Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

QUANTILE SIMULATION OPTIMIZATION WITH STOCHASTIC CO-KRIGING MODEL

Songhao Wang
Szu Hui Ng
William Benjamin Haskell

Department of Industrial Systems Engineering and Management
National University of Singapore
10 Kent Ridge Crescent
Singapore 119260, SINGAPORE

ABSTRACT

Although the mean is a widely-used performance measure for stochastic simulations, the quantiles have become very attractive to measure the variability and the risk of the simulated systems. Optimizing high quantiles, which is typically of interest for decision makers, can be challenging with limited budget as it may cost a large number of simulation runs to obtain estimators for high quantiles with acceptable accuracy. This restricts the application of some traditional Monte Carlo approaches. In this work, we propose a multi-level metamodel (co-kriging) based algorithm to optimize high quantiles more efficiently. Utilizing non-decreasing properties of quantile functions, we first search on some cheaper and informative lower quantiles, and then search the higher quantiles with a focus on the promising regions identified by lower quantiles. This results in less wastage of budget in non-promising regions. A numerical study is presented to show the power of this algorithm.

1 INTRODUCTION

Nowadays, quantile has become an increasingly popular measurement in many fields, including finance, engineering safety, and healthcare, for its capability to provide profiles of the entire output distribution. In risk management, quantile, also termed as Value-at-Risk (VaR), is one of the primary risk measures to quantify the system risk. For instance, in the finance industry, the α-quantile of a loss distribution L represents the lower bound of large losses that the investor can suffer from an activity, where the large losses are defined to be the $(1-\alpha)$-tail of L with α very close to 1 (Hong and Liu 2009).

Optimizing quantiles of random functions is a common practice for decision makers to manage the risk. In this case, searching for x in the decision space $\mathscr{X}$ with the smallest α-quantile of $L(x)$ returns the desired decision. This optimization problem can be challenging for several reasons. First, the loss function $L(x)$ usually has no closed-form and is difficult or expensive to observe from the real system. In many of these cases, some computer models are built to simulate the system behaviors, such as the financial model for risk management. Numerous Monte Carlo methods based on the simulation results have been developed to optimize the quantile (see Hong et al. 2014 for a review). These simulation models, however, can be very complicated and time-consuming to conduct due to the complex nature of the real system. This restricts the number of simulation runs that can be conducted, making it impossible to obtain the results for every considered design. Second, the quantile optimization problems may be non-convex and thus difficult to solve. Third, higher quantiles (close to 1, which are typically of interest) are costly to observe, as it may require a large number of simulation runs to return a quantile estimator at this level with satisfactory precision.

To tackle the first two challenges, we aim to develop a metamodel-based simulation optimization algorithm for quantile measures from black box functions with no convexity restrictions. A metamodel

is a statistical model providing an approximation of the true response surface. Developing metamodels for quantile has been extensively studied (Koenker 2005; Dabo-Niang and Thiam 2010; Chen 2009), of which the quantile regression (QR) (Koenker 2005) is the most widely used model. Recently, the stochastic Gaussian process (GP, also termed as kriging) model (Ankenman et al. 2010; Yin et al. 2011) has been used for quantile metamodeling and shown competitive performance compared with quantile regression model (Chen and Kim 2016). In this work, we also adopt this GP based model.

When optimizing black-box functions, metamodels can serve as fast surrogates of the true baseline functions and guide the search. Based on the GP model, a few optimization approaches has been proposed for deterministic simulations, of which the Efficient Global Optimization (EGO) (Jones et al. 1998) algorithm with Expected Improvement (EI) criterion is the most widely used for its capability to balance between exploration (searching unexplored region) and exploitation (searching the current promising region). For stochastic simulations, some recent works including Two-Stage Sequential Optimization (TSSO) (Quan et al. 2013) tried to combine EI criterion with Optimal Computing Budget Allocation (OCBA) (Chen et al. 2000) technique to iteratively select new possible optimal inputs and reduce the noises of the observations. These algorithms, however, are designed to optimize the mean of the simulation outputs. A similar optimization algorithm for quantiles is proposed in this work.

To address the third challenge, we propose a novel multi-level quantile optimization algorithm, which is the main contribution of this work. Different with traditional approaches which assume only a single quantile level, a multi-level model can leverage on informative lower quantiles to identify the optimum more efficiently. Typically, the lower quantiles are cheaper and easier to estimate and their estimations are likely to be less noisy compared with that of a high quantile (Bahadur 1966). To explain our intuitions more formally, let $0 < \alpha_1 < \alpha_2 < 1$ and $v_{\alpha_1}(L(x)), v_{\alpha_2}(L(x))$ be the two quantiles of which $v_{\alpha_2}(L(x))$ is to be minimized. The idea behind our approach is to first optimize the more accurate $v_{\alpha_1}(L(x))$ to identify promising regions in $v_{\alpha_1}(L(x))$. Based on the non-decreasing properties of quantiles, this helps us identify places to focus sampling on and fit the higher quantile model $v_{\alpha_2}(L(x))$ in the second stage to identify the optimum on $v_{\alpha_2}(L(x))$. If we directly start with optimizing $v_{\alpha_2}(L(x))$, its metamodel can be very inaccurate with limited budget and can mislead the search, resulting in much wasted budget in non-promising regions. Through our two-stage algorithm, we hope to use only a small budget to quickly identify possible promising regions and allocate more budget there. It is noteworthy that those promising and non-promising regions determined by $v_{\alpha_1}(L(x))$ can be informative for $v_{\alpha_2}(L(x))$ for a few reasons. First, the two quantile functions are likely to be correlated since they come from the same distribution $L(x)$ (Wang and Ng 2017). In this case, the shape of the two surfaces can be similar and hence, similar promising regions. Second, since $v_{\alpha_2}(L(x)) > v_{\alpha_1}(L(x))$, if $v_{\alpha_1}(L(x_0))$ is very large for x_0 in some region, we can conclude that the $v_{\alpha_2}(L(x_0))$ will only be even larger. Particularly, when $v_{\alpha_1}(L(x_0))$ is larger than $v_{\alpha_2}(L(x_1))$, it is obvious that x_0 can not be optimum for v_{α_2} and there seems no need to allocate further replications to this region. For ease of exposition, the illustration above only considers one lower quantile function. In the proposed algorithm, a multi-level metamodel can be built with more than one lower quantile to fully benefit from the method.

