Preprint Cambridge Centre for Computational Chemical Engineering ISSN 1473 – 4273

Evaluating Smart Sampling for Constructing Multidimensional Surrogate Models

Sushant S Garud[1], Iftekhar A Karimi[1], George P E Brownbridge[2], Markus Kraft[3,4]

released: 22 June 2017

[1] Department of Chemical and
Biomolecular Engineering
National University of Singapore
4 Engineering Drive 4
Singapore, 117576
Singapore
E-mail: cheiak@nus.edu.sg

[2] CMCL Innovations
Sheraton House, Castle Park
Cambridge, CB3 0AX
United Kingdom

[3] Department of Chemical Engineering
and Biotechnology
University of Cambridge
New Museums Site, Pembroke Street
Cambridge, CB2 3RA
United Kingdom
E-mail: mk306@cam.ac.uk

[4] School of Chemical and
Biomedical Engineering
Nanyang Technological University
62 Nanyang Drive
Singapore, 637459
Singapore

Preprint No. 185

UNIVERSITY OF CAMBRIDGE

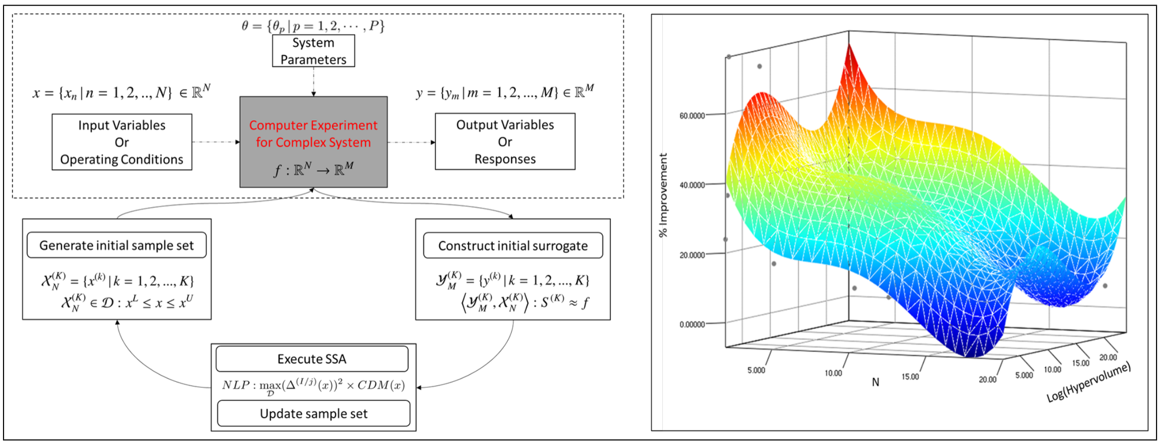
Keywords: Adaptive sampling, Experimental design, Surrogate model, Process flowsheet

Edited by
Computational Modelling Group
Department of Chemical Engineering and Biotechnology
University of Cambridge
New Museums Site
Pembroke Street
Cambridge CB2 3RA
United Kingdom

Fax: + 44 (0)1223 334796
E-Mail: c4e@cam.ac.uk
World Wide Web: `http://como.ceb.cam.ac.uk/`

Abstract

In this article, we extensively evaluate the smart sampling algorithm (SSA) developed by Garud et al. (Computers & Chemical Engineering, 96, 103-114, 2017) for constructing multidimensional surrogate models. Our numerical evaluation shows that SSA outperforms Sobol sampling (QS) for polynomial and kriging surrogates on a diverse test bed of thirteen functions. Furthermore, we compare the robustness of SSA against QS by evaluating them over ranges of domain dimensions and edge length/s. SSA shows consistently better performance than QS making it viable for a broad spectrum of applications. Besides this, we show that SSA performs very well compared to the existing adaptive techniques, especially for the high dimensional case. Finally, we demonstrate the practicality of SSA by employing it for three case studies. Overall, SSA is a promising approach for constructing multidimensional surrogates at significantly reduced computational cost.


$\theta = \{\theta_p \,|\, p = 1, 2, \cdots, P\}$
System Parameters
$x = \{x_n \,|\, n = 1, 2, .., N\} \in \mathbb{R}^N$
Input Variables Or Operating Conditions
Computer Experiment for Complex System
$f : \mathbb{R}^N \rightarrow \mathbb{R}^M$
$y = \{y_m \,|\, m = 1, 2, ..., M\} \in \mathbb{R}^M$
Output Variables Or Responses
Generate initial sample set
$\mathcal{X}_N^{(K)} = \{x^{(k)} \,|\, k = 1, 2, ..., K\}$
$\mathcal{X}_N^{(K)} \in \mathcal{D} : x^L \leq x \leq x^U$
Construct initial surrogate
$\mathcal{Y}_M^{(K)} = \{y^{(k)} \,|\, k = 1, 2, ..., K\}$
$\langle \mathcal{Y}_M^{(K)}, \mathcal{X}_N^{(K)} \rangle : S^{(K)} \approx f$
Execute SSA
$NLP : \max_{\mathcal{D}} (\Delta^{(I/j)}(x))^2 \times CDM(x)$
Update sample set
% Improvement
60.0000
40.0000
20.0000
0.00000
5.000
10.00
15.00
20.00
N
5.000
10.00
15.00
20.00
Log(Hypervolume)

Highlights:

- Extensive numerical evaluation of smart sampling algorithm (SSA) is performed using a diverse test bed of analytical functions.
- Robustness of the performance is examined for SSA over the wide ranges of dimensions and domain sizes.
- Numerical comparison of SSA with existing adaptive approaches is illustrated.
- SSA is employed for three process systems engineering case studies to demonstrate its practical applicability.

Contents

1 Introduction

Process simulators are commonly used to model, study, and analyze complex nonlinear physicochemical systems. However, such simulations are generally computationally intensive, thus, prohibiting their repeated evaluations in a typical analysis procedure. Moreover, the custom-made process simulators are often black-box in nature. Hence, no system information is available to the users without evaluating an instance of this costly simulation. On these accounts, it is beneficial to convert such high-fidelity simulations into computationally inexpensive surrogate models that capture essential features with reasonable numerical accuracy. Surrogate modeling, also known as metamodeling or response surface model, is a technique to generate a mathematical or numerical representation of a complex system based on some sampled input-output data. In a philosophical discussion on the future of computational modeling, Kraft and Mosbach highlight the importance of approximation techniques and experimental designs (sampling techniques) in tackling complex multi-scale systems. The quality of any surrogate approximation depends on a sampling technique used to generate the input-output data and a surrogate modeling technique used to build the approximation. The literature [35] has several forms of surrogate models like polynomial response surface model (PRSM), high dimensional model representation (HDMR), kriging, radial basis functions (RBFs), support vector regression (SVR), artificial neural networks (ANNs), *etc*. Furthermore, many scholars [6, 19, 20] have employed these techniques in the context of various physicochemical systems. Nonetheless, this work focuses on the critical evaluation of a smart and adaptive sampling approach for multidimensional surrogate construction paradigms.

Commonly used sampling techniques employ uniform, quasi-random, or systematic distributions [26, 32]. Examples are factorial design or grid sampling, random sampling, Latin hypercube sampling, orthogonal arrays, Hammersley points, Sobol sampling (QS), *etc*. A recent review by Garud et al. classifies the literature on sampling techniques into three major categories *viz.* static system-free, static system-aided, and adaptive-hybrid. It discusses each of them thoroughly and identifies their advantages and disadvantages. The static techniques are often prone to the curse of dimensionality. Moreover, they can result in under/oversampling and thus, resulting in poor system approximation [17]. In order to tackle these issues, a new upcoming class of modern DoE (design of experiments) called adaptive sampling (sequential sampling) has gained attention from the research community over the past few years. Adaptive sampling approach has two vital advantages over the static ones *viz.* low computational expense and better approximation quality [9]. Typically, an adaptive sampling technique starts with a small set of sample points, and then adds points sequentially based on some user-defined criterion. Such criterion involves an objective (sometimes referred as a *score*) that aims to fill the domain (exploration) as well as improve the overall surrogate quality (exploitation) [9, 17]. We summarize various adaptive approaches from the literature and their vital characteristics like the exploration and exploitation criteria, dependence on the surrogate form, and the placement approach in Table 1. Although, we only discuss the key works from the adaptive sampling literature, Garud et al. has dedicated an entire section for their discussion and the interested readers may refer to it for further details.

