Taylor & Francis
Taylor & Francis Group

Check for updates

Modeling and Active Learning for Experiments with Quantitative-Sequence Factors

Qian Xiao[a] , Yaping Wang[b] , Abhyuday Mandal[a], and Xinwei Deng[c]

[a]Department of Statistics, University of Georgia, Athens, GA; [b]KLATASDS-MOE, School of Statistics, East China Normal University, Shanghai, China; [c]Department of Statistics, Virginia Tech, Blacksburg, VA

ABSTRACT

A new type of experiment that aims to determine the optimal quantities of a sequence of factors is eliciting considerable attention in medical science, bioengineering, and many other disciplines. Such studies require the simultaneous optimization of both quantities and sequence orders of several components which are called quantitative-sequence (QS) factors. Given the large and semi-discrete solution spaces in such experiments, efficiently identifying optimal or near-optimal solutions by using a small number of experimental trials is a nontrivial task. To address this challenge, we propose a novel active learning approach, called QS-learning, to enable effective modeling and efficient optimization for experiments with QS factors. QS-learning consists of three parts: a novel mapping-based additive Gaussian process (MaGP) model, an efficient global optimization scheme (QS-EGO), and a new class of optimal designs (QS-design). The theoretical properties of the proposed method are investigated, and optimization techniques using analytical gradients are developed. The performance of the proposed method is demonstrated via a real drug experiment on lymphoma treatment and several simulation studies. Supplementary materials for this article are available online.

1. Introduction

In modern scientific areas, nontraditional experiments that consider the quantities and sequences for arranging components, called quantitative sequence (QS) factors, are being conducted. For example, both the doses and order-of-addition for multiple drug components as a combination showed significant impacts on the efficacy of cancer treatments (Ding et al. 2015; Wang, Xu, and Ding 2020). In nanocellulose (NC) gel production, a pretreatment process involved swelling agents, different acids and enzymes to release hemicellulose. The sequence in which the pretreatment components were added, along with their quantities, was to be optimized for the NC size (Bharimalla et al. 2015). In the bio-plastics industry, the order in which the compatibilizer and scavenger were mixed with resin, along with their quantities, can cause a significant difference between catfish algae plastic and Solix microalgae plastic. Such QS factors are also used in physical or simulation experiments (a.k.a. computer experiments) in biochemistry (Shinohara and Ogawa 1998), food science (Jourdain et al. 2009) and management science (Panwalkar, Dudek, and Smith 1973).

To illustrate the characteristics of experiments with QS factors, Table 1 presents three runs from an in vitro drug combination experiment (Wang, Xu, and Ding 2020). Three antitumor drugs (A, B, and C) were added every 6 hr in a sequence at different doses. The percentage of tumor inhibition was measured as the response 6 hr after administering the last drug. As indicated in Table 1, different drug doses (comparing Runs 1

and 2) and sequences of adding drugs (comparing Runs 1 and 3) lead to varying responses. Thus, to identify the best drug combination, the doses and sequence for administering drugs should be optimized simultaneously. Notably, this experiment is different from crossover trials (Jones and Kenward 2014). A crossover trial measures all responses after the addition of every drug, and each drug exerts a fixed effect that may be carried over to the next period but does not depend on its order-of-addition. By contrast, only the endpoint efficacy after adding all the drugs will be measured as the response in a QS experiment, and drug effects are assumed to be dependent on the order-of-addition.

For experiments with QS factors, one of the key tasks is finding the optimal settings of quantities and sequences for arranging components to optimize experimental outcomes. In the current literature, researchers frequently enumerate all possible sequences and apply factorial designs to determine the quantities for each sequence (Wang, Xu, and Ding 2020). However, when the number of components is moderate or large, such a strategy may require a prohibitively large number of runs. It may also miss the optimal setting unless a wide range of levels is adopted. To the best of our knowledge, very few studies have been conducted on how to optimize the settings of QS factors via efficient modeling and experimental design. This problem is new and challenging, because QS factors are neither purely continuous nor categorical. To fill this gap, we propose a novel active learning approach, called QS-learning, which can identify good solutions by using only a few sequential experimental trials.

Table 1. Illustration of drug data involving QS factors.

Run	Drug A dosage	order	Drug B dosage	order	Drug C dosage	order	Response
1	3.75 μM	1	95 nM	2	0.16 μM	3	39.91
2	2.80 μM	1	70 nM	2	0.16 μM	3	30.00
3	3.75 μM	3	95 nM	1	0.16 μM	2	34.68

Active learning has attracted considerable attention in recent years (Cohn, Ghahramani, and Jordan 1996; Settles 2009; Deng et al. 2009). It sequentially queries the next data point on the basis of what it has learned from the current ones. Different methods for formulating query strategies have been proposed in the literature, including uncertainty sampling (Lewis and Gale 1994), query-by-committee (Seung, Opper, and Sompolinsky 1992), expected model change (Settles, Craven, and Ray 2007) and variance reductions (Cohn, Ghahramani, and Jordan 1996). Refer to Settles (2009) for a survey. From the experimentation perspective, active learning overlaps with optimal design (Burnaev and Panov 2015) and Bayesian optimization (Frazier 2018). It allocates runs in an adaptive manner, efficiently improving the decision for designing the next experimental trial as more information is acquired over time. Active learning frequently outperforms one-shot experimental designs for optimization when the solution space is large and complex (Kapoor et al. 2007; Burnaev and Panov 2015; Frazier 2018). It involves three major parts: (a) a method for statistical modeling and inference, (b) optimization of some acquisition functions for choosing the next design point, and (c) an initial design for exploring input space. Here, the acquisition function is typically a function that measures the "utility" of the run that will be evaluated next. It often considers "exploration/exploitation" tradeoff, such that balance is achieved between focusing on alternatives that appear to be good and experimenting with little known alternatives. Common choices of acquisition functions include expected improvement (Jones, Schonlau, and Welch 1998), knowledge gradient (Frazier, Powell, and Dayanik 2009), and entropy search (Hennig and Schuler 2012).

In this work, we propose an active learning approach (QS-learning) for experiments with QS factors. It includes a novel mapping-based additive Gaussian process (MaGP) model for prediction and uncertainty quantification, a sequential scheme that uses efficient global optimization algorithms (QS-EGO), and a new class of optimal experimental designs (QS-design) for collecting initial data points. A flowchart of the proposed QS-learning method is presented in Figure 1. The proposed method targets experiments with budget constraints in which practitioners prefer fewer runs. For cases with large data, we develop a variant of QS-learning for computational efficiency.

The key contributions of this work are summarized as follows. First, our proposed MaGP model enables the use of the Gaussian process (GP) in analyzing quantitative and sequence factors, providing desirable predictions and uncertainty quantification. Notably, the classic GP method has been widely used for modeling data with only quantitative factors (Rasmussen and Williams 2006; Kleijnen 2009). Some recent developments have enabled it for both quantitative and qualitative factors (Qian, Wu, and Wu 2008; Deng et al. 2017; Zhang et al. 2019; Xiao et al. 2021). However, this method cannot be easily adapted

for modeling data with QS factors due to the semi-discrete nature of sequence input. Second, we develop a new algorithm (QS-EGO) for efficiently optimizing the expected improvement (EI) acquisition function (Jones, Schonlau, and Welch 1998), which is nontrivial for QS factors. To address a complex solution space with both continuous and semi-discrete characteristics, the proposed QS-EGO combines a genetic algorithm and a new space-filling threshold-accepting (SFTA) algorithm. We derive analytical gradients for model estimation and acquisition function optimization to facilitate computation. Third, we develop a new class of optimal designs (QS-designs) for collecting initial data in the proposed active learning. QS-designs can reduce the number of required sequential runs while simultaneously improving performance. New design criteria are established to search for QS-designs with flexible sizes. We also develop an algebraic construction for QS-designs with certain sizes and prove their desirable properties. In the current experimental design literature, researchers have focused on either quantitative (Wu and Hamada 2021; Joseph 2016) or sequence (Mee 2020; Voelkel 2019; Lin and Peng 2019; Yang, Sun, and Xu 2021) factors, while the proposed QS-designs consider both factors simultaneously.

The remainder of this article is organized as follows. In Section 2, we review several related methods in the literature. In Section 3, we describe the formulation and estimation of the proposed MaGP model in detail. In In Section 4, we discuss the proposed sequential scheme along with QS-EGO. In Section 5, we illustrate the construction of a new class of optimal designs (i.e., QS-designs). A case study of a drug combination experiment on lymphoma is reported in Section 6, and a simulation study on the traveling salesman problem (TSP) is presented in Section 7. We conclude this work with discussions in Section 8. All proofs, technical details, convergence results, and additional numerical studies are included in the supplementary materials.

2. Brief Literature Review

QS factors are commonly observed in drug combination studies (Wang, Xu, and Ding 2020). However, conventional methods often consider only the effects of drug doses (quantitative input), for example, the Hill model (Chou 2006), polynomial model (Jaynes et al. 2013), Hill-based model (Ning et al. 2014), and Kriging or GP model (Xiao, Wang, and Xu 2019). Some recent studies have shown that if several drugs with fixed doses are added in desirable sequences, then such drug combinations will have enhanced efficacy (Ding et al. 2015). To model drug sequences with fixed doses, two types of linear models are proposed: the pairwise ordering (PWO) model (Van Nostrand 1995; Voelkel 2019; Mee 2020) and component-position (CP) model (Yang, Sun, and Xu 2021). We first review the two models and then generalize them for QS factors.

Let us consider a drug combination experiment with n runs and k drugs. For its ith run, let $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,k})^T$ be a vector containing the doses of k drugs and $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \ldots, \alpha_{i,k})^T$ be a vector containing the sequence of applying them. For example, if we add Drug B first, then C and finally A ($k = 3$), then the sequence of adding drugs (B, C, A) is represented by the vector $\boldsymbol{\alpha}_i = (2, 3, 1)^T$. In the PWO model, the features of $\boldsymbol{\alpha}_i$
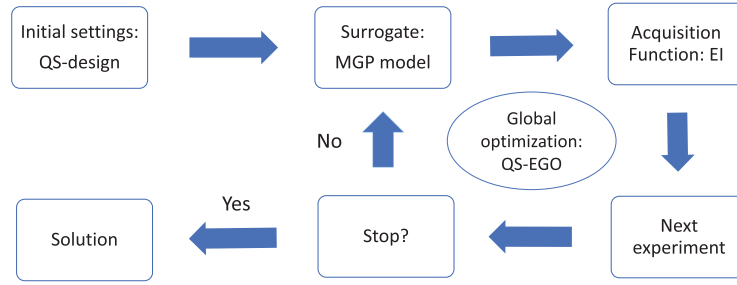
Figure 1. A flow chart of QS-learning for optimization.

are represented by the precedence patterns between all $\binom{k}{2}$ pairs of drugs. Explicitly, let S be the set of all pairs (p, q) for $1 \leq p < q \leq k$ and define the PWO indicator between p and q for any $(p, q) \in S$ as

$$z_{p,q}(\boldsymbol{\alpha}_i) = \begin{cases} 1 & \text{if } p \text{ precedes } q \text{ in } \boldsymbol{\alpha}_i, \\ -1 & \text{if } q \text{ precedes } p \text{ in } \boldsymbol{\alpha}_i. \end{cases}$$

As an illustration, for $\boldsymbol{\alpha}_i = (2, 3, 1)^T$, we have $S = \{(1, 2), (1, 3), (2, 3)\}$, $z_{1,2}(\boldsymbol{\alpha}_i) = -1$, $z_{1,3}(\boldsymbol{\alpha}_i) = -1$, and $z_{2,3}(\boldsymbol{\alpha}_i) = 1$. Based on PWO indicators, the PWO model (Van Nostrand 1995; Voelkel 2019) is defined as

$$f(\mathbf{x}_i^T, \boldsymbol{\alpha}_i^T) = \beta_0 + \sum_{(p,q) \in S} z_{p,q}(\boldsymbol{\alpha}_i) \beta_{p,q} + \epsilon, \quad (1)$$

where the residual ϵ follows the standard assumptions for linear models and parameters β can be estimated via the least squares method. To further capture the two-factor interactions between PWO indicators, Mee (2020) proposed the triplet PWO model. In this work, we consider the PWO model, which often suffices in practice. The triplet PWO model includes as many as $1 + k + k(k-1)/2 + k(k-1)(k-2)/3$ parameters, which often exceed the total number of runs in sequential experiment considered in this work.