More formally, in this paper, we consider the problem where we want to minimize the α_m-quantile, $v_{\alpha_m}(L(x))$, for loss function $L(x)$, where $x \in \mathscr{X} \subset \mathbb{R}_d$ ($\mathscr{X}$ is a compact set):

$$\min_{x \in \mathscr{X}} \quad v_{\alpha_m}(L(x)).$$

To solve this, we propose an algorithm, that leverages on $m-1$ lower quantiles with $0 < \alpha_1 < \alpha_2 < ... < \alpha_m < 1$ to quickly narrow down the promising regions. To develop this, we first propose a stochastic co-kriging model to jointly model the m quantiles and then generalize the TSSO algorithm for multi-level quantile optimization. In the modeling of m quantiles, we extend the standard deterministic co-kriging model (Kennedy and O'Hagan 2000) and apply it to quantile prediction. This differs from the work of Chen et al. (2017) which models the mean. In addition, to avoid the crossing problem in quantile modeling (Koenker 1984; Cole 1988; He 1997; Liu and Wu 2009), we further propose a new penalized GP model applied to stochastic co-kriging model to penalize the model parameters that cause crossing problem.

The rest of this article is organized as follows. In section 2, we review some basics for stochastic co-kriging model and extend it to our non-crossing multi-level quantile metamodeling. In section 3, we propose our optimization algorithm. In section 4, a numerical example is provided to show the power of the method. In section 5, we summarize the work and present some future work.

2 STOCHASTIC CO-KRIGING MODEL FOR QUANTILES

To jointly model these m quantiles, we propose to use the co-kriging model. It was originally developed to model deterministic multi-fidelity problems (where a response can be observed with different fidelities) (Kennedy and O'Hagan 2000; Higdon et al. 2004; Le Gratiet and Cannamela 2015) and has recently been extended to stochastic simulation metamodeling (Chen et al. 2017) for mean performance measures. In this section, we generalize the stochastic co-kriging model for predicting means to predicting quantiles in Section 2.1 and then propose a penalized maximum likelihood estimation (PMLE) approach to prevent the possible crossing problem between different quantile curves in Section 2.2.

Here, we first briefly introduce the notations used and the simulation background. To develop a stochastic kriging model, replicates of the experiments are required. That is, at each design input x, a few simulation runs are required. Throughout this work, we use L to represent the results of these original simulation runs. For instance, where there are n simulations at x, we observe n results: $L_1(x),...,L_n(x)$. With these simulation results, a point estimate (denoted by $\mathscr{Y}$) for the response of interest, which is the α-quantile in this case, can be obtained: $\mathscr{Y}(x) = L_{\llcorner \alpha n \lrcorner}(x)$, where $L_{\llcorner i \lrcorner}(x)$ is the i-th order statistic for $L_1(x),...,L_n(x)$. Due to a limited number of simulation runs that can be conducted, this $\mathscr{Y}(x)$ is a noisy observation for the true response. We further use Y and Z to represent the noisy and noise-free versions of this response and use ε to represent the noise between them ($Y = Z + \varepsilon$). $\mathscr{Y}$ taken at all design inputs can then be used to develop predictive models for Z.

2.1 Stochastic Co-kriging Model Basics and Extensions for Quantiles Predictions

In the standard stochastic co-kriging model, which is designed for means, models at different levels satisfy the following relations:

$$Y_l(x) = Z_l(x) + \varepsilon_l(x) = \rho_{l-1} Z_{l-1}(x) + \delta_l(x) + \varepsilon_l(x), \quad \text{if} \quad 1 < l \leq m,$$

$$Y_l(x) = Z_l(x) + \varepsilon_l(x) = \delta_l(x) + \varepsilon_l(x), \quad \text{if} \quad l = 1,$$

where Y_l and Z_l represent the noisy and noise-free responses at level l, respectively. $\delta_l(x)$ ($l = 1,...,m$) are m independent second-order stationary GPs of mean $f_l^T(x)\beta_l$ and covariance function: $\text{cov}(\delta_l(x_1), \delta_l(x_2)) = \sigma_l^2 \text{corr}_l(x_1, x_2)$, where $\sigma_l^2 = \text{var}(\delta_l(x))$. Here $f_l(x)$ is a $p_l \times 1$ vector of known functions and β_l is a vector of model parameters. For the correlation function, we adopt the popular Gaussian function: $\text{corr}_l(x_1, x_2) = \exp\left\{\sum_{j=1}^{d} \theta_{l,j}(x_{1,j} - x_{2,j})^2\right\}$ for illustration, where $x_{i,j}$ is the jth coordinate of x_i and $\theta_l = (\theta_{l,1},...,\theta_{l,d})$ is the sensitivity parameter. ε_l, $l = 1,...,m$, are m random noises which follow a m dimensional normal distribution with zero mean. These noises are assumed to be independent of δ_l. It is clear that in this model, Z_l is represented by a scaled Z_{l-1} term, $\rho_{l-1} Z_{l-1}$ (ρ_{l-1} can be treated as regression parameter), plus a difference term. This type of autoregressive model is first introduced by Kennedy and O'Hagan (2000) for deterministic multi-fidelity problems.

