

A Bootstrap-Surrogate Approach for Sequential Experimental Design for Simulation Models

Rommel G. Regis(✉)

Saint Joseph's University, Philadelphia, PA 19131, USA
rregis@sju.edu

Abstract. The bootstrap method is a widely used tool for quantifying the uncertainty associated with a given statistical estimator or machine learning method. This paper proposes a novel approach for sequential experimental design that uses the bootstrap in conjunction with an interpolating surrogate model. Consider the problem of fitting a surrogate to a computationally expensive simulation model that yields a numerical output given the values of a set of continuous input variables. To fit a surrogate, initial data points are obtained by running the simulations at a set of space-filling design points. The proposed Bootstrap-Surrogate method improves on this experimental design by sequentially identifying points where the surrogate prediction uncertainty is high and then evaluating the simulation at those points. The surrogate prediction uncertainty at a candidate simulation point is estimated using a weighted combination of two criteria, one based on the bootstrap standard error of the surrogate predictions at the candidate point and the other based on the minimum distance of the candidate point from previous design points. The method is implemented using a radial basis function (RBF) surrogate and tested on groundwater bioremediation models and several test problems. The results show that the Bootstrap-RBF approach generally yields better experimental design points for surrogate modeling (as measured by RMSE on a large test set) than those obtained by an optimum Latin hypercube sample or a sequential experimental design based on a maximin criterion on the groundwater bioremediation models and on the test problems with similar structures.

Keywords: Experimental design · Simulation model · Surrogate model · Bootstrap method · Radial basis functions

1 Introduction

This paper examines the problem of identifying a set of experimental design points that can be used to build a surrogate for a computationally expensive simulation model. Specifically, suppose we have a deterministic simulation model represented by $y = f(x)$ where $x \in [\ell, u] \subset \mathbb{R}^d$ and the functional relationship f

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022 Workshops, LNCS 13377, pp. 498–513, 2022.
https://doi.org/10.1007/978-3-031-10536-4_33

is a black box. That is, the mathematical form of f is not explicitly available and all we have is a simulator that yields a single numerical output y given an input vector $x \in \mathbb{R}^d$. Deterministic simulation models are found in many engineering applications involving finite element or computational fluid dynamics simulations. Our goal is to identify a set of design points in the domain $[\ell, u]$ where the simulation can be run thereby producing data points that can be used to build a surrogate model for $f(x)$. Here, a surrogate model is just an inexpensive approximation of the expensive function $f(x)$. In many cases, the goal is not simply to build a surrogate, but to also to find a global optimum of the expensive function. However, in this paper, the focus is simply on building a surrogate that is as accurate as possible given a limited budget on function evaluations. Once such a surrogate is available along with the corresponding experimental design points, one can then perform optimization on the surrogate, or if additional computational budget is available, use a surrogate-based optimization method to further improve the model in promising regions while searching for the global optimum. Examples of widely used surrogates for simulation and optimization models in the literature include Kriging or Gaussian Process models (e.g., see [3,13,15]), and Radial Basis Function (RBF) models (e.g., see [4,23]).

There is a substantial body of literature on surrogate-based approaches that find an approximate global or local minimum of an expensive simulation-based function (e.g., [24,30]). Among the most popular approaches include Bayesian optimization methods (e.g., [2,22]) and radial basis function methods (e.g., [7,27,31]). These surrogate-based approaches begin with a space-filling experimental design such as a Latin hypercube design that is used to build the initial surrogate for the sequential optimization process. In the context of expensive black-box optimization, the main goal is optimization and expending computational effort to build an accurate global model of $f(x)$ is not always the best approach as has been observed with the success of local surrogate-based approaches in comparison with global surrogate-based methods (e.g., [27]).

There is also a large body of work on experimental designs for black-box simulation-based models along with some optimality criteria for these designs (e.g., see [14,18,28]). Experimental designs can be static/one-shot or adaptive/sequential. A static or one-shot approach tends to be space-filling since the goal is to cover as much of the domain as possible without taking into account any information about the black-box function. In many applications, particularly those involving expensive simulation models, the adaptive or sequential approach is preferred since it incorporates information obtained from previous design points to perform a trade-off between exploration (improving coverage) and exploitation (sampling interesting regions) in selecting the next design point where the simulation will take place [8]. The interesting regions are those where the black-box function is highly nonlinear or those where the uncertainty of the surrogate approximation is high. Sequential experimental design approaches are also referred to as active learning methods in the field of machine learning. Several approaches have been proposed for sequential or adaptive experimental design, including the LOLA (Local Linear Approximation) algorithm

with Voronoi approximation [9], ASRSM (Adaptive Sequential Response Surface Methodology) [1] and Bayesian approaches [20,21]. A review of some of these sequential experimental design approaches can be found in [8] and [16].

This paper presents a novel sequential experimental design approach that uses the bootstrap method [11,12] in conjunction with an interpolating surrogate model. The bootstrap is widely used for quantifying the uncertainty of a statistical estimator or machine learning method. Moreover, the parametric bootstrap and distribution-free bootstrap have been used to estimate the Kriging predictor variance in the context of simulation-optimization [19]. However, it does not appear to have been used solely for sequential experimental design and with other types of surrogates. The proposed Bootstrap-Surrogate experimental design approach is very general and flexible in that it can be used with any surrogate including multivariable polynomials from traditional response surface methodology, Kriging or Gaussian process models, radial basis functions, regression splines, neural networks or support vector regression models. The proposed method begins by running the simulations at a set of space-filling design points. Then, it improves on this experimental design by sequentially identifying candidate points where the surrogate prediction uncertainty is high and then evaluating the simulation at those points. The surrogate prediction uncertainty at a candidate simulation point is estimated using a weighted combination of two criteria, one based on the bootstrap standard error of the surrogate predictions at the candidate point and the other based on the minimum distance of the candidate point from previous design points. The proposed method is implemented using a radial basis function (RBF) surrogate and tested on groundwater bioremediation models and on 17 synthetic test problems. The results show that the Bootstrap-RBF approach generally yields better experimental design points for surrogate modeling (as measured by RMSE on a large test set) than those obtained by an optimum Latin hypercube sample or a sequential experimental design based on a maximin criterion on the groundwater bioremediation models and on the test problems with similar structures.