Jin et al. propose two approaches, namely the maximin scaled distance (MSD) and the

Table 1: *Overview of the adaptive sampling literature. (Mm: maximin distance, CVE: cross validation error, MD: Mahalanobis distance, ME: maximum entropy, VT: Voronoi tessellation, LOLA: local linear approximation, CC: clustering constraint, EE: expected error, NN: nearest neighbor, JK: Jackknifing, DT: Delaunay triangulation, MSE: maximum sampling error, CDM: crowding distance metric, DF: departure function)*

Author/s	**Exploration**	**Exploitation**	**Surrogate Dependence**	**Placement Approach**
Jin et al.	Mm	CVE	×	Optimization
Busby et al.	MD	ME	✓(Kriging)	Score
Crombecq et al.	VT	LOLA	×	Score
Li et al.	CC	EE	✓(Kriging)	Optimization
Xu et al.	VT	CVE	✓ (Kriging)	Optimization
Eason and Cremaschi	NN	JK	×	Score
Ajdari and Mahlooji	DT	CVE	×	Score
Cozad et al.	-	MSE	×	Optimization
Garud et al.	CDM	DF	×	Optimization

cross validation (CV). The former is a modification of maximin distance based sampling that utilizes system information by assigning weights to the important variables while the latter uses CV error [27] to place new sample points. The CV approach can be viewed as a maximum sampling error approach with an additional feature of clustering constraint. Crombecq et al. propose a novel and generic *score* based sequential strategy involving exploration and exploitation. They use a combination of derivative-based local linear approximations and Voronoi tessellations to place new sample points. Although the LOLA-Voronoi strategy has shown some promising results, it can be computationally intensive for large N. A recent work by Eason and Cremaschi proposes an adaptive sampling strategy for ANN surrogates. Instead of generating all sample points in one shot, they choose them gradually based on some score from randomly generated sample sets. The score considers the normalized nearest neighbor distance of a potential point from the current sample points and its normalized expected variance evaluated using jackknifing [14]. Though their selection of sample points is systematic, it is still from randomly generated points. Cozad et al. propose an adaptive sampling for their surrogate modeling tool called ALAMO. They add sample points one at a time to the initial sample set. For each new sample point, they solve a derivative-free optimization problem to maximize the deviation of the surrogate from the real function. This can obviously be compute-intensive, as it requires the evaluation of the real function during optimization.

To this end, the adaptive sampling techniques in the literature can be broadly classified as either score-based or optimization-based. Although the latter strategies aim at the optimal sample placement, the literature suggests that such approaches are employed only with kriging surrogate due to its ready availability of the error estimate. Furthermore, these approaches may not be suitable for a wide range of problems as the performance of kriging may drop significantly with increasing dimensions. This can be tackled by using the surrogate techniques other than kriging. However, the literature clearly points out that surrogate (kriging)-independent approaches are score-based and lack the placement optimality. Therefore, there is a need for surrogate-independent and optimization-based adaptive sampling approach which is generic, robust, and ascertains optimal sample placement. Garud et al. address this exact conundrum by proposing a novel adaptive sampling strategy, namely smart sampling algorithm (SSA). It uses crowding distance metric to identify the unexplored regions while departure function to identify the regions with complex behavior. These two concepts are then combined into an objective to formulate a point placement optimization problem. SSA iteratively solves this optimization problem to place new sample points. SSA has been developed and presented in our previous work [17] along with its application to one dimensional cases. In this work, we present the critical evaluation of SSA for constructing multidimensional surrogate models.

This article is organized as follows. Section 2 gives a brief overview of SSA followed by our evaluation basis and plan in section 3. We present the numerical results in section 4 and section 5 shows the practical application of SSA using three case studies from the chemical and process systems engineering field. Finally, in section 6, we present our conclusions.

2 Overview of SSA

Herein, we present a brief overview of SSA for the sake of completeness. The readers may refer to the article by Garud et al. for the details on the development thought-process. Let $y = f(x)$; $f : \mathbb{R}^N \to \mathbb{R}^M$ for $\mathcal{D} : x^L \leq x \leq x^U$ describe the behavior of a unit/process/system whose experimental or computational quantification is complex and computationally expensive. Thus, we need an analytical or numerical surrogate model $S(x)$ to replace $f(x)$ so that $y \approx S(x)$. Here onwards, we denote $S(x)$ by S for the sake of convenience. To this end, SSA solves the following problem:

Given:

- $y = f(x)$; $f : \mathbb{R}^N \to \mathbb{R}^M$ for $x^L \leq x \leq x^U$. Note that we consider $M = 1$ throughout this article.
- A mathematical form for S.
- Upper limit (K_{max}) on the number of sample points at which $f(x)$ may be evaluated to obtain S, or a desired accuracy for S.

Obtain:

- K_{max} sample points $\{x^{(k)} \,|\, k = 1, 2, \cdots, K_{max}\}$ that give the best S for approximating $f(x)$.
- Or, the sample points that give S with a prescribed accuracy for approximating $f(x)$.

In SSA, we quantify the exploration of $\mathcal{D}$ using crowding distance metric (CDM) as follows.

$$CDM(x) = \sum_{i=1}^{I} (||x - x^{(i)}||_2)^2 \tag{1}$$

where $||\cdot||_2$ is the Euclidean norm. $CDM(x)$ quantifies the relative isolation of x and the greater the $CDM(x)$, the greater its isolation, thus, making the neighborhood of x a potential candidate for sample placement. Additionally, we quantify the exploitation using departure function defined as follows. Let $S^{(I)}(x)$ denote a surrogate constructed using sample set $\mathcal{X}_N^{(I)} = \{x^{(i)} \,|\, i = 1, 2, \cdots, I\}$. Let $x^{(j)}$ be a sample point in $\mathcal{X}_N^{(I)}$, and let $S^{(I/j)}(x)$ be the surrogate constructed using $\mathcal{X}_N^{(I/j)} = \{x^{(i)} \,|\, i = 1, 2, \cdots, I, i \neq j\}$. Then, departure function is given by Eq.(2).

$$\Delta^{(I/j)}(x) = S^I(x) - S^{(I/j)}(x) \qquad j = 1, 2, \cdots, I \tag{2}$$

Qualitatively, it determines the impact of locating a sample point in the neighborhood of $x^{(j)}$ on $S^I(x)$. The larger the departure function value, the greater the placement impact, hence, the more plausible the region for sample placement.

A single objective that combines the above discussed two concepts of CDM and departure function can yield the best new sample point. This is achieved by formulating a point placement optimization problem as follows. Given a sample set $\mathcal{X}_N^I$ and a surrogate $S^I(x)$ constructed using it, we aim to place the new point as far away from existing points as possible, and the new point should have the highest impact on $S^I(x)$. Therefore, we formulate a series of NLPs given in Eq.(3). The optimal solution to this NLP (Eq.(3)) can be a good candidate for new sample point.

$$NLP(j) : \max_{\mathcal{D}} (\Delta^{(I/j)}(x))^2 \times CDM(x) \qquad j = 1, 2, \cdots, I \tag{3}$$

SSA employs these concepts iteratively and adaptively to place new samples, thus, comprising of following key distinct features:

- Single analytical objective consisting exploration and exploitation for placement optimization;
- Inexpensive placement optimization due to surrogate-based objective;
- Placement strategy independent of the surrogate model type and thus, applicable over a wide range of problems.

For $K\,(< K_{max})$ initial sample points and a surrogate model type S, SSA proceeds as follows.

1. Generate a sample set $\mathcal{X}_N^K = \{x^{(i)} \,|\, i = 1, 2, \cdots, K\}$ using any modern DoE technique *e.g.* QS.
2. Compute $\mathcal{Y}_M^K = \{y^{(i)} \,|\, i = 1, 2, \cdots, K\}$ using a sample set $\mathcal{X}_N^K$. Note that throughout our discussion $M = 1$. Thus, it is dropped from the subscript here onwards for the sake of convenience.

3. Set $k = K$.

4. Construct $S^k(x)$ using $\mathcal{X}_N^k$ and $\mathcal{Y}^k$.

5. If $k = K_{max}$, then $S(x) = S^{(k)}(x)$ and STOP. Otherwise, proceed to step 6.

6. Compute $CDM^{(j)} = CDM(x^{(j)})\ \forall\ x^{(j)} \in \mathcal{X}_N^k$ and $j = 1, 2, \cdots, k$ using Eq.(1). Arrange $CDM^{(j)}$ $(j = 1, 2, \cdots, k)$ in descending order and define the order as $p = 1, 2, \cdots, k$.

   (a) Set $p = 1$.

   (b) Construct $S^{(k/p)}(x)$ using data from Steps 1, 2, and $i \neq p$.

   (c) Construct and solve $NLP(p)$ given Eq.(3). Let x^* be the optimal solution. If $||x^* - x^{(i)}||_2 \leq \varepsilon$ for any $i = 1, 2, \cdots, k$, then set $p = p + 1$ and go to Step 6 (b). Otherwise, $x^{(k+1)} = x^*$, evaluate $y^{(k+1)}$ and go to Step 4.

3 Evaluation basis and plan

We now present a detailed plan for the evaluation of SSA for constructing multi-dimensional surrogates. For this, we use two surrogate model types and compare the performance of SSA against a variety of commonly used sampling techniques. This evaluation is performed using a diverse test bed of analytical functions. Additionally, the robustness of SSA is analyzed for the wide ranges of domain sizes and dimensions. Finally, three different performance metrics are employed for these analyses to assure a thorough comparison of sampling techniques.

3.1 Surrogate models

As discussed earlier, one of the key features of SSA is its ability to function with any surrogate modeling technique. Naturally, this requires us to analyze the performance of SSA using different surrogate models. Therefore, we employ PRSM (Eq.(4) and kriging (Eq.(5)) for the evaluation of SSA. These two models are deliberately chosen to show that SSA performs very well for both regression (in case of PRSM) as well as interpolation (in case of kriging) based techniques. Moreover, we use kriging for the comparative illustration of SSA with adaptive techniques in the literature [38] as discussed next in section 3.2. Eq.(4) illustrates second order PRSM and it can easily be generalized to any order $\rho_p \in \mathbb{N}$. The detailed theory and implementation of PRSM can be found in the article by Sikorski et al..

$$y \approx S_{PRSM} = \beta_0 + \sum_{n=1}^{N} \beta_n x_n + \sum_{n=1}^{N} \beta_{nn} x_n^2 + \sum_{n=1}^{N} \sum_{p=n+1}^{N} \beta_{np} x_n x_p \tag{4}$$

Eq.(5) shows a general form for kriging interpolator and we use polynomial basis functions (g_b) with kriging order ρ_k. Kleijnen discusses the theory behind kriging and its implementation in detail.

$$y \approx S_{KRG} = \sum_{b=1}^{B} \beta_b g_b(x) + Z(x) \tag{5}$$

where $Z(x)$ is random process with $\mathbb{E}(Z(x)) = 0$.