Another class of linear models is CP model (Yang, Sun, and Xu 2021), which is defined as

$$f(\mathbf{x}_i^T, \boldsymbol{\alpha}_i^T) = \beta_0 + \sum_{j=1}^{k-1} \sum_{c=1}^{k-1} x_{i,c}^{(j)} \beta_{j,c} + \epsilon_i, \quad (2)$$

where $x_{i,c}^{(j)}$ equals 1 if $\alpha_{i,j} = c$ and 0 otherwise. That is, $x_{i,c}^{(j)}$ is an indicator of whether Drug c is used in the ith run at the jth position. Simply put, CP is a multivariate linear regression model treating each position as a factor with k levels.

Both the PWO and CP models in the current literature work only for sequence factors. In order to establish some benchmark models, we generalize the PWO and CP models via adding covariates for quantitative factors (e.g., doses), such that they can work for QS factors. Both the generalized PWO and CP models can be represented as

$$g(\mathbf{x}_i^T, \boldsymbol{\alpha}_i^T) = \sum_{j=1}^{k} \beta_j' x_{i,j} + f(\mathbf{x}_i^T, \boldsymbol{\alpha}_i^T), \quad (3)$$

where β_j' denotes the coefficients for quantitative factors, and $f(\mathbf{x}_i^T, \boldsymbol{\alpha}_i^T)$ can be either the PWO model in (1) or the CP model in (2).

Linear models may work well under one-shot experimental designs for prediction purposes, but they are less popular in active learning for optimization. Compared with GP models, linear models often perform worse in uncertainty quantification (Smith 2013; Burnaev and Panov 2015). The GP model, where responses are represented by random variables whose probability distributions characterize the beliefs of experimenters about the unknown values, provides a good probabilistic framework for active learning (Rasmussen and Williams 2006; Kapoor et al. 2007; Frazier 2018). It enables the predictive distribution of the outcome of the next experiment and the selection of the best one by maximizing an acquisition function.

3. Mapping Based Additive GP Model

3.1. Model Formulation

Let us consider a QS experiment with n runs and k components (i.e., $c_1, c_2, \ldots, c_k$), where the ith input is denoted as $\mathbf{w}_i = (\mathbf{x}_i^T, \mathbf{o}_i^T)^T$ and the corresponding output is denoted as y_i. Here, $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,k})^T$ is a vector of quantitative values for k components, and $\mathbf{o}_i = (o_{i,1}, \ldots, o_{i,k})^T$ is a vector that contains the orders of components in the arrangement sequence. Notably, $x_{i,h}$ and $o_{i,h}$ ($h = 1, \ldots, k$ and $i = 1, \ldots, n$) are the quantitative and sequence parts of the hth component c_h, respectively. Without loss of generality, we assume that $\mathbf{o}$ is a permutation of the integers 1 to k. As an illustration, the third run in Table 1, $\mathbf{w}_3^T = (\mathbf{x}_3^T, \mathbf{o}_3^T)$, has $\mathbf{x}_3 = (3.75, 95, 0.16)^T$ and $\mathbf{o}_3 = (3, 1, 2)^T$. The vector $\mathbf{o}_3$ represents that Drug A is added in the third place, B is added in the first place, and C is added in the second place, that is, $B \rightarrow C \rightarrow A$. Notably, the $\mathbf{o}$ defined here contains the index orders of the corresponding elements in the vector $\boldsymbol{\alpha}$ defined in Section 2, and they have the same practical meaning.

The order sequence is semi-discrete in nature; hence, the relationship between output and QS input can be complicated. To model such data, we consider the adoption of the GP model because of its flexibility and promising prediction and uncertainty quantification. For an experiment with n runs and k components, we model the output at $\mathbf{w} = (\mathbf{x}^T, \mathbf{o}^T)^T$ as

$$Y(\mathbf{w}) = \mu + \sum_{h=1}^{k} G_h(\mathbf{w}) + \epsilon, \quad (4)$$

where $G_1, \ldots, G_k$ are independent zero-mean GPs with stationary covariance functions, and $\epsilon \sim N(0, \tau^2)$ is a random error. The GP component G_h ($h = 1, \ldots, k$) corresponds to the effect of the hth component c_h on the output. For physical

experiments, we assume homogeneous error variances $\tau^2 > 0$ which may come from measurement errors or some environmental factors. For computer experiments, we take $\tau^2 = 0$, because computer codes provide deterministic output (Fang, Li, and Sudjianto 2005).

In GP models, distances between pairs of input are used to measure their similarities when formulating covariance functions. For the hth ($h = 1, \ldots, k$) component, its sequence input $o_{i,h}$ (for any $i = 1, \ldots, n$) is ordinal. Thus, a distance measure should be specified for sequence input to form the covariance function in the GP component G_h. To address this challenge, we consider mapping the order $o_{i,h}$ ($o_{i,h} \in \{1, \ldots, k\}$) to a t-dimensional latent vector $(\widetilde{o}_{i,h}^{(1)}, \ldots, \widetilde{o}_{i,h}^{(t)})$. Given that the sequence input $\mathbf{o}$ is an assignment of k components to k "fixed" order positions, we should use the same mapping for all components $c_1, \ldots, c_k$ (corresponding to GP components $G_1, \ldots, G_k$, respectively) to quantify the effects of fixed order positions via latent variables. In particular, the t-dimensional mapping ($t = 1, \ldots, k-1$) for the order of any component is defined as

$$
c_1, \ldots, c_k \quad
\begin{matrix} & \widetilde{o}^{(1)} & \widetilde{o}^{(2)} & \ldots & \widetilde{o}^{(t)} \end{matrix}
$$
$$
\begin{bmatrix} 1 \\ 2 \\ \vdots \\ k \end{bmatrix} \rightarrow
\begin{bmatrix} \delta_1^{(1)} & \delta_1^{(2)} & \cdots & \delta_1^{(t)} \\ \delta_2^{(1)} & \delta_2^{(2)} & \cdots & \delta_2^{(t)} \\ \vdots & \vdots & \vdots & \vdots \\ \delta_k^{(1)} & \delta_k^{(2)} & \cdots & \delta_k^{(t)} \end{bmatrix}_{k \times t}, \quad (5)
$$

where $\delta_l^{(j)} = 0$ for all $j \geq l$ to avoid over-parameterization. The interactions among different levels (i.e., orders) can be reflected by the mapping parameters in (5), which are estimated from the data. As all components use the same mapping, the total number of mapping parameters is $t(t+1)/2 + (k-t-1)t$. Specifically, when $t = k-1$, we call it full mapping with a total of $k(k-1)/2$ mapping parameters. When $t = 2$, we call it 2d-mapping, which has $(2k-3)$ mapping parameters.

Example 1. For illustration, consider a QS experiment to find the optimal sequence and quantity to add for $k = 4$ operations in a single production line with four fixed locations to be assigned with different operations. We use the same mapping for all four operations (c_1, c_2, c_3, c_4), which quantifies the effects due to locations (i.e., position orders) to be assigned with operations:

full mapping
$$
c_1, c_2, c_3, c_4 \quad \begin{matrix} & \widetilde{o}^{(1)} & \widetilde{o}^{(2)} & \widetilde{o}^{(3)} \end{matrix}
$$
$$
\begin{bmatrix} \text{order 1} \\ \text{order 2} \\ \text{order 3} \\ \text{order 4} \end{bmatrix} \rightarrow
\begin{bmatrix} 0 & 0 & 0 \\ \delta_2^{(1)} & 0 & 0 \\ \delta_3^{(1)} & \delta_3^{(2)} & 0 \\ \delta_4^{(1)} & \delta_4^{(2)} & \delta_4^{(3)} \end{bmatrix},
$$

2d-mapping
$$
c_1, c_2, c_3, c_4 \quad \begin{matrix} & \widetilde{o}^{(1)} & \widetilde{o}^{(2)} \end{matrix}
$$
$$
\begin{bmatrix} \text{order 1} \\ \text{order 2} \\ \text{order 3} \\ \text{order 4} \end{bmatrix} \rightarrow
\begin{bmatrix} 0 & 0 \\ \delta_2^{(1)} & 0 \\ \delta_3^{(1)} & \delta_3^{(2)} \\ \delta_4^{(1)} & \delta_4^{(2)} \end{bmatrix},
$$

where $\delta_l^{(j)}$ ($j < l$) denotes parameters to be estimated via maximum likelihood estimation (MLE).

The prespecified tuning parameter t ($t \in \{1, \ldots, k-1\}$) controls the flexibility of defining similarities between pairs of order positions. Under full mapping ($t = k-1$), all pairwise distances between order positions can be independently determined. Then, all possible patterns in defining similarities between sequence input can be captured. By contrast, under 1d-mapping ($t = 1$), the mapping in (5) is simplified as order $1 \rightarrow 0$, order $2 \rightarrow \delta_1, \ldots,$ order $k \rightarrow \delta_{k-1}$, or equivalently order $1 \rightarrow 0$, order $2 \rightarrow \delta_1', $ order $3 \rightarrow \delta_1' + \delta_2', \ldots,$ order $k \rightarrow \sum_{i=1}^{k-1} \delta_i'$. Evidently, only the distances between adjacent order positions are independently determined here. For example, when $t = 1$, the distance between orders 1 and 3 (determined via $\delta_1' + \delta_2'$) is dependent on the distance between orders 1 and 2 (determined via δ_1') and the distance between orders 2 and 3 (determined via δ_2'). Such restrictive mapping works for cases where only adjacent orders interact with one another. In this work, we consider $t \geq 2$ to allow a more general pattern of interactions.

An appropriate choice of t provides a tradeoff between model flexibility and computational cost. When many components are involved (i.e., large k), low-dimensional mapping (e.g., 2d-mapping) is often a good choice. It will considerably reduce the number of parameters in the MaGP model, facilitating model estimation in practice. In 2d-mapping, any pairwise distance between order positions can be partially (not fully) determined by other pairwise distances. Thus, this type of mapping can provide certain flexibility to capturing possible patterns for defining similarities between sequence input.

In the ith run $\mathbf{w}_i = (\mathbf{x}_i^T, \mathbf{o}_i^T)^T$, the elements that correspond to the hth component c_h are $(x_{i,h}, o_{i,h})$, where $i = 1, \ldots, n$ and $h = 1, \ldots, k$. From the mapping in (5), we define the distance between the ith and jth runs that correspond to the hth component c_h under the L_2 norm as

$$
d_{i,j}^{(h)} = ||(x_{i,h}, o_{i,h}) - (x_{j,h}, o_{j,h})||
$$
$$
= \sqrt{\theta_h(x_{i,h} - x_{j,h})^2 + \sum_{l=1}^{t} (\widetilde{o}_{i,h}^{(l)} - \widetilde{o}_{j,h}^{(l)})^2}, \quad (6)
$$

where θ_h is the correlation parameter that scales the quantitative input of c_h. Here, the t-dimensional latent vectors $(\widetilde{o}_{i,h}^{(1)}, \ldots, \widetilde{o}_{i,h}^{(t)})$ and $(\widetilde{o}_{j,h}^{(1)}, \ldots, \widetilde{o}_{j,h}^{(t)})$ correspond to the orders $o_{i,h}$ in $\mathbf{w}_i$ and $o_{j,h}$ in $\mathbf{w}_j$, respectively, and their values are determined by the mapping parameters in (5) (denoted as $\boldsymbol{\delta}$). Notably, there is no need to include any correlation parameters to scale latent vectors, because the mapping parameters ($\boldsymbol{\delta}$) are estimated from the data.

Subsequently, we can describe the proposed covariance function for the hth GP component G_h ($h = 1, \ldots, k$) in (4) as

$$
\phi_h(\mathbf{w}_i, \mathbf{w}_j | \sigma_h^2, \theta_h, \boldsymbol{\delta}) = \phi_h((x_{i,h}, o_{i,h}), (x_{j,h}, o_{j,h}) | \sigma_h^2, \theta_h, \boldsymbol{\delta})
$$
$$
= \sigma_h^2 K(d_{i,j}^{(h)}), \quad (7)
$$

where σ_h^2 is the variance parameter corresponding to the hth component c_h, and $K(\cdot)$ is any valid kernel function. Popular kernels include the Matern class with a smoothness parameter $\nu \in (0, \infty)$,

$$
K(d_{i,j}^{(h)}) = \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2}\nu d_{i,j}^{(h)})^\nu k_\nu(\sqrt{2}\nu d_{i,j}^{(h)}), \quad (8)
$$

where k_ν is a modified Bessel function of the second kind. Specifically, we focus on the case of $\nu \to \infty$, that is, the Gaussian kernel, in this work:

$$K(d_{i,j}^{(h)}) = \exp(-(d_{i,j}^{(h)})^2). \tag{9}$$

In (9), we remove a constant multiplier of $1/2$ in the exponent for re-parameterization.