2 Algorithm Description

This section presents the *Bootstrap-Surrogate* method for sequentially generating experimental design points that are needed to build an accurate surrogate model for an expensive black box function. For simplicity, assume that the domain $[\ell, u] \subset \mathbb{R}^d$ of the black-box function $f(x)$ is simply the unit hypercube $[0,1]^d$. This assumption is not restrictive since any function defined over a hyperrectangle in $\mathbb{R}^d$ can be easily transformed to a function whose domain is the unit hypercube $[0,1]^d$. Moreover, in this paper, we focus only on using interpolating surrogate models such as radial basis functions and Kriging (or Gaussian Process) models since these are widely used in expensive simulation-based optimization. However, some of the ideas here can be used for non-interpolating surrogate models such as multivariable polynomial regression models, regression splines and support vector regression models.

The Bootstrap-Surrogate method begins by performing simulations to obtain the actual function values at an initial set of space-filling design points in the domain $[0,1]^d$ such as a Latin hypercube sample [6]. Next, while the computational budget has not been exhausted, the algorithm sequentially adds a new experimental design point as it goes through a loop. To select the next design point where the simulation will be run, generate trial points uniformly at random throughout the domain. Then, generate b bootstrap samples from the set of previous simulation points. Then, for each bootstrap sample, fit a surrogate using the actual function values from the chosen simulation points, and use this surrogate to predict the function value at each of the trial points. One measure of uncertainty of the surrogate prediction at each trial point is then the standard error of the surrogate predictions from the bootstrap samples. This bootstrap approach is similar to that used in [17] in estimating the standard error of a regression coefficient except that, here, we are estimating the standard error of the prediction. Next, for each trial point, calculate its distance to each of the previous simulation points, and determine the smallest of these distances.

The next step is to calculate a measure of uncertainty of the surrogate prediction at each trial point that incorporates two criteria: (i) the standard error of the surrogate predictions from the bootstrap samples; and (ii) the minimum distance from previous simulation points. A minimum distance criterion is incorporated because the bootstrap standard error criterion alone might not yield the best strategy since it is possible to obtain a high bootstrap standard error at trial points that are very close to previous simulation points. However, since we are using an interpolating surrogate model, the prediction error of the surrogate at each of the previous simulations is zero, and for twice continuously differentiable surrogates, the prediction errors at trial points in small neighborhoods of the any previous simulation point are expected to be small. This combined measure of uncertainty is calculated as follows. For each trial point, calculate a continuous ranking between 0 and 1 of the bootstrap standard error criterion, where a rank of 0 is given to the trial point with the largest value of the bootstrap standard error of the surrogate predictions and a rank of 1 is given to the trial point with the smallest value of this bootstrap standard error criterion. Similarly, for each trial point, calculate a continuous ranking between 0 and 1 of the minimum distance criterion, where a rank of 0 is given to the trial point with the maximum value of the minimum distance to the previous simulation points among all trial points, and a rank of 1 is given to the trial point with the smallest value of this minimum distance criterion. Next, for each trial point, calculate a weighted ranking of the standard error criterion and the minimum distance criterion using a weight $0 < w_B < 1$ for the former. We require the weight w_B of the bootstrap standard error criterion to be strictly positive and strictly less than 1, so that both criteria are always incorporated. The next simulation point is then chosen to be the trial point with the best (minimum) weighted ranking of the bootstrap standard error and the minimum distance criteria. Then, the simulator is run at the selected trial point, and then the algorithm iterates until the computational budget in terms of number of simulations is reached. Finally, the algorithm

returns the sequential experimental design points obtained together with the function values and the surrogate approximation to the expensive function.

Below is a pseudo-code of the proposed Bootstrap-Surrogate method for sequential experimental design. Here, n is the counter for the number of simulations, x_n is the nth simulation point, $\mathcal{P}_n$ is the set of data points after n simulations, and $\mathcal{T}_n$ is the set of uniform random trial points where the $(n+1)$st simulation point will be selected. Moreover, $\widehat{\mathrm{SE}}_n^B(x)$ is the standard error of the surrogate predictions at $x \in \mathcal{T}_n$ from the bootstrap samples and $\Delta_n(x)$ is the minimum distance of $x \in \mathcal{T}_n$ from previous simulation points $\{x_1, \ldots, x_n\}$.

Bootstrap-Surrogate Algorithm

Inputs:

- Black box function $f(x)$ defined for all $x \in [0,1]^d$ (e.g., a computationally expensive simulator).
- Size of the experimental design to build, denoted by N.
- Initial space-filling experimental design $\mathcal{I}_0 = \{x_1, \ldots, x_{n_0}\}$ such that $n_0 < N$ (e.g., an optimum Latin hypercube sample).
- An interpolating surrogate model to approximate the black box function (e.g., radial basis functions or Gaussian Process models).
- Number of uniform random trial points to generate on $[0,1]^d$ in each iteration, denoted by t_{rand}.
- Number of bootstrap samples, denoted by b.
- Weight for the bootstrap standard error criterion: $0 < w_B < 1$.

Outputs: An extension of the initial experimental design $\mathcal{I}_0$ together with the function values at the design points and the surrogate approximation to $f(x)$.

(1) *(Initial Experimental Design Simulations)* Perform simulations to obtain $f(x)$ at all $x \in \mathcal{I}_0$. Set the simulation counter $n \leftarrow |\mathcal{I}_0| = n_0$ and the simulation data points $\mathcal{P}_{n_0} = \{(x_1, f(x_1)), \ldots, (x_{n_0}, f(x_{n_0}))\}$.

(2) *(Perform Iterations)* While $n < N$, do:

(a) *(Generate Random Trial Points)* Generate t_{rand} trial points $\mathcal{T}_n$ uniformly at random throughout $[0,1]^d$.

(b) *(Estimate Uncertainty of Surrogate Prediction at Each Trial Point)* Generate b bootstrap samples from the current set of simulation data points $\mathcal{P}_n$. For each bootstrap sample, fit a surrogate using the data points in the bootstrap sample and use this surrogate to predict $f(x)$ for all $x \in \mathcal{T}_n$. Calculate the standard error of the surrogate predicted values, denoted by $\widehat{\mathrm{SE}}_n^B(x)$, which is the bootstrap estimate of the standard error of the surrogate prediction at x.