3.2 Sampling techniques

We compare the performance of SSA against QS, a popular static technique due to its robust performance across dimensions as discussed by Garud et al.. Additionally, we illustrate the comparative performance of SSA against a variety of adaptive techniques like CV-Vor [38], LOLA-Vor [9], SFCVT [29], MIPT [10], and MSE [22]. CV-Vor, SFCVT, and MSE are surrogate (kriging)-dependent optimization-based adaptive techniques while LOLA-Vor is a surrogate-independent score-based adaptive technique. MIPT is a sequential space-filling technique (also known as adaptive exploratory technique). Since the surrogate construction paradigms are limited by the computational time budget, we use a fixed number of true function evaluations (K_{max}) as the basis for our comparison. This number differs with the system under consideration and dimensions of the system as explained next.

3.3 Test functions

The performance of any sampling approach strongly depends on the hyper-volume of $\mathcal{D}$ $\left(V_N(\mathcal{D}) = \prod_{n=1}^{N} d_n \right)$ which is characterized by the dimensions (N) and the edge lengths $\left(d = \{ d_n = x_n^U - x_n^L \,|\, n = 1, 2, \cdots, N \} \right)$. Thus, the robustness of a sampling technique can be determined by evaluating its performance across the ranges of dimensions and edge lengths [18]. This is achieved by formulating a diverse test bed of analytical functions with various domain sizes ($2 \leq d_n \leq 1000$), input dimensions ($2 \leq N \leq 20$), and a variety of function characteristics. In our evaluation procedure, we consider a test bed with thirteen test functions (TF1-TF13) from the literature. It is employed to evaluate the comparative performance of SSA against QS. Furthermore, we use them to study the robustness of SSA compared QS. Table 2 lists these test functions (TF1-TF13), their sources, their domain bounds, and the number of input dimensions for each of them.

Additionally, we use two test cases from the article by Xu et al. *viz.* Peaks function (TF14) and Ackley function (TF15) for illustrating the comparative performance of SSA against the existing adaptive techniques. Table 3 lists these functions along with their input dimensions and their bounds.

3.4 Performance metrics

Typically, the performance of a sampling technique in a surrogate construction paradigm is measured by the quality of the constructed approximation. We quantify this quality

Table 2: *Test functions, their dimensions, and their domain sizes for the numerical evaluation of SSA against QS.*

Legend	N	**Test Function**	**Domain Bound**
TF1 [2]	2	$0.25x_1^4 - 0.50x_1^2 + 0.10x_1 + 0.50x_2^2$	$-10 \leq x \leq 10$
TF2 [15]	2	$(\cos(x_1))^2 + (\sin(x_2))^2$	$-5 \leq x \leq 5$
TF3 [12]	2	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}\cos(x_1)) + 10$	$-5 \leq x_1 \leq 10$ $0 \leq x_2 \leq 15$
TF4 [12]	3	$\sum_{n=1}^{N-1}(100(x_{n+1} - x_n^2)^2 + (x_n - 1)^2)$	$-30 \leq x \leq 30$
TF5 [16]	5	$10\sin(\pi x_1 x_2) + 20(x_3 - 0.50)^2 + 10x_4 + 5x_5$	$-1.5 \leq x \leq 1.5$
TF6 [34]	6	$\sum_{n=1}^{N}(x_n \sin(\sqrt{\lvert x_n \rvert}))$	$-500 \leq x \leq 500$
TF7 [34]	8		
TF8 [21]	10	$0.10\sum_{n=1}^{N}\cos(5\pi x_n) - \sum_{n=1}^{N}x_n^2$	$-1 \leq x \leq 1$
TF9 [21]	12		
TF10 [21]	14		
TF11 [37]	15	$\sum_{n=1}^{N}(x_n^2 - 10\cos(2\pi x_n) + 10)$	$-5.12 \leq x \leq 5.12$
TF12 [37]	17		
TF13 [37]	20		

Table 3: *Test functions, their dimensions, and their domain sizes for the numerical evaluation of SSA against existing adaptive techniques.*

Legend	N	**Test Function**	**Domain Bound**
TF14 [38]	2	$3(1 - x_1)^2\exp(-x_1^2 - (x_2 + 1)^2)$ $-10(\frac{x_1}{5} - x_1^3 - x_2^5)\exp(-x_1^2 - x_2^2)$ $-\frac{1}{3}\exp(-(x_1 + 1)^2 - x_2^2)$	$-5 \leq x \leq 5$
TF15 [38]	10	$-20\exp\left(-0.20\sqrt{\frac{1}{10}\sum_{n=1}^{10}x_n^2}\right)$ $-\exp\left(\frac{1}{10}\sum_{n=1}^{10}\cos(2\pi x_n)\right)$ $+20 + \exp(1)$	$-0.60 \leq x \leq 0.60$

using three error-based performance metrics *viz.* Average Absolute Error (AAE), Root Mean Squared Error (RMSE), and Pooled Error (PE) [17]. AAE depicts the overall magnitude of the error while RMSE captures the sense of its distribution. PE combines these two metrics to provide a single measure of performance. We use randomly generated test set $\mathcal{Q} = \{(x^{(q)}, y^{(q)}) \,|\, q = 1, 2, \cdots, Q\}$ of size Q to compute these metrics for a surrogate S. Typically, Q is a user-defined parameter and often, multiple test sets are generated for the evaluation. To this end, we define the error metrics as follows.

$$\text{AAE} = \frac{\sum_{q=1}^{Q}\lvert y^{(q)} - S(x^{(q)})\rvert}{Q} \tag{6}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{q=1}^{Q}\left(y^{(q)} - S(x^{(q)})\right)^2}{Q}} \tag{7}$$

$$\text{PE} = \sqrt{\text{AAE} \times \text{RMSE}} \tag{8}$$

These error metrics quantify the approximation quality of S developed using a combination of a sampling technique and a surrogate form. Thus, for a given surrogate form, we compute the relative performance of a technique $t \in \mathcal{T}$, where $\mathcal{T}$ is a set of sampling techniques, using normalized metrics given in Eqs(9a)-(9c). For the comparison between SSA and QS, $\mathcal{T} = \{\text{SSA}, \text{QS}\}$. While for the comparison among the adaptive techniques, $\mathcal{T} = \{\text{SSA}, \text{CV-Vor}, \text{LOLA-Vor}, \text{SFCVT}, \text{MIPT}, \text{MSE}\}$. We follow the normalization procedure described by Garud et al. such that normalized metric value lies in $[1, \infty)$. The readers may refer to [3, 17] for further details about calculation and normalization of the metrics.

$$\overline{\text{AAE}}^{(t)} = \frac{\text{AAE}^{(t)}}{\min\limits_{t \in \mathcal{T}}(\text{AAE}^{(t)})} \tag{9a}$$

$$\overline{\text{RMSE}}^{(t)} = \frac{\text{RMSE}^{(t)}}{\min\limits_{t \in \mathcal{T}}(\text{RMSE}^{(t)})} \tag{9b}$$

$$\overline{\text{PE}}^{(t)} = \frac{\text{PE}^{(t)}}{\min\limits_{t \in \mathcal{T}}(\text{PE}^{(t)})} \tag{9c}$$

The lesser the metric value, the better the surrogate quality, hence, the better the sampling technique.

3.5 Evaluation procedure

SSA is implemented in C++ and is integrated with a computational toolkit called the “Model Development Suite” (MoDS) [31]. We perform the sampling and surrogate con-

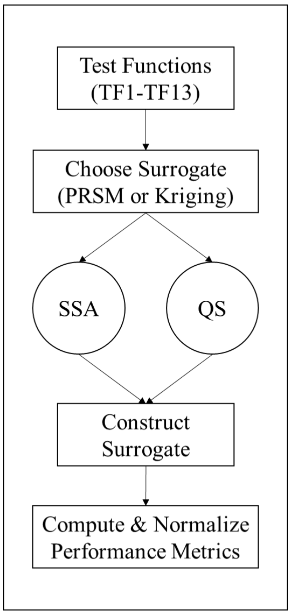


Figure 1: *Flowchart describing the numerical evaluation procedure.*

Table 4: *Experimental settings for the numerical evaluation of SSA against QS.*

Test Function	**Training Set Size** (K_{max})	**Test Set Size** (Q)	**PRSM Order** (ρ_p)	**Kriging Order** (ρ_k)
TF1	25	10, 25, 50	4	4
TF2	25	10, 25, 50	4	4
TF3	25	10, 25, 50	4	4
TF4	50	20, 50, 100	4	4
TF5	125	50, 125, 250	2	3
TF6	150	60, 150, 200	3	3
TF7	200	80, 200, 400	3	3
TF8	250	100, 250, 500	3	3
TF9	300	120, 300, 600	3	3
TF10	350	140, 350, 700	3	3
TF11	375	150, 375, 750	3	3
TF12	425	170, 425, 850	3	3
TF13	500	200, 500, 1000	3	3

struction paradigms in MoDS while the computation and normalization of performance metrics are carried out in Matlab 2012b. To this end, we follow the procedure described in Figure 1 to carry out the numerical evaluation of SSA against QS. For any test function, we choose a type of surrogate model (PRSM or kriging) and construct the surrogates using both QS and SSA-QS *i.e.* SSA initiated with Sobol sampling (Step 1 of SSA in section 2). Henceforth, we denote SSA-QS by SSA for the sake of convenience. We then compute and normalize the performance metrics. These metrics are computed for test sets of three different sizes (Q) to ascertain good performance over the entire domain. Moreover, we repeat the metric evaluation procedure three times for a given Q. The experimental settings for our comparative evaluation between SSA and QS are given in Table 4. It lists the polynomial order ρ_p for PRSM surrogate, polynomial basis functions' order ρ_k for kriging (chosen based on our experience), training and testing set sizes used for each test function.