By (4), (5), (6), (7), and (9), for any two input $\mathbf{w}_i$ and $\mathbf{w}_j$, the covariance function for the MaGP model in (4) can be specified by:

$$\phi(\mathbf{w}_i, \mathbf{w}_j) = \text{cov}(Y(\mathbf{w}_i), Y(\mathbf{w}_j))$$
$$= \sum_{h=1}^{k} \phi_h(\mathbf{w}_i, \mathbf{w}_j | \sigma_h^2, \theta^{(h)}, \boldsymbol{\delta}) + \tau^2 \mathbf{1}(\mathbf{w}_i = \mathbf{w}_j)$$
$$= \sum_{h=1}^{k} \sigma_h^2 \exp\Big\{ -\theta_h(x_{i,h} - x_{j,h})^2 - \sum_{l=1}^{t} (\widetilde{o}_{i,h}^{(l)} - \widetilde{o}_{j,h}^{(l)})^2 \Big\}$$
$$+ \tau^2 \mathbf{1}(\mathbf{w}_i = \mathbf{w}_j), \tag{10}$$

where $\tau^2 \geqslant 0$, and $\mathbf{1}(\cdot)$ is an indicator function. Here, the variance parameter σ_h^2 corresponds to the effect of the hth component, and τ^2 is the variance of the error term $\epsilon \sim N(0, \tau^2)$ in (4). This covariance function combines different dimensions via addition, and the quantity part and sequence part in each dimension via multiplication. It cannot be decomposed into the sum or product of a covariance for purely quantitative factors and a covariance for purely sequence factors. Thus, it is not the separable covariance function as defined in Gneiting (2002).

Given the noise variance τ^2, the MaGP model with the covariance function in (10) includes $n_{\text{par}} = 1 + 2k + kt - t(t+1)/2$ parameters. Specifically, the full-MaGP ($t = k-1$) and the 2d-MaGP ($t = 2$) include $1 + k(k+3)/2$ and $4k - 2$ parameters, respectively.

Theorem 1. Given n input $\mathbf{w}_i = (\mathbf{x}_i^T, \mathbf{o}_i^T)^T$ ($i = 1, \ldots, n$), the covariance matrix of outputs $\mathbf{y} = (Y(\mathbf{w}_1), \ldots, Y(\mathbf{w}_n))^T$ induced by the covariance function in (10) is positive semidefinite.

Theorem 1 holds for any $\mathbf{w}_1, \ldots, \mathbf{w}_n$, including duplicated input, and τ^2 can be 0. For appropriate model inference, $\text{cov}(\mathbf{y})$ must be positive definite, and the following two corollaries shed some light on this aspect.

Corollary 1. Given n input and the noise variance $\tau^2 > 0$, the covariance matrix $\text{Cov}(\mathbf{y})$ induced by the covariance function in (10) is positive definite.

Corollary 2. When the noise variance $\tau^2 = 0$, if no two runs have the same quantitative input (i.e., $\mathbf{x}_i \neq \mathbf{x}_j$ for $i \neq j$), then the covariance matrix $\text{cov}(\mathbf{y})$ induced by the covariance function in (10) is positive definite.

Corollary 1 guarantees the validity of the covariance matrix for modeling physical experiments. For modeling computer experiments, if Latin hypercube designs (Lin and Tang 2015), orthogonal arrays (Hedayat, Sloane, and Stufken 1999), or space-filling designs (Wang, Xiao, and Xu 2018) are used as the quantitative parts of design matrices (where all $\mathbf{x}_i \neq \mathbf{x}_j$ for $i \neq j$), then the covariance matrices in the proposed model

are positive definite by Corollary 2. In Section 5, we propose a new class of optimal designs for QS factors that satisfies the requirements in Corollary 2 and has more attractive properties. Notably, if two runs have the same quantitative part but different sequence parts, then we need to set $\delta_l^{(l-1)} \neq 0$ for $l = 2, \ldots, k$ in the model estimation to guarantee that the covariance matrix is positive definite.

Notably, the warping technique in the literature (Snelson, Ghahramani, and Rasmussen 2004; Xiao and Xu 2021) can be used for ordinal factors, where an ordinal input $o_{i,h}$ is mapped to a quantitative input $f(o_{i,h})$ via a certain transformation function $f_h(\cdot)$. Evidently, such a technique is a special case of, and thus, more restrictive than the 1d-mapping used in the current work. The warping technique frequently considers the case of independent ordinal factors. However, the sequence factors in QS experiments are not independent, because they are required to form sequence input (i.e., permutations of $1, \ldots, k$). Zhang et al. (2019) considered a latent approach for mapping qualitative input to some quantitative vectors in GP. Their method differs from the proposed MaGP in at least two aspects. First, they considered a single GP with a multiplicative covariance structure wherein a single variance parameter is adopted. Such a model structure may not distinguish the specific effects of different qualitative factors. By contrast, the MaGP model considers additive GPs, wherein each GP component has a specific variance parameter that measures the effect of each component. Second, Zhang et al. (2019) set different mapping matrices for various qualitative factors. While, the MaGP model adopts the same mapping for all components, because the order sequence is an assignment of k components to k "fixed" order positions.

3.2. Model Estimation

For parameter estimation, the proposed MaGP model in (4) with the covariance function in (10) contains parameters μ, $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_k^2)^T$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)^T$, $\boldsymbol{\delta} = (\delta_2^{(1)}, \ldots, \delta_k^{(t)})^T$, and τ^2. These parameters can be estimated via the likelihood function. The covariance matrix is denoted by $\boldsymbol{\Phi} = \boldsymbol{\Phi}(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{\delta}, \tau^2) = (\phi(\mathbf{w}_i, \mathbf{w}_j))_{n \times n}$, which follows the covariance function in (7). With some simple algebra, the negative log-likelihood function can be expressed as (up to a constant)

$$\log|\boldsymbol{\Phi}| + (\mathbf{y} - \mu\mathbf{1})^T \boldsymbol{\Phi}^{-1}(\mathbf{y} - \mu\mathbf{1}), \tag{11}$$

where the response vector $\mathbf{y} = (Y(\mathbf{w}_1), \ldots, Y(\mathbf{w}_n))^T$, and $\mathbf{1}$ is an $n \times 1$ column vector of all 1s. For given $\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{\delta}$, and τ^2, the MLE of μ can be obtained explicitly as

$$\widehat{\mu} = (\mathbf{1}^T \boldsymbol{\Phi}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \boldsymbol{\Phi}^{-1} \mathbf{y}. \tag{12}$$

By substituting (12) into (11), the estimation of $\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{\delta}$, and τ^2 can be obtained by

$$[\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{\delta}, \tau^2] = \text{argmin} \big\{ \log |\boldsymbol{\Phi}| + (\mathbf{y}^T \boldsymbol{\Phi}^{-1} \mathbf{y})$$
$$- (\mathbf{1}^T \boldsymbol{\Phi}^{-1} \mathbf{1})^{-1} (\mathbf{1}^T \boldsymbol{\Phi}^{-1} \mathbf{y})^2 \big\}. \tag{13}$$

This minimization problem can be solved using some standard nonlinear optimization algorithms in Matlab or R. Different algorithms or the same algorithm with different initializations may lead to different parameter estimates (Erickson, Ankenman, and Sanchez 2018). In the current work, we

adopt the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm with random initialization (Liu and Nocedal 1989). It is a popular method for estimating GP models. It determines descent direction by preconditioning the gradient with curvature information. Numerical gradients can be used, but they are approximate and are expensive to compute. In the current study, we derive analytical gradients to facilitate fast and exact computation. We report all analytical gradients and the implementation of the optimization algorithm in supplementary materials Sections S1.2 and S3.1, respectively.

Given all the estimated parameters, the prediction mean and variance of the response at the target input $\mathbf{w}_*$ are given by

$$\widehat{Y}(\mathbf{w}_*) = \widehat{\mu} + \boldsymbol{\gamma}^T \boldsymbol{\Phi}^{-1}(\mathbf{y} - \widehat{\mu}\mathbf{1}), \tag{14}$$

$$s^2(\mathbf{w}_*) = \phi(\mathbf{w}_*, \mathbf{w}_*) - \boldsymbol{\gamma}^T \boldsymbol{\Phi}^{-1}\boldsymbol{\gamma} + \frac{(1 - \mathbf{1}^T\boldsymbol{\Phi}^{-1}\boldsymbol{\gamma})^2}{(\mathbf{1}^T\boldsymbol{\Phi}^{-1}\mathbf{1})}. \tag{15}$$

Here, $\boldsymbol{\gamma}$ is the covariance vector $(\phi'(\mathbf{w}^*, \mathbf{w}_i))_{n\times 1}$, where $\phi'(\mathbf{w}^*, \mathbf{w}_i) = \sum_{h=1}^{k} \sigma_h^2 \exp\{-\theta_h(x_{i,h} - x_{*,h})^2 - \sum_{l=1}^{t}(\widetilde{o}_{i,h}^{(l)} - \widetilde{o}_{*,h}^{(l)})^2\}$. Notably, the estimators in (14) and (15) are commonly used in the literature (Rasmussen and Williams 2006; Kleijnen 2009; Gramacy 2020). For an unbiased small-sample estimator of $s^2(\mathbf{w}_*)$, refer to Mehdad and Kleijnen (2015).

For computer experiments with $\tau^2 = 0$, when $\mathbf{w}_*$ is the ith observed input $\mathbf{w}_i$, $\boldsymbol{\gamma}^T$ is the ith row in $\boldsymbol{\Phi}$, and $\boldsymbol{\gamma}^T\boldsymbol{\Phi}^{-1}$ is a row vector with the ith entry being 1 and the others being 0. Evidently, $\widehat{Y}(\mathbf{w}_*) = \widehat{Y}(\mathbf{w}_i) = y_i$, and thus (15) yields $s^2(w^*) = 0$. Therefore, the interpolation property holds. Note that if $\boldsymbol{\Phi}$ is ill-conditioned, then a nugget (or noise) effect may be added, and the interpolation property may not hold; refer to Gramacy and Lee (2012) for details. For physical experiments, the interpolation property does not hold due to the presence of random errors ϵ in (4). In practice, when no replicates are included in physical experiments, the homogeneous noise variance τ^2 of random errors ϵ is often prespecified according to known background information, and different small τ^2 values may not lead to a significant difference in prediction (Xiao, Wang, and Xu 2019). When replicates are included, τ^2 should be estimated via MLE as shown in (13). For some basic derivations of GP model estimation, refer to Rasmussen and Williams (2006) and Roustant, Ginsbourger, and Deville (2012) for a survey.

4. Active Learning for Experiments with QS Factors

In this section, we first introduce a general active learning scheme for experiments with QS factors and then discuss its variant for computational scalability. Given the large and semi-discrete input spaces in such experiments, adapting existing methods for optimization is nontrivial. This issue motivates us to develop a tailored new optimization algorithm.

4.1. EI Optimization with QS Factors

In experimentation, active learning has received considerable attention since the expected improvement (EI) framework, which works for quantitative factors, was proposed by Jones, Schonlau, and Welch (1998). In this section, we adapt the EI acquisition function to work with QS factors and develop an

efficient global optimization algorithm (QS-EGO). This new algorithm adopts the proposed MaGP as the probabilistic model for the input–output relationship, under which we derive the analytical gradients for optimizing EI.

