Statistica Sinica **28** (2018), 761-789
doi:https://doi.org/10.5705/ss.202017.0138

SURROGATE-ASSISTED TUNING FOR COMPUTER EXPERIMENTS WITH QUALITATIVE AND QUANTITATIVE PARAMETERS

Jiahong K. Chen, Ray-Bing Chen, Akihiro Fujii,
Reiji Suda and Weichung Wang

Vpon Inc., National Cheng Kung University, Kogakuin University, The University of Tokyo and National Taiwan University

Abstract: Performance tuning of computer codes is an essential issue in computer experiments. By suitable choosing the values of the tuning parameters, we can optimize the codes in terms of timing, accuracy, robustness, or other performance objectives. As computer software and hardware are becoming more and more complicated, such a tuning process is not an easy task, and there are strong needs for developing efficient and automatic tuning methods. In this article, we consider software auto-tuning problems that involve qualitative and quantitative tuning parameters by solving the resulting optimization problems. Because the performance objective functions in the target optimization problems are usually not explicitly defined, we build up surrogates from the response data and attempt to mimic the true, yet unknown, performance response surfaces. The proposed surrogate-assisted tuning process is an iterative procedure. At each iteration, surrogates are updated and new experimental points are chosen based on the prediction uncertainties provided by the surrogate models until a satisfactory solution is obtained. We propose two surrogate construction methods that adopt two infill criteria for the tuning problems containing qualitative and quantitative parameters. The four variants of the proposed algorithm are used to optimize computational fluid dynamic simulation codes and artificial problems to illustrate the usefulness and strengths of the proposed algorithms.

Key words and phrases: Computer experiments, expected improvement, Gaussian process, infill criteria, performance tuning, qualitative and quantitative parameters, surrogates modeling.

1. Introduction

Numerical simulations have been widely used in natural science, engineering, and social sciences under various scenarios. The successes of numerical simulations rely on the efficiency of computer codes. One key component for producing efficient computer codes is software tuning, which involves finding the optimal parameters within algorithms and codes to achieve optimal performance. Such a tuning process is difficult, if not impossible, to perform manually. It is not trivial because of the increasing complexities of algorithms, software, and computer architectures. Consequently, software automatic performance tuning (auto-tuning in short) plays a key role in computer code efficiency.

Auto-tuning has been studied intensively using various approaches. In early achievements such as ATLAS (Whaley, Petitet and Dongarra (2001)), FFTW (Frigo and Johnson (2005)), SPIRAL (Franchetti et al. (2009)), and ABCLib_Script (Katagiri et al. (2006)), manually created models and exhaustive search (with heuristic pruning) were used. More general-purpose auto-tuners such as Active Harmony (Ţăpuş et al. (2002)) and OpenTuner (Ansel et al. (2014)) have a set of search algorithms, such as exhaustive, random, Hill-Climbing, Nelder-Mead, SA, GA, and PSO. Some researchers used machine learning techniques for auto-tuning, such as regression trees (Bergstra, Pinto and Cox (2012)), neural networks (Falch and Elster (2015); Pellegrini et al. (2010)), decision trees (Pellegrini et al. (2010); Monsifrot, Bodin and Quiniou (2002)), PCA plus k-nearest neighbors (Collins et al. (2013)), and kernel canonical correlation analysis (Ganapathi et al. (2009)). Auto-tuning is also called algorithm configuration in some literature. To solve this type of problems, several iterative algorithms have been developed. At each iteration, one needs to select the configuration candidates, evaluate the performance to compare with the incumbent optimal configuration, and then update the current best parameter setting if necessary. Candidate selection can be performed by Random Online Aggressive Racing (Hutter, Hoos and Leyton-Brown (2011)) or the genetic algorithm (Hutter et al. (2009)).

The aforementioned auto-tuning approaches solve the optimization problems (to choose the next testing parameters or configurations) by direct search type methods without relying on performance models. In contrast, statistical model-based methods are considered in such auto-tuning literature as Ylvisaker and Hauck (2011) and Suda (2011). For example, Gramacy, Taddy and Wild (2013) have studied computer code auto-tuning problems involving qualitative variables (and ordinal variables). They propose a statistical surrogate analysis approach that is based on the "dynamic tree" model. The Sequential Model-Based Algorithm Configuration (SMAC) is proposed in Hutter, Hoos and Leyton-Brown (2011). SMAC uses Gaussian process, tree model, random forest, or others to form a surrogate of the performance model. The method uses efficient global optimization and the expected improvement criterion (Jones, Schonlau and Welch (1998)) to select the next configuration candidate. Furthermore, an intensification step is used to make sure what the best point is. An auto-tuning scheme that searches the parameters based on a discretized spline model is studied in Murata et al. (2015). The idea is to use a "d-spline" as fitting model function. It assumes continuity of performance value over the parameter space. It increases the sampling points and updates the d-spline function until the estimated optimal point is found.

In this article, we focus on auto-tuning problems that involve quantitative and qualitative (Q&Q) tuning parameters. Such a Q&Q auto-tuning problem is the optimization problem

$$\min_{\mathbf{w}\in\mathbf{\Omega}} y(\mathbf{w}) \tag{1.1}$$

by using the following notation. Let $\mathbf{x} = (x_1, \ldots, x_I)^t \in \mathbf{\Omega}_{QN} \subset \mathbb{R}^I$ be the vector containing I quantitative tuning parameters and $\mathbf{z} = (z_1, \ldots, z_J)^t \in \mathbf{\Omega}_{QL} \subset \mathbb{R}^J$ be the vector containing J qualitative tuning parameters, with $\mathbf{\Omega}_{QN}$ and $\mathbf{\Omega}_{QL}$ the quantitative and qualitative experimental domains, respectively. We assume that the qualitative parameters z_j have q_j levels and, thus, a Q&Q auto-tuning problem has M *cases* for $M = \prod_{j=1}^{J} q_j$. Finally, we let $\mathbf{w} = \left(\mathbf{x}^t, \mathbf{z}^t\right)^t \in \mathbf{\Omega}$ for $\mathbf{\Omega} = \mathbf{\Omega}_{QN} \times \mathbf{\Omega}_{QL} \subset \mathbb{R}^{I+J}$ and denote the objective function as $y(\mathbf{w})$.

We take the Algebraic Multigrid (AMG) linear system solver (Fujii, Nishida and Oyanagi (2005); Fujii and Marques (2014); Fujii (2014)) as an example to illustrate (1.1). Linear systems arise in many numerical simulations such as fluid dynamics (Darwish, Sraj and Moukalled (2009)) and elastic analysis (Adams et al. (2004)). Solving these linear systems is usually the most expensive part of the simulations. To accelerate the running time of an AMG solver, both quantitative and qualitative parameters need to be tuned, and the number of all possible combinations of the tuning parameters can be large. For example, in the AMG implementation introduced in Fujii, Nishida and Oyanagi (2003), there are 10^6 different combinations of the tuning parameters. It is thus not practical to test all combinations to find the shortest running time. We propose a systematic way of finding the suitable combinations of the parameters so that the execution time of the AMG solver and the associated numerical simulations can be reduced as much as possible. Example 1 is a particular example of an Q&Q auto-tuning optimization problem. We show how the proposed methods can be applied to solve this problem.

Example 1 (Algebraic Multigrid Linear System Solver)*.* We assume $I = J = 2$, $q_1 = 2$, $q_2 = 3$, $M = 6$, and $\mathbf{w} = (x_1, x_2, z_1, z_2)^t$. The quantitative tuning parameter x_1 is the threshold for strong coupling (Vaněk, Mandel and Brezina

(1996)), and x_2 is the relaxation parameter of smoother (Briggs, Henson and McCormick (2000)). To simplify the tuning procedure, we set a grid over the domain of these two quantitative parameters. In particular, $x_1 \in \{0.02, 0.025, 0.03, \ldots, 0.08, 0.09, 0.11\}$ contains 15 grid points and $x_2 \in \{0.6, 0.8, 1.0, \ldots, 1.8\}$ contains 7 grid points. Consequently, the corresponding quantitative experiment domain $\mathbf{\Omega}_{QN}$ is a two-dimensional domain with 15×7 grid points. The qualitative parameter z_1 can be either the Symmetric Gauss-Seidel smoother (SGS) or the Gauss-Seidel smoother (GS) (Briggs, Henson and McCormick (2000)). The qualitative parameter z_2 can be the V-, F-, or W-cycle in the AMG solver (Briggs, Henson and McCormick (2000)). The objective function $y(\mathbf{w})$ is the total execution time of the AMG solver with the tuning vector $\mathbf{w}$. There are $15 \times 7 \times 2 \times 3 = 630$ possible combinations of $\mathbf{w}$ in total.

Such Q&Q auto-tuning problems are common in various computer simulation codes. To the best of our knowledge, there are few efficient and systematic ways to perform auto-tuning for this type of problem. For example ppOpen-AT (Katagiri, Ohshima and Matsumoto (2014)), an auto-tuning framework, allows library developers to implement auto-tuning functionality only by inserting tuning directives to the codes. Many functionalities have already been prepared this task. For example, d-spline parameter search, loop unrolling, loop splitting, and loop fusion can be implemented only by inserting one or two directive lines. All combinations of transformed code patterns and parameter values are checked for the best performance. As it does not distinguish Q&Q parameters, it is difficult to deal with the problems in an efficient way other than by exhaustive searches. A parameter tuning study targeting a multigrid solver is conducted in Chan et al. (2009) using an auto-tuning language called PetaBricks. The multigrid method is a multi-level method for solving linear equations. The study optimized the combinations of parameter settings at each level for multiple convergence criteria. It succeeded in reducing the parameter space by dynamic programming, was able to choose the optimized parameter setting according to a computing environment, and required convergence levels. This study does not distinguish the Q&Q parameters and this approach is not recommended for our target problem.

While SMAC and other Sequential Model-based Optimization (SMBO) algorithms (Hutter, Hoos and Leyton-Brown (2011); Hutter et al. (2009)) can solve the Q&Q problem, these algorithms are designed to handle different types of problems, and differ from these proposed here. The main difference is that SMBO and SMAC are designed with the assumptions that the responses contain noises. Here we assume responses are deterministic and focus on approaches that can quickly identify a near-optimal parameter setting by minimizing the number of explored points. We take the responses to be deterministic because the performance of our numerical solvers show small variations in response when repeated under the same parameter set-up. For a given AMG solver problem, our goal is to efficiently identify a good parameter set-up with less computational cost. For the Q&Q modeling, a modified Gaussian Process deals with qualitative factors and the corresponding kernel function as proposed in Hutter, Hoos and Leyton-Brown (2011). In our Q&Q Gaussian process, the setting of the kernel function is more general, because we also quantify the effect of the different levels of qualitative factors. An expected improvement criterion for log-transformed response is considered in Hutter, Hoos and Leyton-Brown (2011). We propose modified expected improvement criteria for the different types of surrogate modeling approaches.