Denote $\mathscr{Y}_l(x)$ as the observation (point estimate) for the lth level at x and the design set for Y_l as D_l with $|D_l|$ representing number of inputs in D_l such that for any input $x \in D_l$, the observation $\mathscr{Y}_l(x)$ is available. To fit the co-kriging model, we need observations for all levels: $\mathscr{Y}_1,...,\mathscr{Y}_m$, where $\mathscr{Y}_l$ is a vector of the observations for lth level taken at all inputs in D_l. For efficient parameter estimation, a nested property of the design set: $D_m \subset D_{m-1} \subset ... \subset D_1$ is often assumed in co-kriging metamodeling for multi-fidelity problems (Kennedy and O'Hagan 2000).

To simplify the notation, we define:

$$P_i^j = \prod_{k=i}^{j} \rho_k, \quad P_i^{i-1} = 1.$$

And use $A_k(D_i, D_j)$ to represent the correlation of points in D_i and D_j generated by δ_k with:

$$A_k(D_i, D_j)_{pq} = \text{corr}_k(x_p^i, x_q^j),$$

where x_p^i is the pth design point in D_i, x_q^j is the qth design point in D_j. The predictor and the corresponding predictive variance of $Z_l(x)$ can be derived as:

$$\widehat{Z}_l(x) = h_l(x)^T \widehat{\beta} + t_l(x)^T R^{-1}(\mathscr{Y} - H\widehat{\beta}), \tag{1}$$

$$\text{var}(\widehat{Z}_l(x)) = \text{var}(Z_l(x)) - t_l(x)^T R^{-1} t_l(x) + \zeta_l(x)^T (H^T R^{-1} H)^{-1} \zeta_l(x), \tag{2}$$

where $\zeta_l(x) = h_l(x)^T - t_l(x)^T R^{-1} H$. $h_l(x)^T = (P_1^{l-1} f_1(x)^T, P_2^{l-1} f_2(x)^T, ..., P_{l-1}^{l-1} f_{l-1}(x)^T, f_l(x)^T, \mathbf{0}^T_{p_{l+1}+...+p_m})$. $\widehat{\beta} = (H^T R^{-1} H)^{-1} H^T R^{-1} \mathscr{Y}$ is the generalized least squares estimator for β. $\text{var}(Z_l(x)) = \sigma_l^2 + \sum_{j=1}^{l-1} (P_j^{l-1})^2 \sigma_j^2$. $t_l(x)^T = (t_{l,1}(x)^T, ..., t_{l,m}(x)^T)$ with:

$$t_{l,s}(x) = \sum_{j=1}^{q} \sigma_j^2 P_j^{s-1} P_j^{l-1} A_j(D_s, x), \quad \text{where} \quad q = \min\{s, l\}.$$

$R = R_z + R_\varepsilon$, where R_z, R_ε are symmetric matrices with $m \times m$ blocks:

$$R_z^{(k,s)} = \sum_{j=1}^{q} \sigma_j^2 P_j^{k-1} P_j^{s-1} A_j(D_k, D_s), \quad \text{where} \quad q = \min\{k, s\}.$$

$$R_\varepsilon^{(k,s)} = \begin{pmatrix} \text{diag}(\text{cov}(\varepsilon_k(x_1^s), \varepsilon_s(x_1^s)), ..., \text{cov}(\varepsilon_k(x_{|D_s|}^s), \varepsilon_s(x_{|D_s|}^s))) \\ \mathbf{0}_{(|D_k|-|D_s|)\times|D_s|} \end{pmatrix}, \quad \text{if} \quad k \le s.$$

$\mathscr{Y}^T = (\mathscr{Y}_1(D_1)^T, ..., \mathscr{Y}_m(D_m)^T)$. H has $m \times m$ blocks with:

$$H^{(k,s)} = P_s^{k-1} f_s(D_k)^T, \quad \text{if} \quad k \ge s, \quad H^{(k,s)} = \mathbf{0}_{|D_k|\times p_s}, \quad \text{if} \quad k < s.$$

The above results assume known hyperparamters $\rho_l, \theta_k, \sigma_k^2$, $l = 1, ..., m-1, k = 1, ..., m$, and the covariance matrix for noises, R_ε. To better illustrate the procedure to build the model, we separate the parameters of the model into two categories: model inputs and model parameters. The model inputs include the observations vector $\mathscr{Y}$ and the estimators for the associated noise covariances matrix R_ε. They are directly drawn from the initial simulation results and serve as inputs to the model. The remaining parameters ($\rho = (\rho_1, ..., \rho_{m-1}), \theta = (\theta_1, ..., \theta_m), \sigma^2 = (\sigma_1^2, ..., \sigma_m^2)$) are referred to as model parameters and can be estimated by maximizing the loglikelihood for observation vectors $\mathscr{Y}$. After this, the predictor (1) and predictive variance (2) can be obtained by plugging in the estimated model inputs and parameters. For standard stochastic co-kriging model, the model inputs are just the sample means and sample covariance for the mean estimates (see details in Chen et al. 2017).

To extend this model to our multi-level quantile prediction, we first verify the nested property of the design sets. For any $x \in D_j$ ($\mathscr{Y}_j(x)$ has acceptable precision), the quantile estimators for $\alpha_1, ..., \alpha_{j-1}$ can be naturally obtained from $L_1(x), ..., L_n(x)$ with typically smaller noises. Therefore, $x \in D_l$, $l = 1, ..., j-1$ and the nested property follows. For simplicity, suppose the inputs in design sets have the following order: the first $|D_l|$ design points in D_{l-1} are exactly similar with the points in D_l with similar sequence.

