Online Supplementary Material for 'A Multi-Level Simulation Optimization Approach for Quantile Functions'

Appendix A: Likelihood Function and Estimation of Model Parameters for the Co-Kriging Model

Given the co-kriging model, the point estimate vector $\mathcal{Y}$ follows a multivariate normal distribution $\mathcal{N}(H\widehat{\beta}, R)$. The model parameters can then be straightforwardly estimated by maximizing the loglikelihood function:

$$l(\mathcal{Y}, \theta, \sigma^2) = -\frac{1}{2}\ln(|R|) - \frac{1}{2}(\mathcal{Y} - H\widehat{\beta})^T R^{-1}(\mathcal{Y} - H\widehat{\beta}). \quad (A.1)$$

This approach, however, is to obtain the parameters from models in different levels simultaneously, which involves a multivariate optimization problem. Obviously, this problem becomes more severe as the number of levels increases. To overcome this drawback, we consider more efficient estimation approaches.

As proved by Kennedy and O'Hagan (2000), when the observations have no measurement error ($R_\epsilon = 0$), the likelihood function of the observation vector can be fully decomposed as follows (to differentiate this case with the stochastic problem (measurement error $R_\epsilon \neq 0$), we use $\mathcal{Z}^T = (\mathcal{Z}_1^T, ..., \mathcal{Z}_m^T)$ to represent the observation vector for the deterministic case where $\mathcal{Z}_1$ is the observation vector for the first level):

$$l(\mathcal{Z}, \theta, \sigma^2) = l_1(\mathcal{Z}_1, \theta_1, \sigma_1^2) + \sum_{j=2}^{m} l_m(\mathcal{Z}_j - \mathcal{Z}_{j-1}, \theta_j, \sigma_j^2),$$

where $\mathcal{Z}_{j-1}$ consists of the observations of the inputs in D at level $j-1$. The vector $\mathcal{Z}_j - \mathcal{Z}_{j-1}$ can be shown to follow a multi-normal distribution $\mathcal{N}(F_j\widehat{\beta}_j, R_j)$, where $F_j = \mathbf{f}_l(D)$, $R_j = A_j$, $\widehat{\beta}_j = (F_j^T R_j^{-1} F_j)^{-1} F_j^T R_j^{-1}(\mathcal{Z}_j - \mathcal{Z}_{j-1})$. The function l_m is the loglikelihood for $\mathcal{N}(F_j\widehat{\beta}_j, R_j)$. This decomposition makes optimizing a large scale function l equivalent to optimizing a series of sub problems $(l_1, ..., l_m)$ with fewer parameters in each, and thus greatly reduces the complexity.

This decomposition, however, is not so straightforward to generalize to the stochastic case. When the noise variance of the observations R_ϵ is small, which can be accomplished by increasing simulation replications, the decomposition can serve as an approximation of $(A.1)$ by ignoring the higher-order terms of R_ϵ (see the proof in Appendix B). Although this decomposition approach is not as accurate as the standard approach, which optimizes $(A.1)$ directly, it greatly reduces the complexity. In practice, optimizing $(A.1)$ directly with all model parameters is more difficult and likely to be trapped in sub-optimal regions. A more practical way is to first use the decomposition approach and then treat the optimums found as starting points to apply the standard approach (Fricker et al. 2013).

Appendix B: Proof of Decomposition of loglikelihood

Here we prove the approximation of the loglikelihood in a stochastic co-kriging model. For simplicity, we only consider two levels and assume $D_1 = D_2 = D$, $f_1(x) = f_2(x) = 0$, which can be easily generalized to more complicated multi-level cases. In this simple case,

$$R = R_z + R_\epsilon = \begin{pmatrix} A_1(D) & A_1(D) \\ A_1(D)^T & A_1(D) + A_2(D) \end{pmatrix} + \begin{pmatrix} N_1 & N_2 \\ N_2 & N_3 \end{pmatrix},$$

where N_1, N_3 represent the noise variance matrix for $\mathcal{Y}_1$ and $\mathcal{Y}_2$, respectively, N_2 represents the noise covariance matrix for $\mathcal{Y}_1$ and $\mathcal{Y}_2$.

It can be computed:

$$\begin{aligned} l(\mathcal{Y}_1, \mathcal{Y}_2, , \theta, \sigma^2) &= -\frac{1}{2}\ln(|R|) - \frac{1}{2}\mathcal{Y}^T R^{-1}\mathcal{Y} \\ &= -\frac{1}{2}\ln(|A_1 + N_1|) - \frac{1}{2}\mathcal{Y}_1^T (A_1 + N_1)^{-1}\mathcal{Y}_1 - \frac{1}{2}\ln(|F|) \\ &\quad - \frac{1}{2}(\mathcal{Y}_2 - (A_1^T + N_2)(A_1 + N_1)^{-1}\mathcal{Y}_1)^T F^{-1}(\mathcal{Y}_2 - (A_1^T + N_2)(A_1 + N_1)^{-1}\mathcal{Y}_1), \end{aligned}$$

where $F := 1^2 A_1 + A_2 + N_3 - (A_1^T + N_2)(A_1 + N_1)^{-1}(A_1 + N_2)$.

Suppose that the number of replications at all design points has order $\mathcal{O}(n)$, we find that (see proof of Equation (191), Page 21 from Petersen and Pedersen (2012))

$$(A_1 + N_1)^{-1} \approx A_1^{-1} - A_1^{-1} N_1 A_1^{-1}.$$

It follows that,

$$F \approx \tilde{F} := A_2 + N_3 + N_1 - 2N_2,$$

$$\mathcal{Y}_2 - (A_1 + N_2)(A_1 + N_1)^{-1}\mathcal{Y}_1 \approx \mathcal{Y}_2 - \mathcal{Y}_1.$$

Therefore, suppose the number of replications at design points has order $\mathcal{O}(n)$, we have

$$l(\mathcal{Y}_1, \mathcal{Y}_2, \theta, \sigma^2) = l_1(\mathcal{Y}_1, \theta_1, \sigma_1^2) + l_2(\mathcal{Y}_2 - \mathcal{Y}_1, \theta_2, \sigma_2^2) + \mathcal{O}(1/n),$$

where,

$$\begin{aligned} l_1(\mathcal{Y}_1, \theta_1, \sigma_1^2) &= -\frac{1}{2}\ln(|A_1 + N_1|) - \frac{1}{2}\mathcal{Y}_1^T (A_1 + N_1)^{-1}\mathcal{Y}_1, \\ l_2(\mathcal{Y}_2 - \mathcal{Y}_1, \theta_2, \sigma_2^2) &= -\frac{1}{2}\ln(|\tilde{F}|) - \frac{1}{2}(\mathcal{Y}_2 - \mathcal{Y}_1)^T \tilde{F}^{-1}(\mathcal{Y}_2 - \mathcal{Y}_1). \end{aligned}$$

Therefore, when R_ϵ is small, the decomposition above can serve as an approximation of the likelihood function $(A.1)$.

Appendix C: Proof of Theorem 1

Suppose $0 < \alpha_1 < \alpha_2 < 1$, we prove the consistency and asymptotic unbiasedness of the proposed sectioning covariance estimator for $\mathcal{Y}_1(x)$ and $\mathcal{Y}_2(x)$ with n simulations at x.

First, we refer to Theorem 2.1 from Lin et al. (1980) on the asymptotic covariance for $\mathcal{Y}_1(x)$ and $\mathcal{Y}_2(x)$:

$$\lim_{N\to\infty} N\mathrm{cov}(\mathcal{Y}_1(x), \mathcal{Y}_2(x)) = \frac{\alpha_1(1-\alpha_2)}{f(v_{\alpha_1})f(v_{\alpha_2})},$$

where $v_{\alpha_1}, v_{\alpha_2}$ are the true quantiles and f is the pdf of the underlying distribution. For simplicity, we define $\gamma := \frac{\alpha_1(1-\alpha_2)}{f(v_{\alpha_1})f(v_{\alpha_2})}$.

This proof consists of two parts. In C.1, we prove the consistency and asymptotic unbiasedness of $\widehat{\mathrm{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x))$. In C.2, we derive its MSE.

C.1. Consistency and Asymptotic Unbiasedness of $\widehat{\text{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x))$

Denote the n simulation results as: $\mathbf{L} := \{L(x,\xi_1), ..., L(x,\xi_N)\}$. Recall that N_b is the batch size and N_c is the number of results in each batch, and thus $N_b \times N_c = N$. Define $\mathbf{L}^{(j)} := \{L(x,\xi_{(j-1)N_c+1}), ..., L(x,\xi_{jN_c})\}$ as the jth batch of simulation results and Ψ_i as the operator to take the sample α_i-quantile: $\Psi_i(\mathbf{L}) = L_{\lfloor \alpha_i N \rfloor}$. For example, $\Psi_i(\mathbf{L}^{(j)})$ represents the sample α_i-quantile estimator based on the jth batch. According to Bahadur (1966) and Chen and Kim (2016),

$$\Psi_i(\mathbf{L}) = v_{\alpha_i} + \frac{1}{N}\sum_{j=1}^{N} \psi_i(L(x,\xi_j)) + R_{i,N}, \quad \psi_i(x) = \frac{\alpha_i - \mathbf{1}_{\{x \le v_{\alpha_i}\}}}{f(v_{\alpha_i})}, \ \ i = 1,2, \tag{A.2}$$

where $R_{i,N}$ is the remainder term with $R_{i,N} = O(N^{-3/4}(\text{loglog}N)^{3/4})$. Denote R_i^j as the remainder term in (A.2) for $\Psi_i(\mathbf{L}^{(j)})$, $\varphi_i(\mathbf{L}) = \frac{1}{N}\sum_{i=1}^{N} \psi_i(L(x,\xi_i))$, $\varphi_i^j = \varphi_i(\mathbf{L}^{(j)}) = \frac{1}{N_c}\sum_{i=1}^{N_c} \psi_i(L(x,\xi_{(j-1)N_c+i}))$, $\bar{\varphi}_i = \frac{1}{N_b}\sum_{j=1}^{N_b} \varphi_i^j$ and $\bar{R}_i = \frac{1}{N_b}\sum_{j=1}^{N_b} R_i^j$.