For the comparison among the adaptive techniques, we follow the same procedure for SSA using kriging for two test cases (TF14 and TF15) while we adopt the numerical results for the other adaptive techniques from [38]. As the literature reports only the RMSE values

Table 5: *Experimental settings for the numerical comparison of SSA with adaptive techniques.*

Test Function	**Training Set Size** K_{max}	**Test Set Size** Q	**Kriging Order** (ρ_k)
Peaks	30	5000	2
	40	5000	2
Ackley	100	5000	2
	150	5000	2

computed for $Q = 5000$, we compare these techniques only based on $\overline{\text{RMSE}}$. Table 5 lists the settings for numerical comparison of SSA with the adaptive techniques.

4 Numerical results

4.1 Comparison with Sobol sampling

We now compare the performance of SSA with QS using the performance metrics (Eqs.(9a)-(9c)) discussed earlier. Tables 6 and 7 list the averaged performance metrics computed for SSA and QS using PRSM and kriging surrogates respectively. Clearly, SSA outperforms QS for all the test functions and across all the three metrics for both the surrogates. In the case of PRSM, SSA outperformed QS with the minimum $\overline{\text{PE}}$-based improvement of 9% and the average improvement of around 34% (excluding TF1 where improvement is more than 6 times). Similarly, for the case of kriging surrogate, SSA outperformed QS with the minimum $\overline{\text{PE}}$-based improvement of 6% and the average improvement of around 35% (excluding TF4 where improvement is more than 5 times). Therefore, SSA performs very well in constructing multidimensional surrogates irrespective of the surrogate type.

As discussed earlier, the robustness of a sampling technique is determined by its steady performance across the ranges of N and d. Therefore, we use the metrics from the Tables 6 and 7 to compute the percentage improvement in the performance of SSA compared to QS. This analysis is performed using three metrics, namely $\overline{\text{AAE}}$, $\overline{\text{RMSE}}$, and $\overline{\text{PE}}$ for both the surrogates. To this end, Figures 2a-4a show the improvement in the performance of SSA over N while Figures 2b-4b show this over d in the case of PRSM using $\overline{\text{AAE}}$, $\overline{\text{RMSE}}$, and $\overline{\text{PE}}$ respectively. Similarly, Figures 5a-7a and 5b-7b present the performance improvement in SSA over N and d using the three metrics in the case of kriging.

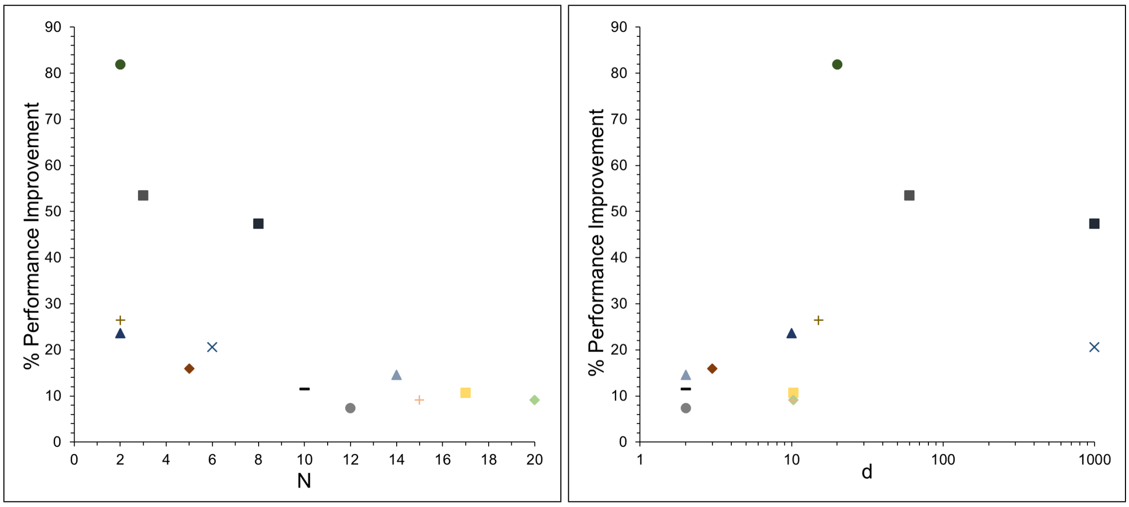
Table 6: *Comparative performance between SSA and QS using PRSM surrogate.*

Test Function	$\overline{\text{AAE}}$		$\overline{\text{RMSE}}$		$\overline{\text{PE}}$	
	QS	**SSA**	**QS**	**SSA**	**QS**	**SSA**
TF1	5.52	1.00	5.99	1.00	6.02	1.00
TF2	1.31	1.00	1.40	1.00	1.35	1.00
TF3	1.36	1.00	1.66	1.00	1.50	1.00
TF4	2.15	1.00	1.99	1.00	2.07	1.00
TF5	1.19	1.00	1.24	1.00	1.22	1.00
TF6	1.26	1.00	1.30	1.00	1.28	1.00
TF7	1.90	1.00	1.93	1.00	1.92	1.00
TF8	1.13	1.00	1.16	1.00	1.14	1.00
TF9	1.08	1.00	1.10	1.00	1.09	1.00
TF10	1.17	1.00	1.21	1.00	1.19	1.00
TF11	1.10	1.00	1.10	1.00	1.10	1.00
TF12	1.12	1.00	1.12	1.00	1.12	1.00
TF13	1.10	1.00	1.10	1.00	1.10	1.00

Table 7: *Comparative performance between SSA and QS using kriging surrogate.*

Test Function	$\overline{\text{AAE}}$		$\overline{\text{RMSE}}$		$\overline{\text{PE}}$	
	QS	SSA	QS	SSA	QS	SSA
TF1	2.13	1.00	2.43	1.00	2.37	1.00
TF2	1.25	1.00	1.30	1.00	1.28	1.00
TF3	1.51	1.00	1.81	1.00	1.65	1.00
TF4	5.55	1.00	4.97	1.00	5.25	1.00
TF5	1.17	1.00	1.24	1.00	1.20	1.00
TF6	1.16	1.00	1.22	1.00	1.19	1.00
TF7	1.73	1.00	1.81	1.00	1.77	1.00
TF8	1.10	1.00	1.11	1.00	1.10	1.00
TF9	1.09	1.00	1.10	1.00	1.10	1.00
TF10	1.25	1.00	1.27	1.00	1.26	1.00
TF11	1.11	1.00	1.12	1.00	1.12	1.00
TF12	1.12	1.00	1.12	1.00	1.12	1.00
TF13	1.07	1.00	1.06	1.00	1.06	1.00

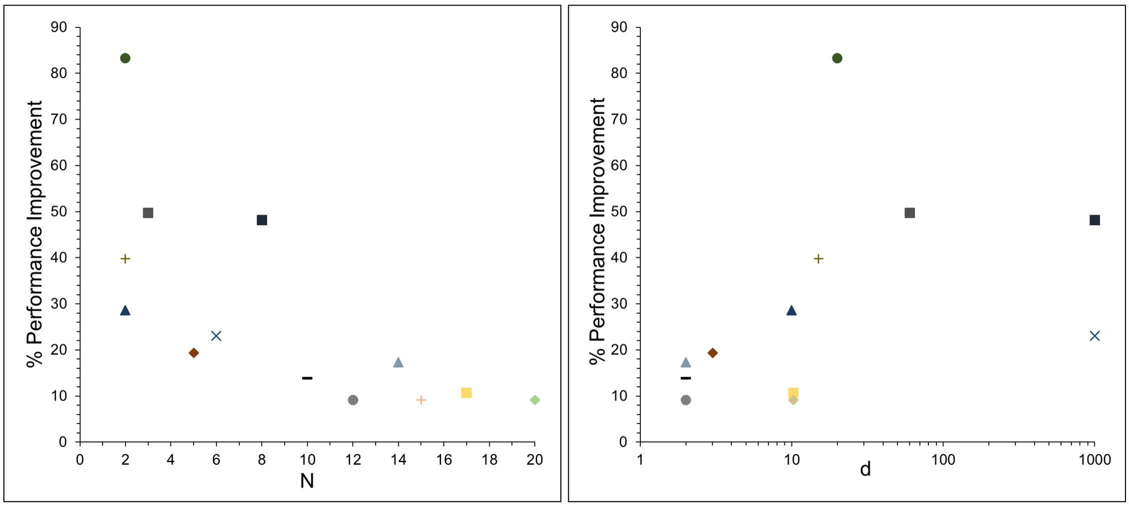
Clearly, SSA is a robust sampling approach as it shows a consistently superior performance compared to QS for the various combinations of N and d irrespective of the surrogate model types. This analysis presents two key findings. First, it shows a better relative performance of SSA for the larger d. This is in agreement with our intuition and can be understood with the following argument. In SSA, the placement is driven by the quality of the surrogate approximation while in QS it is driven by the degree of space-filling. Moreover, the hyper-volume of the domain increases tremendously with increasing d, thus re-