(c) *(Calculate Minimum and Maximum Standard Error of Surrogate Prediction Among Trial Points)* Calculate $\widehat{\mathrm{SE}}_n^{B,\min} = \min\{\widehat{\mathrm{SE}}_n^B(x) \mid x \in \mathcal{T}_n\}$ and $\widehat{\mathrm{SE}}_n^{B,\max} = \max\{\widehat{\mathrm{SE}}_n^B(x) \mid x \in \mathcal{T}_n\}$.

(d) *(Calculate Minimum Distance of Each Trial Point from Previous Simulation Points)* For each $x \in \mathcal{T}_n$, calculate $\Delta_n(x) = \min_{1 \le i \le n} \|x - x_i\|$, where $\|\cdot\|$ is the 2-norm.

(e) *(Calculate Minimum and Maximum of the Minimum Distances from Previous Simulation Points)* Calculate $\Delta_n^{\min} = \min\{\Delta_n(x) \mid x \in \mathcal{T}_n\}$ and $\Delta_n^{\max} = \max\{\Delta_n(x) \mid x \in \mathcal{T}_n\}$.

(f) *(Calculate Standard Error Rank for Each Trial Point)* For each $x \in \mathcal{T}_n$, calculate

$$\mathcal{R}_n^B(x) = \frac{\widehat{\mathrm{SE}}_n^{B,\max} - \widehat{\mathrm{SE}}_n^B(x)}{\widehat{\mathrm{SE}}_n^{B,\max} - \widehat{\mathrm{SE}}_n^{B,\min}}$$

provided $\widehat{\mathrm{SE}}_n^{B,\min} \neq \widehat{\mathrm{SE}}_n^{B,\max}$; otherwise, set $\mathcal{R}_n^B(x) = 1$.

(g) *(Calculate Minimum Distance Rank for Each Trial Point)* For each $x \in \mathcal{T}_n$, calculate

$$\mathcal{R}_n^{\Delta}(x) = \frac{\Delta_n^{\max} - \Delta_n(x)}{\Delta_n^{\max} - \Delta_n^{\min}}$$

provided $\Delta_n^{\min} \neq \Delta_n^{\max}$; otherwise, set $\mathcal{R}_n^{\Delta}(x) = 1$.

(h) *(Calculate Weighted Rank for Each Trial Point)* For each $x \in \mathcal{T}_n$, calculate $\mathcal{R}_n(x) = w_B \mathcal{R}_n^B(x) + (1 - w_B)\mathcal{R}_n^{\Delta}(x)$.

(i) *(Select Next Simulation Point)* Set $x_{n+1} = \operatorname{argmin}\{\mathcal{R}_n(x) \mid x \in \mathcal{T}_n\}$.

(j) *(Evaluate Function at Simulation Point and Update Data Points)* Run the simulator to get the value of $f(x_{n+1})$. Then, set $\mathcal{P}_{n+1} = \mathcal{P}_n \cup \{(x_{n+1}, f(x_{n+1}))\}$ and reset $n \leftarrow n + 1$.

(3) *(Return the Experimental Design Points Found)* Return the experimental design points and function values given by $\mathcal{P}_n$ and the corresponding surrogate approximation to the black box function.

2.1 Radial Basis Function Interpolation

The Bootstrap-Surrogate algorithm for sequential experimental design is implemented using the interpolating RBF model in Powell [23] and the resulting experimental design method is called *Bootstrap-RBF*. In this interpolating RBF model, each data point is a center and the basis functions are not necessarily Gaussian. This type of RBF model has been successfully used in expensive black-box optimization (e.g., see [7,25,27,31]).

Given n distinct design points $x_1, \ldots, x_n \in \mathbb{R}^d$ with their function values $f(x_1), \ldots, f(x_n)$. We use an interpolating RBF model of the form:

$$s_n(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + c^T x + c_0, \ x \in \mathbb{R}^d,$$

where $\|\cdot\|$ is the 2-norm, $\lambda_i \in \mathbb{R}$ for $i = 1, \ldots, n$, $\phi(r) = r^3$ (cubic form), $c \in \mathbb{R}^d$ and $c_0 \in \mathbb{R}$. Here, $c^T x + c_0$ represents a linear polynomial tail in d variables. There are other options for the radial function $\phi(r)$ including $\phi(r) = r^2 \log r$ (thin plate spline) and $\phi(r) = \exp(-\gamma r^2)$ (Gaussian), where γ is a hyperparameter. This model is fit by solving a linear system that is described in [23].

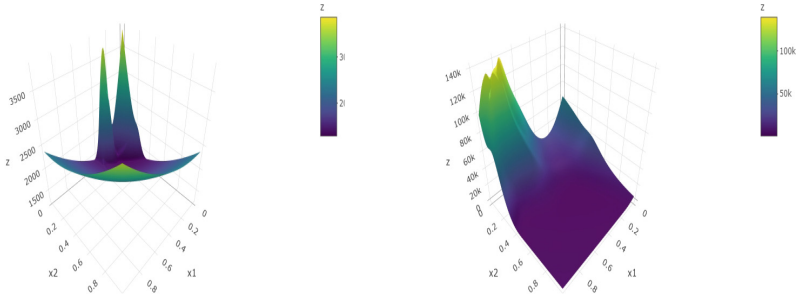

Fig. 1. 3-D surface plots of GWB2A and GWB2B

3 Computational Experiments

3.1 Test Problems

The proposed Bootstrap-RBF method for sequential experimental design is applied to three simulation-based models involving the management of groundwater bioremediation [32] and on 17 synthetic test problems. Groundwater bioremediation is the process by which microorganisms are used to break down harmful chemicals in the groundwater. Groundwater bioremediation simulation models are typically computationally expensive since they often involve numerical solutions of partial differential equations. The synthetic test problems are not simulation-based and are inexpensive to evaluate, but we include them in this study and pretend that each function evaluation is costly by limiting the number of function evaluations allowed. The performance of the proposed method on these synthetic problems will most likely be similar to its performance on simulation-based problems that have similar structure.

The groundwater bioremediation problems, labeled as GWB2A, GWB2B and GWB3, are 2-D and 3-D versions of the simulation-based problem used in [27] where some of the variables have been fixed. Figure 1 shows the 3-D surface plots of the GWB2A and GWB2B problems. The plots reveal that the highly nonlinear part of the surfaces in these problems are mostly limited to a corner or one side of the domain $[0, 1]^2$.