In addition, the observation vector, $\mathscr{Y}$, and associated noise variance estimates, R_ε, should be changed accordingly. Particularly, for the observation $\mathscr{Y}(x_p^i)$, as noted at the beginning of Section 2, we use the sample α_i-quantile at x_p^i (Chen and Kim 2016; Wang and Ng 2017). To estimate the noise covariances, we propose to use a similar sectioning method as Wang and Ng (2017), where they estimated the covariance of the noises for the mean estimator and quantile estimator. Specifically, to estimate the covariance of $\varepsilon_p(x)$ and $\varepsilon_q(x)$, we first divide the simulation results $L_1(x),...,L_n(x)$ into n_b non-overlapping group with n_c replications in each group. Then, the sample α_p-quantile, $\mathscr{Y}_{p,j}(x)$, and sample α_q-quantile, $\mathscr{Y}_{q,j}(x)$ within each group are obtained ($j = 1,...,n_b$). The noise variance estimator is then computed as:

$$\widehat{\text{cov}}(\varepsilon_p(x),\varepsilon_q(x)) = \frac{1}{n_b(n_b-1)}\sum_{j=1}^{n_b}(\mathscr{Y}_p(x)-\mathscr{Y}_{p,j}(x))(\mathscr{Y}_q(x)-\mathscr{Y}_{q,j}(x)).$$

2.2 Preventing the Crossing Problem in Stochastic Quantile Co-kriging Model

Traditional QR models quantile functions at different levels separately, and this can result in possible crossing between different quantile predictive curves. This, for example, will cause the predictive value of 0.95 quantile at some points to be larger than that of 0.99 quantile. This crossing is a widely acknowledged problem in quantile modeling, and can lead to invalid distribution of the response and problematic inferences. For our quantile co-kriging model, preventing crossing to ensure monotonicity not only improves inferences, but more importantly ensures that the multi-level search in the optimization algorithm is valid and efficient. Imagine if the crossing happens between two quantile models, the non-promising region identified by lower quantile model can be misleading, since the higher quantile can be smaller than the lower one and hence can have promising (and even optimal) values in those non-promising regions. Therefore, it is vital to ensure non-crossing in the models before the optimization process. Although in the co-kriging model, the multiple quantiles are modeled jointly, non-crossing is not guaranteed. Note that in the traditional application of the co-kriging model where deterministic or mean responses have been typically modeled, the crossing of the models is not a problem.

In this section, we propose a new penalized version of stochastic co-kriging model to prevent the crossing problem for quantile models. In our multi-level quantile problem, preventing crossing problem is to ensure that:

$$\widehat{Z}_{l+1}(x) - \widehat{Z}_l(x) \geq 0, \quad \forall x \in \mathscr{X}, l = 1,...,m-1.$$

In other words, the difference of the predictive curves for any two successive quantiles should be non-negative across the design space. We start this section by proposing a penalized GP model that can ensure non-negative predictions for single deterministic response (which can be considered as the difference between two quantiles) and then apply it to our multi-level model.

Consider first a deterministic GP model to model a non-negative function W (to distinguish with the model in previous section, we here use W to represent this response and $\mathscr{W}$ as the observations for it):

$$W(x) = f(x)^T\beta + M(x),$$

where $f(x)$ is a $p\times 1$ known function and β is a $p\times 1$ vector of model parameters. $M(x)$ is assumed to be a zero-mean second-order-stationary GP controlled by hyperparameters θ. Given that the true function is a non-negative function, the observation vector we get, $\mathscr{W}$, is non-negative. Traditional GP model for a deterministic function is actually an interpolation of the observations $\mathscr{W} = (\mathscr{W}(x_1),...,\mathscr{W}(x_t))^T$, and the shape of the predictive curve changes with the hyperparameters θ. Therefore, when estimating the θ, we must make sure that the resulting curve should not intersect with surface $W = 0$. In other words, those values of θ that causes the intersection should be eliminated. This intuition can naturally translate into the following penalized method. Specially, instead of optimizing the ordinary loglikelihood of $\mathscr{W}$, we propose to minimize the following penalized likelihood function to get the PMLE estimator for θ:

$$Q(\mathscr{W},\theta) = -l(\mathscr{W},\theta) + P(\mathscr{W},\theta) = \frac{1}{2}\ln(|(R')|) + \frac{1}{2}(\mathscr{W}-F\widehat{\beta}')^T(R')^{-1}(\mathscr{W}-F\widehat{\beta}') + \mu\kappa(\mathscr{W},\theta),$$

where l is the ordinary loglikelihood function, $P = \mu\kappa$ is the penalty term, μ is a non-negative penalty coefficient, $F = (f(x_1), ..., f(x_n))^T$, R is the covariance matrix for $\mathscr{W}$, $\widehat{\beta}' = (F^T R^{-1} F)^{-1} F^T R^{-1} \mathscr{W}$ and

$$\kappa(\mathscr{W}, \theta) = \begin{cases} |\min_{x \in \mathscr{X}}(\widehat{W}(x))|, & \text{if } \min_{x \in \mathscr{X}}(\widehat{W}(x)) < 0 \\ 0, & \text{if } \min_{x \in \mathscr{X}}(\widehat{W}(x)) \geq 0 \end{cases}.$$

With this penalty term, the parameters that cause the predictive curve to go below the $W = 0$ plane will be penalized.

This PMLE estimator can be obtained by applying Algorithm 1, which is similar with the sequential unconstrained minimization technique for constrained optimization problems. Inside this approach involves a sub-optimization problem over the predictive surface $\widehat{W}(x)$. We highlight that this sub-optimization is much easier compared with optimizing the unknown true response surface, since the predictive response function is extremely cheap with explicit form and available gradient. Note also that in the first iteration of Algorithm 1, the ordinary likelihood function is optimized. If the estimators from this ordinary likelihood function results in models that do not cross, then no additional estimate cost is incurred over the existing method. This is likely to happen if the distance between the function and the zero surface is quite large. When $n > 1$, the penalty coefficient μ will be enlarged by μ_0 in each iteration.