With these notations, the proposed covariance estimator is:

$$\begin{aligned}
&\widehat{\text{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x)) = \frac{1}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\Psi_1(\mathbf{L}^{(j)}) - \Psi_1(\mathbf{L}))(\Psi_2(\mathbf{L}^{(j)}) - \Psi_2(\mathbf{L}))\} \\
&= \frac{1}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\Psi_1(\mathbf{L}^{(j)}) - \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_1(\mathbf{L}^{(k)}) + \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_1(\mathbf{L}^{(k)}) - \Psi_1(\mathbf{L})) \\
&(\Psi_2(\mathbf{L}^{(j)}) - \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_2(\mathbf{L}^{(k)}) + \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_2(\mathbf{L}^{(k)}) - \Psi_2(\mathbf{L}))\} \\
&= \sigma_1^2 + \sigma_2^2,
\end{aligned}$$

where we define

$$\sigma_1^2 := \frac{1}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\Psi_1(\mathbf{L}^{(j)}) - \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_1(\mathbf{L}^{(k)}))(\Psi_2(\mathbf{L}^{(j)}) - \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_2(\mathbf{L}^{(k)}))\},$$

$$\sigma_2^2 := \frac{1}{(N_b-1)}\sum_{j=1}^{N_b}\{(\Psi_1(\mathbf{L}) - \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_1(\mathbf{L}^{(k)}))(\Psi_2(\mathbf{L}) - \frac{1}{N_b}\sum_{k=1}^{N_b}\Psi_2(\mathbf{L}^{(k)}))\}.$$

We next derive the asymptotic properties for σ_1^2 and σ_2^2 separately in Sections C.1.1 and C.1.2.

C.1.1. Asymptotic properties for σ_1^2.

Note that

$$\sigma_1^2 = \frac{1}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\varphi_1^j - \bar{\varphi}_1 + R_1^j - \bar{R}_1)(\varphi_2^j - \bar{\varphi}_2 + R_2^j - \bar{R}_2)\}.$$

It is easy to obtain

$$\begin{aligned}
\mathbb{E}[\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\varphi_1^j - \bar{\varphi}_1)(\varphi_2^j - \bar{\varphi}_2)\}] &= \frac{N}{N_b}\text{cov}(\varphi_1^1, \varphi_2^1) \\
&= \frac{N}{N_b N_c}\text{cov}(\psi_1(L(x,\xi_1)), \psi_2(L(x,\xi_1))) = \frac{\alpha_1(1-\alpha_2)}{f(v_{\alpha_1})f(v_{\alpha_2})}.
\end{aligned}$$

Recall $\gamma = \frac{\alpha_1(1-\alpha_2)}{f(v_{\alpha_1})f(v_{\alpha_2})}$. We can show that the value inside the expectation converges to γ in probability. Specifically,

$$\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\varphi_1^j - \bar{\varphi}_1)(\varphi_2^j - \bar{\varphi}_2)\} = \frac{N}{N_b(N_b-1)}(\sum_{j=1}^{N_b}\{\varphi_1^j\varphi_2^j\} - N_b\bar{\varphi}_1\bar{\varphi}_2). \tag{A.3}$$

The first term in $(A.3)$ is asymptotically equal to $\frac{1}{N_b}\sum_{j=1}^{N_b}(\sqrt{N_c}\varphi_1^j)(\sqrt{N_c}\varphi_2^j)$, with:

$$\mathbb{E}[\sqrt{N_c}\varphi_1^j\sqrt{N_c}\varphi_2^j]=\mathbb{E}[\psi_1(L(x,\xi_1))\psi_2(L(x,\xi_1))]=\gamma.$$

Therefore, for all $\epsilon>0$,

$$\begin{aligned}\mathbb{P}(|\frac{1}{N_b}\sum_{j=1}^{N_b}(\sqrt{N_c}\varphi_1^j)(\sqrt{N_c}\varphi_2^j)-\gamma|>\epsilon)&\le\frac{1}{N_b}\frac{1}{\epsilon^2}\mathrm{var}[\sqrt{N_c}\varphi_1^j\sqrt{N_c}\varphi_2^j]\\&\le\frac{1}{N_b}\frac{1}{\epsilon^2}\mathbb{E}[N_c{\varphi_1^j}^2N_c{\varphi_2^j}^2]\\&\le\frac{1}{N_b}\frac{1}{\epsilon^2}\mathbb{E}[N_c^2{\varphi_1^j}^4+N_c^2{\varphi_2^j}^4].\end{aligned}$$

According to Chen and Kim (2016), $\mathbb{E}[{\varphi_i^j}^4]=\mathcal{O}(N_c^{-2})$. The second term in $(A.3)$ is asymptotically equivalent to $N_c\bar{\varphi}_1\bar{\varphi}_2$ and

$$N_c\bar{\varphi}_1\bar{\varphi}_2\le\frac{1}{2}((\sqrt{N_c}\bar{\varphi}_1)^2+(\sqrt{N_c}\bar{\varphi}_2)^2).$$

It is easy to see that,

$$\mathbb{P}[|\sqrt{N_c}\bar{\varphi}_i|>\epsilon]\le\frac{N_c\mathbb{E}[\bar{\varphi}_i^2]}{\epsilon^2}=\frac{N_c\mathrm{var}[\varphi_i^1]}{N_b\epsilon^2}=\frac{\mathrm{var}[\phi_i]}{N_b\epsilon^2}\to0,\quad i=1,2.$$

It follows that $\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(\varphi_1^j-\bar{\varphi}_1)(\varphi_2^j-\bar{\varphi}_2)\}$ converges to γ in probability.

On the other hand, according to the Cauchy-Schwarz inequality,

$$\begin{aligned}&\mathbb{E}[\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(R_1^j-\bar{R}_1)(R_2^j-\bar{R}_2)\}]\\&\le\mathbb{E}[\frac{N}{N_b(N_b-1)}\sqrt{\sum_{j=1}^{N_b}(R_1^j-\bar{R}_1)^2}\sqrt{\sum_{j=1}^{N_b}(R_2^j-\bar{R}_2)^2}]\\&\le\sqrt{\frac{N}{N_b(N_b-1)}\mathbb{E}[\sum_{j=1}^{N_b}(R_1^j-\bar{R}_1)^2]\frac{N}{N_b(N_b-1)}\mathbb{E}[\sum_{j=1}^{N_b}(R_2^j-\bar{R}_2)^2]}.\end{aligned}$$

The second inequality follows the Cauchy-Schwarz inequality applied in probability theory that $|\mathbb{E}[X_1X_2]|^2\le\mathbb{E}[X_1^2]\mathbb{E}[X_2^2]$, where X_1 and X_2 are random variables. According to Duttweiler (1973), $\frac{N}{N_b(N_b-1)}\mathbb{E}[\sum_{j=1}^{N_b}(R_1^j-\bar{R}_1)^2]=\mathcal{O}(N_c^{-1/2})$. Therefore, the above expectation converges to zero. Similarly,

$$\begin{aligned}&\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}\{(R_1^j-\bar{R}_1)(R_2^j-\bar{R}_2)\}\\&\le\sqrt{\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}(R_1^j-\bar{R}_1)^2}\sqrt{\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}(R_2^j-\bar{R}_2)^2}.\end{aligned}$$

According to Chen and Kim (2016), $\frac{N}{N_b(N_b-1)}\sum_{j=1}^{N_b}(R_1^j-\bar{R}_1)^2=\mathcal{O}(N_c^{-1/2}(\log\log N_c)^{3/2})$. It follows that the above quantity converges to zero.

By using the Cauchy-Schwarz to the cross product terms, σ_1^2 is shown to be asymptotically unbiased and converges to γ.

C.1.2. Asymptotic properties for σ_2^2 We next prove the property for σ_2^2.

Through simple computation,

$$\sigma_2^2 = \frac{1}{N_b^2(N_b-1)} \sum_{j=1}^{N_b} \{R_1^j - R_{1,N}\} \sum_{j=1}^{N_b} \{R_2^j - R_{2,N}\}.$$

According to Chen and Kim (2016), $\frac{1}{N_b^2(N_b-1)} \{\sum_{j=1}^{N_b} (R_1^j - R_{1,N})\}^2 \to 0$, $\mathbb{E}[\frac{1}{N_b^2(N_b-1)} \{\sum_{j=1}^{N_b} (R_1^j - R_{1,N})\}^2] \to 0$. It can be easily proved that σ_2^2 converges to 0 in probability and is asymptotically unbiased.

With Section C.1.1 and C.1.2, following the asymptotic properties for σ_1^2 and σ_2^2, it is easy to see the convergency and asymptotic unbiasedness of the proposed covariance estimator.

C.2. MSE of $\widehat{\text{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x))$

We next check the MSE of $\widehat{\text{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x))$. Its bias is easy to see from the proof in C.1 and the squared bias has order $o(N^{-2})$. We next only check the variance of $\widehat{\text{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x))$.

According to the Cauchy-Schwarz inequality,

$$\begin{aligned}
&\text{var}(\widehat{\text{cov}}(\mathcal{Y}_1(x), \mathcal{Y}_2(x))) \\
=&\text{var}(\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} \{(\Psi_1(\mathbf{L}^{(j)}) - \Psi_1(\mathbf{L}))(\Psi_2(\mathbf{L}^{(j)}) - \Psi_2(\mathbf{L}))\}) \\
\leq&\mathbb{E}[\left\{ (\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} \{(\Psi_1(\mathbf{L}^{(j)}) - \Psi_1(\mathbf{L}))(\Psi_2(\mathbf{L}^{(j)}) - \Psi_2(\mathbf{L}))\} \right\}^2] \\
\leq&\mathbb{E}[\{\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_1(\mathbf{L}^{(j)}) - \Psi_1(\mathbf{L}))^2\} \{\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_2(\mathbf{L}^{(j)}) - \Psi_2(\mathbf{L}))^2\}] \\
\leq&\mathbb{E}[\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_1(\mathbf{L}^{(j)}) - \Psi_1(\mathbf{L}))^2] \mathbb{E}[\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_2(\mathbf{L}^{(j)}) - \Psi_2(\mathbf{L}))^2] \\
&+\sqrt{\text{var}[\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_1(\mathbf{L}^{(j)}) - \Psi_1(\mathbf{L}))^2] \text{var}[\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_2(\mathbf{L}^{(j)}) - \Psi_2(\mathbf{L}))^2]}
\end{aligned}$$

The first inequality follows that $\text{var}[X] \leq \mathbb{E}[X^2]$, the second inequality follows Cauchy-Schwarz inequality and the third inequality follows that $\text{cov}[X_1, X_2] = \mathbb{E}[X_1 X_2] - \mathbb{E}[X_1]\mathbb{E}[X_2] \leq \sqrt{\text{var}[X_1]\text{var}[X_2]}$, where X, X_1, X_2 are random variables. From Sections C.1.1 and C.1.2, $\mathbb{E}[\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_i(\mathbf{L}^{(j)}) - \Psi_i(\mathbf{L}))^2] = o(N^{-1})$. According to Chen and Kim (2016), $\text{var}[\frac{1}{N_b(N_b-1)} \sum_{j=1}^{N_b} (\Psi_i(\mathbf{L}^{(j)}) - \Psi_i(\mathbf{L}))^2] = o(N^{-2})$. It follows that the variance of the proposed estimator has order $o(N^{-2})$.