Next, the synthetic problems Skewed Multimodal1 (SKMM1), Skewed Multimodal2 (SKMM2) and the Modified Shekel problems (ModShekel2A, ModShekel2B up to ModShekel3C) have characteristics similar to that of the groundwater bioremediation problems where the highly nonlinear part is confined to a corner or side of the domain. In particular,

$$\text{SKMM1}(x^{(1)}, x^{(2)}) = -3x^{(1)}x^{(2)} \exp\left(-(x^{(1)})^2 - (x^{(2)})^2\right), \ -2 \le x^{(1)}, x^{(2)} \le 5,$$

and

$$\text{SKMM2}(x^{(1)}, x^{(2)}) = \frac{x^{(1)} + x^{(2)}}{2((x^{(1)})^2 + (x^{(2)})^2 + 16)}, \ -20 \le x^{(1)}, x^{(2)} \le 40.$$

Moreover, the Modified Shekel problems are a variation of the 2-D and 3-D Shekel problems in [26] and have mathematical forms similar to that of the well-known Shekel test problems for global optimization (e.g., see [29]). The nine remaining synthetic test problems are well-known problems from the optimization literature such as the Ackley, Beale, Branin, Goldstein-Price, Hartman3 whose domains were rescaled to the unit hypercube. The R codes for these test functions can be obtained from [29].

The GWB3, ModShekel3A, ModShekel3B, ModShekel3C and Hartman3 are all 3-D problems while the rest are 2-D problems. Future work will include extending the capability of the Bootstrap-RBF approach to higher dimensional problems. In particular, since the bootstrap is a computationally intensive procedure, the method could benefit from the use of high-performance computing techniques.

3.2 Experimental Setup

All numerical experiments are conducted using the R software via the RStudio platform on an Intel(R) Core(TM) i7-7700T CPU @ 2.90GHz, 2904 Mhz, 4 Core(s), 8 Logical Processor(s) Windows-based machine. Moreover, the *boot* [5, 10] and *lhs* [6] R packages are used to implement the proposed method. Two instances of the Bootstrap-Surrogate method with the weights for the bootstrap standard error criterion set to $w_B = 0.50$ and $w_B = 0.25$ are implemented using interpolating RBF surrogates with the cubic form. These methods are labeled as Bootstrap-RBF ($w_B = 0.50$) and Bootstrap-RBF ($w_B = 0.25$). Each Bootstrap-RBF algorithm starts with an optimum Latin hypercube sample (LHS) of size $10(d + 1)$ and is extended to an experimental design of size up to $15(d + 1)$. Factors of $d+1$ are used since $d+1$ is the minimum number of points needed to fit a linear model, which is the simplest surrogate that can be used. Moreover, the number of uniform random trial points generated in each iteration is $t_{\text{rand}} = \min(500d, 1000)$ and the number of bootstrap replicates is $b = 1000$.

The Bootstrap-RBF algorithms are compared with two alternative approaches for constructing the experimental design given various computational budgets starting with $10(d+1)+1 = 10d+11$ (one more than the initial space-filling design used by Bootstrap-RBF) up to $15(d+1)$. One approach is simply an optimum Latin hypercube sample that uses up the entire computational budget. This method is labeled as *Optimum LHS*. The other approach is a sequential experimental design method that begins with the same space-filling design of size $10(d+1)$ as the Bootstrap-RBF method and then sequentially adds a new experimental design point that is far from all previous design points. This approach, labeled *Sequential Maximin Design*, has a similar algorithmic structure as the Bootstrap-Surrogate method except it only uses the Minimum Distance criterion in selecting the random trial point where the simulator will be evaluated.

Both the Optimum LHS and the Sequential Maximin Design approaches attempt to spread out the simulation points so they can explore the entire domain. They generally perform well in terms of obtaining good surrogate models on many well-known test problems that are highly nonlinear. In fact, optimal Latin hypercube designs are widely used as starting points for many sequential

surrogate-based optimization methods (e.g., [30]). However, these approaches do not use any information about the function values. Hence, they are not expected to perform as well as the Bootstrap-RBF approach on problems where the highly nonlinear part of the actual surface, which corresponds to a region of high uncertainty for a surrogate, is limited to a small region of the domain such as that of the groundwater bioremediation problems.

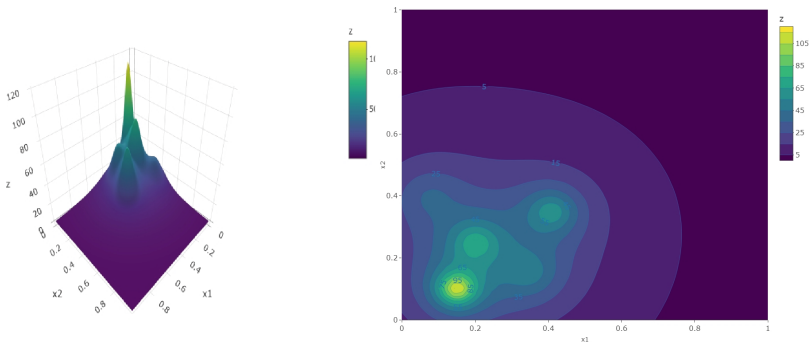

Fig. 2. 3-D surface and contour plots of ModShekel2A

3.3 Results and Discussion

First, we demonstrate the Bootstrap-RBF ($w_B = 0.50$) method on the ModShekel2A test function. Figure 2 shows the 3-D surface plot and contour plot of this function. Next, the top left of Fig. 3 shows the contour plot of the RBF approximation after the $10(d + 1) = 30$ initial experimental design points were evaluated. The top right of the figure shows the corresponding contour plot of the bootstrap estimate of the standard error of the RBF prediction. The bottom left of Fig. 3 shows the contour plot of the RBF approximation after an additional $2(d+1) = 6$ evaluations while the bottom right shows the corresponding contour plot of the bootstrap estimate of the standard error of the RBF prediction. The newly added points (colored red) in the bottom figures tended to be in regions where the uncertainty is high in the top figures.