Algorithm 1 Optimizing PMLE

1: Set initial value: n=1, μ=0
2: Optimize Q and obtain optimal value for θ^*
3: **if** $\kappa(\mathscr{Z}, \theta^*) = 0$ **then**
4: Return θ^*
5: Terminate
6: **else**
7: $\mu \leftarrow \mu + n \times \mu_0$, $n \leftarrow n + 1$
8: Go to 2
9: **end if**

When applied in our case, where the function W (the difference between two quantile functions) is stochastic, this method can also return non-negative predictions by preventing crossing between the predictive curve with the plane $W = 0$. With a slight modification of the penalty function κ, this method can be easily applied in our multi-level quantile problem:

$$\kappa(\varphi) = \begin{cases} |\varphi|, & \text{if } \varphi < 0 \\ 0, & \text{if } \varphi \geq 0 \end{cases},$$

where $\varphi = \min_{x \in \mathscr{X}, l \in \{1,...,m-1\}}(\widehat{Z}_{l+1}(x) - \widehat{Z}_l(x))$. It is easy to see that the parameters will be penalized once crossing happens between any two successive predictive curves among the m quantile models.

With the approaches proposed in Section 2, we can build a co-kriging model for multi-level quantiles with no crossing. In Section 3, we introduce the optimization algorithm based on this metamodel.

3 QUANTILE SIMULATION OPTIMIZATION WITH CO-KRIGING MODEL

This section presents a sequential algorithm used to optimize α_m quantile with a multi-level model built with $\alpha_1, .., \alpha_m$ quantiles. As previously noted, the optimization process is guided by the proposed stochastic quantile cokriging model. The algorithm is sequential where the overall computing budget is iteratively allocated to sequentially search the lower quantile models for promising regions and increasingly narrow down the promising regions to search on higher quantiles. The overall idea is as follows: At the start of the algorithm, when only small budget is first applied, the observations for higher quantiles can be too noisy to

build a trustworthy model. In this stage, only a more reliable lower quantile α_1 model is built and used to guide the search. In other words, we optimize the first level as a start to identify possible promising regions. As the algorithm proceeds, more budget is allocated and the accuracy of quantile estimators improve. As these estimates improve, the algorithm will stepwise increase the level of the quantile metamodels developed, and use the current highest level to guide the search. As a result, the algorithm gradually optimizes a higher and higher quantile with a focus on the promising regions identified by previous levels, to finally optimize α_m quantile.

Within each iteration, we borrow the two-stage framework from the TSSO algorithm, to provide a "division of labor" (Quan et al. 2013) approach in the optimization. In the first stage (Searching Stage), we select a new design based on EI criterion applied to current highest level. This will select a new design point with the highest possibility of achieving a better result than the current best. The second stage (Allocation Stage) is responsible for driving the noises down at already selected designs by allocating additional computing budget to them. This is to improve the model fit and the accuracy of the existing observations, especially those in the optimal regions, to correctly find the optimum. The distribution of budget used in these two stages changes with iteration. At the beginning, more budget is used to search the design space to identify the promising regions. As the process evolves, more budget will be saved for the allocation stage, since the emphasis becomes refining the observations at already selected designs when we are in proximity of the promising regions. These two stages will be discussed in detail in Section 3.2 and 3.3 after an overview of the algorithm given in Section 3.1. Our algorithm is based on the TSSO procedure, and we refer interested readers to Quan et al. (2013) for full details of the algorithm.

3.1 Algorithm Overview

The proposed algorithm is iterative with Searching and Allocation Stage applied in each iteration until the computing budget runs out. Before describing this procedure, we list some parameters in Table 1. The

Table 1: Algorithm parameters list

Parameter	Definition
T	Total number of replications (computing budget) at the beginning
D_0	Initial design set
$\alpha_l, l = 1, ..., m$	Quantiles used for modeling
r_0	Minimum number of replications for a new selected design input
k	Current iteration
$h(k)$	Current highest level available
B_k	Number of available replications in iteration k
D_l	Current design set for lth level
$\mathscr{Y}_l$	Observations for lth level
A	Remaining number of replications(Algorithm terminates when $A = 0$)

first four parameters are user-defined. T is typically determined by the computing budget and for D_0, if no prior knowledge or preference is available, users can apply popular design strategies such as uniform and Latin Hypercube design. The multi-level quantiles used in this algorithm $\alpha_1, ..., \alpha_m$, of which α_m is our objective, should also be specified in advance. As noted above, to optimize a high quantile α_m, we start with the base level α_1. This level should not be too high and we suggest $\alpha_1 \in [0.5, 0.6]$ based on our experience. For the remaining level, we currently consider fixed levels between α_1 and α_m such as $1/2(\alpha_1 + \alpha_m)$. More sophisticated approaches can be applied to select those quantile levels, and this is an area for our future research. r_0 is the number of replications assigned to any new selected design inputs. This r_0 should be chosen to ensure that the observations for the base level have acceptable quality. In practice, starting with a small number of r_0, we can iterative increase its value until the model for α_1 build by D_0 with r_0 replications at each inputs pass the cross-validation test.

The remaining parameters are updated in each iteration. $h(k)$ is the current highest level with acceptable accuracy and it gradually increases with iteration. In each iteration, we choose its value as follows. Denote C_0 as the maximum noise variance of a quantile estimate that can be tolerated (In practice, its value can be chosen as the maximum noise variance of α_1-quantile estimator at D_0). We treat this value as a criterion to examine the quality of the quantile estimator. Specifically, only if $\mathscr{Y}_l(x)$ drawn from simulation results has noises smaller than C_0, we set $x \in D_l$. Denote $h(k)$ as the largest value in $\{1,2,...,m\}$ such that $D_{h(k)} \neq \varnothing$. In this case, we choose $h(k)$ as the current highest level and build a multi-level model for $\alpha_1,...,\alpha_{h(k)}$ in iteration k. B_k changes with iteration and will be introduced in detail in Section 3.3. $D_l, \mathscr{Y}_l$ and A can be easily renewed after the two stages. The algorithm is described in detail in Algorithm 2. In Section 3.2 and 3.3, we go into the details about the Searching and Allocation Stage.