Appendix D: Proof of Theorem 2

Consider model (6) for the non-negative response function W where θ_0 is the true value of the hyperparameter for this model. Denote $l_n(\theta)$ as the ordinary loglikelihood function and $Q_n(\theta)$ as the penalized likelihood function:

$$Q(\theta) = -l(\theta) + \lambda\kappa(\theta).$$

We first compute the order of the penalty term $\kappa(\theta)$. The main idea is that, as the design points get denser and denser, the difference between any unobserved design point x with its nearest design point becomes smaller and so does the difference between the predictive value at x and the positive observation at its nearest

design point. In this case, the predictive value at x becomes more likely to be positive (as it becomes more and more close to a positive value). For simplicity, let the design space be one-dimensional. Nonetheless, this proof can be easily generalized to multi-dimensional case.

For each unobserved $x \in \mathcal{X}$, let $x_0 \in \{D\}$ be the nearest design point to x (if there are more than one nearest point, pick any one):

$$x_0 := \arg\min_{x' \in D} |x' - x|.$$

Further denote h_0 as the maximum of the distance between any unexplored input with its nearest design input:

$$h_0 := \sup_{x \in \mathcal{X} \setminus D} \inf_{x^0 \in D} |x - x^0|.$$

Within a fixed domain, as the design points become dense, $h_0 \to 0$. In other words, there exists a sequence c_n such that $c_n \to \infty$ as $n \to \infty$ and $h_0 = \mathcal{O}(c_n^{-1})$.

The predictor for the deterministic GP model considered here has similar form with (2) by setting noise variance matrix $R_\epsilon = 0$ and the number of levels as 1. Moreover, the predictive value at x_0 is exactly the observation here owing to the interpolation property of the deterministic GP model:

$$\mathcal{W}(x_0) = f(x_0)^T \beta + t(x_0)^T R^{-1}(\mathcal{W} - F\beta),$$

we see that:

$$\widehat{W}(x) = \mathcal{W}(x_0) + (f(x)^T - f(x_0)^T)\beta + (t(x)^T - t(x_0)^T)R^{-1}(\mathcal{W} - F\beta),$$

where $t(x)$ is the covariance vector between x and the design points. For any entry in $f(x)$ and $t(x)$, considering Taylor expansion, we have:

$$f(x)_1 - f(x_0)_1 = f'(x_0)|x - x_0| + o(|x - x_0|) = \mathcal{O}(h) = \mathcal{O}(c_n^{-1}),$$

$$t(x)_1 - t(x_0)_1 = t'(x_0)|x - x_0| + o(|x - x_0|) = \mathcal{O}(h) = \mathcal{O}(c_n^{-1}).$$

Here, we assume that for each θ, the first derivative of f and t is bounded within the design domain, which is valid in most cases. Considering that $f(x)^T\beta + t(x)^T R^{-1}(\mathcal{Z} - F\beta)$ is of order $\mathcal{O}(1)$, we see that $\widehat{W}(x) - \mathcal{W}(x_0) = \mathcal{O}(c_n^{-1})$. In this case, the penalty term $\kappa(\theta) = \mathcal{O}(c_n^{-1})$ for every possible θ.

For the ordinary MLE, according to Yi et al. (2011), under the similar regularity conditions, there exists a solution $\widehat{\theta}_n$ to $l(\theta) = 0$, which is consistent for θ_0 as $n \to \infty$. According to Stein (2012) and Li and Sudjianto (2005), for this series of $\widehat{\theta}_n$, $||\widehat{\theta}_n - \theta|| = \mathcal{O}_p(n^{-1/2})$.

The remaining proof is quite similar to Theorem 1 from Fan and Li (2001). We need to show that for all $\epsilon > 0$, there exists a large constant C, such that:

$$\mathbb{P}\{\inf_{||\mathbf{u}||=C} Q(\theta_0 + n^{-1/2}\mathbf{u}) > Q(\theta_0)\} \geq 1 - \epsilon. \tag{A.4}$$

This shows that there exists a local minimum of Q within the ball $\{\theta_0 + n^{-1/2}\mathbf{u} : ||\mathbf{u}|| \leq C\}$ with probability no less than $1 - \epsilon$. It follows that there is a local minimizer of Q satisfying $||\widehat{\theta}_n - \theta_0|| = \mathcal{O}_p(n^{-1/2})$.

Define $D_n(\mathbf{u}) := Q(\theta_0 + n^{-1/2}\mathbf{u}) - Q(\theta_0)$, we have,

$$\begin{aligned} D_n(\mathbf{u}) &= -l_n(n^{-1/2}\mathbf{u}) + l(\theta_0) + \lambda(\kappa(n^{-1/2}\mathbf{u}) - \kappa(\theta_0)) \\ &= -n^{-1/2} l_n'(\theta_0)\mathbf{u} + \frac{1}{2}\mathbf{u}^T \mathbb{I}_n(\theta_0)\mathbf{u} n^{-1}(1 + \mathcal{O}_p(1)) + \lambda\kappa(\sqrt{n}\mathbf{u}) - \lambda\kappa(\theta_0), \end{aligned}$$

where $\mathbb{I}_n(\theta_0)$ is the fisher information matrix. According to Yi et al. (2011), $n^{-1/2}l'_n(\theta_0) = \mathcal{O}_p(1)$, $\mathbb{I}_n(\theta_0) = \mathcal{O}_p(n)$ and thus the second term has order $\mathcal{O}_p(1)$. The third term is positive and the last term has order $\mathcal{O}_p(c_n^{-1})$. By choosing a sufficiently large C, the second term dominates the first and the last term. It follows that $D_n(\mathbf{u}) > 0$ and $(A.4)$ holds.

Appendix E: Solve the PMLE problem

We choose to solve the PMLE with the sequential unconstrained minimization technique for constrained optimization problems.

Algorithm 1 Optimizing PMLE

1: Set initial value: n=1, λ=10
2: Optimize Q and obtain optimal value for θ^*
3: **if** $\kappa(\mathcal{Z}, \theta^*) = 0$ **then**
4: Return θ^*
5: Terminate
6: **else**
7: $\lambda \leftarrow \lambda \times \lambda_0$, $n \leftarrow n + 1$
8: Go to 2

In this algorithm, the penalty coefficient is enlarged to λ_0^n in iteration n (λ_0 is chosen as 10 in our numerical studies and is shown to work well across different problems). The penalty approach is a classic approach for constrained optimization problem, and the results from this penalty approach will converge to the true solution of the original problem when λ is large enough (Freund 2004). Once we find the solution satisfying the condition in Line 3, we can stop and return this solution. In real applications, it may take quite a number of iterations before reaching this solution. Considering that system optimization is the primary goal and the metamodel is developed to assist in direction search in it (and nor for system inference), a balance of effort should be taken here. To avoid spending too much effort here on estimating penalty coefficients, an upper limit to the number of iterations of PMLE optimization can be added. In our numerical studies, we select the upper limit as 10, since after 10 iteration, the coefficient is already 10^{10}, which is quite large in comparison to the likelihood function value l, and hence, we can expect an acceptable and time-efficient solution to the PMLE.

The PMLE function considered is continuous w.r.t. θ. Note that $\widehat{W}(x, \theta)$ is continuous w.r.t. both x and θ and thus $\min_x \widehat{W}(x, \theta)$ and $\max\{0, |\min_x \widehat{W}(x, \theta)|\}$ are continuous w.r.t. θ. Therefore, the penalty function $P = \lambda\kappa$ is continuous w.r.t θ. Note also that the ordinary likelihood function l is continuous and that $Q = -l + P$. Therefore, the penalized likelihood function Q is continuous w.r.t θ.

Appendix F: An Extended PMLE Approach to Limit the Probability of Crossing

As mentioned in Section 3.2, we observe from extensive tests that with the no-crossing constraint on the predictive mean, the probability of crossing of the entire posterior distributions is quite small. To see this more clearly and verify the practical sufficiency of this constraint, we compute the average probability for the crossing of the posterior distributions in our numerical studies. To compute the average, we repeat each example thirty macro-replications. For each instance, during the optimization process, the co-kriging model is rebuilt in each iteration. We compute the probability of crossing between each two successive levels: $P\left(\widetilde{Z}_{l+1}-\widetilde{Z}_l\leq 0\right)(l=1,\dots m)$. And then take the average of these m-1 probability values (for each two successive levels) as a measure of crossing in this full co-kriging model. We note that as the posterior distribution of $\widetilde{Z}_{l+1},\ \ \widetilde{Z}_l$ is jointly normal, $\widetilde{Z}_{l+1}-\widetilde{Z}_l$ has a normal distribution, and thus this probability can be computed based on the posterior distribution. This is then averaged across 30 macroreplications. The values are summarized in the following table. Through these numerical tests, we find that with this

Table 1 The averaged probability for crossing of the posterior distributions

Example 6.1.1	Example 6.1.2	Ackley	Rastrigin	Levy	Portfolio
4.5620e-07	5.3030e-05	0.0679	0.0965	0.055	2.1184e-06

'non-crossing means' constraint, the crossing of the posterior distributions between different levels has only a small probability of occuring (approximately 0.1 or less). Therefore, we consider this 'non-crossing means' constraint sufficient for making predictions, and we feel it is reasonably adequate for our algorithm when considered together with the results described above.

To fully consider the distributions, we can modify the penalty function to limit the probability of crossing to any given probability level p. We note that the following approach to do this is not used in the current gTSSO-QML algorithm, but it can be treated as an extension of this work. To achieve this, we can introduce a new penalty term $\kappa_2(\mathcal{W},\theta)$ (the terms $\mathcal{W}$, θ have the similar meaning to the penalty function κ in Page 12 of the paper):

$$\kappa_2(\mathcal{W},\theta)=\begin{cases}|\max_{x\in\mathcal{X}}(-\Phi^{-1}(p)-\frac{\widehat{W}(x)}{var(x)})|, & \text{if } \max_{x\in\mathcal{X}}(-\Phi^{-1}(p)-\frac{\widehat{W}(x)}{var(x)})>0\\ 0 & \text{if } \max_{x\in\mathcal{X}}(-\Phi^{-1}(p)-\frac{\widehat{W}(x)}{var(x)})\leq 0\end{cases},$$

where $\hat{W}(x)$ and $var(x)$ are the posterior mean and variance of the difference and Φ is the cdf of the standard normal distribution. This penalty is derived by calculating the cdf of the posterior distribution at 0 and we would like to limit this value to p. Following the similar procedure, this new penalty term can be applied to the multi-level quantile modeling. Compared with the current penalty term κ in Page 12, this penalty requires to additional compute $var(x)$. Moreover, as κ_2 is more restrictive than κ, it is more likely to be violated and require more inner iteration of the PMLE optimization to reduce the violation. Thus, from the practical aspect, we adopt the current constraint.

Appendix G: A Rigorous Approach to Decide the Number of Levels

As mentioned in Section 4, gTSSO-QML is quite robust to the choice of m and thus a moderate number 5 is suggested. Nonetheless, we still provide another more rigorous and objective way to decide m for more broad and practical usage of the algorithm.

