\mu_n(\boldsymbol{x}_n^*) - g^*| > \epsilon_n\}\\
&\quad \leq \ \mathbb{P}\{A_1 \cup A_2 \cup A_3\} \ \leq \ \mathbb{P}\{A_1\} + \mathbb{P}\{A_2\} + \mathbb{P}\{A_3\}
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{P}\{|\mu_n(\boldsymbol{x}_n^*) - g(\boldsymbol{x}_n^*)| > \epsilon_n/2\} + \mathbb{P}\left(\cup_{i=1}^n \{|\mu_n(\boldsymbol{x}_i) - g(\boldsymbol{x}_i)| > \epsilon_n/2\}\right) \\
&\quad + \mathbb{P}\left(\cap_{i=1}^n \{g^* - g(\boldsymbol{x}_i) > \epsilon_n/2\}\right) \\
&\leq \mathbb{P}\{|\mu_n(\boldsymbol{x}_n^*) - g(\boldsymbol{x}_n^*)| > \epsilon_n/2\} + \sum_{i=1}^n \mathbb{P}\{|\mu_n(\boldsymbol{x}_i) - g(\boldsymbol{x}_i)| > \epsilon_n/2\} \\
&\quad + \mathbb{P}\left(\cap_{i=1}^n \{g^* - g(\boldsymbol{x}_i) > \epsilon_n/2\}\right). \qquad (15)
\end{aligned}
$$

For any $\boldsymbol{x} \in \mathcal{X}$, by Lemma 9, with probability one, $k_n(\boldsymbol{x}, \boldsymbol{x}) \leq C_0 n^{-\kappa(n)}$ for some $C_0 > 0$. Then by Lemma 10,

$$
\mathbb{P}\{|\mu_n(\boldsymbol{x}) - g(\boldsymbol{x})| > \epsilon_n/2\} \leq 2\,\mathbb{E}\left[e^{-\epsilon_n^2/(8k_n(\boldsymbol{x},\boldsymbol{x}))}\right] \leq 2e^{-\frac{1}{8C_0}\epsilon_n^2 n^{\kappa(n)}}. \qquad (16)
$$

Combining Equations (15) and (16) and Lemma 11 yields

$$
\begin{aligned}
&\mathbb{P}\{|\mu_n(\boldsymbol{x}_n^*) - g^*| > \epsilon_n\} \\
&\leq 2(n+1)e^{-\frac{1}{8C_0}\epsilon_n^2 n^{\kappa(n)}} + e^{-\frac{\alpha C \pi^{d/2}(\epsilon_n/2)^d n}{\Gamma(d/2+1)\log n}} + 2\sum_{j=1}^d e^{C_j\left(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\right) - \frac{\left(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\right)^2}{2\sigma_j^2}}. \qquad (17)
\end{aligned}
$$

Now let $\epsilon_n = \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}$. Notice that

$$
2(n+1)e^{-\frac{1}{8C_0}\epsilon_n^2 n^{\kappa(n)}} = \frac{2(n+1)}{n^2} \to 0 \text{ as } n \to \infty,
$$

and

$$
e^{-\frac{\alpha C \pi^{d/2}(\epsilon_n/2)^d n}{\Gamma(d/2+1)\log n}} = e^{-\frac{\alpha C (4C_0\pi)^{d/2}(\log n)^{d/2}}{\Gamma(d/2+1)\log n} \cdot n^{1-d\kappa(n)/2}} \to 0 \text{ as } n \to \infty,
$$

since $1 - d\kappa(n)/2 > 2/(d+2)$ by the definition of $\kappa(n)$ in Equation (14). Moreover, it is easy to see that

$$
e^{C_j\left(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\right) - \frac{\left(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\right)^2}{2\sigma_j^2}} \to 0 \text{ as } n \to \infty.
$$

Therefore, it follows that, as $n \to \infty$,

$$
\mathbb{P}\left\{|\mu_n(\boldsymbol{x}_n^*) - g^*| > \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}\right\} \to 0.
$$

This concludes the proof of the proposition. □

Proposition 2 shows that the optimal value of the conditional mean function $\mu_n(\boldsymbol{x}_n^*)$ converge to the global optimal value $g^*$ at the rate $\tilde{O}_p(n^{-1/(d+2)})$ when the Gaussian process satisfies Assumption 4. This rate is the maximum allowable decreasing rate of $\epsilon_n$ such that both the probability that the prediction error of $\mu_n(\boldsymbol{x})$ is greater than $\epsilon_n/2$ and the probability that no point is ever sampled near one of the global optimal solutions (with radius determined by $\epsilon_n/2$) still tend to zero. Based on the established probability bounds in Lemmas 10 and 11, by letting $\epsilon_n = \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}$,

the optimal rate of convergence can be obtained. Notice that this optimal rate is determined by the probability bound of the prediction error of $\mu_n(\boldsymbol{x})$, which is subject to the established rate of convergence of the conditional variance $k_n(\boldsymbol{x}, \boldsymbol{x})$ in Lemma 9. Hence, the dimension $d$ is also included here. Furthermore, it is worth noting that this result holds under both homoscedastic and heteroscedastic simulation noises as long as their variances are known. Therefore, Proposition 2 may be used as a benchmark for comparing rates of convergence of different COvS algorithms.

### 4.3 The Rate of Convergence of the GPS-C Algorithm

In this subsection, our goal is to establish the rate of convergence of the GPS-C algorithm, based on the results developed in previous two subsections. By replacing the unknown $\lambda^2$ with $\tilde{\lambda}^2$, we need to show that $\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*)$, the actual estimated optimal function value, has the same rate of convergence as $\mu_n(\boldsymbol{x}_n^*)$. The result is formally stated in the following Theorem 3.

**Theorem 3.** *Suppose that Assumptions 2, 4 and 5 hold and the GPS-C algorithm is used to solve Problem 1. Then, there exists a constant $C_0 > 0$ such that*

$$\mathbb{P}\left\{|\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*) - g^*| > \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}\right\} \to 0, \tag{18}$$

*as $n \to \infty$, where $\kappa(n)$ is defined in Equation (14). In other words, the rate of convergence of the GPS-C algorithm is $O_p\left(n^{-1/(d+2)} \log^{(b+1)/2} n\right)$ with $b > 2/(d+2)$, or $\widetilde{O}_p\left(n^{-1/(d+2)}\right)$.*

*Proof.* We follow the same arguments as in the proof of Theorem 1. Suppose that in the GPS-C algorithm, a specific mean function $\mu_0$, a specific correlation function $\rho = \exp\left\{-\sum_{j=1}^d \theta_j (x_j - x_j')^2\right\}$ and a specific unconditional variance $\tau_1^2$ are assigned for the Gaussian process, which satisfies Assumption 4. Also, a specific $\tilde{\lambda}^2 > 0$ is chosen. The densities $\{\tilde{f}_i(\cdot) : i = 1, \ldots, \lceil n/r \rceil\}$ of the GPS-C algorithm are used to generate design points $\boldsymbol{X}^n$, and $\boldsymbol{G}^n$ are their corresponding observations.

Consider another imaginary Gaussian process together with the conditional mean function $\mu_n(\cdot)$. We assign the same $\mu_0$ and $\rho$ for the Gaussian process, but let the unconditional variance be $\tau_2^2 = \tau_1^2 \lambda^2 / \tilde{\lambda}^2$. Let $g_{(\mu_0, \rho, \tau_2^2)}$ be the sample path of this Gaussian process. Notice that, by Proposition 2,

$$\mathbb{P}\left\{\left|\mu_n(\boldsymbol{x}_n^*; \mu_0, \rho, \tau_2^2) - g^*_{(\mu_0, \rho, \tau_2^2)}\right| > \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}\right\} \to 0,$$

as long as the design points used are generated from densities lower bounded by a positive constant on $\mathcal{X}$, conditioned on the sample path of the Gaussian process with parameters ($\mu_0$, $\rho$, $\tau_2^2$). By Lemma 1, all the densities $\tilde{f}_i(\cdot)$ used to generate design points have a positive lower bound $\alpha$ on $\mathcal{X}$. Suppose the design points $\boldsymbol{X}^n$ and their associated observations $\boldsymbol{G}^n$ are exactly used by the imaginary Gaussian process. Recall that, by Equation (11), $\mu_n(\boldsymbol{x}; \mu_0, \rho, \tau_2^2) = \tilde{\mu}_n(\boldsymbol{x}; \mu_0, \rho, \tau_1^2)$. So

one can conclude that, as $n \to \infty$,

$$\mathbb{P}\left\{\left|\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*; \mu_0, \rho, \tau_1^2) - g^*_{(\mu_0,\rho,\tau_2^2)}\right| > \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}\right\} \to 0. \tag{19}$$

Therefore, the rate of convergence of the GPS-C algorithm is $O_p\left((\log n)^{1/2} n^{-\kappa(n)/2}\right)$. By the definition of $\kappa(n)$ in Equation (14), $\kappa(n) = 2/(d+2) - b\varepsilon(n)$ with $b > 2/(d+2)$ and $\varepsilon(n)$ satisfies $n^{\varepsilon(n)} = \log n$. So the rate of convergence is $O_p\left(n^{-1/(d+2)} \log^{(b+1)/2} n\right)$. Notice that $\log^{(b+1)/2} n = \left(-(d+2)\log\left(n^{-1/(d+2)}\right)\right)^{(b+1)/2}$. Therefore, the rate of convergence is also $\widetilde{O}_p\left(n^{-1/(d+2)}\right)$. □

Theorem 3 shows that the use of $\tilde{\lambda}^2$ does not affect the rate of convergence of $\tilde{\mu}_n(\boldsymbol{x})$ when the simulation noise is homoscedastic. It is worth emphasizing again that the rate here is not for a specific objective function as typically in the literature. Notice we may interpret the probability statement in Equation (18) as

$$\mathbb{P}\left\{|\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*) - g^*| > \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2}\right\} = \mathbb{E}\left[\mathbb{P}\left\{|\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*) - g^*| > \left(\frac{16C_0 \log n}{n^{\kappa(n)}}\right)^{1/2} \middle| g\right\}\right],$$

where the expectation is taken with respect to the distribution of the random objective function $g$. Therefore, the rate of convergence in Theorem 3 may be viewed as the average rate of convergence (in the typical sense) of all objective functions that are sampled randomly from the Gaussian process that satisfies Assumption 4.

To understand the rate of convergence of the GPS-C algorithm, we compare it with other rate-of-convergence results in the literature. The first is the EGO algorithm of Jones et al. (1998) which, similar to the GPS-C algorithm, uses Gaussian processes to guide the random search to solve deterministic black-box optimization problems. It was proved by Bull (2011) that the rate of convergence (to the global optimum) of the EGO algorithm is $O_p(n^{-1/d})$. Compare to this result, we have two findings. First, the dimension $d$ has a common and significant impact on the rate of convergence of Gaussian process based random search algorithms. Second, the GPS-C algorithm for COvS problems converges slightly slower than the EGO algorithm. However, we think that such difference may be caused by the existence of simulation noises that makes the search and estimation more difficult.

To understand the impact of simulation noises, we note that the Kiefer-Wolfowitz stochastic approximation (KWSA) algorithm also works for COvS problems with simulation noises. For *convex* COvS problems, its rate of convergence is $O_p(n^{-1/3})$, which is independent of the dimension $d$, and it is known to be the best rate of convergence for such problems (Kleinman et al. 1999). When $d = 1$, the rate of convergence of the GPS-C algorithm is $\tilde{O}_p(n^{-1/3})$, which is nearly tight for convex COvS problems. From the comparisons with the EGO and KWSA algorithms, we suspect that the rate of convergence of the GPS-C algorithm is quite tight.

Next we consider simple random search algorithms, such as the ones of Yakowitz et al. (2000) and Chia and Glynn (2013). Even though their practical performances are typically not competitive,

they often reveal important insights on the asymptotic properties, such as convergence and rate of convergence. The random search algorithm with low-dispersion point sets of Yakowitz et al. (2000) solves the COvS problems using a multi-observation approach. Its rate of convergence is $O_p((n/\log n)^{-q/(d+2q)})$ or $\widetilde{O}_p(n^{-q/(d+2q)})$, where the definition of the rate of convergence is similar to ours in this paper and $q$ is a parameter that measures the local Lipschitz condition of the objective function $g$ around the global optimal solution $\boldsymbol{x}^*$. Specifically, $q$ satisfies $\sup_{\boldsymbol{x}\in\mathcal{S}(\boldsymbol{x}^*,t)}(g(\boldsymbol{x}^*) - g(\boldsymbol{x})) \leq Kt^q$ for $t \leq t_0$, for some positive constants $t_0$ and $K$. When $q = 1$, the local Lipschitz condition implies that the first-order derivative of $g$ around $\boldsymbol{x}^*$ is bounded and the global optimal solution can be a boundary point. Notice that the GPS-C algorithm also allows the global optimal solutions to be boundary points. In this situation (i.e, $q = 1$), the rates of convergence of both algorithms are $\widetilde{O}_p\left(n^{-1/(d+2)}\right)$.

When the global optimal solutions are in the interior of the feasible set, this corresponds to the situation of $q = 2$ of Yakowitz et al. (2000) and the situation assumed by Chia and Glynn (2013). Both of them proved that the rate of convergence in this situation is $\widetilde{O}_p(n^{-2/(d+4)})$, which is slightly better than our results. This implies that the rate of convergence of the GPS-C algorithm may be faster if the optimal solutions are in the interior. However, due to the probability inequalities that are used to deal with the Gaussian processes and the complexity of the adaptive sampling, we have not yet found an approach to establishing this. We leave this as a topic for future research. As demonstrated in the numerical experiments in Section 5, the empirical rate of convergence of the GPS-C algorithm appears significantly better than $\widetilde{O}_p\left(n^{-1/(d+2)}\right)$.