We face the following challenges: the response functions involve two different types of parameters; the objective functions are measurable but not explicitly defined; we usually need to determine the optimal parameter by using only a few experimental trials without checking all possible parameter set-ups. To solve the Q&Q auto-tuning problem efficiently, we propose a Surrogate-Assisted Tuning (SAT) framework, along with well-developed statistical technologies; the key is to construct the surrogate model as an approximation of the true objective functions based on few experimental trials. The SAT solves the derived optimization problem in an iterative manner. At the beginning, SAT constructs a surrogate based on a set of experimental points. Then, based on the prediction uncertainty, the next explored point is determined via a certain infill criterion. The surrogate is then updated based on the responses of the explored points. This process is repeated until the (near) optimal solution is found or the available resources have been consumed. The two key components of SAT are surrogate construction and infill criteria. For the surrogate constructions, we rely on Gaussian processes (Sacks et al. (1989)) and qualitative and quantitative Gaussian processes (Qian, Wu and Wu (2008); Zhou, Qian and Zhou (2011)) to construct the surrogates. In the Q&Q setting, we propose two infill criteria that are based on the expected improvement (Jones, Schonlau and Welch (1998)), designed to search the next experimental point for optimizing the objective function and improving the surrogate. When the problem contains only quantitative parameters, the SAT framework is similar to the Design and Analysis of Computer Experiment (DACE) (Sacks et al. (1989)) or the Efficient Global Optimization (EGO) approach (Jones, Schonlau and Welch (1998)).

Table 1. Notation used.

I	number of quantitative parameters
J	number of qualitative parameters
q_j	number of levels of the jth qualitative parameter
M	No. of all possible combinations in qualitative parameters ($M = \prod_{j=1}^{J} q_j$)
n	number of experimental points
N	number of repetitions in numerical experiments
t	the (tth) iteration number in SAT-QQ
T	maximum iteration numbers allowed in SAT-QQ (as the stopping criterion)
N_{mesh}	number of uniform grid points in each dimension of Ω_{QN}

This paper is organized as follows. We introduce the framework of surrogate-assisted tuning in Section 2. The details of the surrogate constructions and infill criteria are discussed in Section 3 and Section 4, respectively. The proposed algorithms are summarized in Section 2. Numerical studies of the proposed algorithms are presented in Section 5. The paper concludes in Section 6. Table 1 presents the notation we use.

2. Surrogate-Assisted Tuning for Q&Q Parameters (SAT-QQ)

We propose solving the Q&Q auto-tuning optimization problem (1.1) using the Surrogate-Assisted Tuning framework as outlined in Algorithm 1. The algorithm is called SAT-QQ and is composed of initialization, tuning iteration, and a stopping criterion. The details of the algorithm are discussed as follows.

In the initialization stage of SAT-QQ, we do the following. In Step (1.1), we select an initial design $\mathbf{D}$ containing n experimental points $\mathbf{w}^{(i)}$ for $i = 1, \ldots, n$. It is common to apply a space-filling design such as Latin hypercube design (LHD) (Santner, Williams and Notz (2003); Chen et al. (2013)), uniform design (Fang et al. (2000)), or a sliced LHD (Qian (2012)). In Step (1.2), we evaluate the corresponding objective functions $y^{(i)} = y(\mathbf{w}^{(i)})$.

In the tuning stage, we do the following. In Step (2.1), we need to construct or update the surrogates. We discuss how to construct the surrogates in the Q&Q domain $\boldsymbol{\Omega}$ in Sections 3. These methods are motivated by the Gaussian process (or kriging) (Sacks et al. (1989)). In Step (2.2), we need to choose the new experimental point. We discuss how to find the next experimental point by using the proposed infill criteria that are motivated by the expected improvement (EI) criterion (Jones, Schonlau and Welch (1998)). In Steps (2.3) and (2.4), we evaluate the corresponding objective functions and update $\mathbf{D}$, $\mathbf{y}$, and $y_{\min}$. The iteration ends when the pre-defined stopping criteria are satisfied. A popular criterion involves the available computing resources such as the total runtime or iteration number.

Algorithm SAT-QQ sketches how we can solve the tuning problems. Depending on the choices of surrogate constructions and the infill criteria, the SAT-QQ has four variants: SDGP-SDEI, SDGP-WDEI, WDQQ-SDEI, and WDQQ-WDEI. We elaborate on how the surrogates are constructed in Section 3, and how the infill criteria are determined in Section 4.

Algorithm 1 Surrogate-Assisted Tuning for Q&Q Parameters (SAT-QQ)

```
(1)  Initialization
        (1.1)  Select an initial design $\mathbf{D} = \{\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(n)}\}$ in $\mathbf{\Omega}$.
        (1.2)  Evaluate the function values $y^{(i)} = y(\mathbf{w}^{(i)})$.
               Set $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(n)})^t$ and $y_{\min} = \min_{\mathbf{w}\in\mathbf{D}} y(\mathbf{w})$.
(2)  Tuning Iteration
     Repeat until stopping criterion is satisfied.
        (2.1)  Construct/Update the surrogate based on $\mathbf{D}$ and $\mathbf{y}$     % surrogate
               by SDGP (Section 3.2) or WDQQ (Section 3.3).
        (2.2)  Determine the next experimental point $\mathbf{w}^+$       % infill criterion
               by SDEI (Section 4.2) or WDEI (Section 4.3.)
        (2.3)  Evaluate the function value $y^+ = y(\mathbf{w}^+)$.
        (2.4)  Update $\mathbf{D} = \mathbf{D} \cup \mathbf{w}^+$ and $\mathbf{y} = (\mathbf{y}; y^+)$ and $y_{\min} = \min(y_{\min}, y^+)$.
(3)  Output
        (3.1)  Output $y_{\min}$ and $\mathbf{w}_{\min} = \arg\min_{\mathbf{w}\in\mathbf{D}} y(\mathbf{w})$.
```

3. Surrogate Construction

Surrogate construction and updates in Step (2.1) of Algorithm 1 play a key role in the SAT-QQ procedure. A Gaussian process (kriging) is commonly used in computer experiments to construct surrogates based on the stationary assumption of quantitative parameters (Sacks et al. (1989)). For example, Mariani et al. (2012) adopted the Gaussian process as the foundation of the OSCAR to deal with an auto-tuning problem. When qualitative parameters are considered, Gramacy, Taddy and Wild (2013) suggested using a dynamic tree model. This surrogate model can also deal with a non-stationary surface. We consider the computer experiments involving both of qualitative and quantitative parameters. Based on the qualitative parameters, two models related to Gaussian processes for surrogate constructions are proposed. The first method is called Sub-Domain with Gaussian Process and introduced in Section 3.2. The second method is called Whole-Domain with Qualitative and Quantitative Gaussian Process and presented in Section 3.3.

3.1. Gaussian process (GP)

We briefly outline how the Gaussian process is constructed on a quantitative domain $\mathbf{\Omega}_{QN}$. For more details on the GP, see Sacks et al. (1989); Santner, Williams and Notz (2003), for example. In the Gaussian process, the response function $y(\mathbf{x})$ is characterized as a stochastic process $Y(\mathbf{x})$ in $\mathbf{\Omega}_{QN}$, so at point $\mathbf{x}$, the response $Y(\mathbf{x})$ is a random variable and the observed response value $y^{(k)} = y(\mathbf{x}^k)$ is treated as a realization at the experimental point $\mathbf{x}^{(k)},\ 1 \le k \le n$. We assume that

$$Y(\mathbf{x}) = f(\mathbf{x})^t\beta + \varepsilon(\mathbf{x}), \tag{3.1}$$

where $f(\mathbf{x}) = (f_1(\mathbf{x}),\ \ldots,\ f_p(\mathbf{x}))^t \in \mathbb{R}^p$ is a pre-specified regression vector, $\beta \in \mathbb{R}^p$ is the corresponding coefficient vector, and $\varepsilon(\mathbf{x})$ is a stationary Gaussian process with mean zero and covariance matrix $\sigma^2 R$. The correlation function of between points $\mathbf{x}_1$ and $\mathbf{x}_2$ is commonly taken as $K_\theta(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\sum_{i=1}^I \theta_i(x_{1i} - x_{2i})^2)$, where the θ_i's are positive parameters and $\theta = (\theta_1, \ldots, \theta_I)$.

Suppose we have n input vectors in the design matrix $X = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}]^t \in \mathbb{R}^{n\times I}$, and let $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(n)})^t \in \mathbb{R}^n$. The unknown parameters, β, σ^2, and θ, are estimated using maximal likelihood. Here $F = [f(\mathbf{x}^{(1)}),\ f(\mathbf{x}^{(2)}), \ldots, f(\mathbf{x}^{(n)})]^t \in \mathbb{R}^{n\times p}$; $R = (R_{ij}) \in \mathbb{R}^{n\times n}$ with $R_{ij} = K_\theta(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$. Thus, given θ, $\widehat{\beta} = (F^tR^{-1}F)^{-1}F^tR^{-1}\mathbf{y}$ and $\widehat{\sigma}^2 = (\mathbf{y} - F\widehat{\beta})^tR^{-1}(\mathbf{y} - F\widehat{\beta})/n$, and θ can be estimated via the minimization problem

$$\arg\min_{\theta>0} \left\{ n\log(\widehat{\sigma}^2) + \log(|R|) \right\}.$$

Thus, the prediction of a particular point $\mathbf{x}$ is

$$\widehat{y}(\mathbf{x}) = f(\mathbf{x})^t\widehat{\beta} + r^tR^{-1}(\mathbf{y} - F\widehat{\beta}), \tag{3.2}$$

where $r \in \mathbb{R}^n$ is the correlation vector with $r_j = K_{\hat{\theta}}(\mathbf{x}, \mathbf{x}^{(j)})$ and the mean squared error (MSE) of $\widehat{y}(\mathbf{x})$ is

$$\widehat{s}^2(\mathbf{x}) = \widehat{\sigma}^2\left[1 - r^tR^{-1}r + u^t(F^tR^{-1}F)^{-1}u\right], \tag{3.3}$$

where $u = F^tR^{-1}r - f(\mathbf{x})$.