As noted in the main paper, the trade-off between the accuracy and complexity has to be noted. Specifically, a larger m can increase the accuracy of the model, but will also increase the co-kriging model complexity. Moreover, we also find that it is not necessary to make m too large, as when m is large, the difference between adjacent intermediate levels are small. In this situation, when the estimate at x for the α_i level is less than C_0, it is likely that the estimate at α_{i+1} level is also smaller than C_0. Therefore, the algorithm can increase its current highest level $h(k)$ by more than one level in each iteration, and thus some intermediate levels will become redundant as they will be skipped by the algorithm. To avoid this, we suggest deciding m from a small value and then increase it by 1 in each step until we find redundant intermediate levels. To achieve this, we choose the value of m with the initial design points after the cross-validation test (which is to determine r_0, C_0 and F. Here, we start from $m=3$ (including the first level, the objective level and one intermediate level) and increases m until there exist any level α_l, which becomes redundant compared with the α_{l-1}-th level. This can be checked as follows. Given m, we can decide the intermediate levels uniformly between α_1 and the objective level. Then, after the initial validation test, for each level l, we calculate A_l which is the number of inital design points whose estimate has noise smaller than C_0 for the α_l-th quantile. If the following set B is non-empty, we deem that there exist redundant levels and we return the current m; Otherwise, there is no redundant levels and we increase m by 1 to introduce more levels and thus increase the correlation between adjacent levels.

$$B := \{l : \ A_l \leq A_{l+1}\}.$$

Initially, when m is small, the difference between levels α_l and α_{l+1} is large and A_l is typically larger than A_{l+1}. However, as m becomes larger, the difference becomes smaller and A_{l+1} approaches A_l. Therefore, when B is non-empty, we some adjacent levels are too close and redundant levels appear. We thus stop increasing m and return the current m. This selection approach for m is presented in Algorithm 2.

Algorithm 2 Choosing m

Set initial value: $m=2$, $\alpha_1 =$ starting level, $B=\emptyset$
while $B=\emptyset$ **do**
    Set $m=m+1$, $\alpha_m =$ objective level
    Compute set A_m
    for $j=2$ to $m-1$ **do**
        Set $\alpha_j = \alpha_1 + \frac{\alpha_m - \alpha_1}{m-1}$
        Compute set A_j
    Update set B
Return m

Appendix H: Sensitivity Analyses of Some Parameters On the Algorithm

Before running the algorithm, there are a few parameters to set. Specifically, the following four parameters must be chosen:

1. r_0: the minimum number of replications for a new design point.
2. m: the number of levels used.
3. C_0: the maximum noise variance of a quantile estimate that can be tolerated.
4. F: the minimum number of design points with acceptable accuracy in E_l to increase $h(k)$ to level l.

Among these four, r_0 is a common parameter for many existing stochastic GP based algorithms (Jalali et al. 2017) and its selection approach (using cross-validation) is well-investigated (Pedrielli et al. 2020). The other three parameters are new in these algorithms and thus the selection of these parameters and their effects on the algorithm should be studied.

In this section, we provide the sensitivity analyses of the algorithm on m, C_0 and F. Specifically, we test different values of these parameters in our numerical studies. Below we provide the results on the portfolio example in Section 6.3 and the 5d Ackley function example in Section 6.2. For the portfolio example, we report the best response obtained with 1,000 replications under different values of parameters. For the Ackley function with larger dimensions and much greater noise, we report the response obtained with 100,000 replications. The results of the other examples are similar and hence the details are not shown here.

H.1. Sensitivity analysis on m

The experiment for each setting of m is conducted 30 times, and the averaged best 0.99 quantile value is plotted in Figure 1 (portfolio example) and 2 (Ackley example). For reference, we also include the case $m = 1$, which is exactly the gTSSO-Q algorithm. We observe that the gTSSO-QML algorithm is quite robust to m. For both examples, gTSSO-QML performs much better than gTSSO-Q. Its performance for y_k for the different cases of $m > 1$ cases is statistically similar (and thus using either method to determine m is acceptable). However, the performance is significantly better when $m = 1$ (gTSSO-Q)

The above experiments all start at an initial level of $\alpha = 0.5$. We next fix $m = 5$ and test the performance for different starting levels of α. The averaged best 0.99 quantile values found are plotted in Figure 3 (the portfolio example) and Figure 4 (the Ackley example). We observe that in general when the starting level is high, the performance becomes worse. This is because the starting level itself then becomes difficult to estimate and optimize, and thus it provides inaccurate information for the higher (and target) levels.

From these sensitivity tests, we find that the gTSSO-QML algorithm is robust to the choice of m, but the starting level should not be too high. Based on our experience from these numerical tests, we recommend a starting level of between 0.5 and 0.6. The quantile estimators at these center quantile values typically have smaller noise variance, and are thus easier to optimize at the start of the algorithm when the number of replications at each design point are small compared with those from much higher levels (we discuss this in Section 1.1).

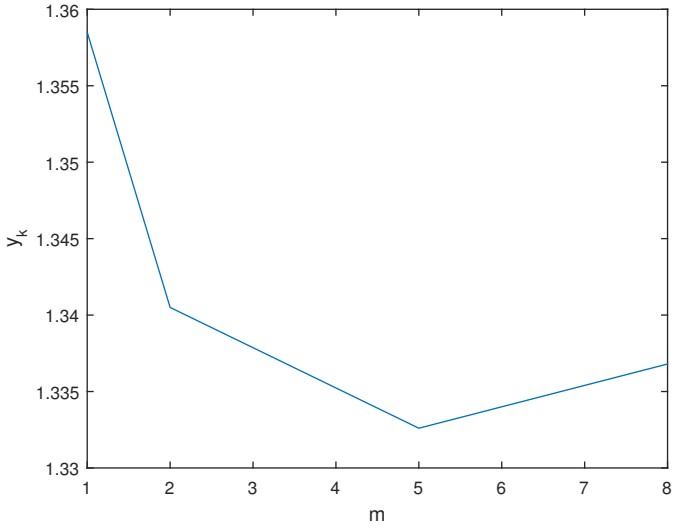

Figure 1 **sensitivity analysis on m in the portfolio example**

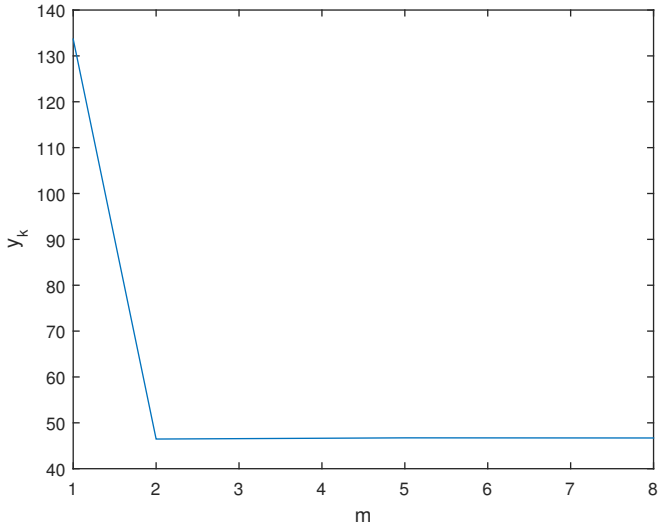

Figure 2 **sensitivity analysis on m in the Ackley example**

H.2. Sensitivity analysis on C_0

An experiment for each setting of C_0 is conducted 30 times and the averaged best 0.99 quantile values found are plotted in Figure 5 (the portfolio example) and Figure 6 (the Ackley example). We note that when C_0 is large, the current highest level $h(k)$ increases quickly to the objective level, without leveraging much of the information from lower levels. This can be observed from Figures 5 and 6, where large C_0 values have poorer performance. From this, we find that C_0 can affect the performance of gTSSO-QML (especially in terms of leveraging information from lower levels), and that larger C_0 values can reduce the benefit of using multiple levels in the first place. Specifically, we note that in the worst case selection of C_0 (with no information gained from lower levels), the algorithm will degenerate to gTSSO-Q.

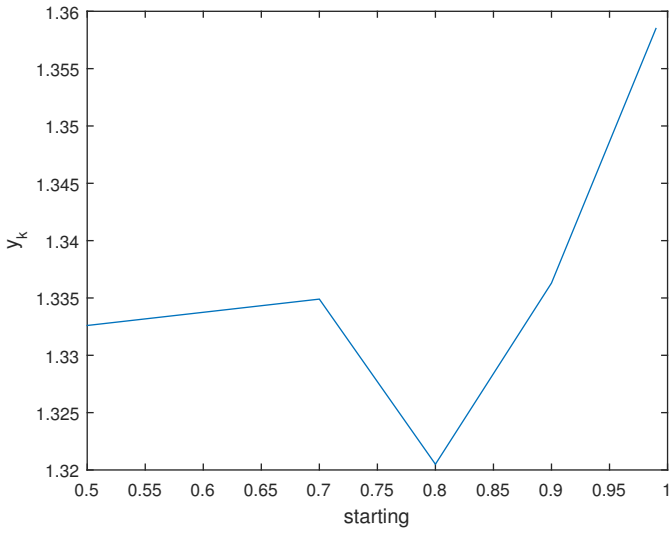

Figure 3 **sensitivity analysis on the starting level in the portfolio example**

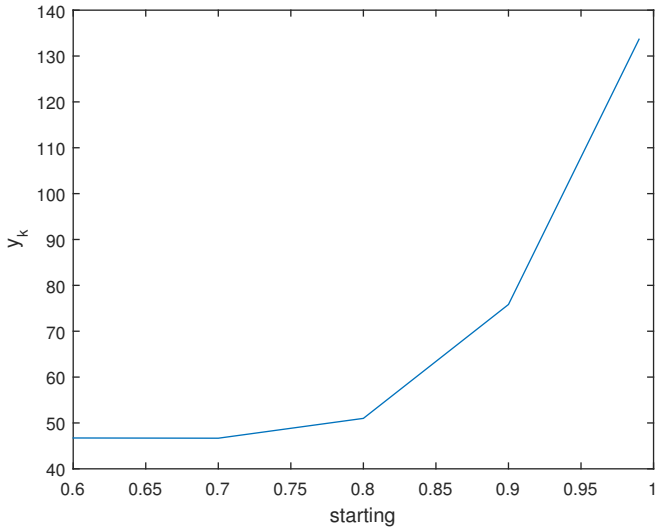

Figure 4 **sensitivity analysis on the starting level in the Ackley example**

Using the recommended approach, the selected values of C_0 are 0.003 in the portfolio example and 0.5 for the Ackley example. As seen from the figures above, these C_0's chosen from the validation approach are (or are close to) the best values found. Hence, the proposed validation-test approach is an easy and good approach to select C_0 that performs well.