(a) *% performance improvement vs. N.*

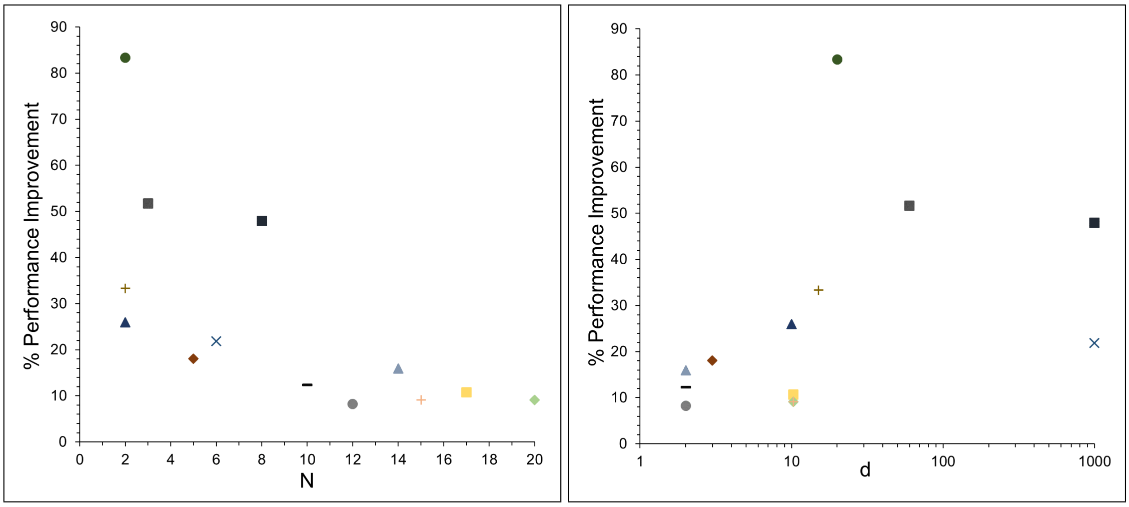
(b) *% performance improvement vs. d.*

Figure 2: *Percentage improvement in the $\overline{\text{AAE}}$-based performance of SSA compared to QS for PRSM surrogate. (Note that Figure 2b uses log scale for the horizontal axis.)*



(a) *% performance improvement vs. N.* **(b)** *% performance improvement vs. d.*

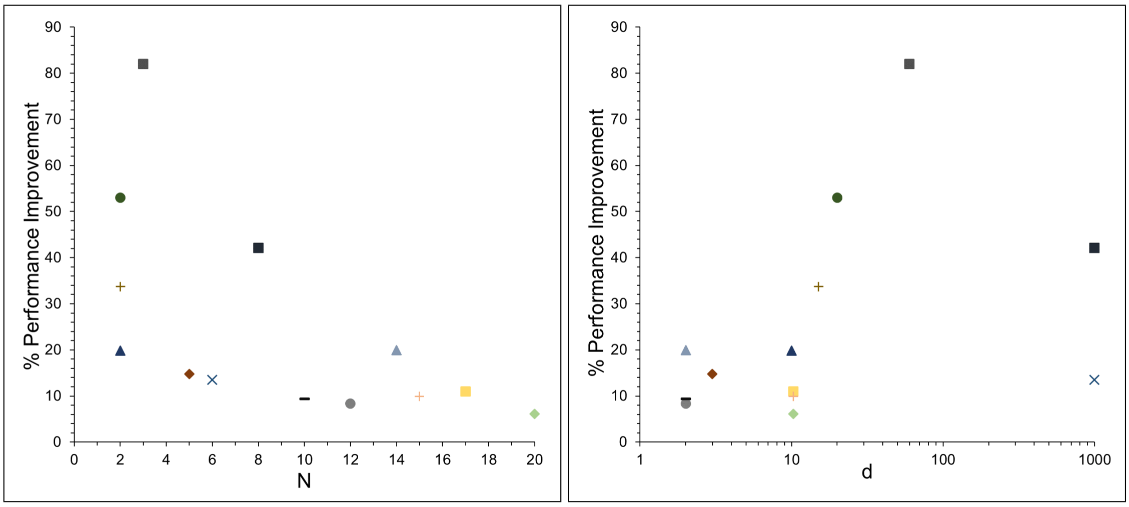
Figure 3: *Percentage improvement in the* $\overline{\mathrm{RMSE}}$*-based performance of SSA compared to QS for PRSM surrogate. (Note that Figure 3b uses log scale for the horizontal axis.)*



(a) *% performance improvement vs. N.* **(b)** *% performance improvement vs. d.*

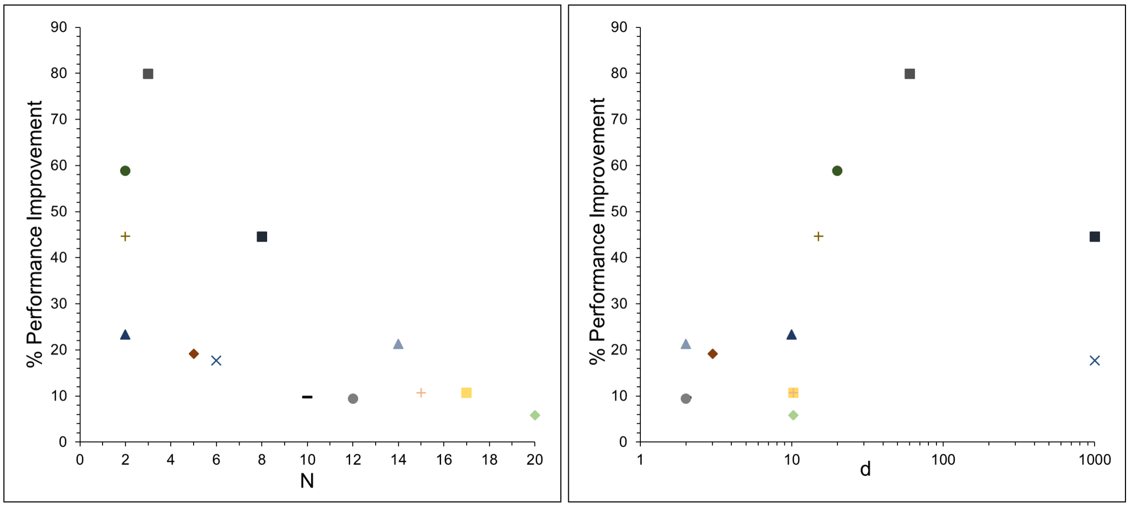
Figure 4: *Percentage improvement in the* $\overline{\mathrm{PE}}$*-based performance of SSA compared to QS for PRSM surrogate. (Note that Figure 4b uses log scale for the horizontal axis.)*

quiring many more samples to achieve the same degree of space-filling. This is why SSA performs much better than QS for the larger d. On the other hand, the relative improvement in the performance of SSA drops for the larger N and to understand this consider the following argument. The approximation ability of a surrogate modeling technique can typically decrease with increasing dimensions. This in turn can affect the performance of SSA since its sample placement strategy is driven by the surrogate approximation. Therefore, a good choice of surrogate modeling technique assures a better relative performance



(a) % *performance improvement vs. N.* **(b)** % *performance improvement vs. d.*

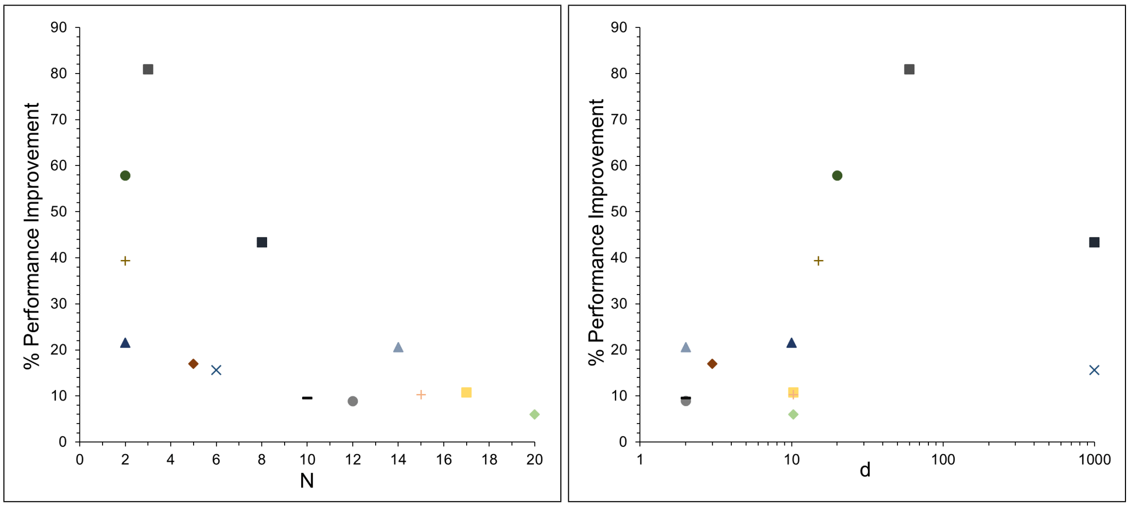
Figure 5: *Percentage improvement in the* $\overline{\text{AAE}}$*-based performance of SSA compared to QS for kriging surrogate. (Note that Figure 5b uses log scale for the horizontal axis.)*



(a) % *performance improvement vs. N.* **(b)** % *performance improvement vs. d.*

Figure 6: *Percentage improvement in the* $\overline{\text{RMSE}}$*-based performance of SSA compared to QS for kriging surrogate. (Note that Figure 6b uses log scale for the horizontal axis.)*

of SSA. Nevertheless, for a given technique, SSA always outperforms QS promising a better approximation at a reduced computational expense.