### 4.4 A Revised GPS-C Algorithm

The last thing to mention in this section is a slightly revised GPS-C algorithm, which can improve the computational efficiency without impacting the rate of convergence of the original GPS-C algorithm. This is achieved by replacing the original Step 2 with the following Step 2′.

**Step 2′ (Calculation).** Set $n = rs$. Let $\boldsymbol{X}^n = \boldsymbol{X}^{r(s-1)} \cup \{\boldsymbol{x}_{r(s-1)+1}, \ldots, \boldsymbol{x}_{rs}\}$ and $\boldsymbol{G}^n = \left([\boldsymbol{G}^{r(s-1)}]^\intercal, G(\boldsymbol{x}_{r(s-1)+1}), \ldots, G(\boldsymbol{x}_{rs})\right)^\intercal$. Calculate $\tilde{\mu}_n(\boldsymbol{x})$ and $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x}')$ according to Equations (4) and (5). Let $\tilde{\boldsymbol{x}}_n^\dagger = \operatorname{argmax}_{\boldsymbol{x}\in\boldsymbol{X}^n} \tilde{\mu}_n(\boldsymbol{x})$ and break the tie arbitrarily if it exists. Then, construct sampling distribution $\tilde{f}_n(\boldsymbol{x})$ according to Equation (8) by replacing $\tilde{c}$ with $\tilde{c}^\dagger = \max_{\boldsymbol{x}\in\boldsymbol{X}^n} \tilde{\mu}_n^{\text{cap}}(\boldsymbol{x})$.

The main difference of this revised GPS-C algorithm to the original one is to replace the function best solution $\tilde{\boldsymbol{x}}_n^*$ with the sample best solution $\tilde{\boldsymbol{x}}_n^\dagger$. In this way, the revised GPS-C algorithm can avoid calculating $\tilde{\boldsymbol{x}}_n^* = \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}} \tilde{\mu}_n(\boldsymbol{x})$ repeatedly (an issue that was discussed at the end of Section 2), which can significantly reduce the computation overhead of the algorithm, especially when the dimension $d$ is large. Although the finite-sample performance of the revised GPS-C algorithm may differ, due to the different sampling distributions and the different way to output the current solution (especially in the early iterations), it can be shown that the rate of convergence is not affected.

The rate of convergence of the revised algorithm can be proved by following the same steps as in this section. Here, we omit the detailed analysis and only provide a sketch of the proof. First it is easy to see that for the revised GPS-C algorithm, Lemma 1 still holds. Let $\boldsymbol{x}_n^\dagger = \text{argmax}_{\boldsymbol{x}\in\boldsymbol{X}^n} \mu_n(\boldsymbol{x})$. With the same $A_2$ and $A_3$ as defined in the proof of Proposition 2, one can have $\mathbb{P}\{|\mu_n(\boldsymbol{x}_n^\dagger) - g^*| > \epsilon_n\} \leq \mathbb{P}\{A_2\} + \mathbb{P}\{A_3\}$. Following the same arguments as in the proof of Proposition 2, we can show that the convergence rate of $\mu_n(\boldsymbol{x}_n^\dagger)$ is the same as $\mu_n(\boldsymbol{x}_n^*)$. By similar arguments as in Theorem 3, the convergence rate of $\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^\dagger)$ is the same as $\mu_n(\boldsymbol{x}_n^\dagger)$. Combining these facts together will complete the proof.

Notice that the revised algorithm only works under Assumption 4. When using the sample best solution $\tilde{\boldsymbol{x}}_n^\dagger$, to prove the almost sure global convergence of the revised algorithm, we need to establish the probability bound of the event that no point is ever sampled near the global optimal solutions, and then use the Borell-Cantelli lemma to prove such event does not happen infinitely. According to the proof of Lemma 11, this requires the boundedness of the derivative surfaces of the Gaussian process. However, for general Gaussian processes that satisfy Assumption 3 but not Assumption 4, such property may not hold and the revised algorithm may not be used.

# 5 Numerical Experiments

In this section we conduct numerical experiments to understand the empirical performances of the GPS-C algorithm, including the convergence, the rate of convergence, the impacts of dimensionality, and the impact of heteroscedastic simulation noises. Throughout this section, we adopt two performance measures. One is the estimated objective function value at the estimated optimal solution, i.e., $\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*)$, and the other is the true objective function value at the estimated optimal solution, i.e., $g(\tilde{\boldsymbol{x}}_n^*)$.

## 5.1 Global Convergence

In this subsection our goal is to understand the global convergence of the GPS-C algorithm. We use the following test problem:

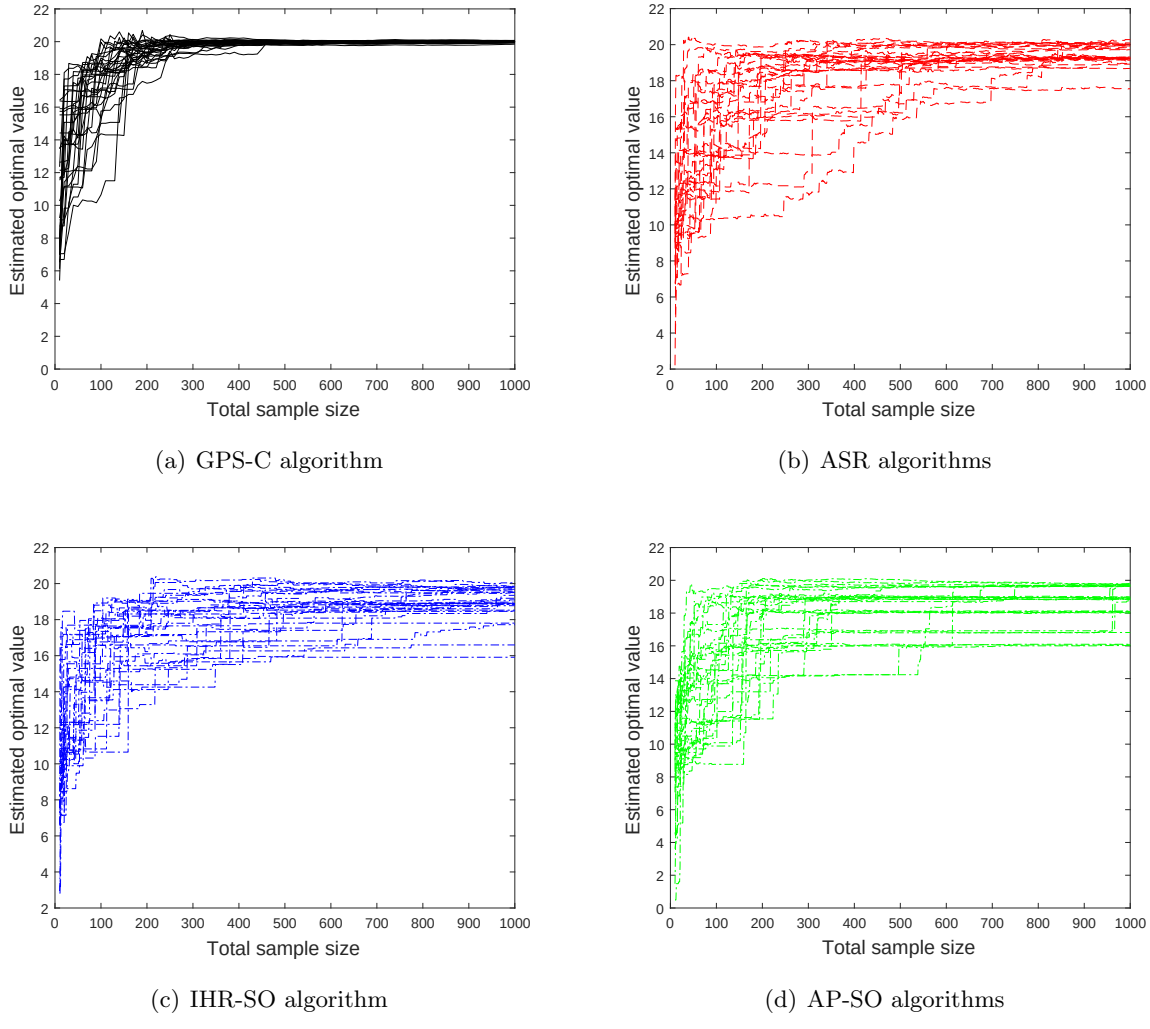
$$\max_{0\leq x_1,x_2\leq 100} g(x_1,x_2) = 10\cdot\frac{\sin^6(0.05\pi x_1)}{2^{2((x_1-90)/50)^2}} + 10\cdot\frac{\sin^6(0.05\pi x_2)}{2^{2((x_2-90)/50)^2}}. \tag{20}$$

This problem was used by Sun et al. (2014) to study the performance of their GPS algorithm for DOvS problems. We extend it to the continuous context. As shown by Sun et al. (2014), this problem has 25 local optimal solutions with the global optimal solution at $(90, 90)$ with $g(90, 90) = 20$ and two second best local optimal solutions at $(90, 70)$ and $(70, 90)$ with $g(90, 70) = g(70, 90) = 18.95$.

We add normally distributed noises with mean 0 and variance 1 (i.e., $\lambda^2 = 1$) to all feasible points. For the GPS-C algorithm, we let $\mu_0(\cdot) = 4$, $\tau^2 = 50$, and the Gaussian correlation function is used with $\theta = (300, 300)^\intercal$. Besides, we set $\tilde{\lambda}^2 = 2$, $r = 10$, $\underline{\tau}^2 = 1$, $\underline{M} = 0$ and $\overline{M} = 40$. The

MCCS scheme is used to sample design points in each iteration with $T = 100$. To find $\tilde{\boldsymbol{x}}_n^*$ in each iteration, we construct a uniform grid in the two dimensional space with step size 0.1, and compare $\tilde{\mu}_n(\boldsymbol{x})$ at all grid points.

For comparison, we also implement the ASR algorithm proposed by Andradóttir and Prudius (2010), and the IHR-SO and the AP-SO algorithms proposed by Kiatsupaibul et al. (2018). The IHR-SO algorithm and the ASR algorithm adopt pure random search samplers; while the AP-SO algorithm adopts a fixed scheme to balance the exploration and exploitation (see Kiatsupaibul et al. (2018) for more details). We use basically the same parameter settings that are used by Kiatsupaibul et al. (2018) and Andradóttir and Prudius (2010). In the IHR-SO and the AP-SO algorithms, we set $\gamma = 0.91$, $\beta = 0.009$, $s = 0.9$, $\kappa_r = 1$, and $R = 1$, where the initial radius $\kappa_r$ and the tuning parameter $R$ is adjusted for Problem (20). In the ASR algorithm, we set $b = 1.1$, $C = 1$, $g = 0.5$, $\delta = 1$, $K = 10$, and $T = 1$.




(a) GPS-C algorithm

(b) ASR algorithms

(c) IHR-SO algorithm

(d) AP-SO algorithms

**Figure 1:** Performance of the GPS-C and other compared algorithms

Each algorithm is run 30 times with a total number of 1000 sampled points. Figure 1 shows

all 30 sample paths of the four algorithms in terms of the $\tilde{\mu}_n(\tilde{\boldsymbol{x}}_n^*)$ as a function of the sample size $n$. Figure 1(a) shows the performance of the GPS-C algorithm. The 30 sample paths indicate that the GPS-C algorithm converges quickly. With less than 400 sample points, the GPS-C algorithm approaches the global optimal function value. Figures 1(b)–1(d) show the performance of the other three algorithms using the same performance measure. It can be observed that for the ASR, the IHR-SO and the AP-SO algorithms, only a proportion of the sample paths can approach the global optimal function value with 1000 simulation observations.

To further illustrate how the GPS-C algorithm works, Figure 2 shows the sampled points and the probability surfaces (i.e., $\mathbb{P}\{Z(\boldsymbol{x}) > \tilde{c}\}$) by different iterations in a typical realization of the algorithm, similar to Sun et al. (2014) (the conditional mean surface (i.e., $\tilde{\mu}_n(\boldsymbol{x})$) and the conditional variance surface (i.e., $\tilde{k}_n(\boldsymbol{x}, \boldsymbol{x})$) are deferred to Figure B1 in Appendix B). The upper panels of Figure 2 show how the algorithm samples design points through different iterations. Many points are sampled in good regions (around the best and second-best solutions), while the algorithm keeps exploring the whole feasible region. This is a good balance between exploration and exploitation, and is guided by the constructed sampling distributions in each iteration (i.e., the probability surfaces in Figure 2). Notice that we impose a lower bound for the sampling probability, which is not shown by the probability surfaces in Figure 2.