Each method is run for 30 trials on each of the 20 test problems (3 groundwater models and 17 synthetic problems). To get a fair and more accurate comparison of performance, the same initial optimum LHS of size $10(d+1)$ is used by the Bootstrap-RBF and Sequential Maximin Design methods, though the optimum LHS varies from trial to trial. The experimental design points obtained by the various methods are compared in terms of the Root Mean Square Error (RMSE) on an independent test set of $1000d$ uniform random points on $[0, 1]^d$ of the final interpolating RBF surrogate on the design points. Tables 1 and 2 show the mean and standard error (over 30 trials) of the RMSE of the final RBF surrogate for

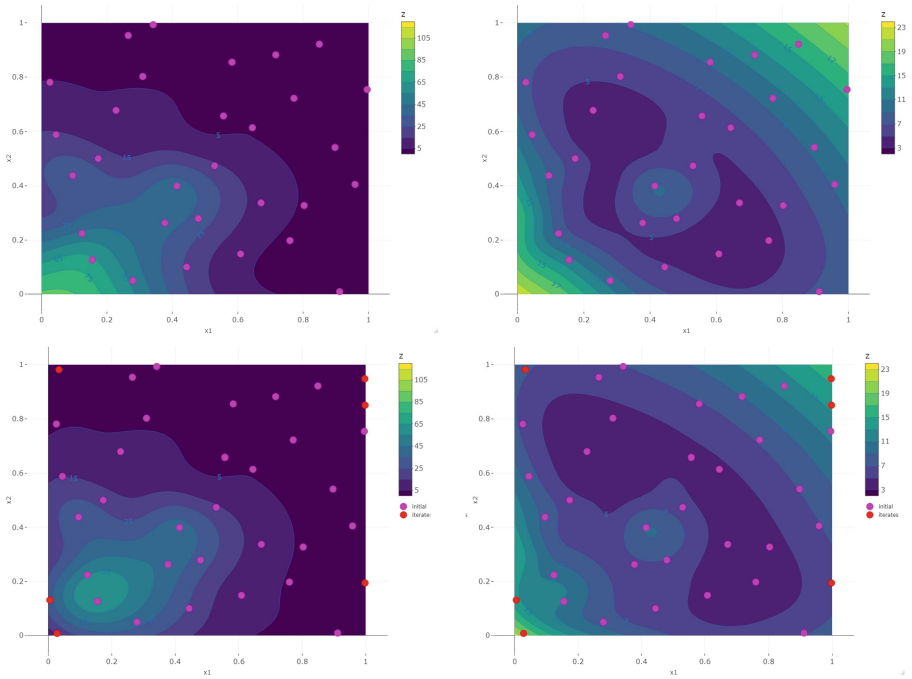

Fig. 3. Contour Plots of RBF approximation (left) and bootstrap estimate of standard error of RBF prediction (right) for ModShekel2A after the initial optimum LHS of $10(d+1) = 30$ points (top) and after $12(d+1) = 36$ simulations (bottom). (Color figure online)

the given function constructed from the experimental design points obtained by Bootstrap-RBF and the alternative methods. The test problems are arranged in the table so that the groundwater bioremediation remediation models are given first, followed by the synthetic test problems whose structures resemble those of the groundwater problems, and finally the well-known synthetic test problems. Moreover, Figs. 4, 5, 6 and 7 show the mean RMSE on the test set for various computational budgets (function evaluations or simulations) beginning with the initial optimum LHS of size $10(d+1)$ up to $15(d+1)$ for some of the problems.

The results show that the Bootstrap-RBF methods are generally better than the Sequential Maximin Design (SeqMaximin) and the Optimum LHS (OptLHS) on the GWB problems and on the synthetic test problems, especially those whose characteristics are similar to the GWB problems. In particular, from Table 2, Bootstrap-RBF ($w_B = 0.25$) is consistently better than OptLHS on all 20 problems, including the three GWB application problems, after $15(d+1)$ simulations. From Table 1, Bootstrap-RBF ($w_B = 0.25$) is also better than OptLHS on 19 of the 20 problems (all except on ModShekel3B) even after a smaller computational budget of only $12(d+1)$ simulations. Moreover, from Table 2, Bootstrap-RBF ($w_B = 0.25$) is better or comparable with SeqMaximin on 16 of the test

Table 1. Mean and standard error (over 30 trials) of the Test Set RMSE of the final RBF surrogate for the given function obtained by various experimental design methods after $12(d+1)$ simulations, where d is the dimension of the problem. Each test set consists of $1000d$ random points from a uniform distribution on $[0,1]^d$. The best result is indicated by a solid blue box and the second best is given by a dashed magenta box.

Test Function	Bootstrap-RBF ($w_B = 0.50$)	Bootstrap-RBF ($w_B = 0.25$)	Sequential Maximin	Optimum LHS
GWB2A	136.5235 (7.7458)	137.2738 (8.1856)	148.381 (9.7442)	163.7199 (10.8559)
GWB2B	3759.2088 (199.0513)	3666.2356 (227.9467)	3851.3788 (213.9667)	4135.8759 (260.661)
GWB3A	6373.6969 (322.7452)	6390.9581 (237.9389)	6206.3263 (321.2515)	6832.68 (391.6532)
SKMM1	0.0992 (0.0052)	0.0994 (0.0053)	0.0989 (0.0054)	0.1054 (0.0075)
SKMM2	0.0076 (6e-04)	0.0082 (5e-04)	0.0082 (6e-04)	0.0084 (5e-04)
ModShekel2A	5.9227 (0.3344)	6.084 (0.4019)	6.0727 (0.3812)	8.3211 (1.2786)
ModShekel2B	4.1322 (0.2522)	4.3843 (0.2955)	4.6127 (0.3063)	5.993 (0.7565)
ModShekel2C	13.3697 (0.8508)	13.4286 (0.8645)	13.6476 (0.8431)	16.2502 (1.5902)
ModShekel3A	1.8575 (0.123)	1.9106 (0.1444)	2.0392 (0.1607)	1.9734 (0.1798)
ModShekel3B	5.3532 (0.9234)	5.4139 (1.0309)	6 (1.3251)	5.4089 (0.5817)
ModShekel3C	3.0242 (0.1967)	3.1252 (0.2139)	3.2585 (0.2252)	3.1983 (0.358)
Ackley	1.0734 (0.036)	1.0453 (0.0324)	1.0334 (0.0302)	1.0635 (0.0393)
Beale	5.30e+06 (2.61e+05)	4.41e+06 (2.27e+05)	4.46e+06 (2.32e+05)	6.30e+06 (2.79e+05)
Scaled Branin	5.9358 (0.5776)	5.0214 (0.3256)	4.796 (0.2539)	6.1424 (0.2908)
Camel6	270.7728 (13.5849)	245.9454 (11.3749)	247.0293 (8.9303)	281.6692 (9.0757)
Scaled Goldstein-Price	2.61e+04 (2.70e+03)	2.11e+04 (1.75e+03)	2.19e+04 (1.75e+03)	3.25e+04 (2.97e+03)
Hartman3	0.3139 (0.0103)	0.3022 (0.0112)	0.301 (0.0102)	0.3132 (0.0113)
Shubert	38.791 (1.6439)	38.6717 (1.6446)	38.7257 (1.5927)	40.4656 (1.7946)
Styblinski-Tang	11.8467 (0.6124)	10.6934 (0.5807)	10.4657 (0.3535)	13.4364 (0.4631)
Zakharov	873.2725 (128.4875)	572.1549 (66.469)	667.718 (65.9827)	1156.8956 (103.4355)