The prediction $\widehat{y}(\mathbf{x})$ is the empirical best linear unbiased predictor (EBLUP), and the interpolation property $\widehat{y}(\mathbf{x}^{(k)}) = y(\mathbf{x}^{(k)})$ and $\widehat{s}^2(\mathbf{x}^{(k)}) = 0$ is satisfied (Lophaven, Nielsen and Sondergaard (2002)). The prediction at a point $\mathbf{x}$ is a now realization of a normal distribution with mean $\widehat{y}(\mathbf{x})$ and variance $\widehat{s}^2(\mathbf{x})$,

$$Y(\mathbf{x}) \sim N(\widehat{y}(\mathbf{x}), \widehat{s}^2(\mathbf{x})). \tag{3.4}$$

3.2. Sub-domain with Gaussian process (SDGP)

One approach to include the qualitative parameters is to apply the "divide-and-conquer" technique on the experimental domain $\mathbf{w}$. Because the Q&Q auto-tuning optimization problem at (1.1) has M cases, we divide the experimental domain $\mathbf{\Omega}$ into M disjoint sub-domains $\mathbf{\Omega}_m$, $m = 1, \ldots, M$. By doing so, each $\mathbf{\Omega}_m$ involves only the quantitative factors and we can apply the Gaussian Process (GP) discussed in Section 3.1 on each sub-domain $\mathbf{\Omega}_m$, independently, to construct M GP-based surrogates. All the derivations regarding GP on Ω_{QN} can then be applied to Ω_m. For example, the prediction of a point $\mathbf{x} \in \Omega_m$ is

$$\widehat{y}_m(\mathbf{x}) = f(\mathbf{x})^t \widehat{\beta}_m + r_m^t R_m^{-1} (\mathbf{y}_m - F_m \widehat{\beta}_m). \tag{3.5}$$

3.3. Whole-domain with qualitative and quantitative Gaussian process (WDQQ)

Another approach is an "all-in-one" type named Whole-Domain Qualitative and Quantitative Gaussian Process (WDQQ). As the SDGP does not take the cross-correlations effects into account, a modification of the Gaussian process was proposed in Qian, Wu and Wu (2008) and Zhou, Qian and Zhou (2011) that incorporates both qualitative and quantitative variables.

In WDQQ, the response model for an explored point $\mathbf{w}$ is represented as

$$Y(\mathbf{w}) = f(\mathbf{w})^t \beta + \varepsilon(\mathbf{w}), \tag{3.6}$$

where $f(\mathbf{w}) = (f_1(\mathbf{w}), \ldots, f_p(\mathbf{w}))^t$ is the vector of the pre-specified functions on $\mathbf{\Omega}$, β is the unknown coefficient vector, and $\varepsilon(\mathbf{w})$ is assumed to be a Gaussian process with mean zero and constant variance σ^2.

The key step here is to define the proper correlation function for the different levels of the qualitative factors. If the M cases are $\mathbf{z}_1, \ldots, \mathbf{z}_M$, with $\mathbf{z}_m = (z_{m1}, \ldots, z_{mJ})^t$, any experimental point can be represented as $\mathbf{w} = (\mathbf{x}^t, \mathbf{z}^t)^t$. Then, following Zhou, Qian and Zhou (2011), for any two points $\mathbf{w}_i = (\mathbf{x}_i^t, \mathbf{z}_i^t)^t$, $i = 1, 2$, the correlation between $\varepsilon(\mathbf{w}_1)$ and $\varepsilon(\mathbf{w}_2)$ is taken as

$$cor\,(\varepsilon(\mathbf{w}_1), \varepsilon(\mathbf{w}_2)) = \tau_{z_1, z_2} K_\theta(\mathbf{x}_1, \mathbf{x}_2), \tag{3.7}$$

where τ_{z_1,z_2} is the cross-correlation between $\mathbf{z}_1$ and $\mathbf{z}_2$. To estimate the unknown parameters $\tau_{r,s}$ and θ, an efficient maximum likelihood estimation (MLE) approach via the hypersphere parameterization was proposed in Zhou, Qian and Zhou (2011) to model the correlations of the qualitative factors.

An unexplored point $\mathbf{w}$, the prediction $\widehat{y}(\mathbf{w})$ and the mean square error $\widehat{s}^2(\mathbf{w})$

can be derived by replacing $\mathbf{x}$ by $\mathbf{w}$ in the formulas of GP:

$$\widehat{y}(\mathbf{w}) = f(\mathbf{w})^t\widehat{\beta} + r^t R^{-1}(\mathbf{y} - F\widehat{\beta}), \tag{3.8}$$

$$\widehat{s}^2(\mathbf{w}) = \widehat{\sigma}^2\left[1 - r^t R^{-1} r + u^t (F^t R^{-1} F)^{-1} u\right]. \tag{3.9}$$

Here, $\widehat{\beta} = (F^t R^{-1} F)^{-1} F^t R^{-1}\mathbf{y} \in \mathbb{R}^p$; $F = \left[f(\mathbf{w}^{(1)}), f(\mathbf{w}^{(2)}), \ldots, f(\mathbf{w}^{(n)})\right]^t \in \mathbb{R}^{n\times p}$; $R = (R_{ij}) \in \mathbb{R}^{n\times n}$ with $R_{ij} = cor\,(\varepsilon(\mathbf{w}_i), \varepsilon(\mathbf{w}_j))$, and $r \in \mathbb{R}^n$ is the correlation vector with $r_j = cor\,(\varepsilon(\mathbf{w}), \varepsilon(\mathbf{w}_j))$.

3.4. A short summary

The SDGP and WDQQ have their own advantages and disadvantages. We compare them theoretically here and numerically in Section 5.

The WDQQ constructs a global surrogate model containing all of the quantitative and qualitative variables. From this viewpoint, the global surrogate may fit the true function better, but it contains more parameters, and the correlation matrix may have more chances to become ill-conditioned. With WDQQ, we have the stationary assumption, while SDGP can deal with some minor non-stationary cases because the variance estimations for the different domains can differ.

The interplay between modeling accuracy and computational efficiency of SDGP and WDQQ depends on the problem. For WDQQ, different sub-domains share the same θ_i's, and the SDGP constructs the surrogate independently for each sub-domain with different θ_i's. If the true surface in each sub-domain is quite different, SDGP has more flexibility to deal with it. One can modify the correlation structures in the WDQQ for this, but this increases the complexity of the covariance matrix and thus increases the computational cost.

We compare the cost of surrogate construction, particularly the cost for solving the MLE problems, in the SDGP and WDQQ. In SDGP, we need to solve the MLE problems with dimension I in each sub-surrogate. In WDQQ, the dimension is $I + M(M-1)/2$, if the product correlation structure (3.7) is used. To compute the prediction value and regression coefficients, a symmetric positive definite linear system $R \cdot x = b$ is solved at the cost of cubic order of the size of R (Trefethen and Bau III (1997)). If n_m is the experimental point size in $\mathbf{\Omega}_m$ and $n = \sum_{m=1}^{M} n_m$ is the total experimental point size in $\mathbf{\Omega}$, computational complexities of SDGP and WDQQ are $O(M \cdot \max_{m=1,\ldots,M} n_m^3)$ and $O(n^3)$, respectively. In the SDGP, when a new experimental point is added, only the corresponding kth sub-surrogate model has to be updated, with cost $O(n_m^3)$, in the WDQQ, the global surrogate must be updated with cost $O(n^3)$.

4. Infill Criteria

A second main component of the SAT-QQ is to select the new experimental point using an infill criterion. The infill criterion is embedded in the iterative tuning process, Step (2.2) of Algorithm 1, where we iteratively select the next experimental point to explore the experimental domain and search for the optimal solution. If the stopping criterion has not been reached, we include the information due to the newly selected point to update the surrogates. To select the next experimental point in Step (2.2), we propose two approaches that are inspired by the expected improvement (EI) criterion (Jones, Schonlau and Welch (1998); Forrester, Sobester and Keane (2008)).

4.1. Expected improvement (EI)

The EI selection criterion is proposed in Jones, Schonlau and Welch (1998) and is widely used in optimization problems with quantitative factors when searching a new experimental point so that the experimental space can be exploited locally and explored globally in an efficient way. We discuss expected improvement (EI) in this section.

Consider the experimental region with only the quantitative variable $\mathbf{\Omega}_{QN}$. In this case, we define the improvement function of a point $\mathbf{x} \in \mathbf{\Omega}_{QN}$ as

$$I(\mathbf{x}) = \max(y_{\min} - Y(\mathbf{x}),\ 0), \tag{4.1}$$

where $y_{\min} = \min\{y^{(1)}, y^{(2)}, \ldots, y^{(n)}\}$ is the observed minimum in $\mathbf{\Omega}_{QN}$. This improvement function can be interpreted as the quantity of improvement at a certain point $\mathbf{x}$, and it is positive only if $Y(\mathbf{x}) < y_{\min}$. Based on the GP assumption, $Y(\mathbf{x})$ satisfies (3.4). Consequently, instead of measuring the improvement quantity $I(\mathbf{x})$ directly, we compute the expected value of the improvement function

$$EI(\mathbf{x}) = \mathbb{E}[I(\mathbf{x})] \tag{4.2}$$

as our new experimental point selection criterion. Thus if $\mathbf{x}^{+}$ is the next experimental point to be explored, and we take

$$\mathbf{x}^{+} = \underset{\mathbf{x} \in \mathbf{\Omega}_{QN}}{\arg\max}\, E(I(\mathbf{x})). \tag{4.3}$$

It can be shown that

$$\mathbb{E}[I(\mathbf{x})] = (y_{\min} - \widehat{y}(\mathbf{x}))\Phi\left(\frac{y_{\min} - \widehat{y}(\mathbf{x})}{\widehat{s}(\mathbf{x})}\right) + \widehat{s}(\mathbf{x})\phi\left(\frac{y_{\min} - \widehat{y}(\mathbf{x})}{\widehat{s}(\mathbf{x})}\right), \tag{4.4}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal distribution function and density function, respectively. Equation (4.4) suggests that the EI infill criterion tries to balance the prediction-based local exploitation $y_{\min} - \widehat{y}(\mathbf{x})$ and the error-based global exploration $\widehat{s}(\mathbf{x})$.

The EI criterion can only be used for the quantitative variables. To link in the information on the qualitative variables, we propose two variants of it.

4.2. Sub-domain expected improvement (SDEI)

This is a direct extension of EI. As in Section 3.2, we divide the experimental domain $\mathbf{\Omega}$ into M disjoint sub-domains $\mathbf{\Omega}_m$. Because each $\mathbf{\Omega}_m$ involves only the quantitative factors, we can apply the EI criterion to each $\mathbf{\Omega}_m$. First, we choose the sub-domain $\mathbf{\Omega}_{m^*}$ with the minimal prediction value,

$$m^* = \underset{m=1,\ldots,M}{\arg\min}\left(\widetilde{y}_m\right), \text{ where } \widetilde{y}_m = \min_{\mathbf{x}\in\mathbf{\Omega}_m}\left(\widehat{y}(\mathbf{x})\right). \tag{4.5}$$

Then, we apply EI criterion on $\mathbf{\Omega}_{m^*}$ to select the next experimental point

$$\mathbf{x}^+ = \underset{\mathbf{x}\in\mathbf{\Omega}_{m^*}}{\arg\max}\left(E(I(\mathbf{x})\right). \tag{4.6}$$

We call this infill criterion the Sub-Domain Expected Improvement (SDEI).