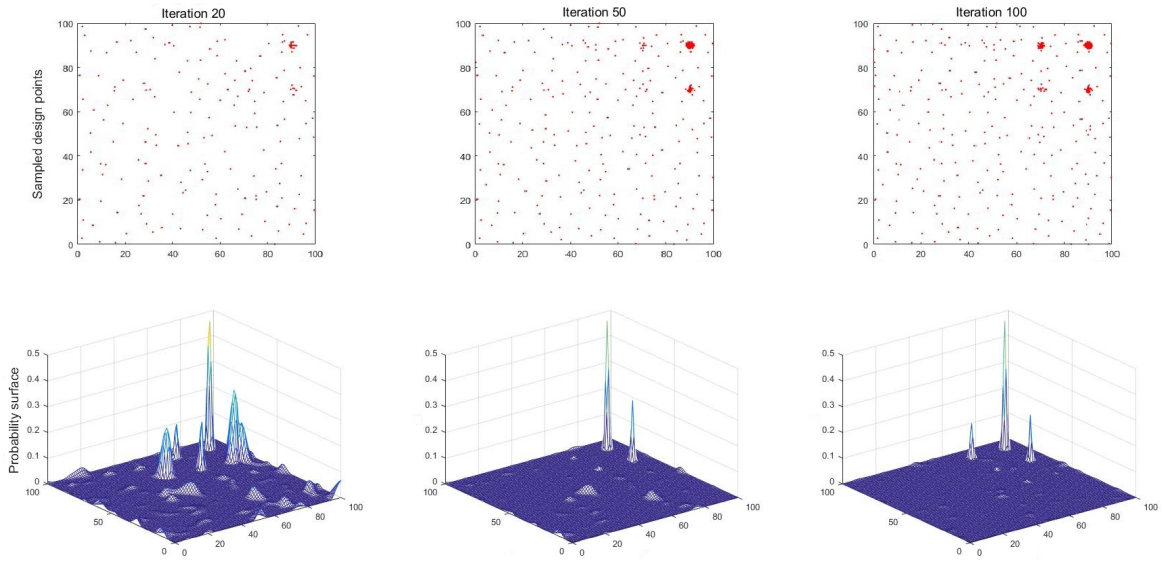



**Figure 2:** The sampled points and probability surface by 20, 50, 100 iterations

Several conclusions can be made from the experiment results in this subsection. First, the global convergence of the GPS-C algorithm is verified by this example. Second, the numerical experiments show that the GPS-C algorithm maintains the advantages of the GPS algorithm in balancing exploration and exploitation, and its finite sample performance appears better than the other three algorithms, which use more rigid sampling schemes.

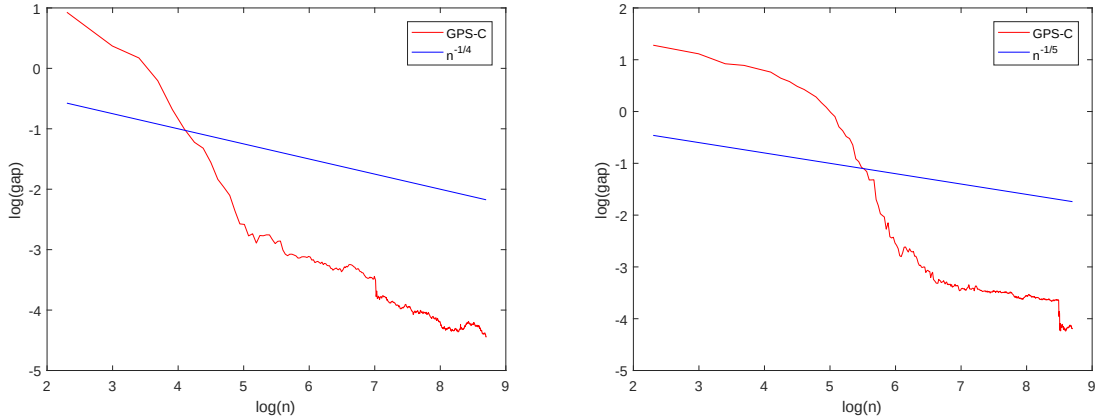
## 5.2 Rate of Convergence

In this subsection our goal is to understand the empirical rate of convergence of the GPS-C algorithm. Since the rate of convergence of the GPS-C algorithm may be viewed as an expectation over all sample paths of the Gaussian process, to display its empirical rate of convergence, we do not use just a specific objective function, but use a number of continuous functions sampled from a Gaussian process on the space $[0,1]^d$.

On a two dimensional space $[0,1]^2$, a Gaussian process satisfying Assumption 4 with $\mu_0 = 1$, $\tau^2 = 4$ and $\theta = (80, 80)$ is used to generate 30 sample paths to represent 30 objective functions, and a random variable following $N(0, 0.25)$ is added at any point as the simulation noise.

To generate these sample paths efficiently, we take observations of this Gaussian process on a uniform grid containing 400 points, and add a normally distributed simulation noise with $\sigma^2 = 0.1$ when fitting its corresponding sample path using the stochastic kriging method. Then, we identify the maximum value of the sample path and run the GPS-C algorithm to search for this maximum value and its corresponding optimal solution. Figure 3(a) shows the average optimality gap $\frac{1}{30}\sum_{i=1}^{30} |\tilde{\mu}_{n,i}(\tilde{\boldsymbol{x}}^*_{n,i}) - g^*_i|$ with respect to the number of observations $n$ on a log-log plot, where $g^*_i$ is the true optimal value of the $i$th objection function. It can be observed that the optimality gap appears to shrink in a rate faster than $n^{-1/4}$, which is its theoretical rate of convergence.

We also replicate this experiment on a three dimensional space $[0,1]^3$ with the parameters of the Gaussian process as $\mu_0 = 1$, $\tau^2 = 9$ and $\theta = (40, 40, 40)$. Figure 3(b) shows a similar result. Due to the numerical difficulties in generating the sample paths and finding the true optimal values, we cannot try problems with higher dimensions and we leave it to next subsection.




(a) Gaussian process on two dimensional space

(b) Gaussian process on three dimensional space

**Figure 3:** Empirically observed rates of convergence of the GPS-C algorithm

### 5.3 The Impact of the Dimensionality

In this subsection, our goal is to understand the performance of the GPS-C algorithm for higher dimensional problems, as well as to understand the impact of dimensionality on the rate of convergence. Because it is difficult to generate high-dimensional test problems using Gaussian processes, we instead use the famous Rosenbrock problem, which is non-convex and is one of the most widely used test problems for COvS algorithms. A $d$-dimensional Reosenbrock problem is as follows:

$$\max_{-10\leq x_1,\ldots,x_d\leq 10} g(x_1,\ldots,x_d) := -10^{-6}\times\sum_{i=1}^{d-1}((1-x_i)^2+100(x_{i+1}-x_i^2)^2), \tag{21}$$

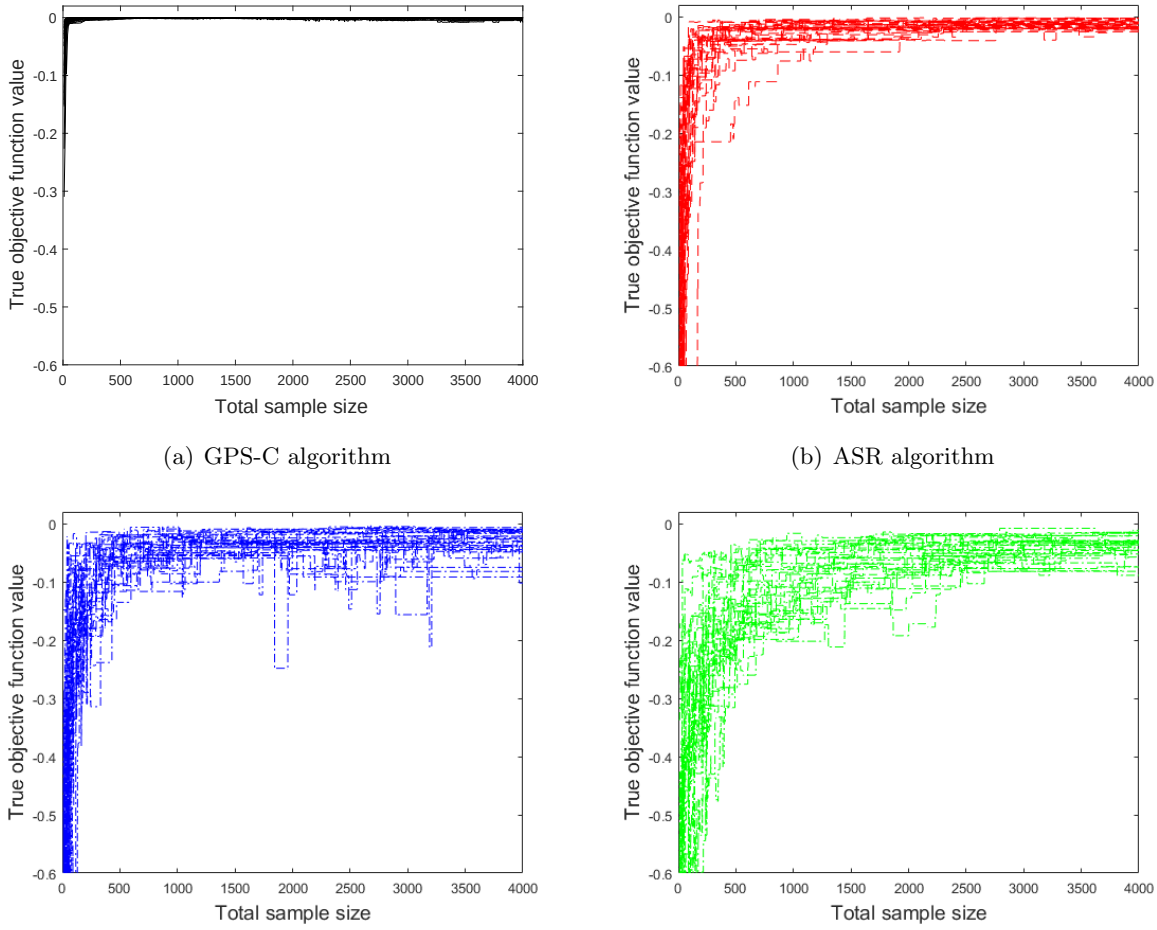
and we add normally distributed noises with mean 0 and variance 0.01 for all feasible points. Kiatsupaibul et al. (2018) used the 10-dimensional Rosenbrock problem to test the performance of their shrinking ball algorithms (i.e., IHR-SO, AP-SO). In this subsection we first use the same 10-dimensional problem to test the performance of the GPS-C algorithm and compare with shrinking ball algorithms, and we then use the Rosenbrock problems with different dimensions to investigate its impact on the rate of convergence.

The 10-dimensional Rosenbrock problem has a global optimum at $(1,1,\ldots,1)^\intercal$ with $g^*=0$. For the GPS-C algorithm, we let $\mu_0(\cdot)=-12$, $\tau^2=10$, and the Gaussian correlation function is used with $\theta=(0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.02)^\intercal$. Besides, we set $\tilde{\lambda}^2=0.1$, $r=10$, $\underline{\tau}^2=0.0001$, $\underline{M}=-20$ and $\overline{M}=3$. The MCCS scheme is used to sample design points in each iteration with $T=100$. To find $\tilde{\boldsymbol{x}}_n^*$ in each iteration, we use the MATLAB solver `fmincon` using the sample best solution of each iteration as its starting point, because the evaluation of a dense grid in 10-dimensional space is too computationally expensive. In addition, we also use the revised algorithm in Section 4.4, and report the sample best solution as the current best solution.

The GPS-C algorithm is run 30 times with a total of 4000 sampled points (i.e., 400 iterations) in each time. Figure 4(a) shows the performance of the GPS-C algorithm in terms of $g(\tilde{\boldsymbol{x}}_n^*)$ as a function of the sample size $n$. It can be observed that the GPS-C algorithm identifies good solutions in early iterations and approaches the global optimal very quickly for all 30 sample paths. This excellent performance can be attributed partly to the characteristics of the GPS-C algorithm, which adaptively balance the exploration and exploitation, and partly to the characteristics of the Rosenbrock function, which is quite flat in the neighborhood of the global optimal solution.

Similarly, we also run the IHR-SO, the AP-SO and the ASR algorithms for 30 times. The parameter settings of the IHR-SO and the AP-SO algorithms are the same as in Kiatsupaibul et al. (2018). For the ASR algorithm, we set $b=1.1$, $C=1$, $g=0.5$, $\delta=0.01$, $K=10$, and $T=0.1$. Figures 4(b)–4(d) show the performances of the ASR, the IHR-SO and the AP-SO algorithms, respectively. From these figures, it is clear that the GPS-C algorithm significantly outperforms the other three algorithms in this example, demonstrating the advantage of adaptive sampling for high-dimensional problems.