Algorithm 2 TSSO for quantile optimization

Input: T, D_0, r_0, $\{\alpha_1,...,\alpha_m\}$

Initialization:
$k=0$; $B_1=r_0$; $D_1=D_0$; $A=T-|D_0|\times r_0$
Get the simulation results for D_0 with r_0 replications for each input
Estimate $\mathscr{Y}_1$ for D_1, let $h(0)\leftarrow 1$
Fit single stochastic GP model $\widehat{Z}_1(x)$ for $\mathscr{Y}_1$, resulting the predictive random variable $\tilde{Z}_1(x)$
Let $z_0^*=\min_{D_1}(\widehat{Z}_1(x))$, $x_0^*=\arg\min_{D_1}(\widehat{Z}_1(x))$
while $A>0$ **do**
    Searching Stage:
    $x_{k+1}=\arg\min \mathbb{E}[(z_k^*-\tilde{Z}_{\alpha_{h(k)}})|\mathscr{Y}_{h(k)},...\mathscr{Y}_{h(1)}]$
    Run r_0 replications and set $D_1\leftarrow D_1\cup\{x_k\}$
    Let $l^*=\arg\max_{l\in 1,..m,\mathscr{Y}_l(x_k)\leq C_0} l$
    Let $D_l\leftarrow D_l\cup\{x_k\}$ and enlarge the corresponding observation vector $\mathscr{Y}_l$, $\forall l\leq l^*$
    Allocation Stage:
    Update B_k
    Use OCBA to allocate the B_k-r_0 replications to selected inputs and run new simulations correspondingly
    For each selected design x_i, decide $l^*=\arg\max_{l\in 1,..m,\mathscr{Y}_l(x_i)\leq C_0} l$. Set $D_l=D_l\cup x_i$ and update the observation vector $\mathscr{Y}_l$, $\forall l\leq l^*$
    Modeling Update
    Set $h(k+1)$ as the largest value in $1,2,...,m$ such that $D_{h(k+1)}\neq\varnothing$
    Fit the co-kriging model for $\alpha_1,...,\alpha_{h(k+1)}$
    Let $z_{k+1}^*=\min_{D_h(k+1)}(\widehat{Z}_{h(k+1)})$, $x_{k+1}^*=\arg\min_{D_h(k+1)}(\widehat{Z}_{h(k+1)})$
    $A\leftarrow A-B_k$, $k\leftarrow k+1$
end while
return $x^*=\arg\min_{D_m}(\mathscr{Y}_m)$

3.2 Searching Stage

In the searching stage, we select the next design input based on the following modified EI criterion:

$$x_{k+1}=\arg\min \mathbb{E}[(z_k^*-\tilde{Z}_{\alpha_{h(k)}})|\mathscr{Y}_{h(k)},...\mathscr{Y}_{h(1)}],$$

where z_k^* is the lowest value of the predictive responses $\widehat{Z}_k$ at $D_{h(k)}$. $\tilde{Z}_{\alpha_{h(k)}}$ is a Gaussian random variable whose mean and variance is the predictive mean and variance of the $\alpha_{h(k)}$-the quantile (see equation 1, 2). With this criterion, we use the current highest level, $h(k)$ to guide our search. Similarly with standard EI

criterion, it balances between exploitation and exploitation. After selecting the new design input, we run r_0 simulations to it and add it to the design and update the observation vector accordingly.

3.3 Allocation Stage

At the beginning of this stage, we update the value of B_k. Since the Searching stage has already cost r_0 replications, the Allocation Stage can spend the remaining $(B_k - r_0)$ replications. As noted before, we want the budget to Allocation Stage increases as the algorithm evolves to refine the observations at selected design inputs. This is intuitive since at the beginning, more budget can be used in the first stage to search the space and when k gets larger, we are more likely to be in proximity of the promising region. In this case, we can reduce the number of new selected designs and assign more budget to the already sampled points in the promising region. We therefore let B_k increases with iteration and update its value as follows:

$$B_1 = r_0; \;\; B_k = \lfloor B_{k-1}(1 + \frac{\max\limits_{x_i \in D_{h(k)}} \mathrm{var}(\widehat{\varepsilon}(x_i))}{\max\limits_{x_i \in D_{h(k)}} \mathrm{var}(\widehat{\varepsilon}(x_i)) + \mathrm{var}(\widehat{Z}_{h(k)}(x_{k+1}))}) \rfloor, \;\; k > 1,$$

where $\mathrm{var}(\widehat{Z}_{h(k)}(x_{k+1}))$ is the predictive variance of $\widehat{Z}_{h(k)}(x_{k+1})$ given by the model built with $\mathscr{Y}$. The increase of B_k is controlled by the relationship between the observations noises, measured by $\max_{x_i \in D_{h(k)}} \mathrm{var}(\widehat{\varepsilon}(x_i))$, and the spatial uncertainty of the GP model, measured by $\mathrm{var}(\widehat{Z}_{h(k)}(x_{k+1}))$. At the beginning, the spatial uncertainty is very large, B_k has a slow growth to save more budget for design selection. When the spatial uncertainty gets smaller, i.e., the design space has been explored better, B_k will experience fast growth so that we can be more concentrate on the selected points in the promising regions.