Without loss of generality, we focus on finding the optimal solution $\mathbf{w}$ to minimize the "black-box" objective function $y(\mathbf{w})$ in either physical or computer experiments. Notably, any maximization problem can be viewed as a minimization problem to the negative objective function $-y(\mathbf{w})$. Let the improvement function be $I(\mathbf{w}) = (y_{\min}^{(n)} - y(\mathbf{w}))_+$ for an input $\mathbf{w}$, where $a_+ = \max(a, 0)$ indicates the nonnegative part of a, and $y_{\min}^{(n)}$ is the minimum response of the n current observations. EI is defined as $E[I(\mathbf{w})] = \int I(\mathbf{w})f_n(y|\mathbf{w})dy$, where $f_n(y|\mathbf{w})$ is the probability density function of the predictive distribution given by the MaGP model based on the n current observations. EI at input $\mathbf{w} = (\mathbf{x}^T, \mathbf{o}^T)^T$ can be expressed in closed form as

$$\text{EI} = E[I(\mathbf{w})] = (y_{\min}^{(n)} - \widehat{Y}(\mathbf{w}))\Phi\left(\frac{y_{\min}^{(n)} - \widehat{Y}(\mathbf{w})}{s(\mathbf{w})}\right)$$
$$+ s(\mathbf{w})\varphi\left(\frac{y_{\min}^{(n)} - \widehat{Y}(\mathbf{w})}{s(\mathbf{w})}\right), \tag{16}$$

where $\Phi(\cdot)$ and $\varphi(\cdot)$ denote the cumulative distribution function and probability density function of the standard normal distribution, respectively. The prediction mean $\widehat{Y}(\mathbf{w})$ and its standard error $s(\mathbf{w}) = \sqrt{s^2(\mathbf{w})}$ are provided in (14) and (15), respectively. EI inherits a tradeoff between exploitation and exploration (Jones, Schonlau, and Welch 1998). The first term in (16) is maximized by the experimental point having the smallest mean value, and thus, it can be interpreted as the exploitation part. Meanwhile, the second term is maximized by the unexplored point having the largest uncertainty, and thus, it can be interpreted as the exploration part.

As shown in Figure 1, the workflow of QS-learning includes the following four steps.

1. Construct an optimal initial design for QS factors with n_0 runs $\mathbf{w}_1, \ldots, \mathbf{w}_{n_0}$. Compute (or simulate) their responses as $y_1, \ldots, y_{n_0}$. Then, fit the MaGP model based on these observations. Set $n = n_0$.
2. Select the next design point $\mathbf{w}_{n+1}$ that maximizes the EI acquisition function in (16) by using the QS-EGO (shown as Algorithm 1), and then compute (or simulate) its response as y_{n+1}.
3. Re-fit the MaGP model based on observations $(\mathbf{w}_1, y_1), \ldots, (\mathbf{w}_{n+1}, y_{n+1})$. Set $n = n + 1$.
4. Repeat Steps 2 and 3 until the stopping criterion is satisfied.

In Step 2, when the number of components k is small, we can enumerate $k!$ sequences (possibly with parallel computing) and identify the optimal $\mathbf{x}$ given each sequence $\mathbf{o}$ that can maximize the EI function. For a large k, such an enumeration may become prohibitively time-consuming. To address this challenge, we propose to iteratively optimize quantitative input $\mathbf{x}$ and sequence input $\mathbf{o}$ given the other, as summarized in Algorithm 1. In both the four-step QS-learning and Algorithm 1, we adopt the stopping criterion used in Jones, Schonlau, and Welch (1998), that is, that the algorithm stops when three consecutive EIs do

Algorithm 1 An efficient optimization of EI for QS factors (QS-EGO)

Initialize the maximum number of rounds N_{round}.

Initialize the current input $\mathbf{w}_c = (\mathbf{x}_c, \mathbf{o}_c)$. Set the current optimal input $\mathbf{w}_{\text{opt}} = \mathbf{w}_c$.

Initialize an empty vector $\mathbf{ei}$ and a scalar $\text{ei}_{\text{opt}} = 0$.

for $i = 1$ to N_{round} **do**

  1. Given $\mathbf{o}_c$, find the quantitative input $\mathbf{x}$ that maximizes EI via a genetic optimization algorithm. Set $\mathbf{x}_c = \mathbf{x}$.

  2. Given $\mathbf{x}_c$, find the sequence input $\mathbf{o}$ that maximizes EI via the SFTA method in Algorithm 2. Set $\mathbf{o}_c = \mathbf{o}$.

  3. Set $\mathbf{w}_c = (\mathbf{x}_c, \mathbf{o}_c)$ and evaluate $E[I(\mathbf{w}_c)]$ defined in (16).

  if $E[I(\mathbf{w}_c)] > \text{ei}_{\text{opt}}$,

    then set $\text{ei}_{\text{opt}} = E[I(\mathbf{w}_c)]$ and $\mathbf{w}_{\text{opt}} = \mathbf{w}_c$.

  Set $\mathbf{ei}[i] = \text{ei}_{\text{opt}}$, where $\mathbf{ei}[i]$ denotes the ith element of $\mathbf{ei}$.

  if the stopping criterion is satisfied,

    then break.

end for

Return $\mathbf{w}_{\text{opt}}$.

Algorithm 2 SFTA algorithm for optimizing $\mathbf{o}$ given $\mathbf{x}_c$

Initialize $n_{\text{step}}^{(1)}$ (number of steps) in SFTA Phase I.

Initialize n_{seq} (number of iterations to compute the threshold sequence), n_{rounds} (number of rounds), and $n_{\text{steps}}^{(2)}$ (number of steps) in SFTA Phase II.

Initialize a starting solution $\mathbf{o}_0$, set current optimal $\mathbf{o}_{\text{opt}} = \mathbf{o}_0$, and let $O_{\text{obs}} = [\mathbf{o}_0^T]^T$.

while $i \leqslant n_{\text{step}}^{(1)}$ **do**

  Generate a random sequence $\mathbf{o}_i$.

  if $H_{\min}(\mathbf{o}_i, O_{\text{obs}})/k > \epsilon$ (ϵ is drawn from Unif(0,1)),

    then let $i = i + 1$, $O_{\text{obs}} = [O_{\text{obs}}, \mathbf{o}_i^T]^T$, and $\delta = f(\mathbf{o}_i) - f(\mathbf{o}_{\text{opt}})$,

  if $\delta < 0$, **then** let $\mathbf{o}_{\text{opt}} = \mathbf{o}_i$.

end while

Set the current solution $\mathbf{o}_c = \mathbf{o}_{\text{opt}}$.

for $j = 1$ to n_{seq} **do**

  Generate a neighbor solution $N(\mathbf{o}_c)$, and let $\Delta_j = |f(o_c) - f(N(\mathbf{o}_c))|$.

end for

Compute the empirical distribution of Δ_j, $j = 1, 2, \ldots, n_{\text{seq}}$, denoted as F.

for $r = 1$ to n_{rounds} **do**

  Generate threshold $\tau_r = F^{-1}(0.5(1 - r/n_{\text{rounds}}))$

  for $i = 1$ to $n_{\text{steps}}^{(2)}$ **do**

    Generate a neighbor solution $N(\mathbf{o}_c)$, and let $\delta = f(N(\mathbf{o}_c)) - f(\mathbf{o}_c)$.

    if $\delta < \tau_r$, **then** let $\mathbf{o}_c = N(\mathbf{o}_c)$.

    if $f(\mathbf{o}_c) < f(\mathbf{o}_{\text{opt}})$, **then** let $\mathbf{o}_{\text{opt}} = \mathbf{o}_c$.

  end for

end for

Return $\mathbf{o}_{\text{step}}$.

---

not produce more than α ($\alpha \in [0.1\%, 1\%]$) improvement over the current best output.

In Algorithm 1, to optimize the quantitative input $\mathbf{x}$ given the sequence input $\mathbf{o}_c$, we adopt a BFGS method in a hybrid genetic optimization algorithm (Mebane, Jr. and Sekhon 2011). Numerical gradients suffice for small cases, but they can be slow for large ones. In this study, we derive the analytical gradients for EI maximization under the proposed MaGP model. These gradients are exact and fast to compute. See supplementary materials Section S1.3 for details.

In Algorithm 1, optimizing the sequence input $\mathbf{o}$ given the quantitative input $\mathbf{x}_c$ is nontrivial, because its solution space is semi-discrete and can be extremely large. To address this issue, we propose the so-called space-filling threshold accepting (SFTA) algorithm for a large k. As detailed in Algorithm 2, the proposed SFTA algorithm includes two phases. The first phase seeks space-filling points that are far from one another to achieve robustness (Johnson, Moore, and Ylvisaker 1990). The second phase inherits from the classic threshold-accepting (TA) algorithm (Dueck and Scheuer 1990), which balances exploration and exploitation.

Specifically, Phase I of SFTA starts from the current optimal sequence input $\mathbf{o}_0$ with the smallest $f(\mathbf{o})$ value in the observed data, where the objective function $f(\mathbf{o})$ is the negative EI value for $\mathbf{o}$ given $\mathbf{x}$. Then, the algorithm iteratively accepts a random sequence $\mathbf{o}_i$ ($i = 1, \ldots, n^{(1)}$) for evaluation with probability $P(\mathbf{o}_i) = H_{\min}(\mathbf{o}_i)/k$, where $H_{\min}(\mathbf{o}_i)$ is the minimum pairwise Hamming distance between the sequence vector $\mathbf{o}_i$ and all observed sequence vectors $\mathbf{o}_1, \ldots, \mathbf{o}_{i-1}$. The Hamming distance is the number of positions at which corresponding symbols are different in two vectors. If a candidate $\mathbf{o}_i$ is far from the observed $\mathbf{o}_1, \ldots, \mathbf{o}_{i-1}$ under the Hamming distance, then it has a high probability of being included for evaluation. Phase II of SFTA starts from the optimal solution $\mathbf{o}_c = \mathbf{o}_{\text{opt}}$ with the smallest $f(\mathbf{o})$ value in Phase I. We define the neighbor solution as $N(\mathbf{o}_c)$ by randomly exchanging two elements of the sequence vector $\mathbf{o}_c$

(current solution). Evidently, all possible $N(\mathbf{o}_c)$'s have a Hamming distance of 2 with $\mathbf{o}_c$. The threshold values for accepting neighbor solutions are generated by empirical distributions of increments (denoted as F) for the objective function f; refer to Dueck and Scheuer (1990) for details on the calculation of threshold values. A neighbor solution is more likely to be accepted early in the search than later in the search, because threshold values decrease. The SFTA algorithm can avoid being trapped at local optima and focus more on exploration in the beginning.

When the number of allowed evaluations is considerably fewer than the total number of possible sequences, random initialization (and generation) of neighbor solutions may not consistently and efficiently explore space. To address this issue, we adopt space-filling samples in Phase I, which provide a good initialization for TA global optimization in Phase II. When parallel computing is available, more than one solution $\mathbf{o}_{\text{opt}}$ in Phase I can be selected as multi-starting points. Refer to supplementary materials Section S3.2 for additional details on Algorithms 1 and 2.

4.2. Fast QS-Learning for Large Experiments

In most literature on experimentation, the costs for estimating surrogate models and assessing acquisition functions are

negligible compared with the costs of conducting experiments (Frazier 2018; Gramacy 2020). In other cases, experiments may be executed rapidly, and researchers will need a fast sequential scheme for a large number of runs (Gramacy 2020).

Classic GP-based active learning approaches have a computational complexity of $O(N^4)$, where N is the total number of runs. The estimation of GP models in each iteration requires $O(n^3)$ computation, where n is the number of data points used. Here, we propose a fast variant of QS-learning with $O(N^3)$ computation. Suppose that the total budgets for the number of runs and computing time are N and T, respectively. The proposed fast QS-learning approach includes the following four steps.

1. Construct an optimal initial design for QS factors with n_0 runs $\mathbf{w}_1, \ldots, \mathbf{w}_{n_0}$, evaluate their responses $y_1, \ldots, y_{n_0}$, and fit the MaGP model based on these observations. Set $n = n_0$. Record the time used for fitting the MaGP model as t. Record the time left from the budget T as T' and the number of runs left from the budget N as N'.
2. For the next $\lceil N't/T' \rceil$ iterations, fix the estimated parameters of the MaGP model and sequentially select runs based on EI by using the fast updating technique illustrated below. Set $n = n + \lceil N't/T' \rceil$ ($\lceil a \rceil$ is the largest integer not exceeding a).
3. Refit the MaGP model (including reestimating all parameters) based on observations $(\mathbf{w}_1, y_1), \ldots, (\mathbf{w}_n, y_n)$. Update the time used for fitting the model as t. Record the time left from the budget as T' and the number of runs left from the budget as N'.
4. Repeat Steps 2 and 3 until the stopping criterion is satisfied.