Figure 5(b) shows the performance of the revised GPS-C algorithm. Compared with the original




(a) GPS-C algorithm

(b) ASR algorithm

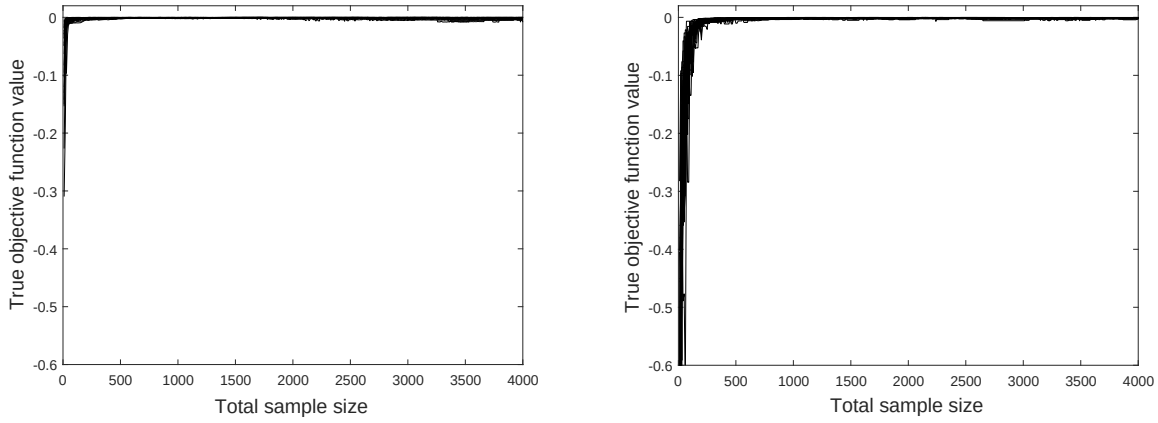
(c) IHR-SO algorithm

(d) AP-SO algorithm

**Figure 4:** Performance of the GPS-C and other compared algorithm for the 10-dimensional Rosenbrock problem

GPS-C algorithm in Figure 5(a), the revised GPS-C algorithm is more volatile in the early iterations. However, the overall performance of the revised algorithm is good, and it is more easy and more efficient to implement as it avoids the calculation of function best solution in high-dimensional problems.

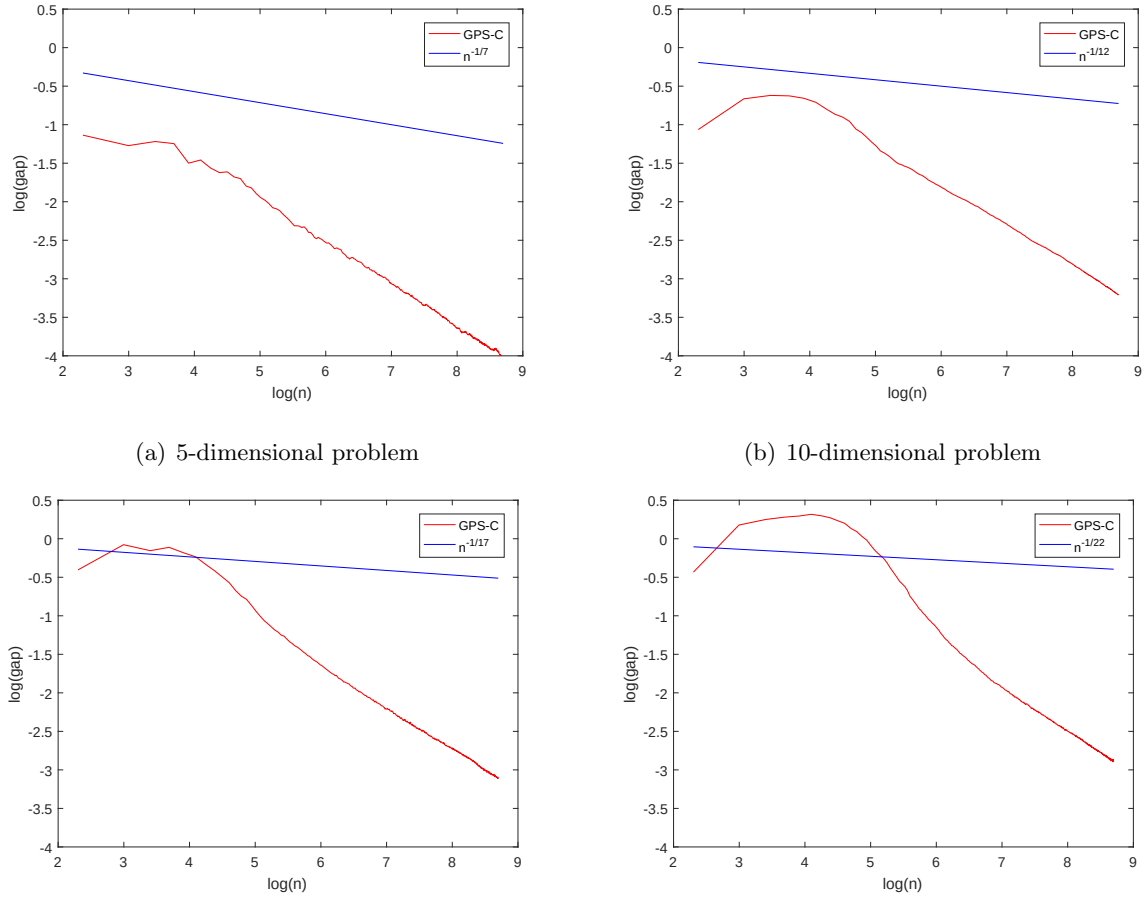
To understand the impact of the dimensionality on the rate of convergence of the GPS-C algorithm, we use the Rosenbrock problems of 5, 10, 15 and 20 dimensions and plot the average optimality gaps with respect to the number of observations $n$ in Figure 6, as we did in Section 5.2. From these plots, we can see that the dimension of the problem has little impact on the rate of convergence of the GPS-C algorithm and the empirically observed rates are much better than the theoretical rates. We suspect that is because the Rosenbrock function is much smoother than the typical sample paths from Gaussian processes satisfying either Assumption 3 or Assumption 4 (see, for instance, the generated sample paths in Figure B2 in Appendix B). As many practical problems in the field of operations research and management science are in general quite smooth, we suspect




(a) GPS-C algorithm (b) Reivised GPS-C algorithm

**Figure 5:** Performance of the GPS-C and the revised GPS-C algorithm for the 10-dimensional Rosenbrock problem

that the performances of the GPS-C algorithm on the Rosenbrock functions are more common and more representative.




(a) 5-dimensional problem (b) 10-dimensional problem

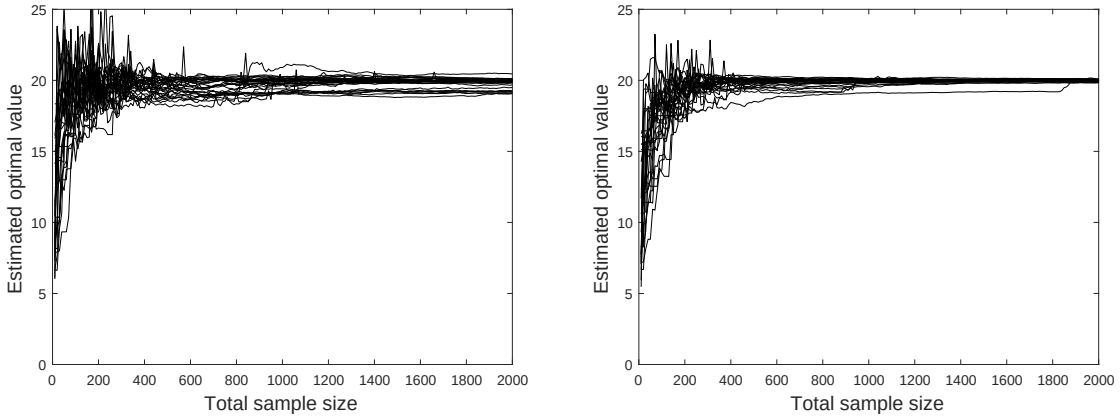
(c) 15-dimensional problem (d) 20-dimensional problem

**Figure 6:** Empirically observed rates of convergence of the GPS-C algorithm for 5-, 10-. 15- and 20-dimensional Rosenbrock problems

### 5.4 Heteroscedastic Simulation Noises

The GPS-C algorithm only has provable convergence and rate of convergence under homoscedastic simulation noises, while many practical COvS problems may have heteroscedastic noises. However, it is easy to see that the GPS-C algorithm can still be applied to solve these problems just by treating the noises to be homoscedastic. We conjecture that this approach works, because the Gaussian process regression estimates the function using the weighted average of all observations and the weights decrease quickly as the distance between the estimated point and the sampled point increases. This works very much like a kernel regression method, which works for heteroscedastic noises. We use the two-dimensional problem in Section 5.1 and the ten-dimensional Rosenbrock problem in Section 5.3 to test the conjecture, except that the added simulation noises are now heteroscedastic. All the parameters for the GPS-C algorithm are the same as in Section 5.1 and Section 5.3.

For the two-dimensional problem, we add two types of zero-mean normal noises, one with the variance $g(\boldsymbol{x})$ and the other with the variance $\frac{1}{4}g(\boldsymbol{x})$ for all $\boldsymbol{x}$ in the feasible region. Notice that the first type of variance is uniformly larger than the second type. These choices are to show the possible effects of different variance values when analyzing the influence of heteroscedasticity on the algorithmic performance. Due to the greatly increased variances of simulation noise in good regions, we run the GPS-C algorithm with a total number of 2000 sampled points. We plot the 30 realizations of the GPS-C algorithm for these two problems in Figure 7, as we did in Figure 1(a). Compare to Figure 1(a) where the variance is set to be 1 for all feasible solutions, we see more volatile realizations in these two cases. In the case with variance $\frac{1}{4}g(\boldsymbol{x})$, i.e., Figure 7(b), the GPS-C algorithm can still find the global optimal solution in all 30 realizations, while in the case with variance $g(\boldsymbol{x})$, i.e., Figure 7(a), the GPS-C algorithm finds the global optimal solution in 24 out of 30 realizations, and only finds the second best solutions in the rest 6 realizations. This is mainly because the increased variances. As shown in Figure B3 in Appendix B, by setting larger $\tilde{\lambda}^2$, the performance of the GPS-C can further be improved in the case with variance $g(\boldsymbol{x})$.



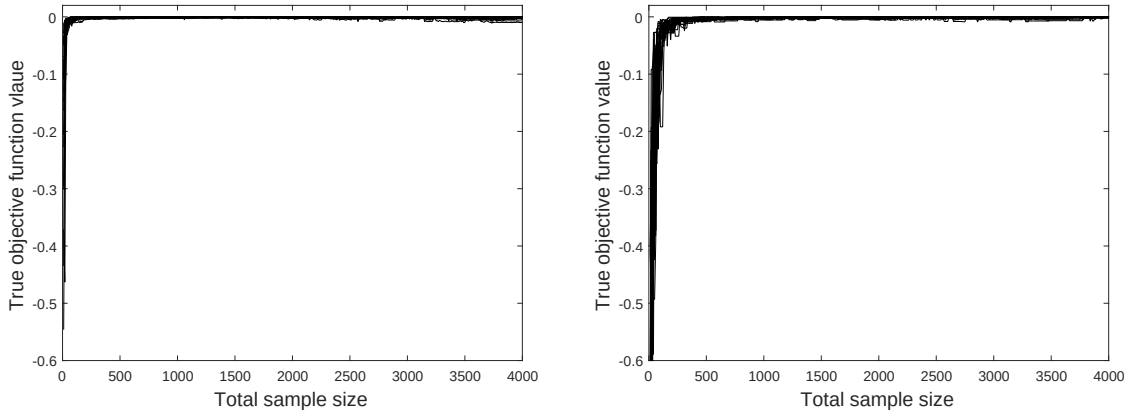

(a) The case with variance $g(\boldsymbol{x})$

(b) The case with variance $\frac{1}{4}g(\boldsymbol{x})$

**Figure 7:** Performance of the GPS-C algorithm under heteroscedastic noises for the two-dimensional problems

For the ten-dimensional Rosenbrock problem, we add zero-mean normal noise with the variance $0.01(1+|g(\boldsymbol{x})|)^2$ for all $\boldsymbol{x}$ in the feasible region. Figure 8 shows the performance of both the GPS-C algorithm and the revised GPS-C algorithm. Compare with Figure 5 where the variance is set to be 0.01 for all feasible solutions, we see similar performances.

From these two examples, we see that the GPS-C algorithm appears to perform well even for COvS problems with heteroscedastic simulation noises.




(a) GPS-C algorithm

(b) Revised GPS-C algorithm

**Figure 8:** Performance of the GPS-C and the revised GPS-C algorithm under heteroscedastic noises for the 10-dimensional Rosenbrock problem

# 6 Conclusions

In this paper we propose a Gaussian process based random search algorithm for COvS problems, i.e., the GPS-C algorithm. This algorithm takes a single observation at each design point, and is capable of balancing the exploration and exploitation adaptively by sampling from distributions constructed based on the Gaussian process. Under homoscedastic but unknown simulation noises, we prove the global convergence of the GPS-C algorithm. Moreover, when the objective functions are sampled from a Gaussian process with Gaussian correlation functions, we establish the rate of convergence of the algorithm, which is $\tilde{O}_p(n^{-1/(d+2)})$. Numerical experiments show that the GPS-C algorithm performs well, even for problems with heteroscedastic simulation noises.

There are several directions to potentially extend this work. First, the global convergence of the GPS-C algorithm under unknown and heteroscedastic simulation noises may be studied. This is an important theoretical extension, though it may be quite challenging. Second, it is interesting to examine whether faster rate of convergence may be established for the GPS-C algorithm by using either sharper inequalities or additional assumptions. Lastly, it may be interesting to further test the performance of the GPS-C algorithm in different scenarios and develop an open-source COvS solver based on the algorithm.

# References


Adler RJ, Taylor JE (2007) *Random Fields and Geometry* (Springer Science & Business Media).

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.

Andradóttir S (2006) An overview of simulation optimization via random search. *Handbooks in operations research and management science* 13:617–631.

Andradóttir S (2015) A review of random search methods. *Handbook of Simulation Optimization* 277–292.

Andradóttir S, Prudius AA (2009) Balanced explorative and exploitative search with estimation for simulation optimization. *INFORMS Journal on Computing* 21(2):193–208.