4.3. Whole-domain expected improvement (WDEI)

In the Whole-Domain Expected Improvement, we link information at different levels of the qualitative factors to define the WDEI for selecting the new experimental point. Let y_m^* be the minimal observed value on the sub-domain $\mathbf{\Omega}_m$, and take

$$y_g^* = \min_{m=1,\ldots,M} y_m^*$$

to be the currently *global* minimal observed response on $\mathbf{\Omega}$. We define the "cross domain improvement function"

$$I_g(\mathbf{w}) = \max(y_g^* - Y(\mathbf{w}),\ 0) \text{ for } \mathbf{w}\in\mathbf{\Omega}, \tag{4.7}$$

where the prediction $Y(\mathbf{w})$ can be obtained from SDGP or WDQQ. The cross domain expected improvement is

$$EI_g(\mathbf{w}) = \mathbb{E}[I_g(\mathbf{w})], \tag{4.8}$$

and next experimental point is chosen as

$$\mathbf{w}^+ = \underset{\mathbf{w}\in\mathbf{\Omega}}{\arg\max}\, EI_g(\mathbf{w}). \tag{4.9}$$

Theorem 1 asserts that the WDEI (4.8) has properties similar to that of the original EI. Because the proof is similar to that for the original EI, we omit its proof.

Table 2. Experiment settings.

Problem	n_0	M	$n_0 \times M$	T	N	N_{mesh}	Regression functions
AMG solver	6	6	36	90	100	12	$f_1 = 1,\ f_2 = x_1,\ f_3 = x_2$
gabor_lv3	6	3	18	90	100	32	$f_1 = 1$

Theorem 1 (Properties of WDEI)**.** *The improvement function in* (4.7) *satisfies (i)* $\mathbb{E}[I_g(\mathbf{w})] > 0$ *if and only if* $\widehat{s}^2(\mathbf{w}) > 0$*, (ii)* $\widehat{s}^2(\mathbf{w}) > 0$*, then*

$$\mathbb{E}[I_g(\mathbf{w})] = \big(y_g^* - \widehat{y}(\mathbf{w})\big)\,\Phi\left(\frac{y_g^* - \widehat{y}(\mathbf{w})}{\widehat{s}(\mathbf{w})}\right) + \widehat{s}(\mathbf{w})\phi\left(\frac{y_g^* - \widehat{y}(\mathbf{w})}{\widehat{s}(\mathbf{w})}\right). \tag{4.10}$$

5. Numerical Experiments

We conducted numerical experiments to study the performance of the proposed algorithms. The algorithms were implemented in MATLAB, and the numerical experiments were conducted on a workstation with Intel Xeon X5570 CPU (8 cores) and 48 GB of main memory. To generate the initial design in Step (1.1) of Algorithm 1, we used Latin hypercube designs (LHD) (McKay, Beckman and Conover (1979)) for the quantitative variables in each case independently. The sizes of LHD in each case were all the same. Let n_0 denote the number of the initial design points in each sub-domain $\mathbf{\Omega}_k$, and let n_0M denote the total number of initial design points. We partition the quantitative domain $\mathbf{\Omega}_{QN}$ into uniform grids with N_{mesh} points for each dimension. The stopping criterion in the Repeat-loop is the maximal number of iterations, denoted by T. Other experimental settings are listed in Table 2.

We have tested the proposed algorithms for two types of tuning problems. In the first problem (TP1), we considered the AMG solver introduced in Example 1. The AMG solver was used to solve anisotropic (ani) or isotropic (iso) problems in physical simulations with Conjugate Gradient (cg) or Biconjugate gradient stabilized (bicgstab) linear system solvers Fujii (2014) and Fujii, Nishida and Oyanagi (2003). Thus we considered four AMG solvers denoted by amg_ani_cg, amg_ani_bicgstab, amg_iso_cg, and amg_iso_bicgstab. The second type of problems (TP2) involved artificial oscillatory surfaces. The surface, gabor_lv3, was constructed using the Gabor functions

$$g_{re} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \psi\right),$$

$$g_{im} = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda} + \psi\right),$$

Table 3. Definitions of the three categories of the qualitative parameter in garbor_lv3.

Qualitative level	λ	ϑ	ψ	σ	γ	Function
1	1.0	$\pi/4$	2	0.5	2.0	g_{re}
2	1.2	$\pi/4$	3	0.5	1.8	g_{re}
3	0.8	$\pi/4$	4	0.5	2.2	g_{im}

Table 4. Results of TP1.

Problem	Meas.	SDGP		WDQQ	
		SDEI	WDEI	SDEI	WDEI
(a)	f_{hit} (%)	85	60	**97**	87
amg_ani_cg	mean(e_R)	2.1238E-02	7.7617E-02	**1.0194E-02**	3.9869E-02
(b)	f_{hit} (%)	27	**28**	13	14
amg_ani_bicgstab	mean(e_R)	3.3310E-02	**2.2610E-02**	4.4969E-02	4.9670E-02
(c)	f_{hit} (%)	58	73	75	**76**
amg_iso_cg	mean(e_R)	1.6217E-02	**2.1621E-03**	7.5348E-03	8.5935E-03
(d)	f_{hit} (%)	60	75	**78**	75
amg_iso_bicgstab	mean(e_R)	2.5353E-02	1.6276E-02	**1.4383E-02**	1.6332E-02

where $x' = x\cos\vartheta + y\sin\vartheta$ and $y' = -x\sin\vartheta + y\cos\vartheta$. Here we considered one qualitative parameter with three qualitative levels. See Table 3 for their definitions. In short, we took $\boldsymbol{\Omega} = [-3,3]^2 \times \{1,2,3\}$.

5.1. Numerical results

We studied the performance of the four variants of SAT-QQ in terms of relative errors, hit frequency, quantile curves, execution time, and comparison with random selection.

Relative errors. Let $y^*_{\min}$ be the minimal objective function value out of all possible parameter combinations and $y^{(t)}_{\min}$ be the observed minimum at the tth iteration. To measure the solution qualities after SAT-QQ has been iterated T times, we computed the relative error

$$e_R = \frac{|y^{(T)}_{\min} - y^*_{\min}|}{|y^*_{\min}|}. \tag{5.1}$$

Tables 4 and 5 show that the four methods SDGP-SDEI, SDGP-WDEI, WDQQ-SDEI, and WDQQ-WDEI achieved similar relative errors after $T = 90$ iterations in the four test problems. Most of the relative errors are in the same order. An exception is TP2-(a), where the infill criterion WDEI significantly outperformed the SDEI by two orders.

Sub-domain hit frequency. To check whether the algorithms can find the

Table 5. Results of TP2.

Problem	Meas.	SDGP		WDQQ	
		SDEI	WDEI	SDEI	WDEI
(a)	f_{hit} (%)	16	18	33	**96**
gabor_lv3	mean(e_R)	1.6619E-01	1.7340E-01	1.3472E-01	**1.6012E-02**

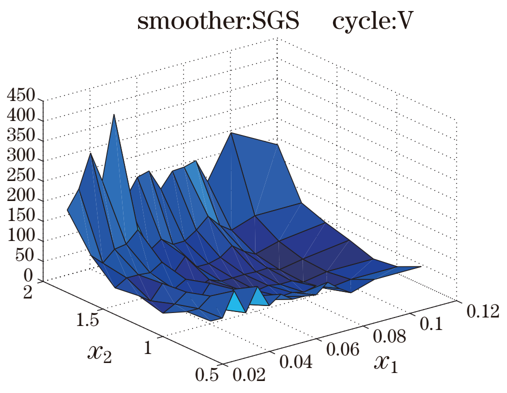
right sub-domain, we used the sub-domain hit frequency

$$f_{hit} = \frac{N_{hit}}{N}, \tag{5.2}$$

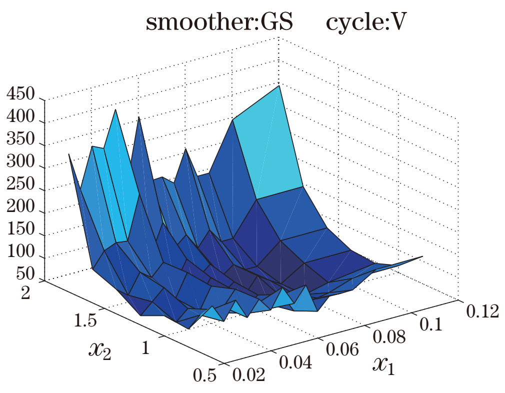
where N_{hit} is the number of SAT-QQ search processes that locate the sub-domain containing the optimal point, and N is the number of repetitions. Thus the hit frequency measures the chance that $\mathbf{w}_{\min}^{(T)}$ and $\mathbf{w}^*$ belong to the same sub-domain, $\mathbf{w}_{\min}^{(T)}$ the argument of the minimal found by the SAT-QQ in the Tth iteration, and $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbf{\Omega}} y(\mathbf{w})$. In our numerical experiments, $\mathbf{w}^*$ was identified by searching the possible arguments in Ω. Here the higher f_{hit} is, the better the result.

Table 4 suggests that WDQQ-SDEI and WDQQ-WDEI have similar performances for TP1-(b) and TP1-(c). WDQQ-SDEI is slightly better for TP1-(a) and TP-(d). For TP1, SDGP-WDEI significantly outperforms the other approaches. To see why, we plot the "true" response surfaces in Figure 1 for the problem amg_ani_cg. The surfaces in each of sub-domain are somewhat simple, there being main trends of some small vibrations, while the true surface of gabor_lv3, shown in Figure 2, is oscillatory. Under TP1, surrogates can capture the main trends and therefore search the optimal points on the "right" sub-domain with a small number of experimental points. In contrast, as shown in Table 5, the true surfaces of TP2 are oscillatory; WDQQ considers both the qualitative and quantitative factors and evaluates the WDEI of all the sub-domains, and this can lead to a better understanding of the surface properties. In the problem TP1-(b), the f_{hit} are low for all four approaches. As the minimal values in each sub-domain are close in value, it is hard to distinguish sub-domains in terms of the objective function values even though the relative errors are small. Thus, all four variants of SAT-QQ can find good experimental points whose objective function values are close to optimal.