problems (all except on the Branin, Goldstein-Price, Hartman3 and Shubert) after $15(d+1)$ simulations. Also, from Table 1, Bootstrap-RBF ($w_B = 0.25$) is better than or comparable with SeqMaximin on 13 of the test problems, including on GWB2A and GWB2B, after $12(d+1)$ simulations. In addition, on most of the problems where the mean RMSE for Bootstrap-RBF ($w_B = 0.25$) is worse than that for SeqMaximin after either $12(d+1)$ or $15(d+1)$ simulations, the performance of the latter is close to that of the former.

Next, from Tables 1 and 2, Bootstrap-RBF ($w_B = 0.50$) is better than OptLHS on 18 of the 20 problems (all except on Ackley and Hartman3) after $12(d+1)$ simulations and also after $15(d+1)$ simulations. Moreover, from Table 2, Bootstrap-RBF ($w_B = 0.50$) has better mean RMSE values than SeqMaximin after $15(d+1)$ simulations on 11 of the 20 problems, including on GWB2A and GWB3 and on all eight synthetic problems whose characteristics are similar to that of the GWB problems. Also, from Table 1, Bootstrap-RBF ($w_B = 0.50$) is better than SeqMaximin after $12(d+1)$ simulations on only 9 of the 20 problems. However, Bootstrap-RBF ($w_B = 0.50$) did better on GWB2A and GWB2B and on 7 of the 8 synthetic problems that have similar structure to the GWB problems.

Table 2. Mean and standard error (over 30 trials) of the Test Set RMSE of the final RBF surrogate for the given function obtained by various experimental design methods after $15(d+1)$ simulations. Each test set consists of $1000d$ random points from a uniform distribution on $[0,1]^d$. The best result is indicated by a solid blue box and the second best is given by a dashed magenta box.

Test Function	Bootstrap-RBF ($w_B = 0.50$)	Bootstrap-RBF ($w_B = 0.25$)	Sequential Maximin	Optimum LHS
GWB2A	93.893 (7.4832)	110.8165 (6.1768)	125.8128 (6.8813)	152.0495 (7.7863)
GWB2B	3318.0344 (197.6043)	2859.8088 (163.5086)	3011.8386 (176.9743)	3823.7482 (259.4735)
GWB3A	5111.4867 (291.8695)	4984.2276 (260.509)	5165.1892 (199.5755)	6212.3128 (388.0434)
SKMM1	0.0668 (0.006)	0.0663 (0.0055)	0.0788 (0.0057)	0.0766 (0.0056)
SKMM2	0.0037 (3e-04)	0.0058 (4e-04)	0.0068 (5e-04)	0.007 (4e-04)
ModShekel2A	5.1081 (0.2705)	5.0901 (0.3025)	5.417 (0.2943)	6.1908 (0.6207)
ModShekel2B	3.9863 (0.3345)	3.9713 (0.2306)	4.0883 (0.2421)	4.5961 (0.3219)
ModShekel2C	11.3725 (0.5673)	11.6233 (0.826)	12.6408 (0.8945)	13.1401 (1.1904)
ModShekel3A	1.6439 (0.1146)	1.7247 (0.1138)	1.8564 (0.1223)	1.7974 (0.1074)
ModShekel3B	4.8914 (0.7565)	5.0113 (0.8666)	5.1751 (0.8841)	5.5127 (1.2035)
ModShekel3C	2.7259 (0.1712)	2.8662 (0.1474)	2.935 (0.1579)	3.1398 (0.331)
Ackley	1.0372 (0.0323)	0.967 (0.0326)	0.9757 (0.0303)	1.0331 (0.0341)
Beale	4.44e+06 (2.90e+05)	3.12e+06 (1.56e+05)	3.26e+06 (1.45e+05)	5.20e+06 (2.29e+05)
Scaled Branin	4.6224 (0.3067)	3.3324 (0.2032)	2.8131 (0.1425)	4.9123 (0.3683)
Camel6	227.5138 (12.7108)	180.504 (6.9274)	181.3427 (5.9533)	229.1421 (8.591)
Scaled Goldstein-Price	1.88e+04 (1.81e+03)	1.53e+04 (1.03e+03)	1.52e+04 (1.14e+03)	2.49e+04 (2.60e+03)
Hartman3	0.2771 (0.0077)	0.2584 (0.0084)	0.2569 (0.0092)	0.2601 (0.0083)
Shubert	37.7721 (1.2746)	38.5864 (1.4775)	38.2659 (1.3992)	39.4475 (1.5612)
Styblinski-Tang	9.313 (0.5173)	6.9234 (0.2679)	7.0488 (0.2591)	9.8954 (0.3673)
Zakharov	405.3347 (59.2295)	325.3212 (30.0241)	395.5428 (36.162)	811.1953 (87.3768)

Overall, it is worth noting that on the GWB application problems and on the 8 synthetic problems with characteristics that are similar to those of the GWB problems, the Bootstrap-RBF methods consistently obtained better mean RMSEs than those for the alternatives. On the remaining 9 synthetic test problems, the Bootstrap-RBF methods also did better than OptLHS. Moreover, Bootstrap-RBF ($w_B = 0.25$) has comparable performance with SeqMaximin on these test problems. However, SeqMaximin obtained better mean RMSEs than Bootstrap-RBF ($w_B = 0.50$) on these test problems though the mean RMSEs are not too far apart. A reasonable explanation is that since the nonlinearity of the surface is spread throughout the domain on the well-known test problems, having a measure of uncertainty through the bootstrap standard error does not provide much of an advantage over an approach that spreads out the design points. In any case, the performance of the Bootstrap-RBF approach does not seem to significantly deteriorate on the other synthetic test problems.