In Step 2, we adopt fast updating of model fit in $O(n^2)$ computing time given all parameters in the MaGP model. Let $\mathbf{\Phi}_n$ be the covariance of the n current input. The key is to update the model when the $(n + 1)$th data point arrives via fast calculation of the covariance $\mathbf{\Phi}_{n+1}$ and its inverse $\mathbf{\Phi}_{n+1}^{-1}$. Similar to a rank one Sherman–Morrison update (Sherman and Morrison 1950), we have

$$\mathbf{\Phi}_{n+1} = \begin{bmatrix} \mathbf{\Phi}_n & \boldsymbol{\gamma} \\ \boldsymbol{\gamma}^T & \phi(\mathbf{w}_{n+1}, \mathbf{w}_{n+1}) \end{bmatrix},$$

$$\mathbf{\Phi}_{n+1}^{-1} = \begin{bmatrix} \mathbf{\Phi}_n^{-1} + \mathbf{g}\mathbf{g}^T v & \mathbf{g} \\ \mathbf{g}^T & v^{-1} \end{bmatrix},$$

where the covariance function ϕ is defined in (10), the covariance vector $\boldsymbol{\gamma} = (\phi'(\mathbf{w}_{n+1}, \mathbf{w}_i))_{n \times 1}$ with $\phi'(\mathbf{w}_{n+1}, \mathbf{w}_i) = \sum_{h=1}^{k} \sigma_h^2 \exp\{-\theta_h(x_{i,h} - x_{n+1,h})^2 - \sum_{l=1}^{t}(\widetilde{o}_{i,h}^{(l)} - \widetilde{o}_{n+1,h}^{(l)})^2\}$ for $i = 1, \ldots, n$, $v = \phi(\mathbf{w}_{n+1}, \mathbf{w}_{n+1}) - \boldsymbol{\gamma}^T \mathbf{\Phi}^{-1} \boldsymbol{\gamma}$ and $\mathbf{g} = -v^{-1}\mathbf{\Phi}_n^{-1}\boldsymbol{\gamma}$. Here, the update on the covariance inverse requires $O(n^2)$ time, and thus, the updates to all relevant quantities for each model fit are of $O(n^2)$ (Gramacy 2020). The total cost of updating with sequential runs from $n = n_0, \ldots, N$ demands flops in $O(N^3)$. Compared with the general QS-learning that reestimates parameters for every sequential run, the fast QS-learning method reestimates them much less frequently, and thus, saves computations for large experiments. Refer to supplementary materials Section S3.3 for details on the stopping criterion and parameter tuning.

Notably, small-sample performance is often more important and relevant than the convergence rate in experimentation, because only a small number of trials are frequently allowed in practice (Fang, Li, and Sudjianto 2005). Asymptotic guarantees may provide minimal information about the practical effectiveness of the method (Sutton and Barto 2018). The discussion of convergence for learning QS experiments is included in Section S2 of the supplementary materials.

5. Optimal Initial Designs for QS-Learning

Desirable initial designs are important in active learning. They may save the total number of runs and lead to better solutions. In this section, we propose a new class of optimal designs for QS factors, called QS-design, which exhibits space-filling and pair-balanced properties. We first propose a general approach for constructing QS-designs with flexible sizes in Section 5.1, and then provide a deterministic algebraic construction for QS-designs with certain sizes in Section 5.2.

5.1. General Construction

The design for QS factors is denoted as $\boldsymbol{D} = (\boldsymbol{X}, \boldsymbol{O})$ where $\boldsymbol{X}$ is the quantitative part, and $\boldsymbol{O}$ is the sequence part. Both parts use components as columns. To construct a desirable design $\boldsymbol{D}$, we will first construct a good sequence design $\boldsymbol{O}$, and then construct a good quantitative design $\boldsymbol{X}$ in combination with $\boldsymbol{O}$.

Sequence designs have two equivalent representations: one with components as columns (denoted as $\boldsymbol{O}$), and the other with order positions as columns (denoted as $\boldsymbol{O}'$). Designs $\boldsymbol{O}$ consist of runs $\mathbf{o}$ defined in Section 3, and designs $\boldsymbol{O}'$ consist of runs $\boldsymbol{\alpha}$ defined in Section 2. For illustration, the following two designs have the same practical meaning:

$$\boldsymbol{O} = \begin{matrix} & A & B & C \\ & \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 2 & 3 & 1 \end{pmatrix} \end{matrix} \Leftrightarrow \boldsymbol{O}' = \begin{matrix} & 1 & 2 & 3 \\ & \begin{pmatrix} A & B & C \\ B & A & C \\ C & A & B \end{pmatrix} \end{matrix}.$$

To identify the optimal sequence design $\boldsymbol{O}$, we first find the optimal $\boldsymbol{O}'$ by minimizing the following v_p criterion:

$$v_p = \left(\rho_1 \sum_{\substack{i=1 \\ i \neq j}}^{k} \sum_{j=1}^{k} \frac{1}{(t_{i,j} + 1)^p} + \rho_2 \sum_{i=2}^{n} \sum_{j=1}^{i-1} \frac{1}{(h_{i,j} + 1)^p} \right)^{\frac{1}{p}}, \tag{17}$$

where $t_{i,j}$ is the number of appearances of the subsequence "(i j)" in rows of $\boldsymbol{O}'$; $h_{i,j}$ is the Hamming distance between the ith and jth rows in $\boldsymbol{O}'$; and ρ_1, ρ_2, and p are tuning parameters. A design is called pair-balanced if it has the same $t_{i,j}$ value for all subsequences (i.e., pairs) of (i, j), where we use $\{1, 2, \ldots, \}$ to denote the levels $\{A, B, \ldots\}$. A pair-balanced design assigns equal importance to all pairwise interactions among components. It also accounts for different precedence patterns where pairs (i, j) and (j, i) are different. It is similar to the "balance" idea in crossover designs (Dean et al. 2015). To find (near) pair-balanced designs, we propose to maximize designs' minimum $t_{i,j}$ values, which is equivalent to minimizing the term

$\sum \sum 1/(t_{i,j}+1)^p$ in (17) for a sufficiently large tuning parameter p. In practice, $p = 15$ often suffices. In the denominators, we add 1 to $t_{i,j}$ (and $h_{i,j}$) to avoid numerical problems when they are equal to 0.

The term $\sum \sum 1/(h_{i,j}+1)^p$ in (17) considers designs' space-filling properties. Here, we adopt the popular maximin distance criterion (Johnson, Moore, and Ylvisaker 1990), which seeks to scatter design points over the experimental domain such that the minimum pairwise distance between points is maximized. The Hamming distance is used here because the elements in O' are categorical. Analogous to the scalar criterion in Morris and Mitchell (1995), minimizing the term $\sum \sum 1/(h_{i,j} + 1)^p$ is asymptotically equivalent to the maximin Hamming distance criterion as p goes toward infinity, where $p = 15$ often suffices. A space-filling O' benefits the exploration of the response surface and is a robust choice for initial points (Frazier 2018). In this study, we set weights $\rho_1 = 0.2$ and $\rho_2 = 0.8$ in (17) emphasizing more on the design's space-filling property.

Example 2. Consider a drug combination experiment consisting of four drug components. Two designs O'_A and O'_B shown below are compared, and they have the same Hamming distance structure. Their $t_{i,j}$ pairs are listed in Table 2. Given the possibility of synergistic or antagonistic interactions between two drugs, the order in which they are administered is important. For example, if Drugs A and B exhibit a strong synergistic effect, then they should be administered in adjacent order. By contrast, if they exhibit a strong antagonistic effect, then their order of administration should be well separated in time. Furthermore, drugs may have different (i.e., immediate, delayed, or cumulative) time course effects (Al-Sallami et al. 2009), and thus, their precedence patterns matter. For example, consider that Drugs A and B have a synergistic interaction, where the effect of A is immediate, whereas the effect of B appears delayed with respect to the concentration–time profile. In such case, B should be added before A to maximize the synergistic effect, because B requires more time to fully exert its work with A. Evidently, subsequences (A, B) and (B, A) may lead to different outcomes in this study. Considering all of the above, it is clear that the pair-balanced design O'_B is better than O'_A, as all possible adjacent pairs in O'_B appear the same number of times in the experiment (i.e., all $t_{i,j}$ values are equal). Finally, we should use O_B, the equivalent form of O'_B, to be the sequence part of the QS-design D.

$$O'_A = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \end{array} \begin{pmatrix} A & B & C & D \\ B & C & D & A \\ C & D & A & B \\ D & A & B & C \end{pmatrix},$$

$$O'_B = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \end{array} \begin{pmatrix} A & B & C & D \\ B & D & A & C \\ C & A & D & B \\ D & C & B & A \end{pmatrix} \Leftrightarrow O_B = \begin{array}{cccc} A & B & C & D \\ \end{array} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \\ 2 & 4 & 1 & 3 \\ 4 & 3 & 2 & 1 \end{pmatrix}.$$

To search for optimal designs O', we adopt a standard TA algorithm (Dueck and Scheuer 1990; Xiao and Xu 2018) by

Table 2. Comparison of designs' $t_{i,j}$ pairs.

$t_{i,j}$ pairs	AB	AC	AD	BA	BC	BD	CA	CB	CD	DA	DB	DC
O'_A	3	0	0	0	3	0	0	0	3	3	0	0
O'_B	1	1	1	1	1	1	1	1	1	1	1	1

using the criterion ν_p in (17) as the objective function. The algorithm starts with a random design O' and defines its neighbor design $N(O')$ by exchanging two random levels in a random row. It can be implemented with the R package "NMOF" (Gilli, Maringer, and Schumann 2019).

After obtaining the optimal O', we propose to construct the optimal $D' = (X, O')$ minimizing the following C_p criterion, which measures a design's space-filling property.

$$C_p = \left(\sum_{i=2}^{n} \sum_{j=1}^{i-1} \frac{1}{(\rho'_1 d_{i,j} + \rho'_2 h_{i,j} + 1)^p} \right)^{\frac{1}{p}}, \qquad (18)$$

where $d_{i,j} = \sqrt{\sum_{l=1}^{k}(x_{il} - x_{jl})^2}$ is the L_2-distance between the rows $\mathbf{x}_i$, and $\mathbf{x}_j$ in X and $h_{i,j}$ is the Hamming distance between the rows $\mathbf{o}'_i$ and $\mathbf{o}'_j$ in O'. Here, we adopt weights $\rho'_1 = \rho'_2 = 0.5$ and the tuning parameter $p = 15$.

To search for the optimal D' given O', we adopt the same TA algorithm. It starts from a design $D'_c = (X_c, O')$, where X_c is the maximin distance Latin hypercube design (LHD) found by the R package "SLHD" (Ba, Myers, and Brenneman 2015) or "LHD" (Wang, Xiao, and Mandal 2020). Notably, the maximin distance LHD is a popular type of space-filling design, which has been proved to be robust for model misspecification and can minimize the theoretical prediction variance of fitted GP models (Gramacy 2020). Here, the C_p criterion in (18) is used as the objective function, and neighbor designs $N(D'_c)$ are defined by exchanging two randomly chosen rows of X_c. Finally, we convert the optimal D' to its equivalent form, that is, the QS-design D. In practice, we may need to normalize quantitative designs X to $[0, 1]$ range.

5.2. Algebraic Construction Method

We develop an algebraic construction for QS-designs whose component sizes k and run sizes n are $p_r - 1$, where p_r is any odd prime number. Denote the $n \times n$ good lattice point design (Zhou and Xu 2015) as D_{glp}, whose ith row is $\mathbf{h} \times i \mod p_r$, where vector $\mathbf{h} = (1, \dots, n)$ and $i = 1, \dots, n$. Design D_{glp} is a Latin square whose rows and columns are both permutations of $1, \dots, n$. To construct $D' = (X, O')$, we propose to use $O' = D_{glp}$ and X as any column permutation of D_{glp}. For illustration, design O'_b in Example 2 is a $4 \times 4 D_{glp}$, where we treat A as 1, B as 2, and so on.

Theorem 2. Let $n = k = p_r - 1$, where p_r is any odd prime number. Then, the n-run sequence design $O' = D_{glp}$ has the following properties (for any $1 \le i \ne j \le n$):
(i) O' is the maximin Hamming distance design, where all $h_{i,j} = n$;
(ii) O' is the pair-balanced design, where all $t_{i,j} = 1$;

(iii) $\boldsymbol{O}^{'}$ is optimal under the ν_p criterion defined in (17) for any positive weights ρ_1 and ρ_2, and it has a ν_p value of:

$$\nu_p(\boldsymbol{O}^{'}) = \left\{ n(n-1)\left(\frac{\rho_1}{2(n+1)^p} + \frac{\rho_2}{2^p} \right) \right\}^{\frac{1}{p}}. \quad (19)$$

Theorem 3. Let $n = k = p_r - 1$ (p_r is any odd prime number), and $\boldsymbol{D}^{'} = (\boldsymbol{X}, \boldsymbol{O}^{'})$, where $\boldsymbol{O}^{'} = \boldsymbol{D}_{\mathrm{glp}}$ and $\boldsymbol{X}$ is any column permutation of $\boldsymbol{D}_{\mathrm{glp}}$. $\boldsymbol{D}^{'}$ has the following properties:
(i) the minimum row-pairwise L_2-distance in $\boldsymbol{X}$ is $\sqrt{n(n+1)(n+2)/12}$;
(ii) it has a upper bound of C_p as defined in (18), that is, $C_p(\boldsymbol{D}^{'}) \leq n^{\frac{1}{p}} C(\rho_1^{'}, \rho_2^{'}, p)$, where

$$C(\rho_1^{'}, \rho_2^{'}, p) = \left(\frac{n/2 - 1}{\left[\rho_1^{'}\left(\frac{1}{12}n(n+1)(n+2) \right)^{\frac{1}{2}} + n\rho_2^{'} + 1 \right]^p} \right.$$
$$\left. + \frac{1}{2\left[\rho_1^{'}\left(\frac{1}{3}n\left(n^2-1\right) \right)^{\frac{1}{2}} + n\rho_2^{'} + 1 \right]^p} \right)^{\frac{1}{p}}$$

is a constant that only depends on adopted weights $\rho_1^{'}$ and $\rho_2^{'}$ in the C_p criterion.