After updating B_k, we can allocate $B_k - r_0$ replications to the selected design inputs. Note that D_1 contains all selected design inputs but $D_{h(k)}$ may not. We next define $D^*_{h(k)} = D_1$ and $\mathscr{Y}^*_{h(k)}$ as the point estimates of $\alpha_{h(k)}$-th quantile at $D^*_{h(k)}$ with noise vector $\varepsilon^*_{h(k)}$. Denote x_b as the current best in $D^*_{h(k)}$, we adopt the following OCBA technique to decide number of new replications n_i assigned to each input $x_i \in D_{h(k)}$:

$$\frac{n_i}{n_j} = \frac{\sqrt{\mathrm{var}(\varepsilon^*_{h(k)}(x_i))}/\lambda_{b,i}}{\sqrt{\mathrm{var}(\varepsilon^*_{h(k)}(x_j))}/\lambda_{b,j}}, \;\; (i, j \neq b); \;\; n_b = \sqrt{\mathrm{var}(\varepsilon^*_{h(k)}(x_b))} \sqrt{\sum_{i \neq b} \frac{n_i}{\mathrm{var}(\varepsilon^*_{h(k)}(x_i))}},$$

where $\mathrm{var}(\varepsilon^*_{h(k)}(x_i))$ is the sample noise variance of the observation at x_i, $\lambda_{b,i}$ is the difference between $\mathscr{Y}^*_{h(k)}(x_i)$ and $\mathscr{Y}^*_{h(k)}(x_b)$. The OCBA technique actually prefers to allocating additional replications to points with low response values and large noises, which is to refine the observations at promising regions and drive the noises down. Next, we can run those additional simulations to the existing inputs and update the observation vector, $\mathscr{Y}$, and current highest level, $h(k)$, accordingly. Finally, a new co-kriging metamodel can be built can goes to next iteration.

As the overall algorithm proceeds, the procedure searches focusing in more promising regions of higher and higher quantiles, and as result, less budget is spent in the non-promising regions sequentially. This can be seen by the budget used for increasing the precision of observation at lower quantiles which essentially also improves the estimates for higher quantiles. When optimizing lower quantiles, more budget is spent in its promising regions and if this region is also promising for α_m, it would have already been sampled and the estimates here have been enhanced. In other words, the algorithm digs into the promising regions identified by the lower quantiles when estimating α_m. We highlight here that this improvement is caused by the specificity of our problem, which does not apply to a general multi-fidelity problems. Since in those problems, the experiments for lower fidelity and higher fidelity are different. Running simulations for one level does not necessarily improve the observations for another level.

4 NUMERICAL EXPERIMENT

To illustrate the benefit we can get from our multi-level optimization algorithm, we conduct a numerical experiment to compare it with the algorithm based on only one quantile level model, the target α_m-quantile model. Without too much modification (fix $m = h(k) = 1$), our algorithm can be easily applied to single-level problem.

In this example, the design space is $\mathscr{X} = [0,1]$. At each $x \in \mathscr{X}$, the loss function $L(x)$ is assumed to follow a normal distribution with the mean $m(x)$ and variance $v(x)$:

$$m(x) = 5(0.2(x-0.02)+1)\cos(13(x-0.02)), \quad v(x) = 10(2+\sin(10\pi x - 0.5)).$$

In this example, we select $r_0 = 20$, $T = 1000$. The initial design a uniform design with $D_0 = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, and here we only consider two quantile levels 0.6 and 0.95 where 0.95 quantile is our objective (shown in Figure 1). .

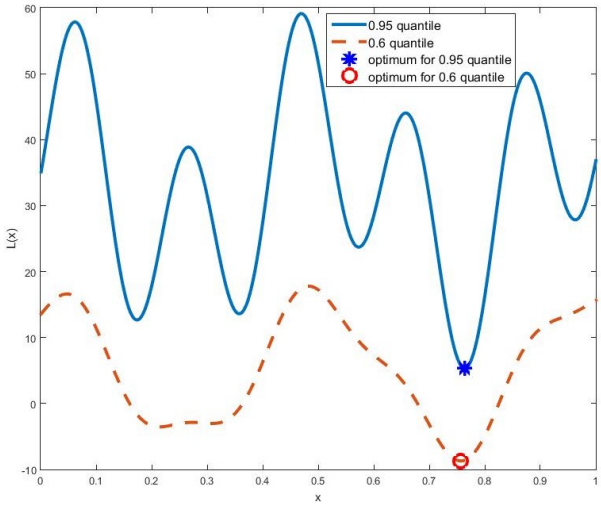

Figure 1: True 0.6 and 0.95 quantile functions

To mitigate the stochastic nature of the problem, all experiments are conducted with 100 macro-replications. To compare the two algorithms, we further define the true selection as: $|x^* - x_0| < 0.035$, where x^* is the optimum found and x_0 represents its true value (0.765). The experiment results for 100 macro-replicaion is summarized in Table 2. The results show that the multi-level algorithm is much better

Table 2: Comparison of the two methods

	multi-level model	single model
Frequency of true selection	91	70
Average Number of selected design points	16.96	14.02

than the single one in terms of true selection. The second criterion we use is the average number of inputs in the design set after the optimization process, and seen from the results, our multi-level tries more design inputs. This is because our algorithm first searches a 0.6 quantile, which has less noisy estimates with limited budget, to determine promising region. In this stage, the increase of B_k will not be that large compared with single model, which directly fits the 0.95 quantile with more noisy observations. Therefore, the multi-level algorithm saves more for searching. In other words, at the beginning, our algorithm is faced with a more accurate surface (0.6 quantile) compared with the one with single model, who deals with a very inaccurate 0.95 surface with limited initial budget. This inaccurate surface can mislead the search and waste budget to unpromising regions. To show this, we here provide the design inputs that

have been selected by the two approaches in one macro-replication run (see in Table 3). In this run, in the first four iterations, our multi-level algorithm first searches 0.6-quantile level ($h(k) = 1$) and then goes to 0.95-quantile level with a focus on the promising regions around 0.75. The single-model based algorithm, however, seems to still focus on providing a space-filling design (likely due to the poor model estimation throughout) and does not end up near the optimum.

Table 3: Comparison of the design inputs selected two methods in one run

multi-level model	0.674	0.295	0.111	0.895	0.987	0.709	0.731	0.732	0.74
single model	0.7	0.5	0.3	0.166	0.108	0.873	0.151		

In this example, although the two curves are not highly correlated, the promising regions of 0.6 quantile is still informative for 0.95 quantile, which makes the proposed method much better. In the case with limited budget and large noises, searching informative lower quantile as a start appears to be a more robust approach.