H.3. Sensitivity analysis on F

The experiment for each setting of F is conducted 30 times and the averaged best 0.99 quantile value found are plotted in Figure 7 (the portfolio example) and Figure 8 (the Ackley example). Similar to C_0, F controls how fast $h(k)$ increases, and a smaller F indicates a faster increase of $h(k)$. Therefore, we find that when F is smaller, the performance is worse (note that when $F = 0$, gTSSO-QML degenerates to gTSSO-Q). From

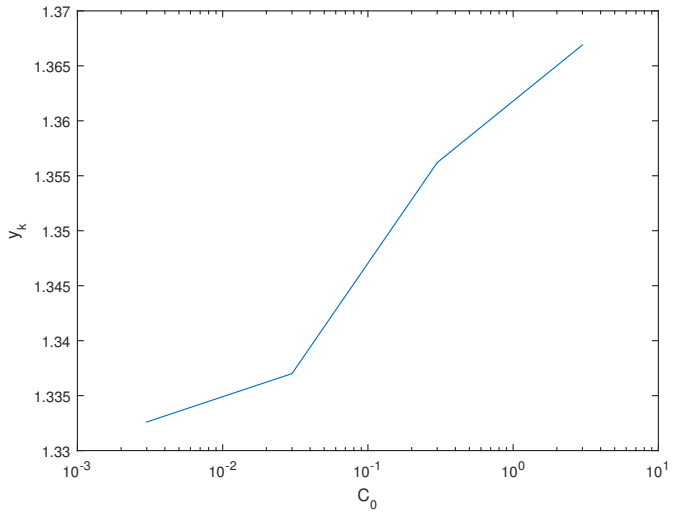

Figure 5 **sensitivity analysis on C_0 in the portfolio example**

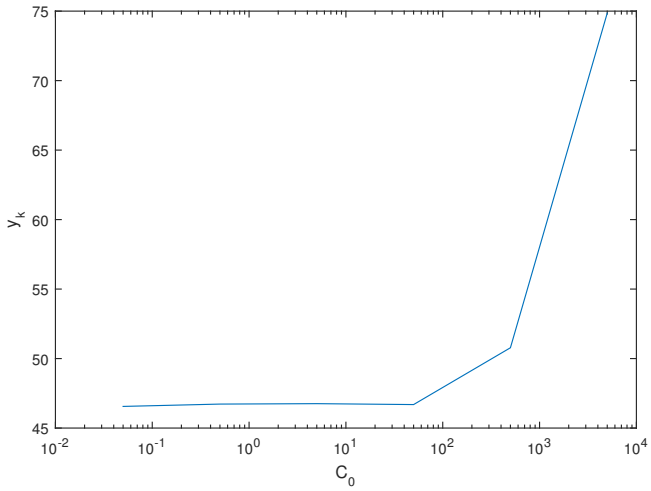

Figure 6 **sensitivity analysis on C_0 in the Ackley example**

this figure, we find that gTSSO-QML is quite robust to F, as long as it is not too small (smaller than 5). We also note that F cannot be chosen to be too large either, since when F is large, the algorithm can become too conservative. In the extreme case where F is larger than the largest number of design points that can be chosen with a given budget, we will never leave the base (starting) level. Using the suggested approach, we choose $F = 10$ in the portfolio example and $F = 12$ in the Ackley example, which have acceptable performance compared with other choices.

Through these sensitivity analyses, we find that the proposed algorithm is quite robust to these parameters. In addition, we also find that the suggested selection approaches of the parameters can provide satisfactory performance of the algorithm.

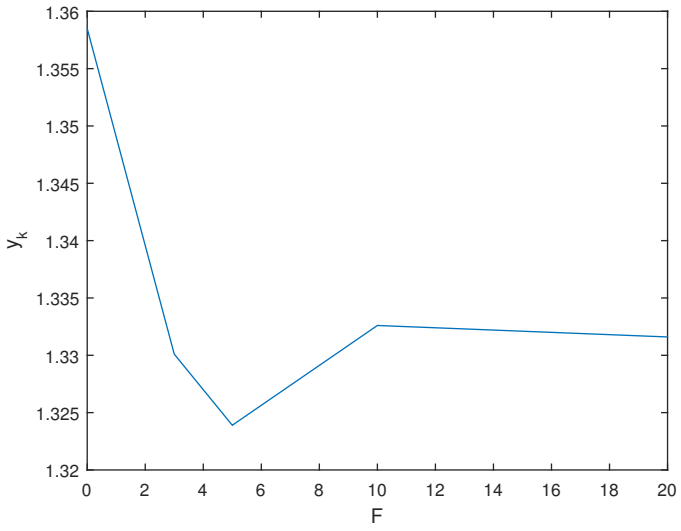

Figure 7 **sensitivity analysis on F in the portfolio example**

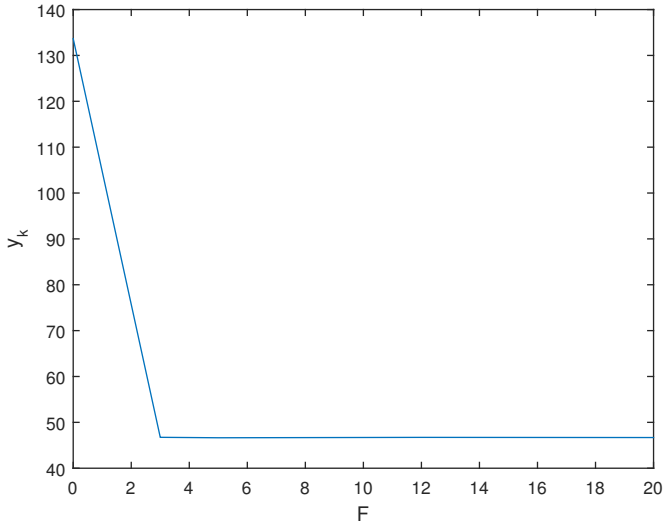

Figure 8 **sensitivity analysis on F in the Ackley example**

Appendix I: The Allocation Rules in the Numerical Studies

In the numerical studies, we test gTSSO with OCBA and equal allocation rules. To satisfy Assumption 2, we select $r_k = 0.1k^{2.01}$. Before stating the details of these two rules, we first introduce how to decide B_k, the number of replications in iteration k.

We follow the recommendation from eTSSO. The budget of Allocation Stage increases with k to refine the point estimates at selected design inputs. This is intuitive since at the beginning, more budget can be used to search the design space and when k gets larger, we are more likely to be in proximity of the promising region. At this point, we can reduce the number of newly selected designs and assign more budget to the already sampled points in the promising region to refine our estimate of the optimum. We, therefore, let

$B_1 = r_0$ and B_k increases with iteration and update its value as follows when $k > 1$:

$$B_k = \max\{\sum_{i=1}^{|D_0|+k} \max\{0, r_k - N_k(x_i)\}, \lfloor B_{k-1}(1 + \frac{\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x'))}{\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x')) + \widehat{s}^2_{h(k)}(x')})\rfloor\},$$

where x' is the design point with the largest number of replications and $\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x'))$ is the sample noise variance of the point estimate at x'. This adaptive scheme B_k is first adopted in the eTSSO algorithm. Its increase is controlled by the relationship between the point estimator noise, measured by $\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x'))$, and the spatial uncertainty of the GP model, measured by the predictive variance $\widehat{s}^2_{h(k)}(x')$ (see equation(3)). At the beginning, when the spatial uncertainty is very large, B_k has a slow growth to save more budget for new design point selection. When the spatial uncertainty gets smaller, i.e., the design space has been better explored, B_k will then experience a faster growth, focusing more on the already selected points in the promising regions. The advantage of this scheme is that it does not require user-defined budgets for each iteration. Furthermore, it can improve the identification of the optimum and lead to efficient use of the computing budget.

This adaptive scheme, however, may experience quite large growth in B_k from our experience. As a result, the optimization process can be overly exploiting. This is also observed in Jalali et al. (2017). When there is limited budget, we may want to be less conservative and explore more regions. In this case, we further fix the adaptive scheme as follows:

$$B_k = \max\{\sum_{i=1}^{|D_0|+k} \max\{0, r_k - N_k(x_i)\}, \min\{\lfloor B_{k-1}(1 + \frac{\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x'))}{\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x')) + \widehat{s}^2_{h(k)}(x')})\rfloor, \beta_k k\}\}.$$

This scheme actually restricts the growth of B_k to linear growth as the iteration. We highlight that only when the budget is limited or relatively small, we recommend this scheme. As the budget in experiments in Section 6.2 is much larger than that in Section 6.3, we choose $\beta_k = 100$ in Section 6.2 and $\beta_k = 10$ in Section 6.3.

We also note that any possible budget increasing schemes, other than the one in eTSSO, that satisfy Condition 1 can also be applied. For example, we can choose $B_k = r_0 + 0.1(D_0| + k - 1)(k^{2.01} - (k-1)^{2.01})$, which is more conservative in the growth of B_k, together with the chosen r_k for equal allocations.

After updating B_k and checking that each design point has at least r_k replications, we allocate remaining replications to the selected design inputs with the selected allocating rule. For OCBA, denote x_b as the current best design point in D_k with respect to the current highest level: $x_b \in \arg\min_{x \in D_k} \mathcal{Y}_{h(k)}(x)$. OCBA decides the number of new replications n_i assigned to each input $x_i \in D_k$ as follows:

$$\frac{n_i}{n_j} = \frac{\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x_i))/\lambda^2_{b,i}}{\widehat{\text{var}}(\mathcal{Y}_{h(k)}(x_j))/\lambda^2_{b,j}}, \quad (i, j \neq b); \quad n_{best} = \widehat{\text{var}}(\mathcal{Y}_{h(k)}(x_b))\sqrt{\sum_{i \neq b} \frac{n_i^2}{\widehat{\text{var}}^2(\mathcal{Y}_{h(k)}(x_i))}},$$

where $\lambda_{b,i} := \mathcal{Y}_{h(k)}(x_i) - \mathcal{Y}_{h(k)}(x_b)$. The OCBA technique actually prefers allocating additional replications to points with low response values and large noises, which is intended to refine the point estimates at promising regions and those with large noise variances. The equal allocation rule is much simpler, where the remaining replications are equally allocated to all design points.