Corollary 3. Let $n = k = p_r - 1$ (p_r is any odd prime number), and $\boldsymbol{X}$ be any column permutation of $\boldsymbol{D}_{\mathrm{glp}}$. Then, the minimum row-pairwise L_2-distance in $\boldsymbol{X}$, denoted as $d(\boldsymbol{X})$, satisfies

$$\frac{d(\boldsymbol{X})}{d_{\mathrm{upper}}} = \sqrt{\frac{n+2}{2n}} > \frac{\sqrt{2}}{2},$$

where $d_{\mathrm{upper}} = n\sqrt{(n+1)/6}$ is the upper bound of $d(\boldsymbol{X})$.

Theorem 2 shows that the proposed sequence design $\boldsymbol{O}^{'}$ is optimal under both the space-filling and pair-balanced criteria. Theorem 3 shows that the constructed $\boldsymbol{D}^{'} = (\boldsymbol{X}, \boldsymbol{O}^{'})$, or equivalently the QS-design $\boldsymbol{D} = (\boldsymbol{X}, \boldsymbol{O})$, has the best space-filling property, that is, the minimized C_p value. Corollary 3 shows that the proposed quantitative design $\boldsymbol{X}$ also exhibits good space-filing property, because it has a large minimum pairwise L_2-distance. Corollary 3 can easily be obtained on the basis of Theorem 3 in Zhou and Xu (2015) along with the proofs for Theorems 2 and 3 in this work. Notably, the upper bound $d_{\mathrm{upper}} = n\sqrt{(n+1)/6}$ may not be achievable for all design sizes.

Although the minimum run size of an initial design in active learning can be as small as 2, many researchers have recommended using initial designs with moderate sizes (Jones, Schonlau, and Welch 1998; Loeppky, Sacks, and Welch 2009; Frazier 2018). We would remark that the run size of QS-design can be flexibly determined. One recommended run size is the number of parameters in the GP part, for example, $4k - 3$ for 2d-MaGP and $k(k + 3)/2$ for full-MaGP. When a larger number of runs is allowed, we recommend a rule-of-thumb run size of $2+k(k+3)/2$ for any t-dimensional MaGP for simplicity, which is one more than the total number of parameters in full-MaGP.

For the special case, when $k = p_r - 1$ and p_r is any odd prime, we find that the k-run QS-design proposed in this section performs very well, as illustrated in Sections S5 and S6 of supplementary materials.

6. Case Study

Lymphoma is cancer that begins in infection-fighting cells of the immune system, called lymphocytes. When a patient has lymphoma, lymphocytes change and grow out of control. In a recent pioneering work (Wang, Xu, and Ding 2020), the researchers conducted a series of drug experiments on lymphoma treatment. Among them, a 24-run in vitro experiment of three Food and Drug Administration (FDA) approved chemotherapeutics, namely, paclitaxel, doxorubicin, and mitoxantrone (denoted as Drugs A, B, and C, respectively), was included. This experiment considered the doses and sequences of drugs. In this experiment, all six sequences of the three drugs were enumerated. For each sequence, two dose levels for A (Level 0: 2.8 μM; Level 1: 3.75 μM) and B (Level 0: 70 nM; Level 1: 95 nM), and a fixed dose level for C (0.16 μM) were considered. The experiment was performed on Raji cells, a human lymphoma cell line. In any treatment (run), each drug was added every 6 hr in a sequence into the Raji cell culture, and the inhibition percentages (the larger, the better the response) were measured 6 hr after the addition of the last drug. The four largest responses in this experiment are 47.18, 44.87, 44.38, and 44.33.

Here, we run the proposed QS-learning to determine if we can use fewer runs (compared with the original 24 runs) to identify the optimal treatment in this experiment. Notably, 2d-MaGP and full-MaGP are the same for $k = 3$ components. Given that the GP part of the model has eight parameters, we construct an eight-run QS-design to collect the initial data. The proposed QS-learning selects seven sequential runs until the stopping rule is satisfied, that is, the last three EIs are all less than 1% of the current best output. Figure 2 shows the plots for EIs and the cumulative maximum responses of the seven sequential runs. The true maximum response, that is, 47.18, is found, along with the third and fourth largest responses, 44.38 and 44.33, respectively. Notably, the initial QS-design does not include settings that lead to the largest four responses. The proposed QS-learning requires 15 runs (8 initial runs plus 7 follow-up runs) to identify the optimal treatment of the original 24-run experiment, saving 37.5% of the budget. Refer to supplementary materials Section S4.1 for the complete data and more details regarding the analysis.

To further evaluate the proposed QS-learning compared with other approaches, we consider three benchmark methods: the random sampling approach (BM$_1$), sequential generalized PWO approach (BM$_2$), and sequential generalized CP approach (BM$_3$). Here, the BM$_1$ method considers a random sampling of 15 runs out of the original 24 runs, which is the same total run size used above under QS-learning. Proving that the probability of including the optimal treatment in such a random sample is only 62.5% will be straightforward. The BM$_2$ and BM$_3$ methods consider sequential experiments based on the generalized PWO and CP approaches, respectively, introduced in Section 2. Both methods will start from 8 initial runs, and then choose the setting with the optimal prediction to be the next experiment

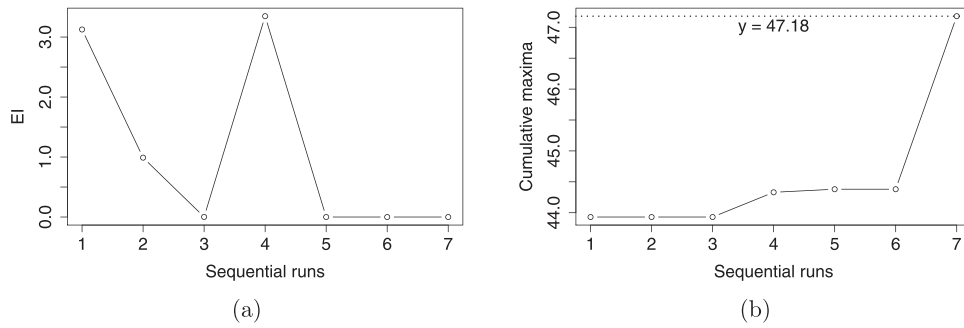**Figure 2.** Plots for (a) EIs and (b) cumulative maximum responses from the QS-learning approach in the case study.
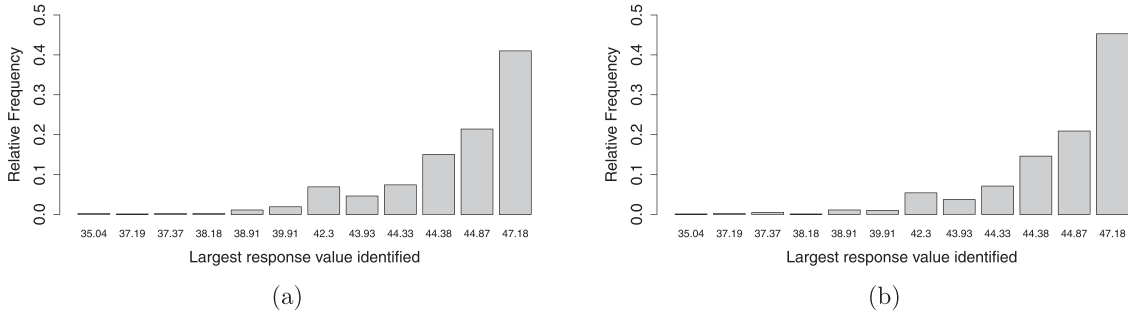

Figure 3. Histograms of the largest response values identified by (a) sequential generalized PWO approach and (b) sequential generalized CP approach in the case study.

trial until no further improvement is achieved. Specifically, when the BM_2 and BM_3 methods start from the QS-design, the largest response identified is only 43.93. In addition, when they start from random designs, we present the plots of their largest responses found from 1000 replications in Figure 3. The probability of successfully identifying the optimal solution (47.18) for BM_2 and BM_3 is less than 50%, and many results are not good in Figure 3.

This 24-run real experiment (Wang, Xu, and Ding 2020) on doses and sequences is a pioneering work in the literature. It serves as a good example to demonstrate the importance of such experiments. Nevertheless, it also has limitations. First, only two dose levels for Drugs *A* and *B* are considered, which does not support the estimation of any curvature effect. In addition, only one dose level for Drug *C* is used, and we cannot estimate its effect. When doses are not restricted to only a few levels and more drugs are included, QS-learning is expected to perform better.

7. Simulation Study

In this section, we evaluate the performance of the proposed QS-learning and its fast variant through a traveling salesman problem (TSP, Applegate et al. (2007)). In Sections S5 and S6 of the supplementary materials, we provide two additional simulations of arranging the four mathematical operations problem (Mee 2020; Yang, Sun, and Xu 2021) and the single machine scheduling problem (Allahverdi, Gupta, and Aldowaisan 1999; Wan and Yuan 2013). These additional results illustrate the advantages of QS-designs (particularly for those from algebraic construction), the superior predictive power of the MaGP model, the difference between 2d-MaGP and full-MaGP, and the general applicability of QS-learning.

TSP is a well-known nonpolynomial-hard problem in combinatorial optimization (Tan et al. 2000; Applegate et al. 2007). Here, we consider a modified TSP that involves the optimization of quantitative input and sequence input. We regard it as a computer experiment wherein the simulator is assumed to be black-box and expensive to evaluate (Fang, Li, and Sudjianto 2005). The cost for evaluating runs is assumed to be considerably higher than that for estimating surrogate models or assessing acquisition functions.

Suppose a salesman needs to travel to *k* cities to sell products, indexed as Cities $1, \ldots, k$. All cities are available for visiting at time zero, and the salesman must visit all cities one by one. The time to travel from City *i* to City *j* $(i \neq j)$ is $s_{i,j}$ days, and $s_{i,j}$ can be different from $s_{j,i}$. The salesman will stay in City *i* for x_i days to sell products. He has a due date to complete the business in City *i*, denoted as d_i. If he misses the due date, then a penalty rate of *f* dollars per day will be charged. After completing his business in each city, he will earn a fixed income of *a* dollars and a variable income of *e* dollars per day when staying in the city. During his entire trip, the expense is *b* dollars per day. In this problem, the target is to identify the optimal traveling schedule that can maximize the profit.