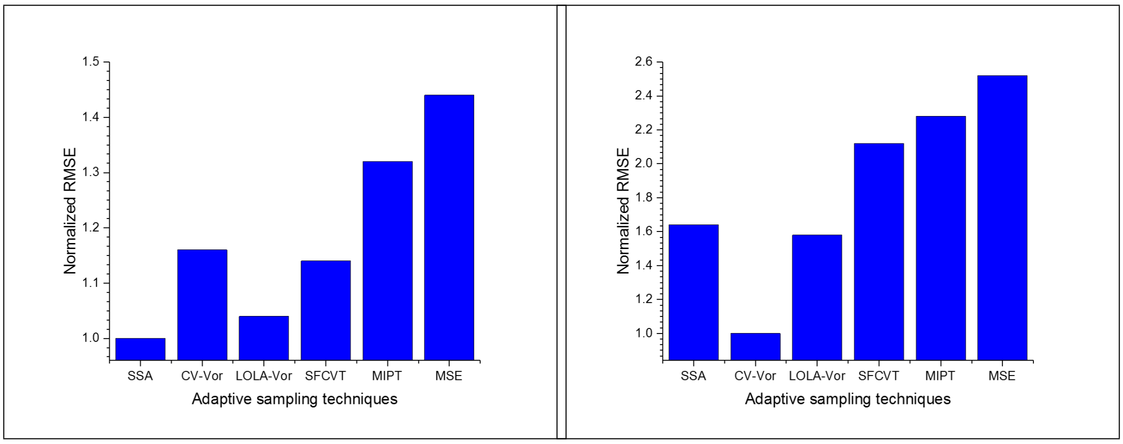
(a) % *performance improvement vs.* N.

(b) % *performance improvement vs. d.*

Figure 7: *Percentage improvement in the* $\overline{\mathrm{PE}}$*-based performance of SSA compared to QS for kriging surrogate. (Note that Figure 7b uses log scale for the horizontal axis.)*

4.2 Comparison with adaptive sampling techniques

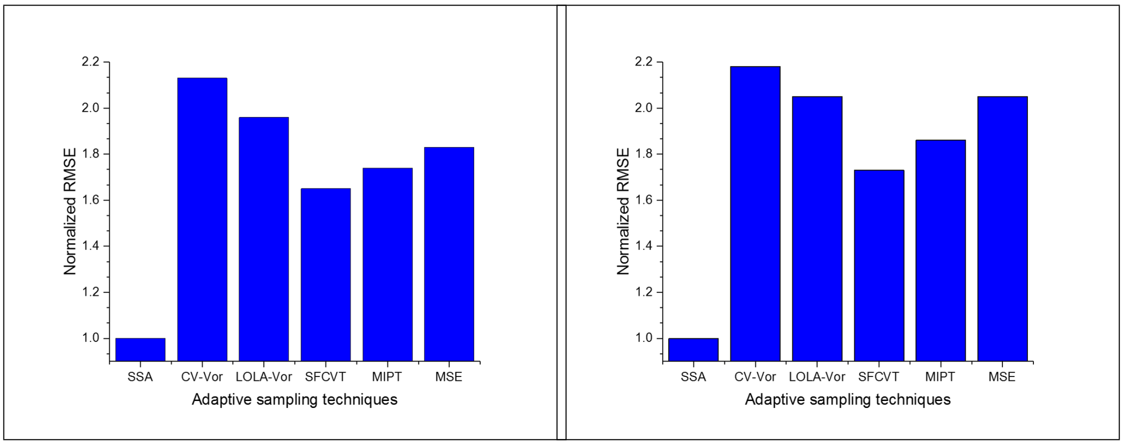
We now illustrate the comparative performance of SSA with the adaptive techniques for two cases *viz.* Peaks (TF14) and Ackley (TF15) functions. The former is a two dimensional function with nonlinearity concentrated in the central region of the domain while the latter is a ten dimensional function with high nonlinearity over the entire domain. We use the experimental settings listed in Table 5 for evaluating SSA and computing RMSE metrics while RMSE values for the other techniques are taken from the literature [38]. We then normalize the RMSE values using Eq. (9b). Figures 8 and 9 show the comparative



(a) $K_{max} = 30$

(b) $K_{max} = 40$

Figure 8: *Numerical comparison of various adaptive techniques using* $\overline{\mathrm{RMSE}}$ *for Peaks function (TF14).*



(a) $K_{max} = 100$ **(b)** $K_{max} = 150$

Figure 9: *Numerical comparison of various adaptive techniques using* $\overline{\mathrm{RMSE}}$ *for Ackley function (TF15).*

performance of SSA with various adaptive techniques for Peaks and Ackley functions respectively. SSA performs the best for $K_{max} = 30$ for Peaks function. However, CV-Vor becomes the best performer for $K_{max} = 40$ followed by LOLA-Vor and SSA performing equally while the rest of the techniques show relatively poor performance with MSE being the worst performer. On the other hand, for Ackley function, SSA performs the best for both $K_{max} = \{100, 150\}$ followed by SFCVT, MIPT, MSE, LOLA-Vor, and CV-Vor. Overall, SSA shows a very good performance compared to the other techniques, especially for a high dimensional case.

5 Case studies

Besides the numerical comparison using analytical functions, the ultimate test of a technique is its practical applicability to real-life case studies. Thus, we employ SSA to the following three cases from the literature of chemical and process systems engineering: (i) Biodiesel production process, (ii) Multi-component distillation column, and (iii) Carbon capture unit. We follow the same evaluation procedure as described earlier in Figure 1 and compare its performance with QS using PRSM surrogate.

5.1 Biodiesel production process

Biodiesel is industrially produced via transesterification of vegetable oils with alcohol (typically methanol or ethanol) in presence of base catalyst. In transesterification, triglycerides of fatty acids react with alcoxide (resulting from alcohol and base) to form methyl esters (biodiesel) and glycerol. Following equation only shows a general transesterification reaction, however, its detailed mechanism and kinetics can be found in a recent article

Table 8: *Experimental settings for approximating the exchangers in the biodiesel plant simulation.*

Output varibale	**Input variable**	**Description of Input Variable**	**Default Value**	**Lower bound**	**Upper bound**
y_1 (Duty of 10E01)	x_1	Molar flow rate of OIL	30 Kmol/h	27 Kmol/h	33 Kmol/h
	x_2	Temperature of OIL	303.15 K	300.15 K	306.15 K
	x_3	Temperature of 10E01	343.15 K	336.15 K	350.15 K
y_2 (Duty of 10E02)	x_1	Molar flow rate of OIL	30 Kmol/h	27 Kmol/h	33 Kmol/h
	x_2	Temperature of OIL	303.15 K	300.15 K	306.15 K
	x_3	Temperature of 10D01	333.15 K	327.15 K	339.15 K
	x_4	Volume of 10D01	45 m^3	40.5 m^3	49.5 m^3
	x_5	Temperature of 10E01	303.15 K	300.15 K	306.15 K
	x_6	Molar flow rate of MEOH	180 Kmol/h	162 Kmol/h	198 Kmol/h
	x_7	Temperature of MEOH	303.15 K	300.15 K	306.15 K
	x_8	Temperature of 10E02	363.15 K	354.15 K	372.15 K

Output varibale	**Input variable**	**Description of Input Variable**	**Default Value**	**Lower bound**	**Upper bound**
y_3 (Duty of 10E03)	x_1	Molar flow rate of OIL stream	30 Kmol/h	27 Kmol/h	33 Kmol/h
	x_2	Temperature of OIL	303.15 K	300.15 K	306.15 K
	x_3	Temperature of 10D01	333.15 K	327.15 K	339.15 K
	x_4	Volume of 10D01	45 m^3	40.5 m^3	49.5 m^3
	x_5	Temperature of 10D02F	363.15 K	354.15 K	372.15 K
	x_6	Temperature of 10E01	303.15 K	300.15 K	306.15 K
	x_7	Molar flow rate of MEOH	180 Kmol/h	162 Kmol/h	198 Kmol/h
	x_8	Temperature of MEOH	303.15 K	300.15 K	306.15 K
	x_9	Temperature of 10D02D	303.15 K	300.15 K	306.15 K
	x_{10}	Temperature of 10E02	363.15 K	354.15 K	372.15 K
	x_{11}	Temperature of 10E03	343.15 K	336.15 K	350.15 K