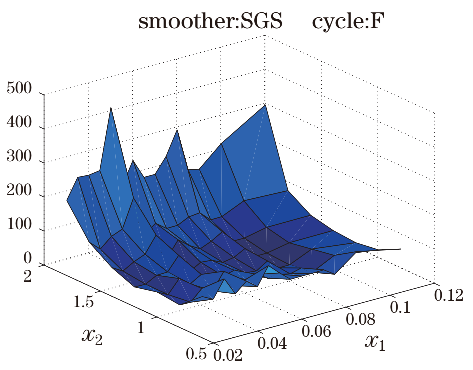
Quantile curves. Because the performance of the SAT-QQ depends on the initial designs and number of iterations, we repeated the numerical experiments N times with different initial space-filling designs and then plot the quantile curves to analyze how $y_{\min}^{(t)}$ converges with the iterations. The quantile curves



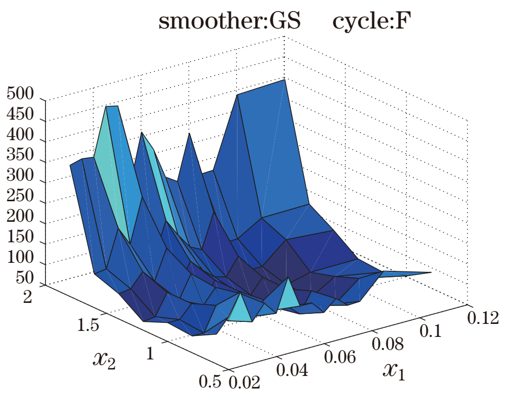
(a) (SGS, V)



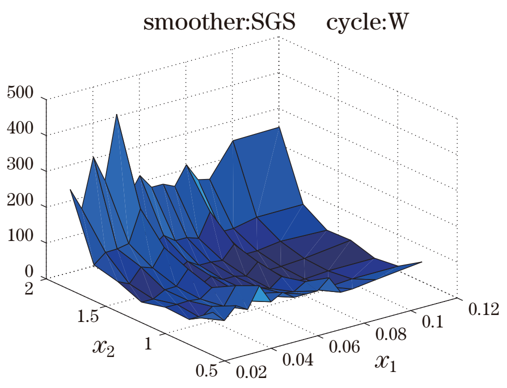
(b) (GS, V)



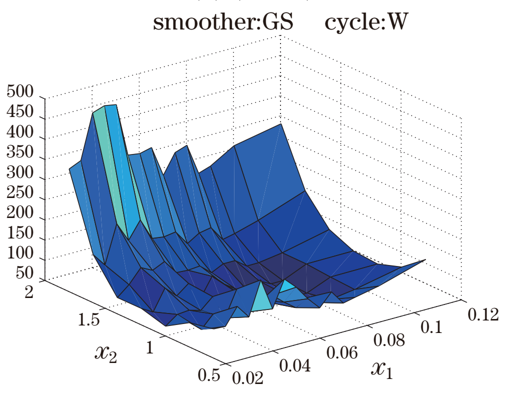
(c) (SGS, F)



(d) (GS, F)



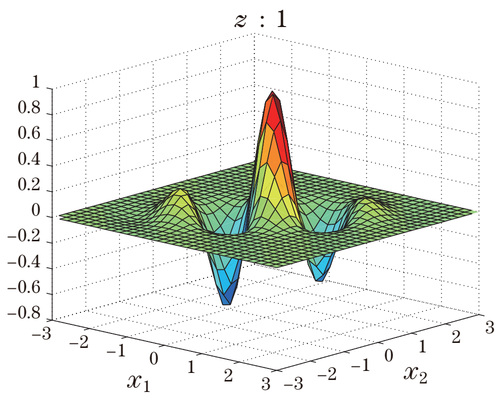
(e) (SGS, W)



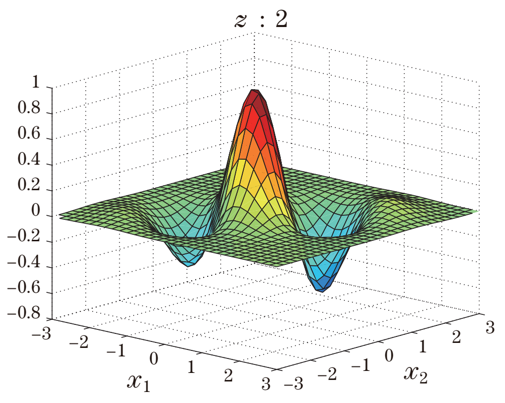
(f) (GS, W)

Figure 1. True surface of amg_ani_cg of each (Smoother, Cycle).

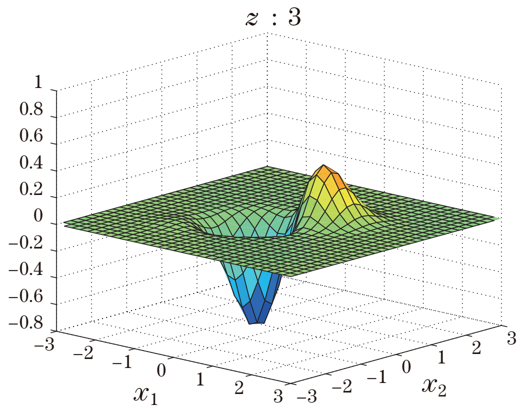
show the 5-percentile, 95-percentile, and median of the N values of $y_{\min}^{(t)}$ in the tth iteration. In TP1, SDGP has a better chance of achieving smaller objective



(a) $z = 1$



(b) $z = 2$

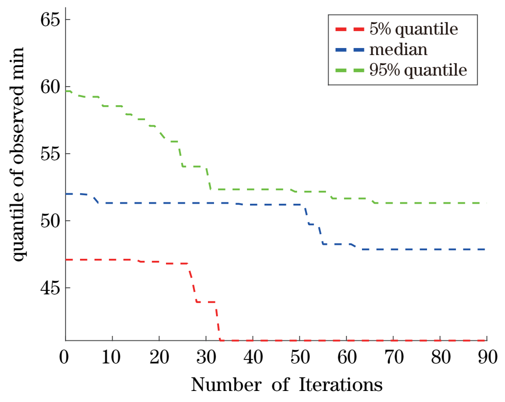


(c) $z = 3$

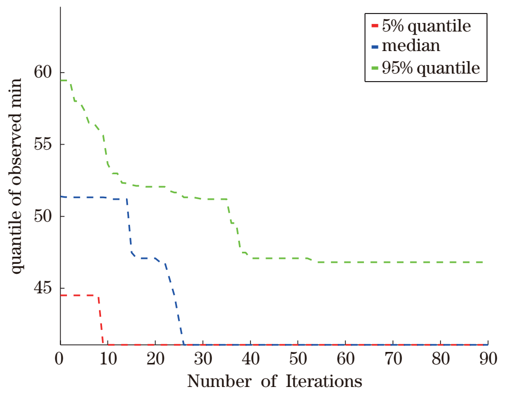
Figure 2. True surface of gabor_lv3 with each qualitative level.

functions with fewer iterations, as shown in the median and 5% quantile curves presented in Figures 3, 4, 5, and 6. However, it can stick in a "wrong" sub-domain due to the lack of a global view across the sub-domains. In such cases, the surrogates of the other sub-domains may not be updated. Consequently, the objective function values may not decrease as shown in the 95% quantile curves. Meanwhile, the 95% quantile curves produced by WDQQ decrease if the number of iterations is large enough. In TP2, the quantile curves due to WDQQ-WDEI decrease faster than other approaches, as shown in Figures 7. This parallels to the results and explanations in the discussion of relative errors.

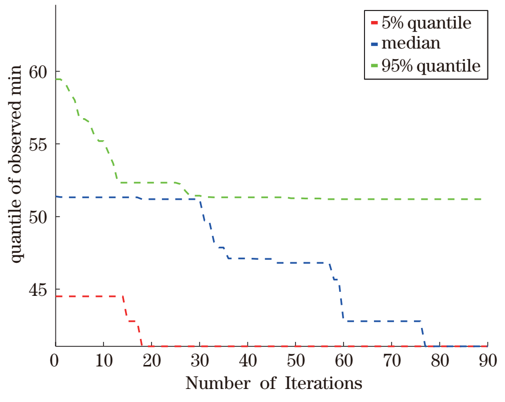
Timing. Table 6 lists the computational cost in time (seconds) for constructing the surrogates of the $N = 100$ repetitions. The table shows that WDQQ is much more expensive than SDGP, regardless of whether SDEI or WDEI is used.



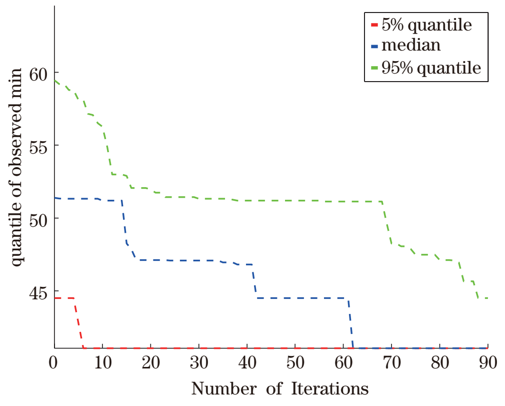
(a) SDGP+SIEI



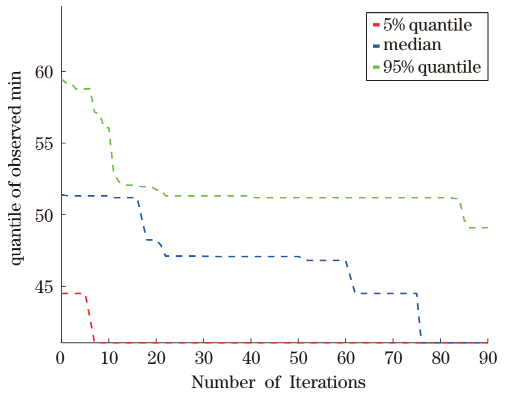
(b) SDGP+SDEI



(c) SDGP+WDEI



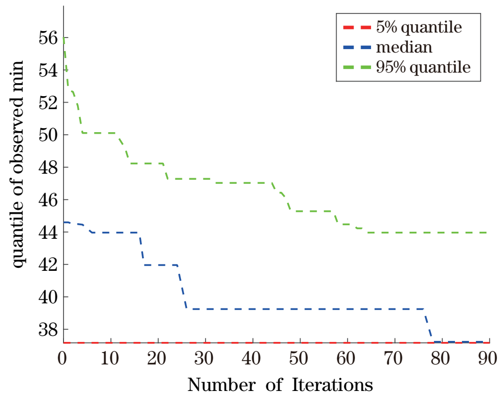
(d) WDQQ+SDEI



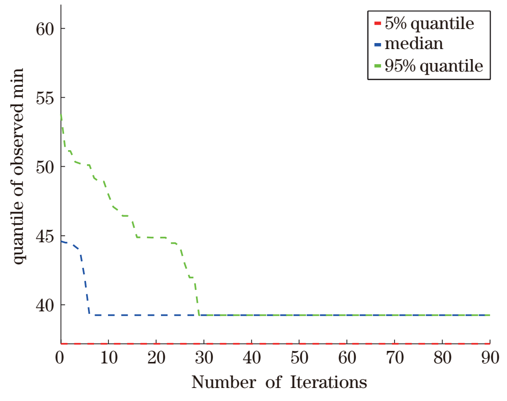
(e) WDQQ+WDEI

Figure 3. Quantile curves of amg_ani_cg.