Let us define $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ as the sequence of cities visited, and the corresponding order sequence is $\mathbf{o} = (o_1, \ldots, o_k)$, where City α_i is visited at order $o_{\alpha_i} = i$ $(i = 1, \ldots, k)$. $\alpha_0 = 0$ is defined to be the starting point at time 0. The completion time of the business in City α_i is $C(\mathbf{x}, \alpha_i) = \sum_{l=1}^{i}(s_{\alpha_{l-1}, \alpha_l} + x_{\alpha_l})$, and the tardiness (days passed the due time) for City α_i is $T(\mathbf{x}, \alpha_i) = \max(0, C(\mathbf{x}, \alpha_i) - d_{\alpha_i})$. Thus, the profit function that involves the days-staying-in-cities $\mathbf{x}$ and the sequence-of-cities-visited $\boldsymbol{\alpha}$ (or equivalently $\mathbf{o}$) is

$$F(\mathbf{x}, \boldsymbol{\alpha}) = ka + e \sum_{i=1}^{k} x_i - bC(\mathbf{x}, \alpha_k) - f \sum_{j=1}^{k} T(\mathbf{x}, \alpha_j).$$
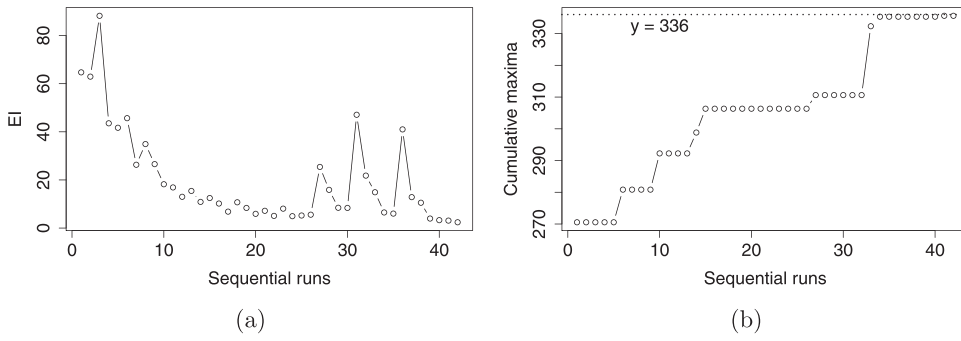
Figure 4. Plots for (a) EIs and (b) cumulative maximum responses from the QS-learning approach under 2d-MaGP in Example 3.
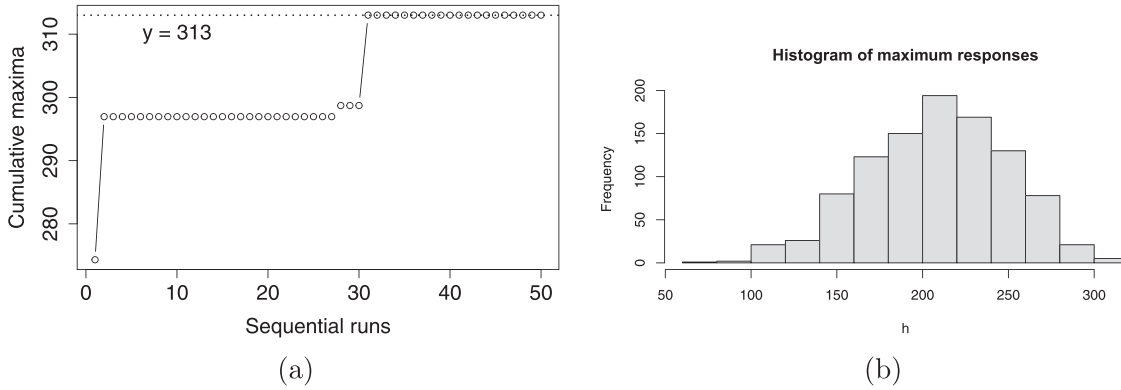

Figure 5. Plots for (a) cumulative maximum responses from the fast QS-learning approach and (b) maximum responses in random samples of 96 trials in Example 3.

Example 3. Consider the above TSP with $k = 8$ cities. Here, we set $a = 20$, $e = 10$, $b = 2$, $f = 15$, due dates $(d_1, \ldots, d_8) = (26, 10, 42, 23, 25, 12, 44, 10)$, and staying days $x_i \in [1, 4]$ for $i = 1, \ldots, 8$. The traveling time $s_{i,j}$ $(i < j)$ is sampled from a uniform distribution $U(0.5, 3)$, and set $s_{j,i} = (1 + 0.1 \times \epsilon_{ji})s_{i,j}$, where ϵ_{ji} is sampled from the standard normal distribution. Refer to Section S4.2 of supplementary materials for additional details of this simulation.

Such a TSP does not have a known analytical solution. We consider the proposed QS-learning to identify the (nearly) optimal setting that will maximize the profit function via a few experimental trials. It starts from the 46-run (the rule-of-thumb run size illustrated in Section 5.2) QS-design and selects 42 sequential runs under the 2d-MaGP model before the stopping criterion is satisfied. The maximum response identified is 336, which is found at the 41th sequential run. The optimal setting includes $\mathbf{x}_* = (1.14, 3.44, 2.48, 2.86, 3.78, 4.00, 3.11, 4.00)$ and $\mathbf{o}_* = (8, 6, 2, 1, 4, 5, 3, 7)$. Figure 4 displays EIs and cumulative maximum responses of sequential runs. After the 10th sequential run, the ordinal parts in all the runs are either $\mathbf{o} = (8, 6, 2, 1, 4, 5, 3, 7)$ or $\mathbf{o} = (8, 6, 2, 1, 4, 5, 7, 3)$, both of which are good candidates. Such an observation indicates the stability of QS-EGO (i.e., Algorithm 1 in Section 4.1). In practice, 2d-MaGP is preferred over full-MaGP when many components are involved, because it is more computationally efficient.

To make comparisons, we first consider the random sampling approach (BM$_1$), which uses a large random sample of 4,032,000 observations, where a random Latin hypercube design is used for the quantitative part and a hundred replicates of all possi-

ble sequences are used for the sequence part. The maximum response found is 325, which is clearly worse than that identified by the QS-learning by using only 88 runs. Next, we consider the sequential generalized PWO (BM$_2$) and CP (BM$_3$) approaches starting from random initial designs with the required sizes (i.e., 37 and 38 runs, respectively). We replicate BM$_2$ and BM$_3$ 1000 times. Their average results are 254 and 264, and their best results are 321 and 324, respectively. Their performances are clearly inferior.

Finally, we evaluate the performance of the fast QS-learning approach introduced in Section 4.2. It starts from the same QS-design as above, and we set a total time budget of 1 hr. It includes 96 runs in total, where the adopted 2d-MaGP model is estimated for only 12 times. Its cumulative maximum responses are reported in Figure 5(a). On average, each sequential run takes about 1 min, while the general QS-learning takes about 4 min in this example. The maximum response found here is 313, which is still much better than the average results of BM$_2$ and BM$_3$ (i.e., 254 and 264, respectively). In addition, we show the histogram of maximum responses found from 1000 random samples of 96 trials in Figure 5(b), where the average value is 208 and the largest value is 309. Evidently, the fast QS-learning approach appears to exhibit reasonably good performance.

8. Discussion

In this work, we propose an active learning approach to identify (nearly) optimal solutions for experiments with QS factors. Analyzing such experiments is challenging due to their

semi-discrete and possibly extremely large solution spaces and complex input–output relationships. The proposed QS-learning includes a novel MaGP surrogate model, an efficient sequential scheme (QS-EGO), and a new class of optimal experimental designs (QS-designs), providing a systematic solution for analyzing QS experiments. The theoretical properties of the proposed method are investigated, and techniques for optimization by using analytical gradients are developed. A case study on lymphoma treatment and several simulations are presented to illustrate the advantages of the proposed method.

In this work, we focus on the widely used EI framework, which works well empirically. In the current literature, the upper confidence bound (UCB) is another popular framework for working with GP models in active learning, particularly for purely discrete input spaces (Srinivas et al. 2012; Djolonga, Krause, and Cevher 2013; Berkenkamp, Schoellig, and Krause 2019; Vakili, Picheny, and Durrande 2020). Results on convergence rates have been established for GP-UCB and its variants. For example, Srinivas et al. (2012) proved a cumulative regret bound of $n^{(v+d(d+1))/(2v+d(d+1))}$ by using a Matern kernel of smoothness v on a d-dimensional space. Vakili, Picheny, and Durrande (2020) further improved the bound to $O(n^{(d-v)/d})$ for $d > v$, $O(\log(n))$ for $d = v$, and some constants for $d < v$. All these results require finite or general compact input spaces, but the space in the QS-experiment (a joint one of continuous and sequence spaces) does not satisfy this requirement. Considering QS-learning under the UCB framework and studying its convergence rate will be interesting topics for future research.

Embedding some of the currently popular non-separable covariance functions (Gneiting 2002) into the MaGP model is also interesting. For example, one may consider

$$\text{cov}(Y(\mathbf{x}_i, \mathbf{o}_i), Y(\mathbf{w}_j, \mathbf{o}_j)) = \frac{\sigma^2}{\psi(||\mathbf{o}_i - \mathbf{o}_j||^2)^{1/d}} \varphi\left(\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\psi(||\mathbf{o}_i - \mathbf{o}_j||^2)}\right),$$

where $||\cdot||$ is the L_2 norm (or other norms), $\varphi(\cdot)$ is a completely monotonic function (e.g., $\varphi(t) = \exp(-ct^\gamma)$), and $\psi(t)$ is a positive function with a completely monotonic derivative (e.g., $\psi(t) = (at^\alpha + 1)^\beta$). Such a structure may have an interpretation for certain choices of φ and ψ (Haslett and Raftery 1989). A planned future study can investigate the appropriate choices for φ and ψ and their parameters to explain QS experiments.

In active learning and other nonsequential learning methods, optimal designs for experiments with QS factors are important but not well addressed. In this work, we propose criteria for QS-designs that consider designs' space-filling and pair-balanced properties. The current literature presents various types of space-filling designs, including maximin distance designs (Johnson, Moore, and Ylvisaker 1990; Xiao and Xu 2017), minimax distance designs (Chen et al. 2015), uniform designs (Fang and Lin 2003), MaxPro designs (Joseph, Gul, and Ba 2015) and uniform projection designs (Sun, Wang, and Xu 2019), which can all be used as the quantitative part in the QS-design. For the sequence part, desirable properties beyond the pair balance can be studied analogous to component orthogonal arrays (Yang, Sun, and Xu 2021), order-of-addition orthogonal arrays (Voelkel 2019) and optimal fractional PWO designs (Peng, Mukerjee, and Lin 2019; Chen, Mukerjee, and Lin 2020). Moreover, some desirable structures that connect

the quantitative and sequence parts of QS-designs can be investigated (Deng, Hung, and Lin 2015).

Funding

Supplementary Materials

Supplementary Materials include: (1) all proofs, (2) details on analytical gradients for parameter estimation and expected improvement optimization, (3) convergence results, (4) implementation details for QS-learning, (5) additional details on numerical studies, and (6) additional simulation studies on arranging four mathematical operations and single machine scheduling problems.

ORCID

Qian Xiao 🄳 https://orcid.org/0000-0001-7869-7109
Yaping Wang 🄳 https://orcid.org/0000-0002-2494-9072
Xinwei Deng 🄳 https://orcid.org/0000-0002-1560-2405

References

Al-Sallami, H. S., Pavan Kumar, V. V., Landersdorfer, C. B., Bulitta, J. B., and Duffull, S. B. (2009), "The Time Course of Drug Effects," *Pharmaceutical Statistics: The Journal of Applied Statistics in the Pharmaceutical Industry*, 8, 176–185. [9]

Allahverdi, A., Gupta, J. N., and Aldowaisan, T. (1999), "A Review of Scheduling Research Involving Setup Considerations," *Omega*, 27, 219–239. [11]

Applegate, D. L., Bixby, R. E., Chvatal, V., and Cook, W. J. (2007), *The Traveling Salesman Problem: A Computational Study*, Princeton, NJ: Princeton University Press. [11]

Ba, S., Myers, W. R., and Brenneman, W. A. (2015), "Optimal Sliced Latin Hypercube Designs," *Technometrics*, 57, 479–487. [9]

Berkenkamp, F., Schoellig, A. P., and Krause, A. (2019), "No-Regret Bayesian Optimization with Unknown Hyperparameters," *Journal of Machine Learning Research*, 20, 1–24. [13]

Bharimalla, A. K., Deshmukh, S. P., Patil, P. G., Vigneshwaran, N. (2015), "Energy Efficient Manufacturing of Nanocellulose by Chemo-and Bio-Mechanical Processes: A Review," *World Journal of Nano Science and Engineering*, 5, 204–212. [1]

Burnaev, E., and Panov, M. (2015), "Adaptive Design of Experiments based on Gaussian Processes," in *International Symposium on Statistical Learning and Data Sciences*, pp. 116–125, Springer. [2,3]

Chen, J., Mukerjee, R., and Lin, D. K. J. (2020), "Construction of Optimal Fractional Order-of-Addition Designs via Block Designs," *Statistics & Probability Letters*, 161, 108728. [13]

Chen, R.-B., Chang, S.-P., Wang, W., Tung, H.-C., and Wong, W. K. (2015), "Minimax Optimal Designs via Particle Swarm Optimization Methods," *Statistics and Computing*, 25, 975–988. [13]

Chou, T.-C. (2006), "Theoretical Basis, Experimental Design, and Computerized Simulation of Synergism and Antagonism in Drug Combination Studies," *Pharmacological Reviews*, 58, 621–681. [2]

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996), "Active Learning with Statistical Models," *Journal of Artificial Intelligence Research*, 4, 129–145. [2]