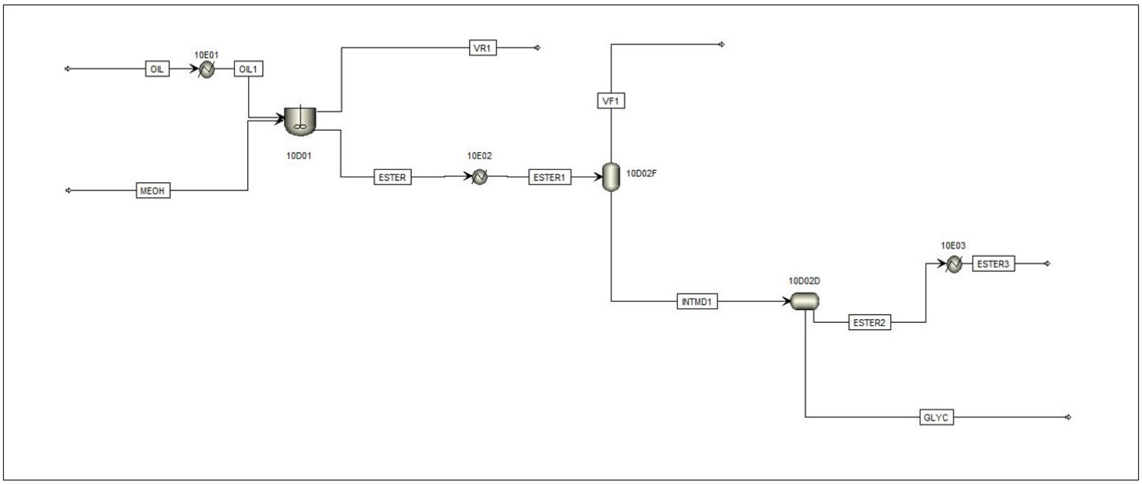


Figure 10: *Aspen Plus flowsheet of biodiesel production plant.*

by Likozar and Levec.

$$\text{Triglyceride} + 3\,\text{Methanol} \longrightarrow 3\,\text{MethylEster} + \text{Glycerol}$$

Here, we simulate the biodiesel production process designed by Lurgi GmbH using Aspen Plus v8.6 [36]. Palm oil (simulated using tripalmitin) is used as a feedstock for biodiesel production. It is preheated to 343.15 K and is fed to the reactor with a mixture of methanol and sodium hydroxide. Transesterification is carried out in a continuously stirred tank reactor (CSTR) at 1 bar with high mixing intensity. This ascertains the homogeneity in the reactor, thus removing the mass transfer limitations. The product mixture from the reactor is heated in 10E02 and passed to the flash drum (10D02F) to remove excess methanol. Then the liquid stream from the flash drum is fed to the decanter 10D02D to separate the side product *i.e.* glycerol. Finally, crude biodiesel from the decanter is sent downstream for further processing which is not considered in this case study.

To this end, we employ SSA and QS to approximate the duties of the three exchangers from this flowsheet simulation (Figure 10) using PRSM surrogates. Table 8 shows the responses (duties of three exchangers *viz.* 10E01, 10E02, and 10E03), their respective inputs, the default values of the input variables, and their bounds. Overall, we construct six surrogates based on two surrogate model-sampling technique combinations and three responses. We use $K_{max} = \{25, 200, 275\}$ for approximating the duties of 10E01, 10E02, and 10E03 respectively, and $Q = \{10, 50, 50\}$ for evaluating the performances of these ap-

Table 9: *Performance of SSA against QS for approximating the exchanger duties in the biodiesel production process.*

Legend	$\overline{\text{AAE}}$		$\overline{\text{RMSE}}$		$\overline{\text{PE}}$	
	QS	**SSA**	**QS**	**SSA**	**QS**	**SSA**
10E01	1.02	1.00	1.02	1.00	1.02	1.00
10E02	1.12	1.00	1.14	1.00	1.13	1.00
10E03	1.08	1.00	1.11	1.00	1.09	1.00

proximations. Tables 9 shows the comparative performance and clearly, SSA outperforms QS in approximating all the three exchangers with the maximum $\overline{\text{PE}}$ based improvement of 13%. Note that a small amount of improvement is observed for 10E01 due to its linear response. However, the improvement significantly rises for the other two exchangers. This shows that SSA can be successfully employed to reduce computational expense in approximating the typical heaters or coolers in process simulations.

5.2 Multi-component distillation column

Now, we consider one of the most complex and nonlinear process units for approximation *viz.* a multicomponent distillation column. This case study is adopted from the article by Dhole and Linnhoff and its Aspen Plus flowsheet is shown in Figure 11. The feed stream (denoted as FEED in Figure 11) at 373.15 K and 2 bar consists of 5 components namely N-Heptane, N-Octane, N-Nonane, N-Decane, and N-Pentadecane. This is preheated in 10E01 and then fed to the distillation column 10D01. The column 10D01 aims to separate N-Heptane and N-Octane as distillate products and the rest as still products. The column operates at 2 bar and has 30 stages. We simulate the column for the following two design specifications: (i) 99% recovery of Octane in the distillate stream by adjusting the reflux ratio and (ii) 98.5% recovery of Nonane in the still stream by adjusting boil-up ratio.

To this end, we wish to approximate the reboiler duty, condenser duty, reflux ratio, and boil-up ratio of the column as a function of component flow rates of the feed, temperature of the feed, and the preheater duty. Table 11 lists the input-output variables considered for the approximation, their defaults values, and their lower and upper bounds. Overall, we construct eight surrogates based on two surrogate modeling-sampling technique combinations for four responses. We use $K_{max} = 175$ for constructing the surrogates and three randomly generated test sets of size $Q = 50$. Tables 10 presents the comparative performance of SSA against QS. Clearly, SSA performs very well compared to QS with respect to all the three metrics. Overall, the maximum $\overline{\text{PE}}$-based improvement of 52% is achieved with SSA against QS across all the four responses.

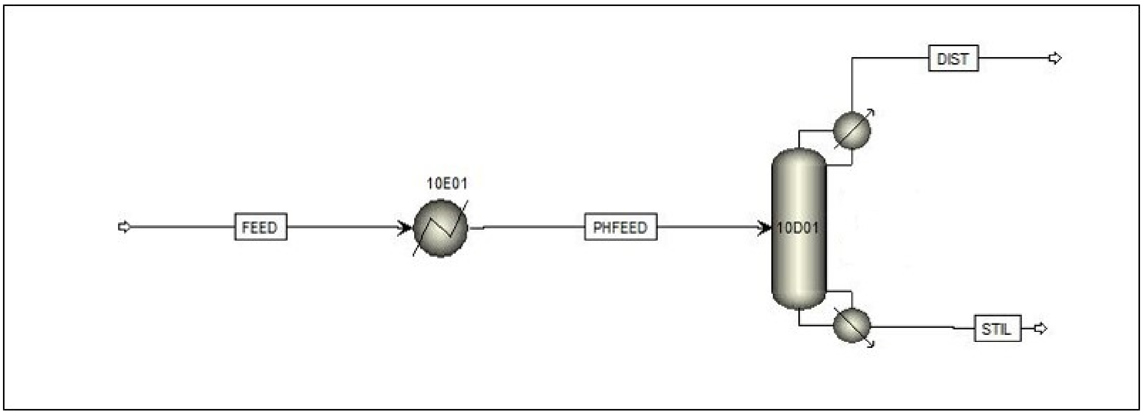


Figure 11: *Aspen Plus flowsheet of multi-component distillation column.*

Table 11: *Experimental settings for approximating the multi-component distillation column simulation.*

Output Variable	Input Variable	Description of Input Variable	Default Value	Lower Bound	Upper Bound
y_1 Heat duty of reboiler in 10D01 y_2 Heat duty of condenser in 10D01 y_3 Reflux ratio in 10D01 y_4 Boil up ratio 10D01	x_1	Temperature of FEED	373.15 K	363.15 K	383.15 K
	x_2	Molar flow rate of heptane in FEED	0.05 Kmol/s	0.04 Kmol/s	0.06 Kmol/s
	x_3	Molar flow rate of octane in FEED	0.05 Kmol/s	0.04 Kmol/s	0.06 Kmol/s
	x_4	Molar flow rate of nonane in FEED	0.05 Kmol/s	0.04 Kmol/s	0.06 Kmol/s
	x_5	Molar flow rate of decane in FEED	0.05 Kmol/s	0.04 Kmol/s	0.06 Kmol/s
	x_6	Molar flow rate of pentadecane in FEED	0.05 Kmol/s	0.04 Kmol/s	0.06 Kmol/s
	x_7	Heat duty of 10E01	25 MMBtu/h	20 MMBtu/h	30 MMBtu/h

Table 10: *Performance of SSA against QS using PRSM for approximating the multicomponent distillation column.*

Response	$\overline{\text{AAE}}$		$\overline{\text{RMSE}}$		$\overline{\text{PE}}$	
	QS	**SSA**	**QS**	**SSA**	**QS**	**SSA**
Condenser Duty	1.56	1.00	1.48	1.00	1.52	1.00
Reboiler Duty	1.57	1.00	1.47	1.00	1.51	1.00
Reflux Ratio	1.57	1.00	1.47	1.00	1.52	1.00
Boil-up Ratio	1.56	1.00	1.48	1.00	1.52	1.00

5.3 Carbon capture unit

Herein, we simulate and approximate a typical CO_2 capture unit (CCU) in natural gas processing industry. It removes CO_2 from a gaseous mixture of CH_4, C_2H_6, C_3H_8, N_2, CO_2, and H_2S using diethanolamine (DEA) in a reactive absorption column. We use this case study from Aspen Plus v8.6 user guide and is simulated using operational data from a natural gas treatment unit at Pyote, Texas [5]. Simulation of this reactive absorption column entails three critical aspects: (i) Simulation of the ionic species with Electrolyte NRTL fluid package, (ii) Incorporation of the electrolyte transport property models, and (iii) The reaction kinetics and equilibrium calculations in the absorption column. Additionally, the following system of equilibrium and kinetic reactions is modeled within the absorption column.

$$\mathrm{DEAH^+ + H_2O \rightleftharpoons DEA + H_3O^+}$$
$$\mathrm{2\,H_2O \rightleftharpoons H_3O^+ + OH^-}$$