Appendix J: Proof of Lemma 1

At iteration k, all selected design points have been allocated at least $r_k > k$ iterations in the proposed algorithm. Consider a design point $x_0 \in D_k$. Recall that $(A.2)$ states that

$$\mathcal{Y}_m(x_0) = v_{\alpha_m}(L(x_0)) + \frac{1}{N_k(x_0)} \sum_{j=1}^{N_k(x_0)} \psi_m(L(x_0,\xi_j)) + R_{N_k(x_0)}, \quad \psi_m(L(x_0,\xi_j)) = \frac{\alpha_m - 1_{\{L(x_0,\xi_j) \le v_{\alpha_m}(L(x_0))\}}}{f_{x_0}(v_{\alpha_m}(L(x_0)))},$$

where $N_k(x_0)$ is the number of replications assigned to x_0 by iteration k. It follows that $\mathcal{Y}_m(x_0)$ has variance:

$$\mathrm{var}(\mathcal{Y}_m(x_0)) = \frac{1}{N_k(x_0)} \mathrm{var}(\psi_m(L(x_0,\xi))) + \mathrm{var}(R_{N_k(x_0)}) = \frac{\alpha_m(1-\alpha_m)}{N_k(x_0) f^2_{x_0}(v_{\alpha_m}(L(x_0)))} + \mathrm{var}(R_{N_k(x_0)}).$$

According to Duttweiler (1973), $\mathbb{E}[R^2_{N(x_0)}] \approx N(x_0)^{-3/2} f^{-2}_{x_0}(v_{\alpha_m}(L(x_0)))(2\alpha_m(1-\alpha_m)/\pi)^{1/2}$. Recall that $f_x(v_{\alpha_m}(L(x))) > f^*$ for all $x \in \mathcal{X}$. Under Assumption 2, $N(x_0) \ge r_k > k$ for all $x \in D_k$ in iteration k. Therefore, for all $x_0 \in D_k$,

$$\mathrm{var}(\mathcal{Y}_m(x_0)) \le \frac{\alpha_m(1-\alpha_m)}{N_k(x_0) f^2_x(v_{\alpha_m}(L(x)))} + \mathbb{E}[R^2_{N(x_0)}] \le \frac{\alpha_m(1-\alpha_m)}{r_k (f^*)^2} + \frac{(2\alpha_m(1-\alpha_m))^{1/2}}{r_k^{3/2}(f^*)^2 \pi^{1/2}} := \varrho_k. \tag{A.5}$$

The first inequality holds since $\mathrm{var}(R_{N_k(x_0)}) \le \mathbb{E}[R^2_{N(x_0)}]$. We see that ϱ_k does not depend on x_0 and furthermore $\varrho_k \to 0$ as $k \to \infty$. Note that P_k is an upper bound for $\mathrm{var}(\mathcal{Y}_m(x_0))$ in iteration k for all $x_0 \in D_k$. It follows that, the noise variance for the α_m-quantile estimators at all design points in D_k tends to zero uniformly as $k \to \infty$. Therefore, there exists a large value K that does not depend on x such that when $k > K$, $\mathrm{var}(\mathcal{Y}_m(x_0)) < C_0$ for any design point $x_0 \in D_k$ (under Assumption 3(iii), we may use the true value of $\mathrm{var}(\mathcal{Y}_m(x_0))$, instead of its estimate (4), to examine the quality of the point estimate). Therefore, in iterations $k > K$ of eTSSO-QML, we only use a single-level model for the objective level. In this case, the model at the objective level is accurate enough such that we may optimize it without leveraging the lower levels.

Appendix K: Proof of Lemma 2

According to Lemma 1, there exists a large number K such that for iterations $k > K$, $\mathrm{var}(\mathcal{Y}_m(x_0)) < C_0$ for any design point $x_0 \in D_k$ and thus eTSSO-QML adopts a single-level model for the objective level. In this proof, we suppose $k > K$ and omit the subscript l in equations (2) and (3). Denote the EI function in iteration k as $T_k(x)$ where:

$$T_k(x) = \widehat{s}_k(x)\phi(\frac{y^*_k - \widehat{\mu}_k(x)}{\widehat{s}_k(x)}) + (y^*_k - \widehat{\mu}_k(x))\Phi(\frac{y^*_k - \widehat{\mu}_k(x)}{\widehat{s}_k(x)}),$$

where y^*_k is the current best objective value. For ease of exposition, we write $\widehat{\mu}_k(x)$ and $\widehat{s}^2_k(x)$ (as there is only one level α_m here, the subscript in $s^2_k(x)$ does not represent the level but the iteration number) as to denote the predictor (2) and predictive variance (3) obtained from the single-level GP model for the objective level.

The proof of this Lemma follows that of Theorem 1 from Locatelli (1997). It can be divided into three parts. In Section K.1, we find an upper bound for $T_k(x)$ at any unobserved point $x \in \mathcal{X} \setminus D_k$. This upper bound depends on the the nearest design point to x. Intuitively, if x is very close to a design point x_0, the uncertainty at x should be small since its response has a large correlation with x_0. As a result, the expected improvement we get from observing the response at x should be small. In Section K.2, we show how to construct a region around any design point where $T_k(x)$ is bounded above by a threshold c. Finally, in Section proLe23, we apply Lemma 1 and Theorem 1 from Locatelli (1997) to prove that the design points are dense.

K.1. Upper bound for $T_k(x)$

According to Assumption 3(i), the true baseline quantile function is bounded. We may select a large enough value M such that the predictor $\widehat{\mu}_k(x)$ and the true quantile value $v_{\alpha_m}(x)$ are constrained in $(-M, M)$, for all $x \in \mathcal{X}$, for all k (in practice, with a loose bound $(-M, M)$, we can set $\widehat{\mu}_k(x) = -M$ if $\widehat{\mu}_k(x) < -M$ and set $\widehat{\mu}_k(x) = M$ if $\widehat{\mu}_k(x) > M$. This avoids severe underestimation or overestimation caused by a limited number of replications (Sun et al. 2014)).

In iteration $k > K$, consider an unknown point x, then we must have:

$$y_k^* - \widehat{\mu}_k(x) < 2M.$$

Through simple computation of the partial derivatives of $T_k(x)$, we find that $\frac{\partial T_k(x)}{\partial (y_k^* - \widehat{\mu}_k(x))} > 0$ and $\frac{\partial T_k(x)}{\partial \widehat{s}_k(x_0)} > 0$. As $T_k(x)$ is an increasing function of $y_k^* - \mu_k(x)$, we have:

$$T_k(x) \le \widehat{s}_k(x)\phi(\frac{2M}{\widehat{s}_k(x)}) + 2M\Phi(\frac{2M}{\widehat{s}_k(x)}) := P_k(x).$$

Define $x_0 := \arg\max_{x' \in D_k} \text{corr}(x, x')$, which is the design point with the largest correlation with x. Denote the covariance matrix in (2) as $R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$, where $R_{11} = \sigma^2 + \text{var}(\mathcal{Y}_m(x_0))$ is the variance at x_0, $R_{21} = R_{12}^T$ is the $(|D_k| - 1) \times 1$ covariance vector of the spatial process between x_0 and the remaining design points $D_k \setminus \{x_0\}$, and R_{22} is the $(|D_k| - 1) \times (|D_k| - 1)$ covariance matrix at $D_k \setminus \{x_0\}$. Further denote $t(x) = (\sigma^2 \text{corr}(x, x_0), t_2^T(x))^T$, where $t_2(x)$ is the $(|D_k| - 1) \times 1$ response covariance vector between x and $D_k \setminus \{x_0\}$. We find that for iteration $k > K$

$$\begin{aligned}
\widehat{s}_k^2(x) &= \sigma^2 - t(x)^T R^{-1} t(x) + \zeta(x)^T (H^T R_z^{-1} H)^{-1} \zeta(x) \\
&= \sigma^2 - \frac{\sigma^4 \text{corr}(x, x_0)^2}{R_{11} + \text{var}(\mathcal{Y}_m(x_0))} - \Gamma^T P \Gamma + \zeta(x)^T (H^T R_z^{-1} H)^{-1} \zeta(x) \\
&\le \sigma^2 - \frac{\sigma^4}{\sigma^2 + \varrho_k} \text{corr}(x, x_0)^2 + \zeta(x)^T (H^T R_z^{-1} H)^{-1} \zeta(x),
\end{aligned}$$

where $\Gamma = R_{21} R_{11}^{-1} \sigma^2 \text{corr}(x, x_0) - t_2(x)$ and $P^{-1} = R_{22} - R_{21} R_{11}^{-1} R_{12}$. The inequality holds since P^{-1} is positive definite (as P is a covariance matrix of the responses at $D_k \setminus \{x_0\}$ given x_0, which is symmetric and positive definite). For the last term, we recall that $\zeta(x) = h(x) - t(x)^T R_z^{-1} H$ from (3). For the commonly used constant mean function $h(x) = 1$, $t(x)^T R_z^{-1} H$ is the GP prediction at x given observation vector H. Following the same procedure as in Appendix D, we have that $h(x_0) - t(x)^T R_z^{-1} H = \mathcal{O}(|x - x_0|)$. As $h(x_0) = h(x) = 1$, it follows that $|\zeta(x)| = |h(x) - t(x)^T R_z^{-1} H| = \mathcal{O}(|x - x_0|)$. In this case, we can select a value M_1 such that $|\zeta(x)| < M_1 |x - x_0|$. Moreover, we can check that $(H^T R_z^{-1} H)^{-1} < \sigma^2 + C_0$. Therefore $\zeta(x)^T (H^T R_z^{-1} H)^{-1} \zeta(x) = (H^T R_z^{-1} H)^{-1} |\zeta(x)|^2 < M_1^2 (\sigma^2 + C_0) |x - x_0|^2$. Denote M_2 as $M_1^2(\sigma^2 + C_0)$, then we have $\zeta(x)^T (H^T R_z^{-1} H)^{-1} \zeta(x) < M_2 |x - x_0|^2$. For a general mean function h, the proof follows the same reasoning.

Define $\widehat{s}_{k0}^2(x; x_0) := \sigma^2 - \frac{\sigma^4}{\sigma^2 + \varrho_k} \text{corr}(x, x_0)^2 + M_2 |x - x_0|^2$. We can see that $\widehat{s}_k^2(x) \le \widehat{s}_{k0}^2(x; x_0)$. Moreover, as $\text{corr}(x, x_0)$ increases as the distance between x and x_0 decreases, we see that $\widehat{s}_{k0}^2(x; x_0)$ increases as the distance between x and x_0 increases.

As $P_k(x)$ is an increasing function of $\widehat{s}_k(x)$, we have:

$$T_k(x) \le P_k(x) \le \widehat{s}_{k0}(x; x_0)\phi(\frac{2M}{\widehat{s}_{k0}(x; x_0)}) + 2M\Phi(\frac{2M}{\widehat{s}_{k0}(x; x_0)}) := Q_k(x; x_0).$$

Since $\frac{\partial Q_k(x; x_0)}{\partial \widehat{s}_{k0}(x; x_0)} > 0$ and the fact that $\widehat{s}_{k0}^2(x; x_0)$ increases as the distance between x and x_0 increases, we see that $Q_k(x; x_0)$ increases as the distance between x and x_0 increases.