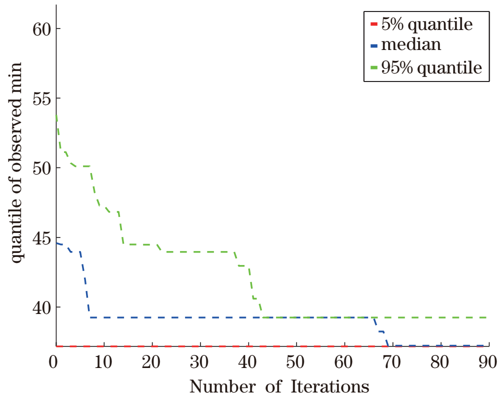
The main cost of each surrogate construction is the optimization problem for parameter estimation in SDGP and WDQQ, respectively. Then too, as WDQQ



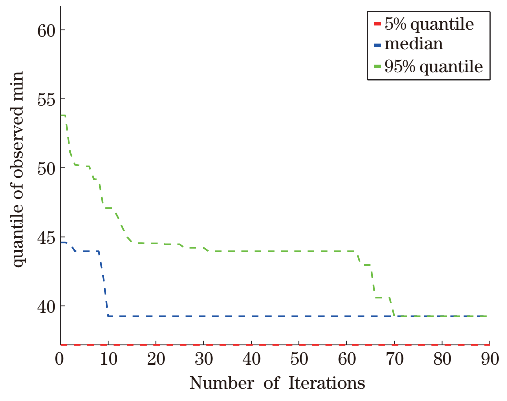
(a) SDGP+SIEI



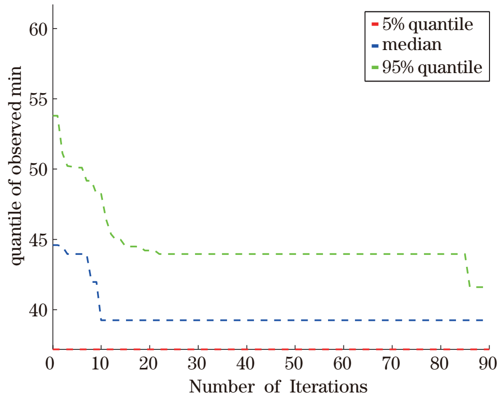
(b) SDGP+SDEI



(c) SDGP+WDEI



(d) WDQQ+SDEI



(e) WDQQ+WDEI

Figure 4. Quantile curves of amg_ani_bicgstab.

integrates information from all the sub-domains, the size of the correlation matrix, R, is larger than that of SDGP. Thus it bears a greater computational cost

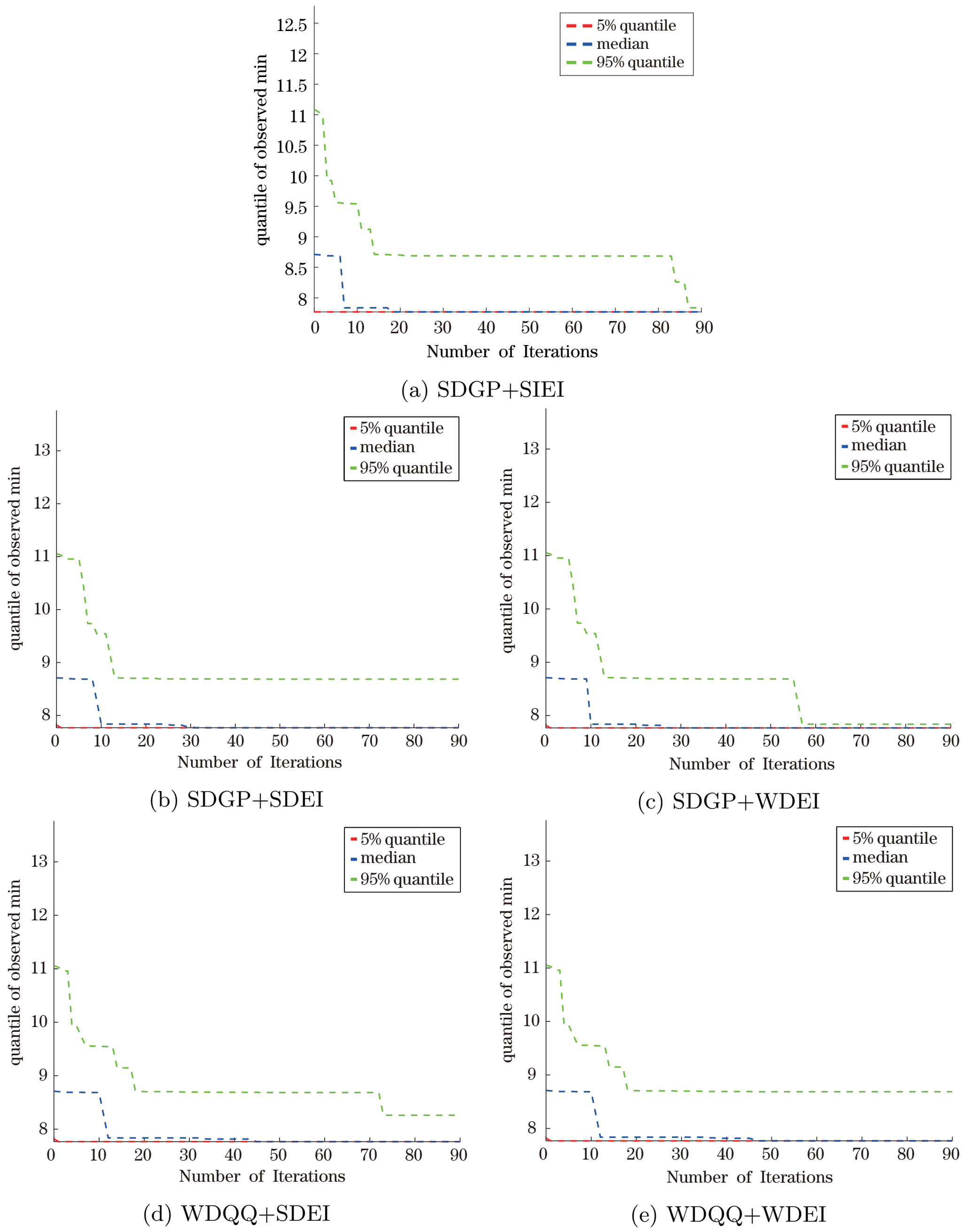


Figure 5. Quantile curves of amg_iso_cg.

due to the computation of the inverse of R. The computational complexity is shown in Section 3.4. If the available tuning resource in time is limited, the

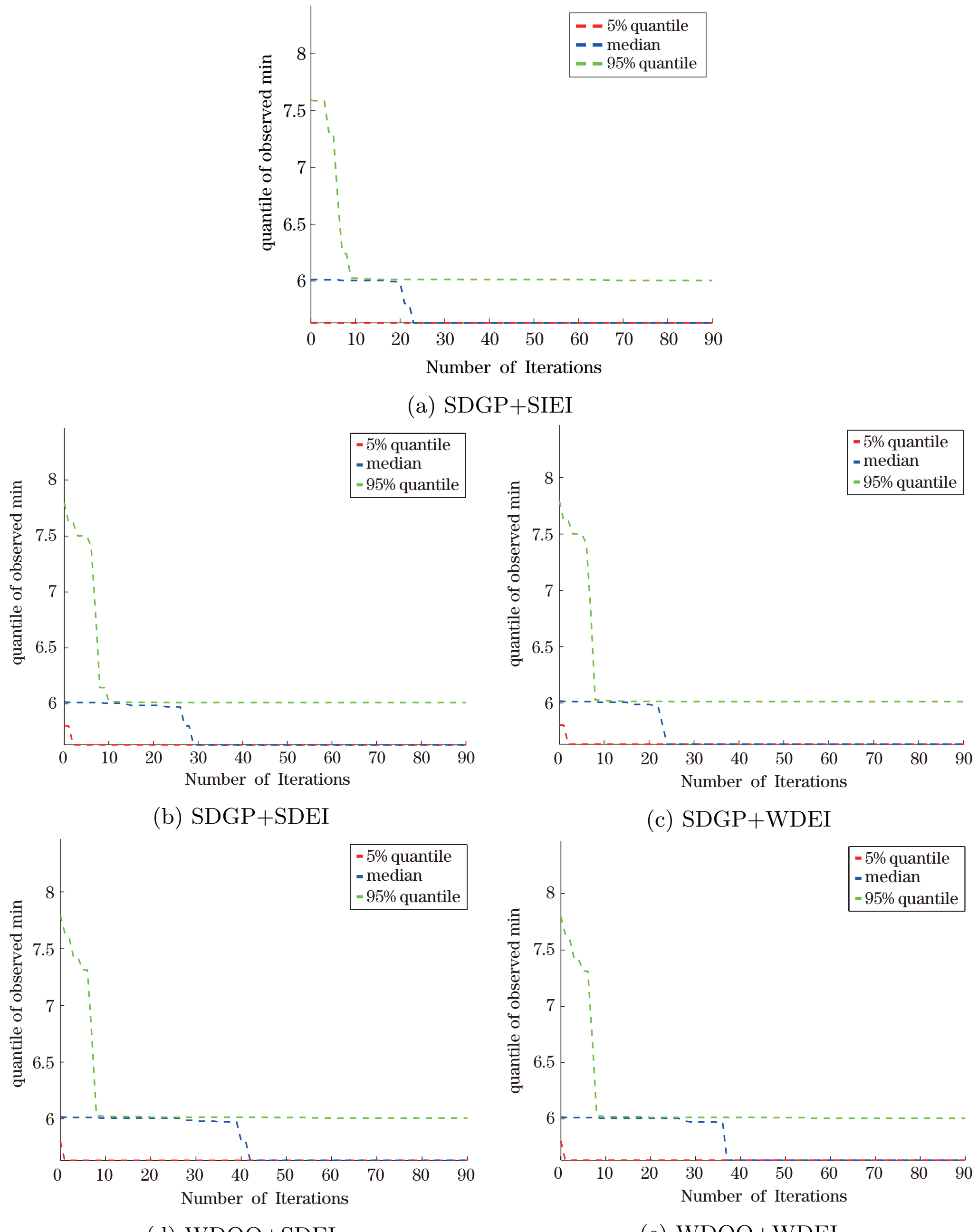
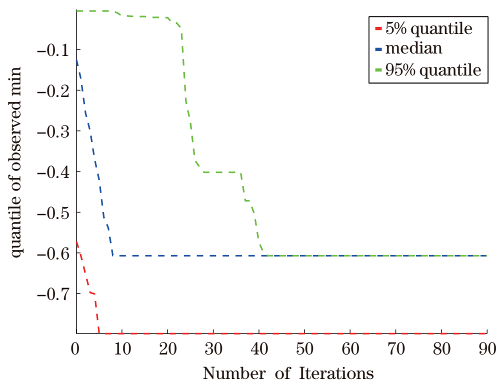