5 CONCLUSION

In this paper, we propose a co-kriging based multi-level quantile simulation optimization algorithm. The main idea is to leverage on the lower quantiles, which are typically easier and cheaper to obtain, to help optimize some higher quantile function. To develop the metamodel, we first generalize the stochastic co-kriging model and apply it to quantile prediction. Next, to prevent possible crossing problem among multiple quantile curves, we propose a PMLE approach. This helps to fulfill the non-decreasing property of quantiles and ensure our multi-level search in the optimization algorithm is valid. Based on this metamodel, we propose our optimization algorithm, which considers selecting new inputs and improving the accuracy of already taken observations simultaneously. It starts with searching lower quantiles and gradually goes to optimizing higher quantiles. Through a numerical example, we clearly see the benefit of our multi-level algorithm. When optimizing much noisier quantile functions with limited budget, starting with some informative and less noisy lower quantile can provide much better results.

This work provides a different view of optimizing quantile function, yet there are still a few questions worth investigating. First, as we currently only consider fixed levels in our search, more sophisticated approaches to more intelligently select the number of levels and the levels themselves should be further explored to fully benefit from the proposed algorithm. Second, although the numerical example showed good results and empirical convergence, more detailed convergence of the algorithm should be studied.

REFERENCES

Ankenman, B., B. L. Nelson, and J. Staum. 2010. "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58(2):371–382.

Bahadur, R. R. 1966. "A Note on Quantiles in Large Samples". *The Annals of Mathematical Statistics* 37(3):577–580.

Chen, C.-H., J. Lin, E. Yücesan, and S. E. Chick. 2000. "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization". *Discrete Event Dynamic Systems* 10(3):251–270.

Chen, E. J. 2009. "Metamodels for Estimating Quantiles of Systems with One Controllable Parameter". *Simulation* 85(5):307–317.

Chen, X., and K.-K. Kim. 2016. "Efficient VaR and CVaR Measurement via Stochastic Kriging". *INFORMS Journal on Computing* 28(4):629–644.

Chen, X., S. Hemmati, and F. Yang. 2017. "Stochastic Co-Kriging for Steady-State Simulation Metamodeling". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan. et al., 1750–1761. Piscataway, New Jersey: IEEE.

Cole, T. 1988. "Fitting Smoothed Centile Curves to Reference Data". *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 151(3):385–418.

Dabo-Niang, S., and B. Thiam. 2010. "Robust Quantile Estimation and Prediction for Spatial Processes". *Statistics & Probability Letters* 80(17):1447–1458.

He, X. 1997. "Quantile Curves without Crossing". *The American Statistician* 51(2):186–192.

Higdon, D., M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne. 2004. "Combining Field Data and Computer Simulations for Calibration and Prediction". *SIAM Journal on Scientific Computing* 26(2):448–466.

Hong, L. J., and G. Liu. 2009. "Simulating Sensitivities of Conditional Value at Risk". *Management Science* 55(2):281–293.

Hong, L. J., Z. Hu, and G. Liu. 2014. "Monte Carlo Methods for Value-at-Risk and Conditional Value-at-Risk: A review". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 24(4):22.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global optimization* 13(4):455–492.

Kennedy, M. C., and A. O'Hagan. 2000. "Predicting the Output from a Complex Computer Code when Fast Approximations Are Available". *Biometrika* 87(1):1–13.

Koenker, R. 1984. "A Note on L-Estimates for Linear Models". *Statistics & Probability Letters* 2(6):323–325.

Koenker, R. 2005. *Quantile Regression*. Number 38. Cambridge university press.

Le Gratiet, L., and C. Cannamela. 2015. "Cokriging-Based Sequential Design Strategies Using Fast Cross-Validation Techniques for Multi-Fidelity Computer Codes". *Technometrics* 57(3):418–427.

Liu, Y., and Y. Wu. 2009. "Stepwise Multiple Quantile Regression Estimation Using Non-Crossing Constraints". *Statistics and Its Interface* 2(3):299–310.

Quan, N., J. Yin, S. H. Ng, and L. H. Lee. 2013. "Simulation Optimization via Kriging: a Sequential Search Using Expected Improvement with Computing Budget Constraints". *IIE Transactions* 45(7):763–780.

Wang, S., and S. H. Ng. 2017. "A Joint Gaussian Process Metamodel to Improve Quantile Predictions". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan. et al., 1891–1902. Piscataway, New Jersey: IEEE.

Yin, J., S. H. Ng, and K. M. Ng. 2011. "Kriging Metamodel with Modified Nugget-Effect: The Heteroscedastic Variance Case". *Computers & Industrial Engineering* 61(3):760–777.

AUTHOR BIOGRAPHIES

SONGHAO WANG is a Ph.D. student in the Department of Industrial Systems Engineering and Management at National University of Singapore. He received his B.S. in Modern Mechanics from University of Science and Technology of China in 2015. His research interests include simulation metamodeling and optimization. His email address is wangsonghao@u.nus.edu.

SZU HUI NG is an Associate Professor in the Department of Industrial Systems Engineering and Management at the National University of Singapore. She holds B.S., M.S., and Ph.D. degrees in Industrial and Operations Engineering from the University of Michigan. Her research interests include computer simulation modeling and analysis, design of experiments, and quality and reliability engineering. She is a member of IEEE and INFORMS and a senior member of IIE. Her email address is isensh@nus.edu.sg and her web page is https://www.isem.nus.edu.sg/staff/ngsh/index.html .

WILLIAM BENJAMIN HASKELL received his B.S. degree in Mathematics and M.S. degree in Econometrics from the University of Massachusetts Amherst in 2006, and his Ph.D. degree in Operation Research from the University of California Berkeley in 2011. Currently, he is an assistant professor in the Department of Industrial Systems Engineering and Management at the National University of Singapore. His research interests include convex optimization, risk-aware decision making, and dynamic programming. His email address is wbhaskell@gmail.com