Andradóttir S, Prudius AA (2010) Adaptive random search for continuous simulation optimization. *Naval Research Logistics (NRL)* 57(6):583–604.

Ankenman B, Nelson BL, Staum J (2010) Stochastic kriging for simulation metamodeling. *Operations Research* 58(2):371–382.

Azaïs JM, Wschebor M (2009) *Level Sets and Extrema of Random Processes and Fields* (John Wiley & Sons).

Baumert S, Ghate A, Kiatsupaibul S, Shen Y, Smith RL, Zabinsky ZB (2009) Discrete hit-and-run for sampling points from arbitrary distributions over subsets of integer hyperrectangles. *Operations Research* 57(3):727–739.

Baumert S, Smith RL (2002) Pure random search for noisy objective functions. *Technical Report* 01–03.

Bect J, Bachoc F, Ginsbourger D, et al. (2019) A supermartingale approach to gaussian process based sequential design of experiments. *Bernoulli* 25(4A):2883–2919.

Box GE, Wilson KB (1951) On the experimental attainment of optimum conditions. *Journal of the royal statistical society: Series b (Methodological)* 13(1):1–38.

Boyd S, Boyd SP, Vandenberghe L (2004) *Convex optimization* (Cambridge university press).

Bull AD (2011) Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research* 12(10):2879–2904.

Chang KH, Hong LJ, Wan H (2013) Stochastic trust-region response-surface method (strong)—a new response-surface framework for simulation optimization. *INFORMS Journal on Computing* 25(2):230–243.

Chia YL, Glynn PW (2013) Limit theorems for simulation-based optimization via random search. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 23(3):1–18.

Devroye L (1978) The uniform convergence of nearest neighbor regression function estimators and their application in optimization. *IEEE Transactions on Information Theory* 24(2):142–151.

Durrett R (2010) *Probability: Theory and Examples* (Cambridge University Press), 4th edition.

Ensor KB, Glynn PW (1997) Stochastic optimization via grid search. *Lectures in Applied Mathematics-American Mathematical Society* 33:89–100.

Fan Q, Hu J (2018) Surrogate-based promising area search for lipschitz continuous simulation optimization. *INFORMS Journal on Computing* 30(4):677–693.

Fitzpatrick P (2009) *Advanced calculus*, volume 5 (American Mathematical Soc.).

Forrester AI, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: mathematical, physical and engineering sciences* 463(2088):3251–3269.

Frazier PI (2018) Bayesian optimization. *Recent Advances in Optimization and Modeling of Contemporary Problems*, 255–278 (INFORMS).

Gut A (2013) *Probability: A graduate course*, volume 75 (Springer Science & Business Media).

Harold J, Kushner G, George Y (1997) Stochastic approximation algorithms and applications.

Hu J, Fu MC, Marcus SI (2007) A model reference adaptive search method for global optimization. *Operations Research* 55(3):549–568.

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13(4):455–492.

Kiatsupaibul S, Smith RL, Zabinsky ZB (2018) Single observation adaptive search for continuous simulation optimization. *Operations Research* 66(6):1713–1727.

Kiefer J, Wolfowitz J, et al. (1952) Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* 23(3):462–466.

Kleijnen JP (1998) Experimental design for sensitivity analysis, optimization, and validation of simulation models. *Handbook of simulation* 173:223.

Kleinman NL, Spall JC, Naiman DQ (1999) Simulation-based optimization with stochastic approximation using common random numbers. *Management Science* 45(11):1570–1578.

Osorio C, Bierlaire M (2013) A simulation-based optimization framework for urban transportation problems. *Operations Research* 61(6):1333–1345.

Potthoff J (2010) Sample properties of random fields iii: Differentiability. *Communications on Stochastic Analysis* 4(3):335–353.

Rasmussen CE, Williams CK (2006) *Gaussian Processes for Machine Learning* (MIT Press).

Robbins H, Monro S (1951) A stochastic approximation method. *The annals of mathematical statistics* 400–407.

Santner TJ, Williams BJ, Notz WI (2003) *The Design and Analysis of Computer Experiments* (New York: Springer).

Shapiro A, Dentcheva D, Ruszczyński A (2009) *Lectures on stochastic programming: modeling and theory* (SIAM).

Shen H, Hong LJ, Zhang X (2018) Enhancing stochastic kriging for queueing simulation with stylized models. *IISE Transactions* 50(11):943–958.

Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control* 37(3):332–341.

Stroock DW (2010) *Probability Theory: An Analytic View* (Cambridge University Press).

Sun L, Hong LJ, Hu Z (2014) Balancing exploitation and exploration in discrete optimization via simulation through a gaussian process-based search. *Operations Research* 62(6):1416–1438.

Sun W, Hu Z, Hong LJ (2018) Gaussian mixture model-based random search for continuous optimization via simulation. *2018 Winter Simulation Conference (WSC)*, 2003–2014 (IEEE).

Tao T (2009) *Analysis*, volume 185 (Springer).

Yakowitz S, L'ecuyer P, Vázquez-Abad F (2000) Global stochastic optimization with low-dispersion point sets. *Operations Research* 48(6):939–950.

Yu T, Zhu H (2020) Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689* .

Zhang X, Shen H, Hong LJ, Ding L (2019) Knowledge gradient for selection with covariates: Consistency and computation. *arXiv preprint arXiv:1906.05098* .

# Appendix A

Appendix A contains the proofs of Lemmas 4, 6 and 9–11, which are critical results in this paper.

### Proof of Lemma 4

First notice that $k_n(\boldsymbol{x},\boldsymbol{x}) = \mathrm{Var}\left(g(\boldsymbol{x})|\{\boldsymbol{X}^n,\boldsymbol{G}^n\}\right) \geq 0$. From Equation (3), it is easy to see that

$$k_{n+1}(\boldsymbol{x},\boldsymbol{x}) = k_n(\boldsymbol{x},\boldsymbol{x}) - \frac{[k_n(\boldsymbol{x},\boldsymbol{x}^{n+1})]^2}{k_n(\boldsymbol{x}^{n+1},\boldsymbol{x}^{n+1})+\lambda^2} \leq k_n(\boldsymbol{x},\boldsymbol{x}), \tag{22}$$

which implies that $k_n(\boldsymbol{x},\boldsymbol{x})$ decreases in $n$. Also note from Equation (3) that reordering the sampling decision-observation pairs $(\boldsymbol{x}_1,G(\boldsymbol{x}_1)),\ldots,(\boldsymbol{x}_n,G(\boldsymbol{x}_n))$ does not alter $k_n(\boldsymbol{x},\boldsymbol{x})$. Fix an $\boldsymbol{x}\in\mathcal{X}$. Then, for any $\epsilon>0$, $\boldsymbol{x}\in\mathcal{X}\cap\mathcal{S}(\boldsymbol{x},\epsilon)\subset\mathcal{X}$, and

$$k_n(\boldsymbol{x},\boldsymbol{x}) \leq k_{s_n(\boldsymbol{x},\epsilon)}(\boldsymbol{x},\boldsymbol{x}) \leq \tau^2 - \frac{s_n(\boldsymbol{x},\epsilon)\min_{\boldsymbol{x}'\in\mathcal{X}\cap\mathcal{S}(\boldsymbol{x},\epsilon)}[k_0(\boldsymbol{x},\boldsymbol{x}')]^2}{s_n(\boldsymbol{x},\epsilon)\tau^2+\lambda^2}, \tag{23}$$

where the second inequality follows from Lemma 3. According to Assumption 3, $[k_0(\boldsymbol{x},\boldsymbol{x}')]^2 = \tau^4\rho^2(|\boldsymbol{x}-\boldsymbol{x}'|)$. Since $\|\boldsymbol{x}-\boldsymbol{x}'\|\leq\epsilon$, Assumption 3 also implies that $\rho(|\boldsymbol{x}-\boldsymbol{x}'|)\geq\rho(\epsilon\mathbf{1})$, where $\mathbf{1}\in\mathbb{R}^d$ is the vector of all ones. Following Equation (23),

$$k_n(\boldsymbol{x},\boldsymbol{x}) \leq \tau^2 - \frac{s_n(\boldsymbol{x},\epsilon)\tau^4\rho^2(\epsilon\mathbf{1})}{s_n(\boldsymbol{x},\epsilon)\tau^2+\lambda^2}.$$

By Lemma 2, $s_n(\boldsymbol{x},\epsilon)\xrightarrow{\text{a.s.}}\infty$ as $n\to\infty$, so, $\limsup_{n\to\infty}k_n(\boldsymbol{x},\boldsymbol{x})\leq\tau^2[1-\rho^2(\epsilon\mathbf{1})]$, with probability one. Sending $\epsilon\to0$, we have $\rho(\epsilon\mathbf{1})\to1$, thus $\limsup_{n\to\infty}k_n(\boldsymbol{x},\boldsymbol{x})\leq0$, with probability one. Recall that $k_n(\boldsymbol{x},\boldsymbol{x})\geq0$, then, with probability one, $\limsup_{n\to\infty}k_n(\boldsymbol{x},\boldsymbol{x}) = \liminf_{n\to\infty}k_n(\boldsymbol{x},\boldsymbol{x}) = 0$, which implies $k_n(\boldsymbol{x},\boldsymbol{x})\xrightarrow{\text{a.s.}}0$ as $n\to\infty$. □

### Proof of Lemma 6

Notice that under Assumption 4, for $\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}$, $k_0(\boldsymbol{x},\boldsymbol{x}')=\tau^2\rho(\boldsymbol{\delta})=\tau^2\exp\left\{-\sum_{j=1}^d\theta_j\delta_j^2\right\}$, where $\boldsymbol{\delta}=\boldsymbol{x}-\boldsymbol{x}'$. Fix $j=1,\ldots,d$. Then $\ddot{\rho}_j(\boldsymbol{\delta})=\frac{\partial^2\rho(\boldsymbol{\delta})}{\partial\delta_j^2}=2\theta_j(2\theta_j\delta_j^2-1)\rho(\boldsymbol{\delta})$ exists and is continuous with $\ddot{\rho}_j(\mathbf{0})=-2\theta_j$. Moreover, it can be checked that $\ddot{\rho}_j(\boldsymbol{\delta})/\ddot{\rho}_j(\mathbf{0})=(1-2\theta_j\delta_j^2)\rho(\boldsymbol{\delta})$ as a function of $\boldsymbol{\delta}$ satisfies Assumption 3 (i) and (iii). So it can be concluded that, with probability one, the Gaussian process $f_{\mathcal{GP}}$ has $j$-th partial differentiable sample path (Santner et al. 2003, pp. 39–40), and this $j$-th partial derivative is continuous on $\mathcal{X}$ (Adler and Taylor 2007, Theorem 1.4.1). Hence, with probability one, the sample paths of Gaussian process $f_{\mathcal{GP}}$ are continuously differentiable. One can also reach the conclusion by directly invoking Potthoff (2010, Corollary 2.11 and Theorem 3.2).

On the other hand, for each $j=1,\ldots,d$, $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ is still a Gaussian process, as differentiation is a linear operator (Azaïs and Wschebor 2009, p. 29), with mean function $\dot{\mu}_0(\boldsymbol{x})_j$ and covariance function $-\tau^2\ddot{\rho}_j(\boldsymbol{\delta})=2\tau^2\theta_j(1-2\theta_j\delta_j^2)\rho(\boldsymbol{\delta})$, where $\boldsymbol{\delta}=\boldsymbol{x}-\boldsymbol{x}'$ (Santner et al. 2003, pp. 39–40).