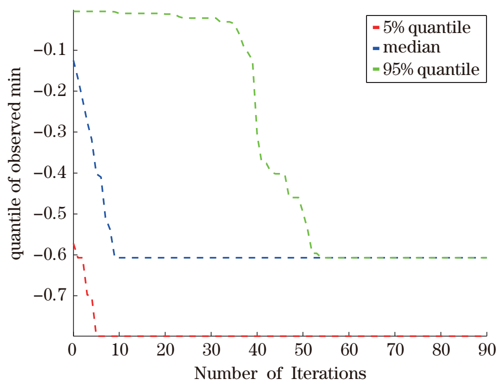
Figure 6. Quantile curves of amg_iso_bicgstab.

SDGP is preferred for surrogate construction.

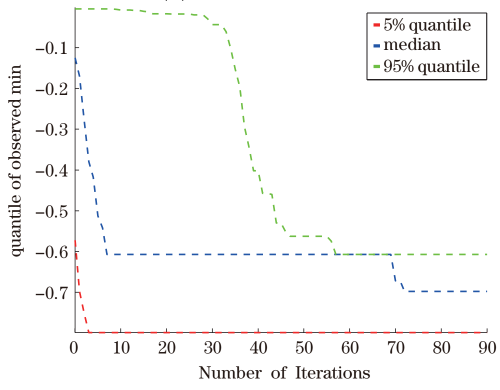
Results of a random method. We attempted to solve the target Q&Q



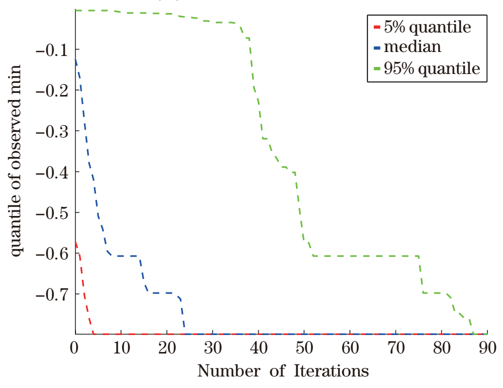
(a) SDGP+SDEI



(b) SDGP+WDEI



(c) WDQQ+SDEI



(d) WDQQ+WDEI

Figure 7. Quantile curves of gabor_lv3.

Table 6. Total surrogae construction time (seconds) of $N = 100$ repetations.

Problem	SDGP		WDQQ	
	SDEI	WDEI	SDEI	WDEI
gabor_lv3	195.49	248.38	1,252.90	4,039.98
amg_ani_cg	100.26	104.09	1,125.21	1,148.83
amg_ani_bicgstab	101.91	105.37	1,142.38	1,176.22
amg_iso_cg	97.79	103.57	1,162.95	1,162.90
amg_iso_bicgstab	97.75	103.86	1,162.02	1,169.26

auto-tuning problem (1.1) by randomly selecting experiment points. In each of the test problems, we let the total number of experimental points be n_0M, $n_0M+T/3$, $n_0M+2T/3$, or n_0M+T. These experimental points were distributed into the M sub-domains evenly, and the experimental points were chosen as the LHD points in each sub-domain. One hundred repetitions were performed in each

Table 7. Results of randomly selected points.

Test Problems	Experimental Points	mean(e_R)	std(e_R)
gabor_lv3	n_0M	0.9997	0.0015
	$n_0M + T/3$	0.4860	0.2257
	$n_0M + 2T/3$	0.3642	0.1377
	$n_0M + T$	0.2778	0.1470
amg_ani_cg	n_0M	1.3126	1.0689
	$n_0M + T/3$	0.1850	0.1060
	$n_0M + 2T/3$	0.1481	0.0898
	$n_0M + T$	0.1481	0.0898
amg_ani_bicgstab	n_0M	0.8426	1.1412
	$n_0M + T/3$	0.1603	0.0984
	$n_0M + 2T/3$	0.1135	0.1048
	$n_0M + T$	0.1135	0.1048
amg_iso_cg	n_0M	1.4279	0.7150
	$n_0M + T/3$	0.3155	0.1880
	$n_0M + 2T/3$	0.1833	0.1297
	$n_0M + T$	0.0794	0.0869
amg_iso_bicgstab	n_0M	0.9754	0.7068
	$n_0M + T/3$	0.1166	0.1179
	$n_0M + 2T/3$	0.0687	0.0293
	$n_0M + T$	0.0559	0.0238

numerical experiment. The means and standard deviations of the corresponding e_R are listed in Table 7. While more experimental points yield better results, the proposed methods outperform the random results, see Tables 4, 5, and 7.

Results of a straightforward combination of GP and EI. To solve the target Q&Q auto-tuning problem, we can apply the GP and EI on each subdomain and then select the next exploring point as the point with the maximal EI value among all sub-domains. This approach is different from the SDEI introduced in Section 4.2 and this infill approach can be treated as a straightforward generalization of EI over multiple sub-domains. We used this straightforward GP-EI based method to solve the test problem TP1 and report the numerical results in Table 8. Comparing Tables 4 and 8, the performance of this GP-EI-based method is mixing. It performs reasonably well in term of f_{hit} for TP1-(b,c,d). However, for TP1-(a), it performs much worse than the four proposed variants. Furthermore, this straightforward method does not perform well in term of e_R for all four cases associated with TP1. Numerical results suggest that more information associated the qualitative factors can improve the computational performance.

Table 8. Results of TP1 by a straightforward combination of GP and EI.

Means	(a) amg_ani_cg	(b) amg_ani_bicgstab	(c) amg_iso_cg	(d) amg_iso_bicgstab
f_{hit} (%)	55	25	81	77
mean(e_R)	1.420E-01	3.690E-02	5.900E-03	1.490E-0.2

Table 9. Results of TP1 by a tree method with EI criterion based on 30 replications.

Means	(a) amg_ani_cg	(b) amg_ani_bicgstab	(c) amg_iso_cg	(d) amg_iso_bicgstab
f_{hit} (%)	67	53	73	90
mean(e_R)	1.724E-01	7.900E-02	1.280E-02	6.730E-0.3

Results of tree-type methods. Tree-type methods, such as classification and regression trees (CART) (Breiman et al. (1984); Chipman, George and McCulloch (1998)), can be used to model the qualitative variables. In such methods, the qualitative variables are represented by a set of indicator variables. After variable transformation, we use these indicator variables to build the tree and apply the quantitative variables for model fitting in the leaves of the tree (Gramacy and Taddy (2010)). For the infill criterion, the EI criterion was adopted to select the next explored point when revisited the first test problem TP1 to illustrate the performance of the tree-based methods. A linear model was used to describe the relationship between the quantitative factors in each leaf of the tree model. Here the function `btlm` in the `tgp` R-package (Gramacy and Taddy (2016)) was adopted. We repeated this tree method 30 times by randomly generating a Latin hypercube design. In each replication, we iterated the tree method 90 times. The results are summarized in Table 9. From Table 9, we see the tree method performs quite well in the measurement of f_{hit} for all four cases. For the mean values of e_R, however, the proposed approaches outperform the first three cases TP1-(a), (b), (c). For TP1-(d), the tree method performs better by obtaining lower e_R mean value.

The Bayesian treed Gaussian process model was also used in the leaf of the tree model. This approach was implemented by the `tgp` R-package (Gramacy and Taddy (2016)). In its latest version, the `tgp` extends the treed Gaussian process models for Bayesian nonstationary and semiparametric nonlinear regressions and designs. The tree structure can be constructed based on the qualitative factors. The stationary or Bayesian non-stationary Gaussian process models in the nodes were used to describe the relationship between the quantitative factors. Thus, the

Table 10. Results of TP1 by a treed Gaussian process method with EI criterion based on 30 replications.

Means	(a) amg_ani_cg	(b) amg_ani_bicgstab	(c) amg_iso_cg	(d) amg_iso_bicgstab
f_{hit} (%)	100	90	57	60
mean(e_R)	0.000E-00	8.116E-03	2.896E-02	2.516E-0.2
CPU time (s)	1,541.08	1,677.38	2,042.08	2,039.19

latest `tgp` can be used to solve our tuning problems. In `tgp`, the function `btgpllm` was used to implement the tree construction based on the qualitative factors and the surrogate fitting via the treed Gaussian process for the quantitative factors in each leaf. The results obtained by 30 replications are summarized in Table 10. For TP1-(a) and (b), this treed Gaussian process can capture the true model quite well and achieves extremely high f_{hit} rate and lower e_R. The proposed SAT-QQ shows better performances in TP1-(c) and (d). The table also shows the execution time taken by the `tgp` on a PC with Intel Core i7-2600 CPU (faster than the Intel Xeon X5570 CPU used in other numerical experiments) and 8GB of memory. Comparing Tables 6, the treed Gaussian process approach takes longer CPU time for the same number of replications.

6. Conclusion

For the four variants of SAT-QQ, numerical results suggest the following findings. WDQQ-WDEI leads to good solutions in general. If computational resources are limited, we suggest using SDGP-WDEI. If we have prior knowledge that the true response surfaces are "simple and smooth" (rather than oscillatory), SDGP type methods are recommended. Otherwise, WDQQ is a proper choice because it considers the correlations between sub-domains.

The variants of SAT-QQ are based on the stationary assumption. Such information about a data set is usually unknown a priori, and may be costly to verify. For example, in Gramacy, Taddy and Wild (2013), the non-stationary of the responses was confirmed based on a large training point set. The proposed methods remain applicable in some cases. One example is the test problem TP1. While whether the testing data satisfies the stationary assumption is available, see Figure 1, the response surfaces clearly demonstrate main trends that are likely stationary. Consequently, the proposed methods can identify region of interest that are associated with the lower timing results.