K.2. Local Region Centered at Design Points with Bounded $T_k(x)$

Considering any design point $x_0 \in D_k$. Based on the bound $Q_k(x; x_0)$, we may construct a region, $R(x_0, c)$, containing x_0 defined by:

$$R(x_0, c) = \{x \in \mathcal{X} | Q_k(x; x_0) < c\}.$$

From the proof in Section K.1, we find that $Q_k(x; x_0)$ decreases as the distance between x and x_0 decreases and that $Q_k(x; x_0) \to 0$ as $x \to x_0$. Therefore, for any value of c, $R(x_0, c)$ is a region centered at x_0, such that $T_k(x) < c$, for all $x \in R(x_0, c)$.

K.3. Proof of Density

To prove the density of the design points, we deploy Lemma 1 and Theorem 1 from Locatelli (1997). Specifically, we consider the following stopping rule in the algorithm:

– *Stopping Rule*: The algorithm stops when the maximum of $T_k(x)$ is smaller than some pre-defined threshold c.

We can see that with this stopping rule, the points in $R(x_0, c)$ will never be selected in iteration $k > K$. From Theorem 1 in Locatelli (1997), the algorithm will terminate within a finite number of design points for any given c. However, we assume infinite budget so that the algorithm does not stop within finitely many iterations. To achieve this, similar to the procedure in Locatelli (1997), once the algorithm stops, we decrease the value of threshold c to ensure that the maximum of $T_k(x)$ is larger that the updated c. Thus, the condition of the stopping rule is not met and the algorithm continues. To finish the proof, we directly use the results of Lemma 1 from Locatelli (1997) that the design points will be dense everywhere in $\mathcal{X}$ if the threshold value c keeps decreasing.

The above result is proved with known parameters σ^2 and θ. However, when the parameters are estimated, the above proof still goes through under Assumption 3(ii). Specifically, the estimated values of these parameters will influence the size of the region $R(x_0, c)$. With bounded values of $\widehat{\sigma}^2$ and $\widehat{\theta}$, the region should be nonempty and the proof will continue to hold.

Appendix L: Proof of Theorem 3

We prove that $\widehat{\mathcal{Y}}_k \to v_{\alpha_m}(L(x^*))$ w.p.1 as $k \to \infty$. Equivalently, we prove that $\lim_{n\to\infty} \mathbb{P}(\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x^*))| > \delta\}) = 0$, for all $\delta > 0$. Recall that $\widehat{\mathcal{Y}}_k = \mathcal{Y}_m(\widehat{x}_k)$, where $\widehat{x}_k = \arg\min_{x\in D_k} \mathcal{Y}_m(x)$ is the observed best point within the design set. Define $x_k^* := \arg\min_{x\in D_k} v_{\alpha_m}(L(x))$, which is the true best point within the design set. This proof is divided into three parts. In Section L.1, we prove that $\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*)) \to 0$ w.p.1. This is to correctly identify the best points within the design set. In Section L.2, we prove that $v_{\alpha_m}(L(x_k^*)) \to v_{\alpha_m}(L(x^*))$, which ensures the true optimum within the design set tends to the true global minimum. In Section L.3, we combine the proofs from L.1 and L.2 to finish the convergence proof.

L.1. Proof that $\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*)) \to 0$ w.p.1 as $k \to \infty$

We prove that $\lim_{n\to\infty} \mathbb{P}(\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}\}) = 0$ for all $\delta > 0$. To show this, we verify the following sufficient condition $\sum_{k=1}^{\infty}\mathbb{P}[|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*)) > \frac{\delta}{2}|] < \infty$ (Theorem 7.5, Pishro-Nik (2016)). For all $\delta > 0$,

$$\begin{aligned}
&\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k) - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}] \\
=&\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k) - v_{\alpha_m}(L(\widehat{x}_k)) + v_{\alpha_m}(L(\widehat{x}_k)) - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}] \\
<&\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k) - v_{\alpha_m}(L(\widehat{x}_k))| > \frac{\delta}{4}] + \mathbb{P}[|v_{\alpha_m}(L(\widehat{x}_k)) - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{4}]. \qquad (A.6)
\end{aligned}$$

We bound the first term in $(A.6)$ as follows. Under Assumption 2, the accumulated number of replications at each input $x \in D_k$, $N_k(x) > r_k$ and $r_k \to \infty$ as $k \to \infty$. According to Bahadur's representation (Kiefer 1967):

$$\mathcal{Y}_m(x) = v_{\alpha_m}(L(x)) + \frac{1}{N(x)}\sum_{i=1}^{N(x)}\psi(L(x,\xi_i)) + R_{N(x)},$$

where $N(x)$ is the total number of replications at x, $\psi(L(x,\xi_i)) := \frac{\alpha_m - \mathbf{1}_{\{L(x,\xi_i) < v_{\alpha_m}(L(x))\}}}{f_x(v_{\alpha_m}(L(x)))}$ and $R_{N(x)}$ is the remainder term. Therefore, for all $x \in D_k$, we have that

$$\begin{aligned}
\mathbb{P}[|\mathcal{Y}_m(x) - v_{\alpha_m}(L(x))| > \frac{\delta}{4}] &= \mathbb{P}[|\frac{1}{N(x)}\sum_{i=1}^{N(x)}\psi(L(x,\xi_i)) + R_{N(x)}| > \frac{\delta}{4}] \\
&< \mathbb{P}[|\frac{1}{N(x)}\sum_{i=1}^{N(x)}\psi(L(x,\xi_i))| > \frac{\delta}{8}] + \mathbb{P}[|R_{N(x)}| > \frac{\delta}{8}].
\end{aligned}$$

Through simple computation, we find $\mathbb{E}[\psi(L(x,\xi_i))] = 0$ and $\text{var}[\psi(L(x,\xi_i))] = \frac{\alpha(1-\alpha)}{f_x^2(v_{\alpha_m}(L(x)))}$, and thus by Chebychev's inequality, we have

$$\mathbb{P}[|\frac{1}{N(x)}\sum_{i=1}^{N(x)}\psi(L_i(x))| > \frac{\delta}{8}] < \frac{64\alpha(1-\alpha)}{N(x)\delta^2 f_x^2(v_{\alpha_m}(L(x)))} < \frac{64\alpha(1-\alpha)}{r_k\delta^2(f^*)^2},$$

(recall $f^* < f_x(v_{\alpha_m}(L(x)))$ for all $x \in \mathcal{X}$ by Assumption 3(i)). On the other hand, for the remainder $R_{N(x_0)}$, we have

$$\mathbb{P}[|R_{N(x)}| > \frac{\delta}{8}] = \mathbb{P}[|R_{N(x)}|^2 > \frac{\delta^2}{64}] \le \frac{\mathbb{E}[R_{N(x)}^2]}{\delta^2/64} < \frac{64(2\alpha_m(1-\alpha_m))^{1/2}}{\delta^2 r_k^{3/2}(f^*)^2\pi^{1/2}}.$$

The first inequality holds by Chebychev's inequality and the second inequality holds by the same reasoning with $(A.5)$. It follows that for all $x \in D_k$

$$\mathbb{P}[|\mathcal{Y}_m(x) - v_{\alpha_m}(L(x))| > \frac{\delta}{4}] < \frac{64\alpha(1-\alpha)}{r_k\delta^2(f^*)^2} + \frac{64(2\alpha_m(1-\alpha_m))^{1/2}}{\delta^2 r_k^{3/2}(f^*)^2\pi^{1/2}} < \frac{64}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha) + \frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right).$$

Therefore,

$$\begin{aligned}
&\mathbb{P}[\max_{x\in D_k}|\mathcal{Y}_m(x) - v_{\alpha_m}(L(x))| > \frac{\delta}{4}] \\
\le &\sum_{i=1}^{k+|D_0|}\mathbb{P}[|\mathcal{Y}_m(x_i) - v_{\alpha_m}(L(x_i))| > \frac{\delta}{4}] \\
< &\frac{64(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha) + \frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right).
\end{aligned}$$

With this inequality, we see that

$$\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k)-v_{\alpha_m}(L(\widehat{x}_k))|>\frac{\delta}{4}]\le\mathbb{P}[\max_{x\in D_k}|\mathcal{Y}_m(x)-v_{\alpha_m}(L(x))|>\frac{\delta}{4}]<\frac{64(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right),$$

$$\mathbb{P}[|\mathcal{Y}_m(x_k^*)-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{4}]\le\mathbb{P}[\max_{x\in D_k}|\mathcal{Y}_m(x)-v_{\alpha_m}(L(x))|>\frac{\delta}{4}]<\frac{64(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right).$$

Now we bound the second term in (*A*.6). Define sets $A_k:=\{|\mathcal{Y}_m(\widehat{x}_k)-v_{\alpha_m}(L(\widehat{x}_k))|\le\frac{\delta}{9}\}$ and $B_k:=\{|\mathcal{Y}_m(x_k^*)-v_{\alpha_m}(L(x_k^*))|\le\frac{\delta}{9}\}$ for all $k\ge0$. We note that

$$\begin{aligned}&\mathbb{P}[|v_{\alpha_m}(L(\widehat{x}_k))-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{4}]\\=&\mathbb{P}[\{|v_{\alpha_m}(L(\widehat{x}_k))-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{4}\}\cap\{A_k\cap B_k\}]+\mathbb{P}[\{|v_{\alpha_m}(L(\widehat{x}_k))-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{4}\}\cap\{A_k\cap B_k\}^{\complement}].\end{aligned}$$

We prove that the first term is zero by contradiction. When $v_{\alpha_m}(L(\widehat{x}_k))-v_{\alpha_m}(L(x_k^*))\ge\frac{\delta}{4}$, as $|\mathcal{Y}_m(\widehat{x}_k)-v_{\alpha_m}(L(\widehat{x}_k))|\le\frac{\delta}{9}$ (set A_k) and $|\mathcal{Y}_m(x_k^*)-v_{\alpha_m}(L(x_k^*))|\le\frac{\delta}{9}$ (set B_k), it must be that $\mathcal{Y}_m(\widehat{x}_k)>\mathcal{Y}_m(x_k^*)$. This inequality contradicts the fact that $\widehat{x}_k$ is the best observed point at iteration k, i.e., $\widehat{x}_k=\arg\min_{x\in D_k}\mathcal{Y}_m(x)$. It follows that the first term is 0. For the second term, we see that

$$\begin{aligned}&\mathbb{P}[\{|v_{\alpha_m}(L(\widehat{x}_k))-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{4}\}\cap\{A_k\cap B_k\}^{\complement}]\\<&\mathbb{P}[\{A_k\cap B_k\}^{\complement}]=1-\mathbb{P}[A_k\cap B_k]<2-\mathbb{P}[A_k]-\mathbb{P}[B_k]<\frac{648(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right).\end{aligned}$$

The last inequality follows because $1-\mathbb{P}[A_k]=\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k)-v_{\alpha_m}(L(\widehat{x}_k))|>\frac{\delta}{9}]<\frac{324(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right)$ (and similarly for $1-\mathbb{P}[B_k]$). Therefore,

$$\mathbb{P}[|v_{\alpha_m}(L(\widehat{x}_k))-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{4}]<\frac{648(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right).$$

As a result,

$$\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k)-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{2}]<\frac{712(k+|D_0|)}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right).$$

By Assumption 2, we have $\sum_{k=1}^{\infty}\frac{k}{r_k}<\infty$. With this assumption, we see that $\sum_{k=1}^{\infty}\frac{|D_0|}{r_k}=|D_0|\sum_{k=1}^{\infty}\frac{1}{r_k}<\infty$. Therefore,

$$\sum_{k=1}^{\infty}\mathbb{P}[|\mathcal{Y}_m(\widehat{x}_k)-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{2}]<\frac{712}{r_k\delta^2(f^*)^2}\left(\alpha(1-\alpha)+\frac{(2\alpha_m(1-\alpha_m))^{1/2}}{\pi^{1/2}}\right)\sum_{k=1}^{\infty}\frac{(k+|D_0|)}{r_k}<\infty.$$

It follows that $\lim_{n\to\infty}\mathbb{P}(\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k-v_{\alpha_m}(L(x_k^*))|>\frac{\delta}{2}\})=0$.