Notice that the correlation function of $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ is $-\tau^2\ddot{\rho}_j(\boldsymbol{\delta})/(-\tau^2\ddot{\rho}_j(\mathbf{0})) = (1-2\theta_j\delta_j^2)\rho(\boldsymbol{\delta})$ which satisfies Assumption 3 (i) and (iii). Therefore, for $j = 1, \cdots, d$, $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ has continuous sample paths with probability one. □

## Proof of Lemma 9

Let $\gamma_n$, $p_n$ and $r_n$ be as defined in Lemma 8 with $\gamma \in (0,1)$ and $b-1 > a$. Notice that $k_n(\boldsymbol{x},\boldsymbol{x})$ decreases in $n$ (see Equation (22) in the proof of Lemma 4), and does not depend on the ordering of the sampling decision-observation pairs $(\boldsymbol{x}_1, G(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_n, G(\boldsymbol{x}_n))$ (see Equation (3)). Fix an $\boldsymbol{x} \in \mathcal{X}$. Notice that $\boldsymbol{x} \in \mathcal{X} \cap \mathcal{S}(\boldsymbol{x}, r_n) \subset \mathcal{X}$. Then,

$$k_n(\boldsymbol{x},\boldsymbol{x}) \le k_{s_n(\boldsymbol{x},r_n)}(\boldsymbol{x},\boldsymbol{x}) \le \tau^2 - \frac{s_n(\boldsymbol{x},r_n)\min_{\boldsymbol{x}'\in\mathcal{X}\cap\mathcal{S}(\boldsymbol{x},r_n)}[k_0(\boldsymbol{x},\boldsymbol{x}')]^2}{s_n(\boldsymbol{x},r_n)\tau^2+\lambda^2},$$

where the second inequality follows from Lemma 3. According to Assumption 4, $k_0(\boldsymbol{x},\boldsymbol{x}') = \tau^2\exp\left\{-\sum_{j=1}^d \theta_j|x_j - x_j'|^2\right\}$. Let $\theta^* = \max\{\theta_1,\ldots,\theta_d\}$, then $\sum_{i=j}^d \theta_j|x_j-x_j'|^2 \le \theta^*\|\boldsymbol{x}-\boldsymbol{x}'\|^2$. So, for $\boldsymbol{x}'$ in $\mathcal{S}(\boldsymbol{x},r_n)$, $k_0(\boldsymbol{x},\boldsymbol{x}') \ge \tau^2\exp\left\{-\theta^*\|\boldsymbol{x}-\boldsymbol{x}'\|^2\right\} \ge \tau^2\exp\{-\theta^* r_n^2\}$. By Lemma 8, with probability one, $s_n(\boldsymbol{x},r_n)$ is $\Omega(n^{p_n})$ with $\gamma\in(0,1)$ and $b-a-1>0$, i.e., $s_n(\boldsymbol{x},r_n) \ge cn^{p_n}$ for some $c>0$. Then,

$$\begin{aligned} k_n(\boldsymbol{x},\boldsymbol{x}) &\le \tau^2 - \frac{s_n(\boldsymbol{x},r_n)\tau^4}{s_n(\boldsymbol{x},r_n)\tau^2+\lambda^2}\times\exp\{-2\theta^* r_n^2\} \\ &\le \tau^2 - \frac{cn^{p_n}\tau^4}{cn^{p_n}\tau^2+\lambda^2}\times\exp\{-2\theta^* r_n^2\} \end{aligned}$$

where the second inequality holds with probability one, and is due to the fact that $x\tau^2/(x\tau^2+\lambda^2)$ increases in $x>0$. Notice that $e^x \ge 1+x$. Hence,

$$\begin{aligned} k_n(\boldsymbol{x},\boldsymbol{x}) &\le \tau^2 - \frac{cn^{p_n}\tau^4}{cn^{p_n}\tau^2+\lambda^2}\times\left(1-2\theta^* r_n^2\right) \\ &= \frac{cn^{p_n}\tau^2}{cn^{p_n}\tau^2+\lambda^2}\times\left(\frac{\lambda^2}{c}n^{-p_n} + 2\tau^2\theta^* r_0^2 n^{-\frac{2(1-\gamma_n)}{d}}\right) \\ &\le \text{constant}\times\left(\frac{\lambda^2}{c}n^{-p_n} + 2\tau^2\theta^* r_0^2 n^{-\frac{2(1-\gamma_n)}{d}}\right) \\ &\le \text{constant}\times n^{-\min\{p_n,\frac{2(1-\gamma_n)}{d}\}}, \end{aligned}$$

for sufficiently large $n$. Thus we have that $k_n(\boldsymbol{x},\boldsymbol{x})$ is $O(n^{-\min\{p_n,\frac{2(1-\gamma_n)}{d}\}})$ almost surely.

By letting $p_n = \frac{2(1-\gamma_n)}{d}$, i.e., $\gamma - b\varepsilon(n) = \frac{2(1-\gamma+a\varepsilon(n))}{d}$, we have $\gamma = [2+(2a+db)\varepsilon(n)]/(d+2)$. Take $a = -db/2$. It makes $\gamma = 2/(d+2)$, which satisfies $\gamma\in(0,1)$. Let $b > 2/(d+2)$. Then $b-a = b(d+2)/2 > 1$, which satisfies $b-1>a$. Finally, $\min\{p_n, \frac{2(1-\gamma_n)}{d}\} = p_n = \gamma - b\varepsilon(n) = 2/(d+2) - b\varepsilon(n)$, with $b > 2/(d+2)$. □

## Proof of Lemma 10

For any $n \geq 1$, it can be obtained that

$$\mathbb{P}\{|\mu_n(\boldsymbol{x}) - g(\boldsymbol{x})| > \epsilon_n\} \leq \mathbb{P}\{\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}) > \epsilon_n\} + \mathbb{P}\{\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}) < -\epsilon_n\}. \tag{24}$$

Applying the Chernoff bound, we have

$$\begin{aligned}
\mathbb{P}\{\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}) > \epsilon_n\} &= \mathbb{E}\left[\mathbb{P}\{\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}) > \epsilon_n | \{\boldsymbol{X}^n, \boldsymbol{G}^n\}\}\right] \\
&\leq \mathbb{E}\left[\min_{t>0} e^{-t\epsilon_n} \cdot \mathbb{E}[e^{t(\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}))} | \{\boldsymbol{X}^n, \boldsymbol{G}^n\}]\right] \\
&= \mathbb{E}\left[\min_{t>0} e^{-t\epsilon_n} \cdot e^{\frac{t^2}{2} k_n(\boldsymbol{x},\boldsymbol{x})}\right] \\
&= \mathbb{E}\left[\min_{t>0} e^{\frac{t^2}{2} k_n(\boldsymbol{x},\boldsymbol{x}) - t\epsilon_n}\right],
\end{aligned}$$

where the second inequality is due to Equation (6) and the moment-generating function of normal random variable. Notice that $\frac{t^2}{2} k_n(\boldsymbol{x},\boldsymbol{x}) - t\epsilon_n$ is minimized at $t = \epsilon_n / k_n(\boldsymbol{x},\boldsymbol{x})$ with value $-\epsilon_n^2/(2k_n(\boldsymbol{x},\boldsymbol{x}))$, then

$$\mathbb{P}\{\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}) > \epsilon_n\} \leq \mathbb{E}\left[e^{-\epsilon_n^2/(2k_n(\boldsymbol{x},\boldsymbol{x}))}\right]. \tag{25}$$

In the similar way, we can also get

$$\mathbb{P}\{\mu_n(\boldsymbol{x}) - g(\boldsymbol{x}) < -\epsilon_n\} \leq \mathbb{E}\left[e^{-\epsilon_n^2/(2k_n(\boldsymbol{x},\boldsymbol{x}))}\right]. \tag{26}$$

Combining Equations (24), (25) and (26) finishes the proof. □

## Proof of Lemma 11

Step 1. Arbitrarily choose an optimal solution $\boldsymbol{x}^* \in \mathcal{X}^*$. For $j = 1, 2, \ldots, d$, let $\dot{g}(\boldsymbol{x})_j$ be the sample path of the derivative surface $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$. According to the mean value theorem, it can be obtained that

$$g(\boldsymbol{x}) = g(\boldsymbol{x}^*) + \nabla g(\boldsymbol{\xi})^\intercal (\boldsymbol{x} - \boldsymbol{x}^*), \tag{27}$$

where $\nabla g(\boldsymbol{x}) = (\dot{g}(\boldsymbol{x})_1, \ldots, \dot{g}(\boldsymbol{x})_d)^\intercal$ denotes the gradient of $g(\boldsymbol{x})$ and $\boldsymbol{\xi} \in \mathcal{X}$ (Fitzpatrick 2009, Theorem 15.29). By the Cauchy-Schwarz inequality, it then follows that

$$|g(\boldsymbol{x}^*) - g(\boldsymbol{x})| = \left|\nabla g(\boldsymbol{\xi})^\intercal (\boldsymbol{x} - \boldsymbol{x}^*)\right| \leq \|\nabla g(\boldsymbol{\xi})\| \|\boldsymbol{x}^* - \boldsymbol{x}\|. \tag{28}$$

Recall that $\mathcal{X}$ is compact under Assumption 5. Based on Lemma 6, $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ has continuous sample paths on $\mathcal{X}$, for all $j = 1, \ldots, d$, and is thus almost surely bounded on $\mathcal{X}$ (Adler and Taylor 2007, Theorem 1.5.4). So,

$$\dot{g}^* = \max_{j=1,\ldots,d} \left\{\sup_{\boldsymbol{x} \in \mathcal{X}} |\dot{g}(\boldsymbol{x})_j|\right\},$$

is well defined. Since

$$\|\nabla g(\boldsymbol{\xi})\| = \left[\sum_{j=1}^{d} (\dot{g}(\boldsymbol{\xi})_j)^2\right]^{1/2} \leq \sqrt{d}\dot{g}^*,$$

then

$$|g(\boldsymbol{x}^*) - g(\boldsymbol{x})| \leq \sqrt{d}\dot{g}^* \|\boldsymbol{x}^* - \boldsymbol{x}\|. \tag{29}$$

So, we have

$$\begin{aligned}
&\mathbb{P}\left(\cap_{i=1}^n \{g^* - g(\boldsymbol{x}_i) > \epsilon_n\}\right) \\
=&\mathbb{P}\left(\cap_{i=1}^n \{g(\boldsymbol{x}^*) - g(\boldsymbol{x}_i) > \epsilon_n\}\right) \\
\leq&\mathbb{P}\left(\cap_{i=1}^n \{\sqrt{d}\dot{g}^* \|\boldsymbol{x}^* - \boldsymbol{x}_i\| > \epsilon_n\}\right) \\
\leq&\mathbb{P}\left(\cap_{i=1}^n \left\{\|\boldsymbol{x}^* - \boldsymbol{x}_i\| > \frac{\epsilon_n}{(\log n)^{1/d}}\right\}\right) + \mathbb{P}\left(\sqrt{d}\dot{g}^* \geq (\log n)^{1/d}\right).
\end{aligned} \tag{30}$$

We now establish the bounds for the two terms respectively.

Step 2. For the first term in the right-hand side of Equation (30), let $\delta_n(\epsilon_n) = \epsilon_n/(\log n)^{1/d}$, and construct a ball $\mathcal{S}(\boldsymbol{x}^*, \delta_n(\epsilon_n))$. Then, we can have

$$\begin{aligned}
\mathbb{P}\left(\cap_{i=1}^n \left\{\|\boldsymbol{x}^* - \boldsymbol{x}_i\| > \frac{\epsilon_n}{(\log n)^{1/d}}\right\}\right) &= 1 - \mathbb{P}\left(\cup_{i=1}^n \{\|\boldsymbol{x}^* - \boldsymbol{x}_i\| \leq \epsilon_n/(\log n)^{1/d}\}\right) \\
&= 1 - \mathbb{P}\left(\sum_{i=1}^n \mathbb{1}_{\{\boldsymbol{x}_i \in \mathcal{S}(\boldsymbol{x}^*, \delta_n(\epsilon_n))\}} > 0\right).
\end{aligned} \tag{31}$$

By Lemma 7, for small enough $\delta_n(\epsilon_n) > 0$, $\nu(S(\boldsymbol{x}^*, \delta_n(\epsilon_n))) \cap \mathcal{X}) \geq C\nu(S(\boldsymbol{x}^*, \delta_n(\epsilon_n)))$. Since each design point $\boldsymbol{x}_i \in \boldsymbol{X}^n$ is generated from density function $\psi_i$, which satisfies $\psi_i \geq \alpha > 0$ on $\mathcal{X}$, for $i = 1, \ldots, n$,

$$\begin{aligned}
\mathbb{P}\left\{\boldsymbol{x}_i \in \mathcal{S}(\boldsymbol{x}^*, \delta_n(\epsilon_n))\right\} &\geq \alpha\nu(\mathcal{S}(\boldsymbol{x}^*, \delta_n(\epsilon_n)) \cap \mathcal{X}) \\
&\geq \alpha C\nu(\mathcal{S}(\boldsymbol{x}^*, \delta_n(\epsilon_n))) = \frac{\alpha C \pi^{\frac{d}{2}} \delta_n(\epsilon_n)^d}{\Gamma(\frac{d}{2}+1)},
\end{aligned}$$

where the equality is due to the volume formula of a $d$-dimensional ball. Let $B_i$, $i = 1, \ldots, n$, be i.i.d. Bernoulli random variables with parameter $\frac{\alpha C \pi^{d/2} \delta_n(\epsilon_n)^d}{\Gamma(d/2+1)} \in (0, 1)$. Then,

$$\mathbb{P}\left(\sum_{i=1}^n \mathbb{1}_{\{\boldsymbol{x}_i \in \mathcal{S}(\boldsymbol{x}^*, \delta_n(\epsilon_n))\}} > 0\right) \geq \mathbb{P}\left(\sum_{i=1}^n B_i > 0\right) = 1 - \left(1 - \frac{\alpha C \pi^{\frac{d}{2}} \delta_n(\epsilon_n)^d}{\Gamma(\frac{d}{2}+1)}\right)^n. \tag{32}$$

Combing Equations (31) and (32), we have

$$\mathbb{P}\left(\cap_{i=1}^n \{\|\boldsymbol{x}^* - \boldsymbol{x}\| > \frac{\epsilon_n}{(\log n)^{1/d}}\}\right) \leq \left(1 - \frac{\alpha C \pi^{\frac{d}{2}} \delta_n(\epsilon_n)^d}{\Gamma(\frac{d}{2}+1)}\right)^n \leq \exp\left\{-\frac{\alpha C \pi^{\frac{d}{2}} \delta_n(\epsilon_n)^d n}{\Gamma(\frac{d}{2}+1)}\right\}$$

$$= \exp\left\{-\frac{\alpha C \pi^{\frac{d}{2}} \epsilon_n^d n}{\Gamma(\frac{d}{2}+1)\log n}\right\}, \tag{33}$$

where the second inequality is due to $e^x \geq 1 + x$.