Dean, A., Morris, M., Stufken, J., and Bingham, D. (2015), *Handbook of Design and Analysis of Experiments* (Vol. 7), Boca Raton, FL: CRC Press. [8]

Deng, X., Hung, Y., and Lin, C. D. (2015), "Design for Computer Experiments with Qualitative and Quantitative Factors," *Statistica Sinica*, 25, 1567–1581. [13]

Deng, X., Joseph, V. R., Sudjianto, A., and Wu, C. J. (2009), "Active Learning through Sequential Design, with Applications to Detection of Money Laundering," *Journal of the American Statistical Association*, 104, 969–981. [2]

Deng, X., Lin, C. D., Liu, K.-W., and Rowe, R. (2017), "Additive Gaussian Process for Computer Models with Qualitative and Quantitative Factors," *Technometrics*, 59, 283–292. [2]

Ding, X., Matsuo, K., Xu, L., Yang, J., and Zheng, L. (2015), "Optimized Combinations of Bortezomib, Camptothecin, and Doxorubicin show Increased Efficacy and Reduced Toxicity in Treating Oral Cancer," *Anti-Cancer Drugs*, 26, 547–554. [1,2]

Djolonga, J., Krause, A., and Cevher, V. (2013), "High-Dimensional Gaussian Process Bandits," in *Advances in Neural Information Processing Systems*, eds. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (Vol. 26), Curran Associates, Inc. [13]

Dueck, G., and Scheuer, T. (1990), "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," *Journal of Computational Physics*, 90, 161–175. [7,9]

Erickson, C. B., Ankenman, B. E., and Sanchez, S. M. (2018), "Comparison of Gaussian Process Modeling Software," *European Journal of Operational Research*, 266, 179–192. [5]

Fang, K.-T., Li, R., and Sudjianto, A. (2005), *Design and Modeling for Computer Experiments*, Boca Raton, FL: Chapman and Hall/CRC. [4,8,11]

Fang, K.-T., and Lin, D. K. J. (2003), "Uniform Experimental Designs and their Applications in Industry," in *Handbook of Statistics* (Vol. 22), eds. F. Nielsen, A. S. R., Srinivasa Rao, and C. R. Rao, pp. 131–170, Amsterdam: North-Holland. [13]

Frazier, P., Powell, W., and Dayanik, S. (2009), "The Knowledge-Gradient Policy for Correlated Normal Beliefs," *INFORMS Journal on Computing*, 21, 599–613. [2]

Frazier, P. I. (2018), "Bayesian Optimization," in *Recent Advances in Optimization and Modeling of Contemporary Problems*, pp. 255–278, Catonsville, MD: Informs. [2,3,8,9,10]

Gilli, M., Maringer, D., and Schumann, E. (2019), *Numerical Methods and Optimization in Finance*, Amsterdam: Academic Press. [9]

Gneiting, T. (2002), "Nonseparable, Stationary Covariance Functions for Space–Time Data," *Journal of the American Statistical Association*, 97, 590–600. [5,13]

Gramacy, R. B. (2020), *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*, Boca Raton, FL: Chapman and Hall/CRC. [6,8,9]

Gramacy, R. B., and Lee, H. K. (2012), "Cases for the Nugget in Modeling Computer Experiments," *Statistics and Computing*, 22, 713–722. [6]

Haslett, J., and Raftery, A. E. (1989), "Space-Time Modelling with Long-Memory Dependence: Assessing Ireland's Wind Power Resource," *Journal of the Royal Statistical Society*, Series C, 38, 1–21. [13]

Hedayat, A. S., Sloane, N. J. A., and Stufken, J. (1999), *Orthogonal Arrays: Theory and Applications*, New York: Springer. [5]

Hennig, P., and Schuler, C. J. (2012), "Entropy Search for Information-Efficient Global Optimization," *Journal of Machine Learning Research*, 13, 1809–1837. [2]

Jaynes, J., Ding, X., Xu, H., Wong, W. K., and Ho, C.-M. (2013), "Application of Fractional Factorial Designs to Study Drug Combinations," *Statistics in Medicine*, 32, 307–318. [2]

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148. [7,9,13]

Jones, B., and Kenward, M. G. (2014), *Design and Analysis of Cross-over Trials* (3rd ed.), Boca Raton, FL: CRC Press. [1]

Jones, D. R., Schonlau, M., and Welch, W. J. (1998), "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, 13, 455–492. [2,6,10]

Joseph, V. R. (2016), "Space-Filling Designs for Computer Experiments: A Review," *Quality Engineering*, 28, 28–35. [12]

Joseph, V. R., Gul, E., and Ba, S. (2015), "Maximum Projection Designs for Computer Experiments," *Biometrika*, 102, 371–380. [13]

Jourdain, L. S., Schmitt, C., Leser, M. E., Murray, B. S., and Dickinson, E. (2009), "Mixed Layers of Sodium Caseinate+ Dextran Sulfate: Influence of Order of Addition to Oil- Water Interface," *Langmuir*, 25, 10026–10037. [1]

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. (2007), "Active Learning with Gaussian Processes for Object Categorization," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE. [2,3]

Kleijnen, J. P. (2009), "Kriging Metamodeling in Simulation: A Review," *European Journal of Operational Research*, 192, 707–716. [2,6]

Lewis, D. D., and Gale, W. A. (1994), "A Sequential Algorithm for Training Text Classifiers," in *SIGIR'94*, pp. 3–12, Springer. [2]

Lin, C. D., and Tang, B. (2015), "Latin Hypercubes and Space-Filling Designs," in *Handbook of Design and Analysis of Experiments*, eds. A. Dean, M. Morris, J. Stufken, and D. Bingham, 593–625, Boca Raton, FL: CRC Press. [5]

Lin, D. K. J., and Peng, J. (2019), "Order-of-Addition Experiments: A Review and Some New Thoughts," *Quality Engineering*, 31, 49–59. [2]

Liu, D. C., and Nocedal, J. (1989), "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, 45, 503–528. [6]

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009), "Choosing the Sample Size of a Computer Experiment: A Practical Guide," *Technometrics*, 51, 366–376. [10]

Mebane, Jr., W. R., and Sekhon, J. S. (2011), "Genetic Optimization using Derivatives: The rgenoud package for R," *Journal of Statistical Software*, 42, 1–26. [7]

Mee, R. W. (2020), "Order-of-Addition Modeling," *Statistica Sinica*, 30, 1543–1559. [2,3,11]

Mehdad, E., and Kleijnen, J. P. (2015), "Classic Kriging versus Kriging with Bootstrapping or Conditional Simulation: Classic Kriging's Robust Confidence Intervals and Optimization," *Journal of the Operational Research Society*, 66, 1804–1814. [6]

Morris, M. D., and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402. [9]

Ning, S., Xu, H., Al-Shyoukh, I., Feng, J., and Sun, R. (2014), "An Application of a Hill-based Response Surface Model for a Drug Combination Experiment on Lung Cancer," *Statistics in Medicine*, 33, 4227–4236. [2]

Panwalkar, S., Dudek, R., and Smith, M. (1973), "Sequencing Research and the Industrial Scheduling Problem," in *Symposium on the Theory of Scheduling and its Applications*, pp. 29–38, Springer. [1]

Peng, J., Mukerjee, R., and Lin, D. K. J. (2019), "Design of Order-of-Addition Experiments," *Biometrika*, 106, 683–694. [13]

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008), "Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors," *Technometrics*, 50, 383–396. [2]

Rasmussen, C. E., and Williams, C. K. (2006), *Gaussian Processes for Machine Learning*, Cambridge, MA: MIT Press. [2,3,6]

Roustant, O., Ginsbourger, D., and Deville, Y. (2012), "Dicekriging, Diceoptim: Two R Packages for the Analysis of Computer Experiments by Kriging-based Metamodelling and Optimization," *Journal of Statistical Software*, 51, 1–54. [6]

Settles, B. (2009), "Active Learning Literature Survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison. [2]

Settles, B., Craven, M., and Ray, S. (2007), "Multiple-Instance Active Learning," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1289–1296. [2]

Seung, H. S., Opper, M., and Sompolinsky, H. (1992), "Query by Committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294. [2]

Sherman, J., and Morrison, W. J. (1950), "Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix," *The Annals of Mathematical Statistics*, 21, 124–127. [8]

Shinohara, A., and Ogawa, T. (1998), "Stimulation by Rad52 of Yeast Rad51-Mediated Recombination," *Nature*, 391, 404–407. [1]

Smith, R. C. (2013), *Uncertainty Quantification: Theory, Implementation, and Applications* (Vol. 12), Philadelphia, PA: SIAM. [3]

Snelson, E., Ghahramani, Z., and Rasmussen, C. E. (2004), "Warped Gaussian Processes," in *Advances in Neural Information Processing Systems*, pp. 337–344. [5]

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. (2012), "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting," *IEEE Transactions on Information Theory*, 58, 3250–3265. [13]

Sun, F., Wang, Y., and Xu, H. (2019), "Uniform Projection Designs," *Annals of Statistics*, 47, 641–661. [13]

Sutton, R. S., and Barto, A. G. (2018), *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press. [8]

Tan, K.-C., Narasimhan, R., Rubin, P. A., and Ragatz, G. L. (2000), "A Comparison of Four Methods for Minimizing Total Tardiness on a Single Processor with Sequence Dependent Setup Times," *Omega*, 28, 313–326. [11]

Vakili, S., Picheny, V., and Durrande, N. (2020), "Regret Bounds for Noise-Free Bayesian Optimization," arXiv preprint arXiv:2002.05096. [13]

Van Nostrand, R. (1995), "Design of Experiments Where the Order of Addition is Important," in *ASA Proceedings of the Section on Physical and Engineering Sciences*, pp. 155–160. [2,3]

Voelkel, J. G. (2019), "The Design of Order-of-Addition Experiments," *Journal of Quality Technology*, 51, 230–241. [2,3,13]

Wan, L., and Yuan, J. (2013), "Single-Machine Scheduling to Minimize the Total Earliness and Tardiness is Strongly NP-Hard," *Operations Research Letters*, 41, 363–365. [11]

Wang, A., Xu, H., and Ding, X. (2020), "Simultaneous Optimization of Drug Combination Dose-Ratio Sequence with Innovative Design and Active Learning," *Advanced Therapeutics*, 3, 1900135. [1,2,10,11]

Wang, H., Xiao, Q., and Mandal, A. (2020), *LHD: Latin Hypercube Designs (LHDs) Algorithms*, R package version 1.3.3. [9]

Wang, L., Xiao, Q., and Xu, H. (2018), "Optimal Maximin L_1-Distance Latin Hypercube Designs based on Good Lattice Point Designs," *The Annals of Statistics*, 46, 3741–3766. [5]

Wu, C. J., and Hamada, M. S. (2021), *Experiments: Planning, Analysis, and Optimization* (3rd ed.), Hoboken, NJ: Wiley. [2]

Xiao, Q., Mandal, A., Lin, C. D., and Deng, X. (2021), "EzGP: Easy-to-Interpret Gaussian Process Models for Computer Experiments with Both Quantitative and Qualitative Factors," *SIAM/ASA Journal on Uncertainty Quantification*, 9, 333–353. [2]

Xiao, Q., Wang, L., and Xu, H. (2019), "Application of Kriging Models for a Drug Combination Experiment on Lung Cancer," *Statistics in Medicine*, 38, 236–246. [2,6]

Xiao, Q., and Xu, H. (2017), "Construction of Maximin Distance Latin Squares and Related Latin Hypercube Designs," *Biometrika*, 104, 455–464. [13]

——— (2018), "Construction of Maximin Distance Designs via Level Permutation and Expansion," *Statistica Sinica*, 28, 1395–1414. [9]

——— (2021), "A Mapping-based Universal Kriging Model for Order-of-Addition Experiments in Drug Combination Studies," *Computational Statistics & Data Analysis*, 157, 107155. [5]

Yang, J.-F., Sun, F., and Xu, H. (2021), "A Component-Position Model, Analysis and Design for Order-of-Addition Experiments," *Technometrics*, 63, 212–224. [2,3,11,13]

Zhang, Y., Tao, S., Chen, W., and Apley, D. W. (2019), "A Latent Variable Approach to Gaussian Process Modeling with Qualitative and Quantitative Factors," *Technometrics*, 62, 291–302. [2,5]

Zhou, Y., and Xu, H. (2015), "Space-Filling Properties of Good Lattice Point Sets," *Biometrika*, 102, 959–966. [9,10]