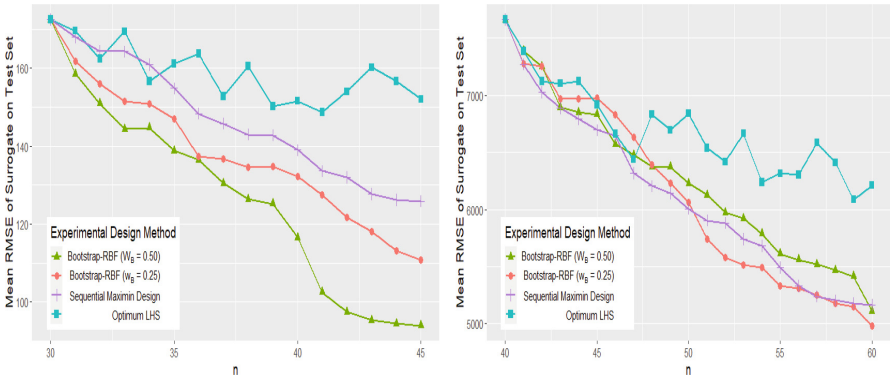

Fig. 4. Results on GWB2A (left) and GWB3 (right)

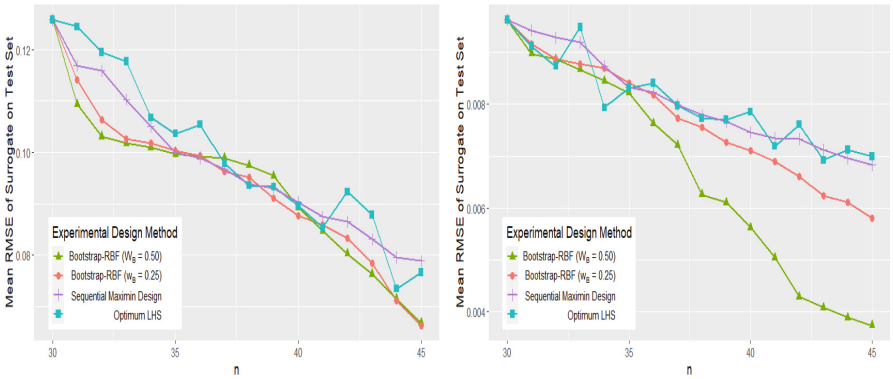

Fig. 5. Results on SKMM1 (left) and SKMM2 (right)

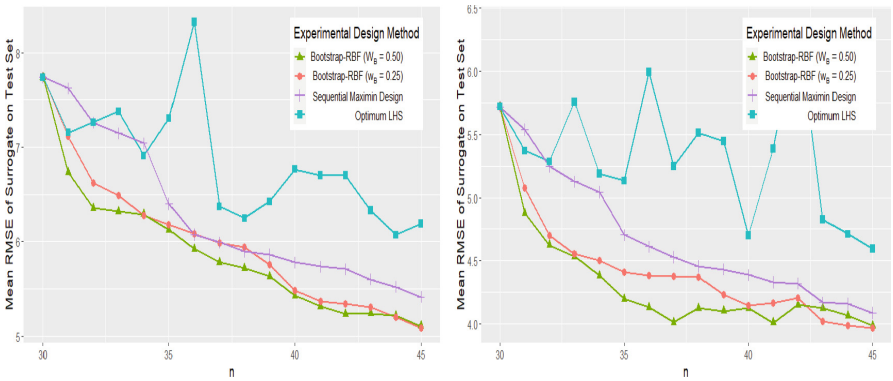

Fig. 6. Results on Modified Shekel2A (left) and Modified Shekel2B (right)

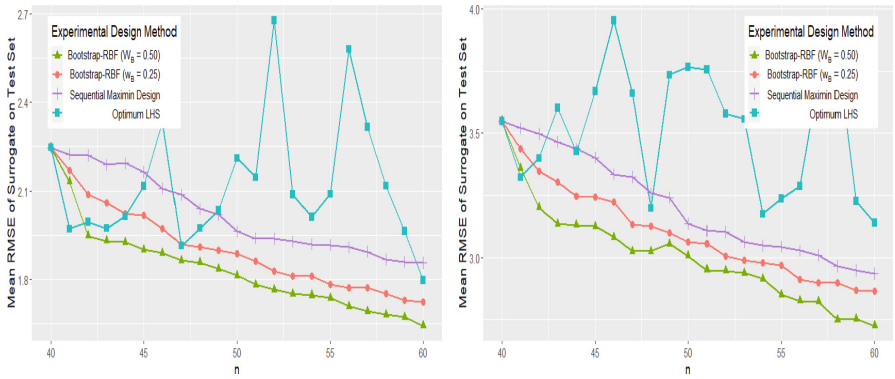

Fig. 7. Results on Modified Shekel3A (left) and Modified Shekel3C (right)

4 Summary and Conclusions

This paper proposed the Bootstrap-Surrogate approach for sequential experimental design for simulation-based functions where a bootstrap standard error criterion and a minimum distance criterion are used to quantify the uncertainty of a surrogate model prediction at a given point. It begins by evaluating the simulation-based function at the points of a space-filling experimental design and then successively identifies the next simulation point as a point where the aggregate uncertainty measure is high. The method is implemented using a radial basis function (RBF) surrogate and using two different weights for the bootstrap standard error criterion. The resulting Bootstrap-RBF methods are compared with an Optimum Latin Hypercube Sample (LHS) and a Sequential Maximin Design (SeqMaximin) on three groundwater bioremediation (GWB) problems and on a series of synthetic test problems. The methods are compared in terms of the RMSE of the final surrogate built using the experimental design points obtained on an independent test set on each of the test problems. The results show that the Bootstrap-RBF methods consistently performed better than OptLHS and SeqMaximin on the three GWB problems and on eight synthetic test problems similar to the GWB problems when given a computational budget of up to $15(d+1)$ simulations. Moreover, the Bootstrap-RBF methods also performed better than OptLHS and are generally competitive with SeqMaximin on another nine widely used test problems. The results suggest that the advantage of the Bootstrap-RBF approach is on problems where the nonlinearity of the function is limited to a smaller subregion of the domain, which is a characteristic of the simulation-based problems involving groundwater bioremediation. However, it also works reasonably well in comparison to a more static approach of using a fixed optimum space-filling design on other test problems where the nonlinearity is spread throughout the entire domain.