L.2. Proof that $v_{\alpha_m}(L(x_k^*))\to v_{\alpha_m}(L(x^*))$ w.p.1 as $k\to\infty$

According to Theorem 1.3 from Torn and Zilinskas (1989), for a deterministic search (given starting point x_0, the design points are determined), the algorithm converges if the design points are everywhere dense, i.e., $v_{\alpha_m}(L(x_k^*))\to v_{\alpha_m}(L(x^*))$ if D_k is dense in $\mathcal{X}$. When the design points are random (x_k and x_k^* are random variables), denseness of the design points is not sufficient to guarantee almost sure convergence. We next prove that in our algorithm, $v_{\alpha_m}(L(x_k^*))\to v_{\alpha_m}(L(x^*))$ as $k\to\infty$ w.p.1. Equivalently, we prove that for all $\delta>0$, $\mathbb{P}[|v_{\alpha_m}(L(x_k^*))-v_{\alpha_m}(L(x^*))|>\delta,i.o.]=0$ (*i.o.* is shorthand for infinitely often).

For $\epsilon>0$, we can select a region S around x^* such that for all $x\in S$, $|v_{\alpha_m}(L(x))-v_{\alpha_m}(L(x^*))|\le\epsilon$ (under the assumption that the baseline function $v_{\alpha_m}(L(x))$ is continuous). We next prove that there exists a

large value K_1 such that at least one design point is selected in S before iteration K_1. It then follows that $\mathbb{P}[|v_{\alpha_m}(L(x_k^*)) - v_{\alpha_m}(L(x^*))| > \epsilon, i.o.] = 0$.

If any points in S are selected in some iteration $k \leq K_1 - 1$, the condition holds. Now suppose no points in S are selected before iteration K_1. In this case we can find a lower bound $\widehat{s}_0^2$ for the predictive variance $\widehat{s}_{K_1}^2(x^*)$ at x^*, which is the value of $\widehat{s}_{K_1}^2(x^*)$ if all the design points in $\mathcal{X} \setminus S$ are observed with no noise. Subsequently, we see

$$T_{K_1}(x^*) > \widehat{s}_0 \phi(\frac{-2M}{\widehat{s}_0}) - 2M\Phi(\frac{-2M}{\widehat{s}_0}) := t_0.$$

In other words, we can find a lower bound for the EI function value at x^*, t_0. Note that t_0 is the EI function value if the predictive response value is $2M$ larger than the current best value and the predictive variance is $\widehat{s}_0$. As the EI function is always positive if $\widehat{s}_0 > 0$, we see that $t_0 > 0$. From the proof in Section K, we see that if we keep reducing the value of c in the stopping rule to t_0, then within a finite number of iterations, the EI function values at all points in $\mathcal{X} \setminus S$ will become smaller than t_0. As a result, when K_1 is large enough, $T_{K_1}(x) < t_0$ for all $x \in \mathcal{X} \setminus S$ while $T_{K_1}(x^*) > t_0$. Therefore, the next design point must belong to S, and we finish the proof.

L.3. Proof that $\widehat{\mathcal{Y}}_k \to v_{\alpha_m}(L(x^*))$ w.p.1 as $k \to \infty$

Since

$$\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x^*))| > \delta\} \subset \{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}\} \cup \{|v_{\alpha_m}(L(x_k^*)) - v_{\alpha_m}(L(x^*))| > \frac{\delta}{2}\},$$

it follows that

$$\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x^*))| > \delta\} \subset \left\{\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}\}\right\} \cup \left\{\cup_{k=n}^{\infty}\{|v_{\alpha_m}(L(x_k^*)) - v_{\alpha_m}(L(x^*))| > \frac{\delta}{2}\}\right\}.$$

From Section L.1, we have that, for all $\delta > 0$,

$$\lim_{n\to\infty} \mathbb{P}(\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}\}) = 0.$$

Moreover, since $v_{\alpha_m}(L(x_k^*)) \to v_{\alpha_m}(L(x^*))$ as $k \to \infty$ w.p.1 (Section L.2), we have that, for all $\delta > 0$,

$$\lim_{n\to\infty} \mathbb{P}(\cup_{k=n}^{\infty}\{|v_{\alpha_m}(L(x_k^*)) - v_{\alpha_m}(L(x^*))| > \frac{\delta}{2}\}) = 0.$$

Therefore,

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{P}[\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x^*))| > \delta\}] \\
&\leq \lim_{n\to\infty} \mathbb{P}[\{\cup_{k=n}^{\infty}\{|\widehat{\mathcal{Y}}_k - v_{\alpha_m}(L(x_k^*))| > \frac{\delta}{2}\}\}] + \lim_{n\to\infty} \mathbb{P}[\{\cup_{k=n}^{\infty}\{|v_{\alpha_m}(L(x_k^*)) - v_{\alpha_m}(L(x^*))| > \frac{\delta}{2}\}\}] = 0.
\end{aligned}$$

It follows that $\widehat{\mathcal{Y}}_k \to v_{\alpha_m}(L(x^*))$ w.p.1 as $k \to \infty$.

Appendix M: Details of the Numerical Studies

M.1. The parameters selected for the numerical experiments

We provides some parameters chosen with the suggested approaches for our numerical studies in Table 2.

M.2. The random seeds for the numerical experiments

Below in Table 3 we provide the random seeds we use for different experiments in 'MatLab'.

Table 2 The parameters selected for the numerical experiments

Parameters	Example 6.1.1	Example 6.1.2	Ackley/Levy/Rastrigin	Portfolio
r_0	20	20	50	50
C_0	1.5	20	0.5	0.003
F	6	6	12	10
m	2	2	5	5

Table 3 The random seeds for the numerical experiments

Example 6.1.1	Example 6.1.2	Ackley/Levy/Rastrigin	Portfolio
666	666	666	666666

M.3. The candidate points set and initial design points set

The section provides the candidate points and initial design points sets for the numerical experiments.

M.3.1. Example 6.1.1 and Example 6.1.2 The candidate points set is generated with 1000 evenly spaced point in [0,1] and the initial design point set consists of 6 evenly spaced points in [0,1].

M.3.2. Ackley/Levy/Rastrigin Example The candidate points are generated as follows:

- Generate 10000 points in $[-10, 10]^5$ with Latin Hypercube design strategy
- Generate another 50 points in $[0, 1]^5$ with Latin Hypercube design strategy

The initial design points are generated with 25 Latin Hypercube design points. These two sets are provided in the *pointsA.mat* file in the online supplements.

M.3.3. Portfolio Example The candidate points are generated as follows. Each one of the two dimensions is discretized with 40 evenly spaced points and the Cartesian product of these points are used as the candidate points. The initial design points are 9 evenly spaced points. These points are provided in the *pointsB.mat* file in the online supplements.

We also note that the true quantile values have no closed form. We thus estimate these values with 5×10^8 simulations at each point: at each points, we generate the 5×10^8 random function values and the sample 0.99 quantile is treated as the true quantile value. These values are also provided in *pointsB.mat*.

Appendix N: The test functions in Section 6.2

The test functions F_1 to F_3 used are (in the d-dimensional input space):

$$\text{Ackley: } F_1(x) = -20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)) + 20 + \exp(1).$$

$$\text{Rastrigin: } F_2(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)].$$

$$\text{Levy : } F_3(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{d-1}(\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)],$$

$$\text{where, } \omega_i = 1 + \frac{x_i - 1}{4}, \forall i = 1, ..., d.$$

References

Bahadur RR (1966) A note on quantiles in large samples. *The Annals of Mathematical Statistics* 37(3):577–580.

Chen X, Kim KK (2016) Efficient var and cvar measurement via stochastic kriging. *INFORMS Journal on Computing* 28(4):629–644.

Duttweiler D (1973) The mean-square error of bahadur's order-statistic approximation. *The Annals of Statistics* 446–453.

Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association* 96(456):1348–1360.

Freund RM (2004) Penalty and barrier methods for constrained optimization. *Lecture Notes, Massachusetts Institute of Technology* .

Fricker TE, Oakley JE, Urban NM (2013) Multivariate gaussian process emulators with nonseparable covariance structures. *Technometrics* 55(1):47–56.

Jalali H, Van Nieuwenhuyse I, Picheny V (2017) Comparison of kriging-based algorithms for simulation optimization with heterogeneous noise. *European Journal of Operational Research* 261(1):279–301.

Kennedy MC, O'Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1):1–13.

Kiefer J (1967) On bahadur's representation of sample quantiles. *The Annals of Mathematical Statistics* 38(5):1323–1342.

Li R, Sudjianto A (2005) Analysis of computer experiments using penalized likelihood in gaussian kriging models. *Technometrics* 47(2):111–120.

Lin PE, Wu KT, Ahmad IA (1980) Asymptotic joint distribution of sample quantiles and sample mean with applications. *Communications in Statistics-Theory and Methods* 9(1):51–60.

Locatelli M (1997) Bayesian algorithms for one-dimensional global optimization. *Journal of Global Optimization* 10(1):57–76.

Pedrielli G, Wang S, Ng SH (2020) An extended two-stage sequential optimization approach: Properties and performance. *European Journal of Operational Research* .

Petersen KB, Pedersen MS (2012) The matrix cookbook (version: November 15, 2012).

Pishro-Nik H (2016) Introduction to probability, statistics, and random processes .

Stein ML (2012) *Interpolation of spatial data: some theory for kriging* (Springer Science & Business Media).

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

Torn A, Zilinskas A (1989) *Global optimization* (Springer-Verlag New York, Inc.).

Yi G, Shi J, Choi T (2011) Penalized gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics* 67(4):1285–1294.