There are several further research directions. In our study, the number of all possible cases (M) is small. For large M, WDQQ might be problematic in terms of the parameter estimation in its correlation matrix. One needs to simplify the correlation structure based on some prior knowledge. SDGP can be used to fit the surrogates for all sub-domains, but it ignores the relations between sub-domains. Numerical results show that the performance of the tree-based methods and the proposed methods is mixed. Further study could deepen the understanding of these methods and provide more choices for the auto-tuning problems involving qualitative and quantitative parameters.

The SAT-QQ can be viewed as a sequential design procedure, because the next point is selected based on an EI criterion from unexplored point set. The proposed methods share a similarity with active learning. Active learning is also a sequential procedure that iteratively adds un-label points to update the learning model. The point selection criterion is the key. For example, in Deng et al. (2009), the selection approach is a combination of stochastic approximation and D-optimal criterion. In particular, one identifies a D-optimal design point from a small number of candidates based on the learning model fitting. It is worth investigating whether we can integrate some selection approaches in active learning with the proposed EI-type criterion to improve computational efficiency, especially for the case with many candidate experimental points.

A further study could explore modifying SAT-QQ for on-line tuning problems. Due to the lack of the training set, the key component is still the surrogate construction. Since the stationary assumption might not hold, the type of the surrogate model that should be used is a challenge problem. In addition, How to efficiently update the surrogate in each SAT-QQ iteration will be an issue, especially for large experimental domains. Please send correspondences regarding this paper to Ray-Bing Chen (rbchen@mail.ncku.edu.tw) and Weichung Wang (wwang@ntu.edu.tw).

Acknowledgment

The authors are grateful to an associate editor and the referees for helpful comments and suggestions. This work was supported in part by the Ministry of Science and Technology, the Taida Institute of Mathematical Sciences, the Center for Advanced Study in Theoretical Sciences, and the Mathematics Division of the National Center for Theoretical Sciences in Taiwan. This work is also partially supported by the grant MOST 104-2918-I-006-005-.

References

Adams, M. F., Bayraktar, H. H., Keaveny, T. M. and Papadopoulos, P. (2004). Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom. In *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*. IEEE Computer Society.

Ansel, J., Kamil, S., Veeramachaneni, K., Ragan-Kelley, J., Bosboom, J., O'Reilly, U.-M. and Amarasinghe, S. (2014). Opentuner: An extensible framework for program autotuning. In *Proceedings of the 23rd International Conference on Parallel Architectures and Compilation*. ACM.

Bergstra, J., Pinto, N. and Cox, D. (2012). Machine learning for predictive auto-tuning with boosted regression trees. In *Innovative Parallel Computing (InPar), 2012*. IEEE.

Breiman, L., Friedman, J. H., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

Briggs, W. L., Henson, V. E. and McCormick, S. F. (2000). *A Multigrid Tutorial: Second Edition*. Philadelphia, PA, USA: SIAM.

Chan, C., Ansel, J., Wong, Y. L., Amarasinghe, S. and Edelman, A. (2009). Autotuning multigrid with petabricks. In *High Performance Computing Networking, Storage and Analysis, Proceedings of the Conference on*. IEEE.

Chen, R.-B., Hsieh, D.-N., Hung, Y. and Wang, W. (2013). Optimizing latin hypercube designs by particle swarm. *Statistics and Computing* **23**, 663–676.

Chipman, H. A., George, E. I. and McCulloch, R. E. (1998). Bayesian cart model search (with discussion and a rejoinder by the authors). *Journal of the American Statistical Association*, 935960.

Collins, A., Fensch, C., Leather, H. and Cole, M. (2013). Masif: Machine learning guided auto-tuning of parallel skeletons. In *High Performance Computing (HiPC), 2013 20th International Conference on*. IEEE.

Darwish, M., Sraj, I. and Moukalled, F. (2009). A coupled finite volume solver for the solution of incompressible flows on unstructured grids. *Journal of Computational Physics* **228**, 180–201.

Deng, X. W., Joseph, V. R., Sudjianto, A. and Wu, C. F. J. (2009). Active learning through sequential design, with applications to detection of money laundering. *Journal of the American Statistical Association* **104**, 969–981.

Falch, T. L. and Elster, A. C. (2015). Machine learning based auto-tuning for enhanced OpenCL performance portability. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*. IEEE.

Fang, K.-T., Lin, D. K. J., Winker, P. and Zhang, Y. (2000). Uniform design: Theory and application. *Technometrics* **42**, 237–248.

Forrester, A., Sobester, A. and Keane, A. (2008). *Engineering Design via Surrogate Modelling: a Practical Guide*. John Wiley & Sons.

Franchetti, F., Püschel, M., Voronenko, Y., Chellappa, S. and Moura, J. M. F. (2009). Discrete Fourier transform on multicore. *Signal Processing Magazine, IEEE* **26**, 90–102.

Frigo, M. and Johnson, S. G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE* **93**, 216–231.

Fujii, A. (2014). AMGS: Algebraic multigrid solvers. `Http://hpcl.info.kogakuin.ac.jp/lab/software/amgs`.

Fujii, A. and Marques, O. (2014). Axis communication method for algebraic multigrid solver. *IEICE Transactions on Information and Systems* **97**, 2955–2958.

Fujii, A., Nishida, A. and Oyanagi, Y. (2003). A parallel AMG algorithm based on domain decomposition. *IPSJ Transactions on Advanced Computing System* **44**, 9–17.

Fujii, A., Nishida, A. and Oyanagi, Y. (2005). Evaluation of parallel aggregate creation orders: Smoothed aggregation algebraic multigrid method. In *High Performance Computational Science and Engineering*. Springer, pp. 99–122.

Ganapathi, A., Datta, K., Fox, A. and Patterson, D. (2009). A case for machine learning to optimize multicore performance. In *First USENIX Workshop on Hot Topics in Parallelism*.

Gramacy, R. B., Taddy, M. and Wild, S. M. (2013). Variable selection and sesitivity analysis using dynamic trees, with an application to computer code performance tuning. *The Annals of Applied Statistics* **7**, 51–80.

Gramacy, R. B. and Taddy, M. A. (2010). Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an r package for treed gaussian process models. *Journal of Statistical Software*, 1–48.

Gramacy, R. B. and Taddy, M. A. (2016). *tgp: Bayesian Treed Gaussian Process Models*. R Foundation for Statistical Computing, Vienna, Austria. `Https://cran.r-project.org/web/packages/tgp/index.html`.

Hutter, F., Hoos, H. H. and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*. Springer.

Hutter, F., Hoos, H. H., Leyton-Brown, K. and Murphy, K. P. (2009). An experimental investigation of model-based parameter optimisation: Spo and beyond. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*. ACM.

Jones, D. R., Schonlau, M. and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. of Global Optimization* **13**, 455–492.

Katagiri, T., Kise, K., Honda, H. and Yuba, T. (2006). Abclibscript: A directive to support specification of an auto-tuning facility for numerical software. *Parallel Computing* **32**, 92–112.

Katagiri, T., Ohshima, S. and Matsumoto, M. (2014). Auto-tuning of computation kernels from an fdm code with ppOpen-AT. In *Embedded Multicore/Manycore SoCs (MCSoc), 2014 IEEE 8th International Symposium on*. IEEE.

Lophaven, S. N., Nielsen, H. B. and Sondergaard, J. (2002). Aspects of the Matlab toolbox DACE. Tech. Rep. IMM-REP-2002-13, Technical University of Denmark.

Mariani, G., Palermo, G., Zaccaria, V. and Silvano, C. (2012). Oscar: An optimization methodology exploiting spatial correlation in multicore design spaces. *IEEE Trans. Comput.-Aided Des. Intergr. Circuits* **31**, 740–753.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.

Monsifrot, A., Bodin, F. and Quiniou, R. (2002). A machine learning approach to automatic production of compiler heuristics. In *Artificial Intelligence: Methodology, Systems, and Applications*. Springer, pp. 41–50.

Murata, R., Irie, J., Fujii, A., Tanaka, T. and Katagiri, T. (2015). Enhancement of incremental performance parameter estimation on ppOpen-AT. In *Embedded Multicore/Many-core Systems-on-Chip (MCSoC), 2015 IEEE 9th International Symposium on.* IEEE.

Pellegrini, S., Fahringer, T., Jordan, H. and Moritsch, H. (2010). Automatic tuning of MPI runtime parameter settings by using machine learning. In *Proceedings of the 7th ACM International Conference on Computing Frontiers.* ACM.

Qian, P. Z. G. (2012). Sliced latin hypercube designs. *Journal of the American Statistical Association* **107**, 393–399.

Qian, P. Z. G., Wu, H. and Wu, C. F. J. (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50**, 383–396.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science* **4**, 409–423.

Santner, T. J., Williams, B. and Notz, W. (2003). *The Design and Analysis of Computer Experiments.* Springer-Verlag.

Suda, R. (2011). A Bayesian method of online automatic tuning. In *Software Automatic Tuning.* Springer, pp. 275–293.

Ţăpuş, C., Chung, I.-H., Hollingsworth, J. K. et al. (2002). Active harmony: Towards automated performance tuning. In *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing.* IEEE Computer Society Press.

Trefethen, L. N. and Bau III, D. (1997). *Numerical Linear Algebra.* SIAM.

Vaněk, P., Mandel, J. and Brezina, M. (1996). Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing* **56**, 179–196.

Whaley, R. C., Petitet, A. and Dongarra, J. J. (2001). Automated empirical optimizations of software and the atlas project. *Parallel Computing* **27**, 3–35.

Ylvisaker, B. and Hauck, S. (2011). Probabilistic auto-tuning for architectures with complex constraints. In *Proceedings of the 1st International Workshop on Adaptive Self-Tuning Computing Systems for the Exaflop Era.* ACM.

Zhou, Q., Qian, P. Z. G. and Zhou, S. (2011). A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics* **53**, 266–273.

7F, No.337, Fuxing N. Rd., Songshan Dist., Taipei 105, Taiwan, R.O.C.

E-mail: jrpg0618@gmail.com

Department of Statistics, National Cheng Kung University, Tainan, 70101, Taiwan, R.O.C.

E-mail: rbchen@mail.ncku.edu.tw

1-24-2 Nishishinjuku, Shinjuku-ku, Tokyo 163-8677, Japan.

E-mail: fujii@cc.kogakuin.ac.jp

7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan.

E-mail: reiji@is.s.u-tokyo.ac.jp

No. 1, Sec. 4, Roosevelt Road, Taipei, 10617 Taiwan, R.O.C.

E-mail: wwang@ntu.edu.tw

(Received March 2017; accepted April 2017)