Future work will consider adapting the method to work well on higher dimensional problems and on other practical simulation-based problems. Moreover, the

Bootstrap-Surrogate approach has great potential for sequential surrogate-based optimization and provides an alternative to Bayesian global optimization techniques that also use an uncertainty measure in selecting simulation points.

References

1. Alaeddini, A., Yang, K., Murat, A.: ASRSM: a sequential experimental design for response surface optimization. Qual. Reliab. Eng. Int. **29**(2), 241–258 (2013)
2. Archetti, F., Candelieri, A.: Bayesian Optimization and Data Science. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24494-1
3. Bouhlel, M.A., Bartoli, N., Regis, R.G., Otsmane, A., Morlier, J.: Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. Eng. Optim. **50**(12), 2038–2053 (2018)
4. Buhmann, M.: Radial Basis Functions: Theory and Implementations (Cambridge Monographs on Applied and Computational Mathematics). Cambridge University Press, Cambridge (2003)
5. Canty, A., Ripley, B.: boot: Bootstrap R (S-Plus) Functions. R package version 1.3-28 (2021). https://CRAN.R-project.org/package=boot
6. Carnell, R.: lhs: Latin Hypercube Samples. R package version 1.1.3 (2021). https://CRAN.R-project.org/package=lhs
7. Costa, A., Nannicini, G.: RBFOpt: an open-source library for black-box optimization with costly function evaluations. Math. Program. Comput. **10**(4), 597–629 (2018). https://doi.org/10.1007/s12532-018-0144-7
8. Crombecq, K.: Surrogate Modelling of Computer Experiments with Sequential Experimental Design. Ph.D. thesis. Ghent University (2011)
9. Crombecq, K., Gorissen, D., Deschrijver, D., Dhaene, T.: A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. SIAM J. Sci. Comput. **33**(4), 1948–1974 (2011)
10. Davison, A.C., Hinkley, D.V.: Bootstrap Methods and Their Applications. Cambridge University Press, Cambridge (1997). ISBN 0-521-57391-2
11. Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Stat. **7**, 1–26 (1979)
12. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall/CRC, Boca Raton (1993)
13. Forrester, A.I.J., Sobester, A., Keane, A.J.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, Hoboken (2008)
14. Garud, S.S., Karimi, I.A., Kraft, M.: Design of computer experiments: a review. Comput. Chem. Eng. **106**, 71–95 (2017)
15. Gramacy, R.B.: Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences. Chapman and Hall/CRC, New York (2020)
16. Greenhill, S., Rana, S., Gupta, S., Vellanki, P., Venkatesh, S.: Bayesian optimization for adaptive experimental design: a review. IEEE Access **8**, 13937–13948 (2020)
17. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning with Applications in R. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-7138-7
18. Joseph, V.R.: Space-filling designs for computer experiments: a review. Qual. Eng. **28**(1), 28–35 (2016)

19. Kleijnen, J.P.C.: Simulation-optimization via Kriging and bootstrapping: a survey. J. Simul. **8**(4), 241–250 (2014)
20. Kleinegesse, S., Drovandi, C., Gutmann, M.U.: Sequential Bayesian experimental design for implicit models via mutual information. Bayesian Anal. **16**(3), 773–802 (2021)
21. Pandita, P., Tsilifis, P., Awalgaonkar, N.M., Bilionis, I., Panchal, J.: Surrogate-based sequential Bayesian experimental design using non-stationary Gaussian Processes. Comput. Methods Appl. Mech. Eng. **385**, 114007 (2021)
22. Pourmohamad, T., Lee, H.K.H.: Bayesian Optimization with Application to Computer Experiments. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-82458-7
23. Powell, M.J.D.: The theory of radial basis function approximation in 1990. In: Light, W. (ed.) Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions, pp. 105–210. Oxford University Press, Oxford (1992)
24. Regis, R.G.: A survey of surrogate approaches for expensive constrained black-box optimization. In: Le Thi, H.A., Le, H.M., Pham Dinh, T. (eds.) WCGO 2019. AISC, vol. 991, pp. 37–47. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-21803-4_4
25. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng. Optim. **46**(2), 218–243 (2014)
26. Regis, R.G.: Trust regions in Kriging-based optimization with expected improvement. Eng. Optim. **48**(6), 1037–1059 (2016)
27. Regis, R.G., Shoemaker, C.A.: A stochastic radial basis function method for the global optimization of expensive functions. Informs J. Comput. **19**(4), 497–509 (2007)
28. Santner, T.J., Williams, B.J., Notz, W.I.: Space-filling designs for computer experiments. In: The Design and Analysis of Computer Experiments. SSS, pp. 145–200. Springer, New York (2018). https://doi.org/10.1007/978-1-4939-8847-1_5
29. Surjanovic, S., Bingham, D.: Virtual Library of Simulation Experiments: Test Functions and Datasets (2013). http://www.sfu.ca/~ssurjano. Accessed 12 Apr 2022
30. Vu, K.K., D'Ambrosio, C., Hamadi, Y., Liberti, L.: Surrogate-based methods for black-box optimization. Int. Trans. Oper. Res. **24**, 393–424 (2017)
31. Wild, S.M., Regis, R.G., Shoemaker, C.A.: ORBIT: optimization by radial basis function interpolation in trust-regions. SIAM J. Sci. Comput. **30**(6), 3197–3219 (2008)
32. Yoon, J.-H., Shoemaker, C.A.: Comparison of optimization methods for groundwater bioremediation. J. Water Resour. Plan. Manag. **125**(1), 54–63 (1999)