Step 3. For the second term in the right-hand side of Equation (30), we need to bound the tail probability of $\dot{g}^*$. As mentioned before, $\dot{f}_{\mathcal{GP}}(\boldsymbol{x})_j$ is a Gaussian process with continuous and thus bounded sample paths on $\mathcal{X}$ with probability one, for each $j = 1, \ldots, d$. Then by the Borell-TIS inequality (Adler and Taylor 2007, Section 2.1), for $j = 1, \ldots, d$, and sufficiently large $t$,

$$\mathbb{P}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}} (\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j) > t\right\} \leq \exp\left\{C_j t - \frac{t^2}{2\sigma_j^2}\right\},$$

where $C_j$ is a constant depending only on $\mathbb{E}[\sup_{\boldsymbol{x}\in\mathcal{X}}(\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j)]$, and $\sigma_j^2 = \sup_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[(\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j)^2]$. Let $a_j = \sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{\mu}_0(\boldsymbol{x})_j|$. Notice that $\sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j| + a_j \geq \sup_{\boldsymbol{x}\in\mathcal{X}}(|\dot{g}(\boldsymbol{x})_j| - |\dot{\mu}_0(\boldsymbol{x})_j|) + a_j \geq \sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j|$. So, for sufficiently large $t$,

$$\begin{aligned}
\mathbb{P}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j| > t\right\} &\leq \mathbb{P}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j| + a_j > t\right\} = \mathbb{P}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j| > t - a_j\right\} \\
&\leq 2\,\mathbb{P}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}} (\dot{g}(\boldsymbol{x})_j - \dot{\mu}_0(\boldsymbol{x})_j) > t - a_j\right\} \\
&\leq 2\exp\left\{C_j(t-a_j) - \frac{(t-a_j)^2}{2\sigma_j^2}\right\}. \qquad (34)
\end{aligned}$$

Replacing $t$ with $(\log n)^{1/d}/\sqrt{d}$ where $n$ is sufficiently large, we have

$$\begin{aligned}
\mathbb{P}\left(\sqrt{d}\dot{g}^* \geq (\log n)^{\frac{1}{d}}\right) &= \mathbb{P}\left(\max_{j=1,\ldots,d}\left\{\sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j|\right\} \geq (\log n)^{\frac{1}{d}}/\sqrt{d}\right) \\
&\leq \sum_{j=1}^{d} \mathbb{P}\left(\sup_{\boldsymbol{x}\in\mathcal{X}} |\dot{g}(\boldsymbol{x})_j| \geq (\log n)^{\frac{1}{d}}/\sqrt{d}\right) \\
&\leq 2\sum_{j=1}^{d} \exp\left\{C_j\Big(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\Big) - \frac{\Big(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\Big)^2}{2\sigma_j^2}\right\}, \qquad (35)
\end{aligned}$$

where the second inequality is by Equation (34). Finally, combining Equations (30), (33) and (35), we have

$$\begin{aligned}
&\mathbb{P}\left(\cap_{i=1}^n \{g^* - g(\boldsymbol{x}_i) > \epsilon_n\}\right) \\
&\leq \exp\left\{-\frac{\alpha C \pi^{\frac{d}{2}} \epsilon_n^d n}{\Gamma(\frac{d}{2}+1)\log n}\right\} + 2\sum_{j=1}^{d} \exp\left\{C_j\Big(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\Big) - \frac{\Big(\frac{(\log n)^{1/d}}{\sqrt{d}} - a_j\Big)^2}{2\sigma_j^2}\right\},
\end{aligned}$$

for sufficiently large $n$. So the proof is completed. □

# Appendix B

Appendix B contains the proofs of Lemmas 1, 2 and 8 and some figures in the numerical experiments.

### Proof of Lemma 1

Let $Z \sim \mathcal{N}(0,1)$, then

$$\begin{aligned}
\mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{c}\} &= \mathbb{P}\{\tilde{\mu}_n^{\text{cap}}(\boldsymbol{x}) + [\tilde{k}_n^{\text{cap}}(\boldsymbol{x},\boldsymbol{x})]^{1/2} Z > \tilde{c}\} \\
&= \mathbb{P}\left\{ Z > \frac{\tilde{c} - \tilde{\mu}_n^{\text{cap}}(\boldsymbol{x})}{[\tilde{k}_n^{\text{cap}}(\boldsymbol{x},\boldsymbol{x})]^{1/2}} \right\} \\
&\geq \mathbb{P}\left\{ Z > \frac{\overline{M} - \underline{M}}{\underline{\tau}} \right\} \\
&= 1 - \Phi((\overline{M} - \underline{M})/\underline{\tau}),
\end{aligned}$$

where the inequality is due to the facts that $\tilde{c} \leq \overline{M}$, $\tilde{\mu}_n^{\text{cap}}(\boldsymbol{x}) \geq \underline{M}$, and $\tilde{k}_n^{\text{cap}}(\boldsymbol{x},\boldsymbol{x}) \geq \underline{\tau}^2$. Besides, since $\tilde{\mu}_n^{\text{cap}}(\boldsymbol{x}) \leq \tilde{c}$, $\mathbb{P}\{Z(\boldsymbol{x}) > \tilde{c}\} \leq 0.5$. Hence,

$$\tilde{f}_n(\boldsymbol{x}) = \frac{\mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{c}\}}{\int_{\mathcal{X}} \mathbb{P}\{\tilde{Z}(\boldsymbol{x}) > \tilde{c}\} \mathrm{d}\boldsymbol{z}} \geq \frac{1 - \Phi((\overline{M} - \underline{M})/\underline{\tau})}{\int_{\mathcal{X}} 0.5 \ \mathrm{d}\boldsymbol{z}} = \alpha.$$

□

### Proof of Lemma 2

Since each point in $\boldsymbol{X}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is sampled from a density which is lower bounded by the common constant $\alpha > 0$, for any point in $\boldsymbol{X}^n$, say $\boldsymbol{x}_i$,

$$\mathbb{P}\left\{\boldsymbol{x}_i \in \mathcal{S}(\boldsymbol{x}, \epsilon)\right\} \geq \int_{\mathcal{X} \cap \mathcal{S}(\boldsymbol{x},\epsilon)} \alpha \ \mathrm{d}\boldsymbol{z} = \alpha \nu(\mathcal{X} \cap \mathcal{S}(\boldsymbol{x}, \epsilon)) > 0,$$

where $\nu(\cdot)$ denotes the $d$-dimensional volume and is positive due to Assumption 1 and the subsequent remark. Let $\alpha_1 = \alpha\nu(\mathcal{X} \cap \mathcal{S}(\boldsymbol{x}, \epsilon))$, and $B_i$, $i = 1, 2, \ldots$, be independent Bernoulli random variables with parameter $\alpha_1$. Let $\mathbb{1}_B$ denote the indicator function defined on event $B$. Then,

$$s_n(\boldsymbol{x}, \epsilon) = \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_i \in \mathcal{S}(\boldsymbol{x},\epsilon)} \geq \sum_{i=1}^{n} B_i.$$

Since $B_i$ are i.i.d. Bernoulli random variables, by the law of large numbers, $\frac{1}{n}\sum_{i=1}^{n} B_i \xrightarrow{\text{a.s.}} \alpha_1$ as $n \to \infty$. Hence, $\sum_{i=1}^{n} B_i \xrightarrow{\text{a.s.}} \infty$, which then implies $s_n(\boldsymbol{x}, \epsilon) \xrightarrow{\text{a.s.}} \infty$ as $n \to \infty$. □

### Proof of Lemma 8

Since $\gamma_n = \gamma - a\varepsilon(n) \to \gamma$ as $n \to \infty$, it can be obtained that $r_n \to 0$. Notice that $\gamma \in (0, 1)$. Then, there exists some $N \in \mathbb{N}$ such that $\gamma_n \in (0, 1)$ and $r_n/(3\sqrt{d}) \leq \epsilon$ for all $n \geq N$, where $\epsilon$ is a sufficiently small positive constant which satisfies Equation (13) in Lemma 7. Suppose that for

each $n > N$, we partition each coordinate of $\mathbb{R}^d$ into segments of length $r_n/(3\sqrt{d})$, and by doing so we obtain closed subsets, which are referred as grid boxes that together cover $\mathbb{R}^d$. For each $\boldsymbol{x} \in \mathcal{X}$, let $T_{\boldsymbol{x}}$ be the grid box containing $\boldsymbol{x}$. Define $H_{\boldsymbol{x}}$ as the union of $T_{\boldsymbol{x}}$ and all the other grid box adjacent to $T_{\boldsymbol{x}}$ (i.e., with common vertex, edge or surface). Under the Eclidean distance used in this paper, $H_{\boldsymbol{x}}$ covers all points which are at most $r_n/(3\sqrt{d})$ from $T_{\boldsymbol{x}}$, and hence it can be obtained that $\mathcal{S}(\boldsymbol{x}, r_n/(3\sqrt{d})) \subset H_{\boldsymbol{x}}$. Thus, for $n \geq N$,

$$\nu(H_{\boldsymbol{x}} \cap \mathcal{X}) \geq \nu(\mathcal{S}(\boldsymbol{x}, r_n/(3\sqrt{d})) \cap \mathcal{X}) \geq C_1 \cdot \nu(\mathcal{S}(\boldsymbol{x}, r_n/(3\sqrt{d}))) = C_2 \cdot (r_n)^d = \Omega\left(n^{-(1-\gamma_n)}\right), \tag{36}$$

where $C_1$ and $C_2$ are positive constants, the second inequality follows from Lemma 7, and the last equality follows from the choice that $r_n = r_0 n^{-\frac{1-\gamma_n}{d}}$.

Because $\mathcal{X}$ is bounded by Assumption 5 and $r_n$ is of order $\Omega(n^{-\frac{1-\gamma_n}{d}})$, each dimension needs to be partitioned into $O(n^{\frac{1-\gamma_n}{d}})$ segments, and thus the total number of grid boxes $T_{\boldsymbol{x}}$ necessary to cover $\mathcal{X}$ is $O(n^{1-\gamma_n})$. Because $T_{\boldsymbol{x}} \subset H_{\boldsymbol{x}}$ for each $\boldsymbol{x} \in \mathcal{X}$, obviously $\mathcal{X}$ can be covered with a set of $H_{\boldsymbol{x}}$, whose cardinality is $l(n) = O(n^{1-\gamma_n})$. Denote such set as $\mathcal{H}(n) = \{H_k(n)\}_{k=1}^{l(n)}$. By Equation (36), it can be obtained that for each $n \geq N$, $\nu(H_k(n) \cap \mathcal{X})$ is $\Omega(n^{-(1-\gamma_n)})$ for all $k$. Note that for each $n \geq N$ and $\boldsymbol{x} \in \mathcal{X}$, we can find $1 \leq k \leq l(n)$ such that $\boldsymbol{x} \in H_k(n)$. Also note that the maximum distance between any two points in $H_k(n)$ is $r_n$. We can obtain that $H_k(n) \cap \mathcal{X} \subset \mathcal{S}(\boldsymbol{x}, r_n)$ for each $\boldsymbol{x} \in H_k(n)$. To summarize, for sufficiently large $n$, (i.e., $n \geq N$), we obtain the three properties about $\mathcal{H}(n)$:

(i) $l(n)$ is $O(n^{1-\gamma_n})$.

(ii) $\nu(H_k(n) \cap \mathcal{X})$ is $\Omega(n^{-(1-\gamma_n)})$ for all $k$, where $\nu(\cdot)$ is the $d$-dimensional volume.

(iii) For $\boldsymbol{x} \in \mathcal{X}$, if $H_k(n)$ is the box in $\mathcal{H}(n)$ such that $\boldsymbol{x} \in H_k(n)$, then $H_k(n) \cap \mathcal{X} \subset \mathcal{S}(\boldsymbol{x}, r_n) \subseteq \mathcal{S}(\boldsymbol{x}, r_i),\ i = 1, \ldots, n$.

Recall that $p_n = \gamma - b\varepsilon(n)$. Let $s(n)$ be an integer-valued function of $n$ with order $\Theta(n^{p_n})$, and $N_k(n) = \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_i \in H_k(n) \cap \mathcal{X}}$ be the number of sampled points that fall into $H_k(n) \cap \mathcal{X}$ among all $n$ points. By the property (iii), if $\boldsymbol{x} \in H_k(n)$,

$$\{N_k(n) \geq s(n)\} \subseteq \{s_n(\boldsymbol{x}, r_n) \geq s(n)\}.$$

Because $\mathcal{H}(n)$ covers $\mathcal{X}$ and property (iii) holds for all $\boldsymbol{x} \in \mathcal{X}$, we can then have

$$D(n) = \bigcap_{k=1}^{l(n)} \{N_k(n) \geq s(n)\} \subseteq \bigcap_{\boldsymbol{x} \in \mathcal{X}} \{s_n(\boldsymbol{x}, r_n) \geq s(n)\}.$$

Taking complement set on both sides yields

$$D(n)^{\mathsf{c}} = \bigcup_{k=1}^{l(n)} \{N_k(n) < s(n)\} \supseteq \bigcup_{\boldsymbol{x} \in \mathcal{X}} \{s_n(\boldsymbol{x}, r_n) < s(n)\}. \tag{37}$$

Let us first look at the probability $\mathbb{P}\{N_k(n) < s(n)\}$. For a fixed $n$, consider $H_k(n) \in \mathcal{H}(n)$. Since each design point $\boldsymbol{x}_i \in \boldsymbol{X}^n$ is generated from density $\psi_i \geq \alpha > 0$ on $\mathcal{X}$, for $i = 1, \ldots, n$, then

$$\mathbb{P}\left\{\boldsymbol{x}_i \in H_k(n)\right\} \geq \alpha \nu(\mathcal{X} \cap H_k(n)).$$

Notice that $\nu(H_k(n) \cap \mathcal{X}) \geq \frac{c_0}{n^{1-\gamma_n}}$ for some constant $c_0 > 0$, from the property (ii). Let $B_i$, $i = 1, 2, \ldots$, be independent Bernoulli random variables with parameter $\frac{\alpha c_0}{n^{1-\gamma_n}}$. Then,

$$N_k(n) = \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_i \in H_k(n) \cap \mathcal{X}} \geq \sum_{i=1}^{n} B_i,$$

So, by letting $c = \frac{1}{\alpha c_0}$,

$$\begin{aligned}
\mathbb{P}\{N_k(n) < s(n)\} &= \mathbb{P}\left\{\sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}_i \in H_k(n) \cap \mathcal{X}} < s(n)\right\} \\
&\leq \mathbb{P}\left\{\sum_{i=1}^{n} B_i < s(n)\right\} \\
&\leq \sum_{j=0}^{s(n)-1} \binom{n}{j} \left(\frac{1}{cn^{1-\gamma_n}}\right)^j \left(1 - \frac{1}{cn^{1-\gamma_n}}\right)^{n-j}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbb{P}\{D(n)^{\mathsf{c}}\} = \mathbb{P}\left(\bigcup_{k=1}^{l(n)} \{N_k(n) < s(n)\}\right) &\leq \sum_{k=1}^{l(n)} \mathbb{P}\{N_k(n) < s(n)\} \\
&\leq l(n) \sum_{j=0}^{s(n)-1} \binom{n}{j} \left(\frac{1}{cn^{1-\gamma_n}}\right)^j \left(1 - \frac{1}{cn^{1-\gamma_n}}\right)^{n-j} \\
&= l(n) \left(1 - \frac{1}{cn^{1-\gamma_n}}\right)^n \sum_{j=0}^{s(n)-1} \binom{n}{j} \left(cn^{1-\gamma_n} - 1\right)^{-j}.
\end{aligned}$$

By the property (i), $l(n) \leq Cn^{1-\gamma_n}$ for some constant $C > 0$. Recall that $s(n)$ is $\Theta(n^{p_n})$, so $s(n) \leq \bar{a} n^{p_n}$ for some constant $\bar{a} > 0$. We then follow the similar steps as in the proof of Lemma 4 in Andradóttir and Prudius (2010) to show

$$\sum_{n=n_0}^{\infty} \mathbb{P}\{D(n)^{\mathsf{c}}\} < \infty, \tag{38}$$

for sufficiently large $n_0$. Note that $p_n = \gamma - b\varepsilon(n)$, so $p_n \to \gamma \in (0,1)$ as $n \to \infty$. Recall that $\gamma_n \to \gamma \in (0,1)$ as $n \to \infty$, either. Hence, let $n$ be sufficiently large so that $s(n) \le n/2$ and $cn^{1-\gamma_n} > 2$. Then, for $j < s(n) \le n/2$, $\binom{n}{j} \le \binom{n}{s(n)} \le n^{s(n)} \le n^{\bar{a}n^{p_n}}$. So,

$$\mathbb{P}\{D(n)^{\mathsf{c}}\} \le Cn^{1-\gamma_n+\bar{a}n^{p_n}} \left(1 - \frac{1}{cn^{1-\gamma_n}}\right)^n \sum_{j=0}^{s(n)-1} \left(cn^{1-\gamma_n} - 1\right)^{-j}.$$

Since $cn^{1-\gamma_n} > 2$, then

$$\sum_{j=0}^{s(n)-1} \left(cn^{1-\gamma_n} - 1\right)^{-j} \le \sum_{j=0}^{\infty} \left(\frac{1}{cn^{1-\gamma_n} - 1}\right)^j = \frac{cn^{1-\gamma_n} - 1}{cn^{1-\gamma_n} - 2}.$$

Thus,

$$\mathbb{P}\{D(n)^{\mathsf{c}}\} \le Cn^{1-\gamma_n+\bar{a}n^{p_n}} \left(1 - \frac{1}{cn^{1-\gamma_n}}\right)^n \times \frac{cn^{1-\gamma_n} - 1}{cn^{1-\gamma_n} - 2} \le \text{constant} \times n^{1+\bar{a}n^{p_n}} \left(1 - \frac{1}{cn^{1-\gamma_n}}\right)^n,$$

for sufficiently large $n$. Observe that $n^{1-\gamma_n} \to \infty$ as $n \to \infty$ since $\gamma_n \to \gamma \in (0,1)$. Then, $\left(1 + 1/(-cn^{1-\gamma_n})\right)^{-cn^{1-\gamma_n}} \to e$ as $n \to \infty$, which implies that $\left(1 + 1/(-cn^{1-\gamma_n})\right)^{cn^{1-\gamma_n}} \to 1/e < 1$. So, there exists $0 < \beta < 1$ such that for sufficiently large $n$, $\left(1 + 1/(-cn^{1-\gamma_n})\right)^{n^{1-\gamma_n}} \le \beta$, which further implies that $\left(1 - 1/(cn^{1-\gamma_n})\right)^n \le \beta^{n^{\gamma_n}}$. Note that $b - 1 > a$. We can find $\delta > 0$ such that $b - 1 - \delta > a$. Observe that $\log\left(n^{1+\bar{a}n^{p_n}}\right) = \log n + \bar{a}n^{\gamma-(b-1)\varepsilon(n)}$. It can be obtained that $\log\left(n^{1+\bar{a}n^{p_n}}\right)/n^{\gamma-(b-1-\delta)\varepsilon(n)} \to 0$ as $n \to \infty$, which implies that $n^{1+\bar{a}n^{p_n}} \le e^{n^{\gamma-(b-1-\delta)\varepsilon(n)}}$ for sufficiently large $n$. Then,

$$\begin{aligned}\mathbb{P}\{D(n)^{\mathsf{c}}\} \le \text{constant} \times e^{n^{\gamma-(b-1-\delta)\varepsilon(n)}} \times \beta^{n^{\gamma_n}} &= \text{constant} \times \exp\Big(n^{\gamma-(b-1-\delta)\varepsilon(n)} + n^{\gamma-a\varepsilon(n)} \log\beta\Big) \\ &\le \text{constant} \times \exp\Big(-n^{\gamma-(b-1-\delta)\varepsilon(n)}\Big),\end{aligned}$$

for sufficiently large $n$, where the second inequality comes from the facts that $\log\beta < 0$ and $b - 1 - \delta > a$. Notice that $\gamma - (b-1-\delta)\varepsilon(n) \to \gamma \in (0,1)$ as $n \to \infty$. So we can find $t \in (0,1)$ such that $t \le \gamma - (b-1-\delta)\varepsilon(n)$ for sufficiently large $n$. Thus,

$$\mathbb{P}\{D(n)^{\mathsf{c}}\} \le \text{constant} \times \exp\Big(-n^{\gamma-(b-1-\delta)\varepsilon(n)}\Big) \le \text{constant} \times \exp\big(-n^t\big).$$

So, to prove Equation (38), by the integral test for convergence, it suffices to prove $\int_{n_0}^{\infty} e^{-x^t}\,\mathrm{d}x < \int_0^{\infty} e^{-x^t}\,\mathrm{d}x < \infty$, for $t \in (0,1)$. Note that by the change of variable in integral,

$$\int_0^{\infty} e^{-x^t}\,\mathrm{d}x = \int_0^{\infty} \frac{1}{t} y^{\frac{1}{t}-1} e^{-y}\,\mathrm{d}y = \frac{1}{t}\Gamma(\frac{1}{t}) < \infty,$$

where $\Gamma(\cdot)$ is the gamma function. Thus, Equation (38) is proved.

According to Equations (37) and (38), for any $\boldsymbol{x} \in \mathcal{X}$,

$$\sum_{n=n_0}^{\infty} \mathbb{P}\{s_n(\boldsymbol{x}, r_n) < s(n)\} \leq \sum_{n=n_0}^{\infty} \mathbb{P}\left(\bigcup_{\boldsymbol{x}\in\mathcal{X}} \{s_n(\boldsymbol{x}, r_n) < s(n)\}\right) \leq \sum_{n=n_0}^{\infty} \mathbb{P}\{D(n)^{\mathsf{c}}\} < \infty,$$

for sufficiently large $n_0$. Then, by Borel-Cantelli lemma, $\mathbb{P}\{s_n(\boldsymbol{x}, r_n) < s(n) \text{ infinitely often}\} = 0$. Since $s(n)$ is $\Theta(n^{p_n})$, we then immediately have $\mathbb{P}\{s_n(\boldsymbol{x}, r_n) \text{ is } \Omega(n^{p_n})\} = 1$. □

## Figures of numerical results in Section 5

This part contains some figures of numerical results in Section 5, including the figures of conditional mean surfaces, conditional variance surfaces, some examples of generate sample paths and the performance of the GPS-C algorithm under heteroscedastic context.

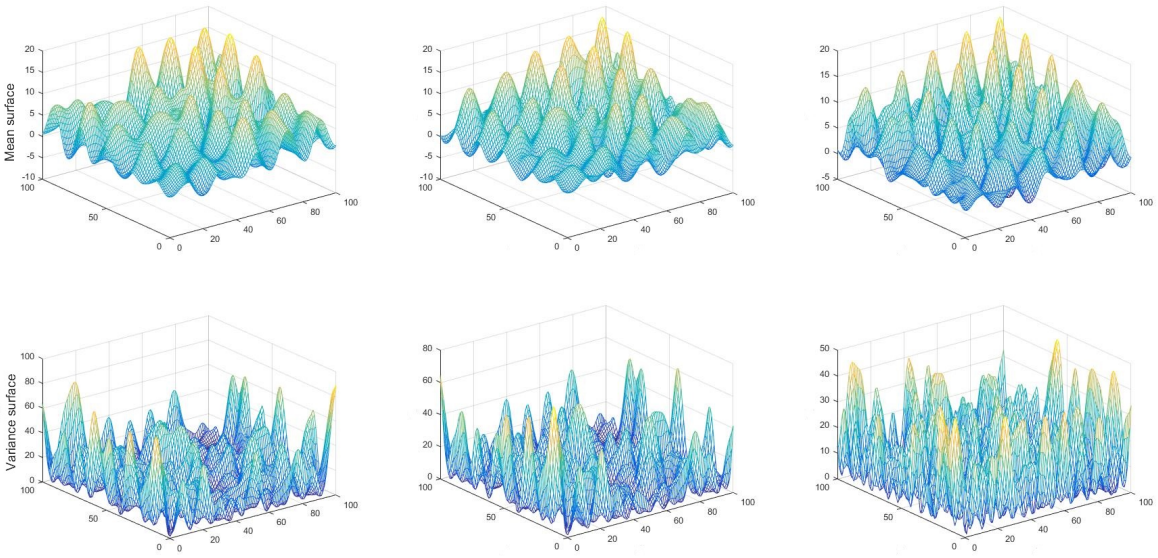
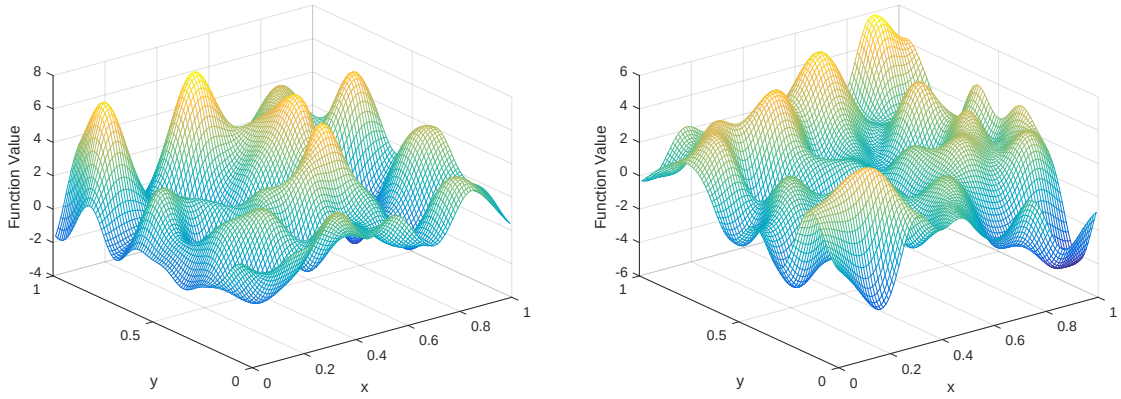



**Figure B1:** The conditional mean surface and conditional variance surface by 20, 50, 100 iterations




(a) Example 1 (b) Example 2

**Figure B2:** Two examples of the generated sample paths with $\mu_0 = 1$, $\sigma^2 = 4$ and $\theta = (80, 80)$

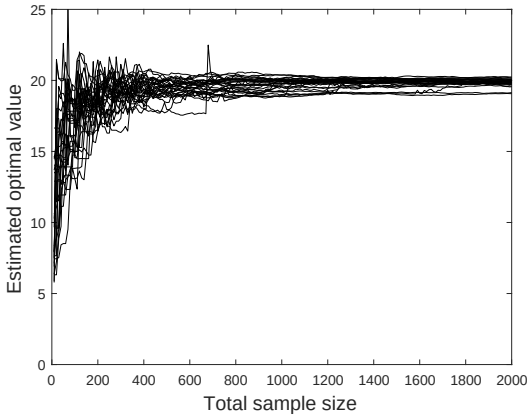



**Figure B3:** The performance of the GPS-C algorithm under the case with variance $g(\boldsymbol{x})$ when $\tilde{\lambda}^2 = 9$