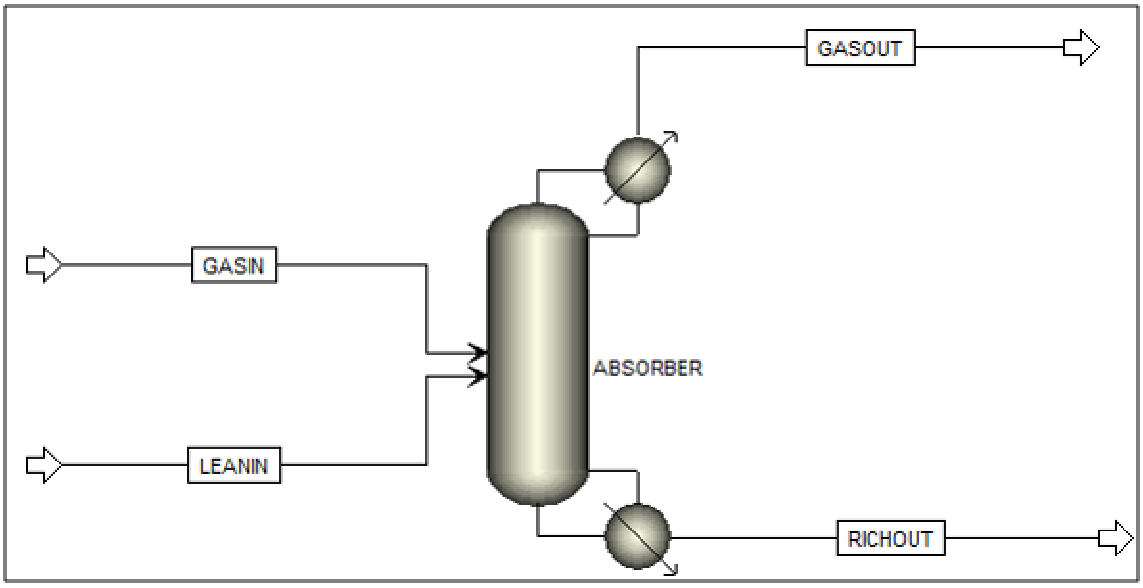


Figure 12: *Aspen Plus flowsheet of the reactive absorption column in CCU.*

$$\mathrm{HCO_3{}^- + H_2O \rightleftharpoons H_3O^+ + CO_3{}^{-2}}$$

$$\mathrm{CO_2 + OH^- \longrightarrow HCO_3{}^-}$$

$$\mathrm{HCO_3{}^- \longrightarrow CO_2 + OH^-}$$

$$\mathrm{DEA + CO_2 + H_2O \longrightarrow DEACOO^- + H_3O^+}$$

$$\mathrm{DEACOO^- + H_3O^+ \longrightarrow DEA + CO_2 + H_2O}$$

$$\mathrm{H_2O + H_2S \rightleftharpoons HS^- + H_3O^+}$$

$$\mathrm{H_2O + HS^- \rightleftharpoons S^{-2} + H_3O^+}$$

The equilibrium reactions are taken from [23, 24] while the kinetic reactions and their rate constants are taken from [33]. Figure 12 shows the Aspen Plus simulation of the absorption column where gaseous mixture (GASIN) at 295.35 K and 900 psig is fed through the bottom of the absorber and liquid DEA absorbent stream at 312.04 K and 900 psig is fed through the top. This column has 20 stages with 1.66 m as tray diameter and 0.06 m weir height for Glitch Ballast type of tray. We use “rate-based” calculation mode in Aspen Plus and the readers may refer to Aspen Plus user guide for further details. The default component flow rates of the feed streams, their lower and upper bounds are listed in Table 13. Finally, we employ SSA for developing the approximation of this simulation using $K_{max} = 225$. Tables 12 show the averaged comparative performance metrics computed for three different test sets of size $Q = 50$ and clearly, SSA outperforms QS.

Table 12: *Performance between SSA and QS for approximating CCU.*

Legend	$\overline{\mathrm{AAE}}$		$\overline{\mathrm{RMSE}}$		$\overline{\mathrm{PE}}$	
	QS	**SSA**	**QS**	**SSA**	**QS**	**SSA**
CO_2 molar flow	1.03	1.00	1.15	1.00	1.08	1.00

Based on these case studies, it is evident that SSA performs equally well in approximating real life simulations as it does for the analytical functions. Thus, it has an immense potential in reducing computational cost associated with approximating typical unit operations like heat exchangers, distillation columns, reaction systems, thermodynamic property packages, absorption columns *etc.*

Table 13: *Experimental settings for approximating the reactive absorption column simulation.*

Output Variable	**Input Variable**	**Description of Input Variable**	**Default Value**	**Lower Bound**	**Upper Bound**
y_1 Component Molar Flow of CO_2 in GASOUT	x_1	Temperature of GASIN	295.35 K	288.15 K	303.15 K
	x_2	Component Molar Flow of CO_2 in GASIN	48 Kmol/h	40 Kmol/h	60 Kmol/h
	x_3	Component Molar Flow of CH_4 in GASIN	3400 Kmol/h	3200 Kmol/h	3600 Kmol/h
	x_4	Temperature of LEANIN	312.04 K	308.15 K	318.15 K
	x_5	Component Molar Flow of H_2O in LEANIN	1740 Kmol/h	1600 Kmol/h	1900 Kmol/h
	x_6	Component Molar Flow of DEA in LEANIN	120 Kmol/h	90 Kmol/h	150 Kmol/h
	x_7	Component Molar Flow of CO_2 in LEANIN	1.23 Kmol/h	0.5 Kmol/h	2 Kmol/h

6 Conclusions

In this article, we extensively evaluated a novel adaptive sampling approach, namely smart sampling [17] for constructing multidimensional surrogate approximations. We draw following conclusions from our numerical investigation.

1. SSA shows an excellent performance compared to QS for approximating a variety of test functions using polynomial and kriging surrogates.
2. It performs more robustly compared to QS over ranges of domain dimensions and edge length/s for both the surrogates highlighting its viability for a broad spectrum of applications.
3. SSA also performs better than the existing adaptive sampling approaches like CV-Vor, LOLA-Vor, SFCVT, MIPT, and MSE, especially for the high dimensional case.
4. Finally, SSA is successfully applied to three practical case studies *viz.* biodiesel production process, multi-component distillation column, and reactive absorption column in CO_2 capture unit and it shows an excellent performance for approximating commonly encountered processes and units such as exchangers, distillation column, absorption column, a system of kinetic reactions *etc.*

Overall, SSA is a generic, robust, optimal, and surrogate-independent adaptive sampling approach that has an immense potential to reduce computational expenses associated with surrogate construction paradigms.

Acknowledgement

This publication is made possible by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

Nomenclature

Abbreviations

AAE: average absolute error
ANN: artificial neural network
CC: clustering constraint
CCU: carbon capture unit
CDM: crowding distance metric
CSTR: continuously stirred tank reactor
CV: cross validation
CVE: cross validation error
DEA: diethanolamine
DF: departure function
DoE: design of experiments
DT: Delaunay triangulation
EE: expected error
HDMR: high dimensional model representation
JK: Jackknifing
LOLA: local linear approximation
MD: Mahalanobis distance
ME: maximum entropy
Mm: maximin distance
MoDS: model development suite
MSD: maximum scaled distance
MSE: maximum sampling error
NLP: nonlinear programming problem
NN: nearest neighbor
PE: pooled error
PRSM: polynomial response surface model
QS: Sobol sampling
RBF: radial basis function
RMSE: root mean squared error
SSA: smart sampling algorithm
SVR: support vector regression
VT: Voronoi tessellation

Notation

Subscripts

b: index for the basis functions in kriging
m: index for elements of response/output variables' vector
n: index for elements of design/input variables' vector

Superscripts

i: index for elements of set
j: index for elements of set
k: index for elements of set
t: index for elements in set of sampling techniques
L: lower bound
U: upper bound

Parameters

K: size of initial sample set
K_{max}: maximum number of sample points
N: total number of input domain dimensions
M: total number of output domain dimensions

Continuous Variables

x: vector of input/design variables
y: vector of output/response variables

Symbols

d_n: edge length of nth dimension of $\mathcal{D}$
d: vector of edge lengths of $\mathcal{D}$
$\mathcal{D}$: domain
Δ: departure function
ε: minimum allowed distance between two points
$\mathbb{E}$: expectation
f: computationally costly function
g_b: basis function in kriging
$\mathbb{N}$: set of natural numbers
ρ_k: kriging order
ρ_p: PRSM order
Q: test set size
$\mathcal{Q}$: test set

$\mathbb{R}$: set of real numbers
S: surrogate model form
$\mathcal{T}$: set of sampling techniques
$V_N(\mathcal{D})$: hyper-volume of $\mathcal{D}$
$\mathcal{X}_N^{(K)}$: N dimensional sample set of size K
$\mathcal{Y}_M^{(K)}$: M dimensional response set of size K
Z: random process

References

[1] A. Ajdari and H. Mahlooji. An adaptive exploration-exploitation algorithm for constructing metamodels in random simulation using a novel sequential experimental design. *Communications in Statistics-Simulation and Computation*, 43(5):947–968, 2014. doi:10.1080/03610918.2012.720743.

[2] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *Journal of Optimization Theory and Applications*, 47(1):1–16, 1985. doi:10.1007/BF00941312.

[3] S. Bhushan and I. A. Karimi. Heuristic algorithms for scheduling an automated wet-etch station. *Computers & Chemical Engineering*, 28(3):363–379, 2004. doi:10.1016/S0098-1354(03)00192-3.

[4] D. Busby, C. L. Farmer, and A. Iske. Hierarchical nonlinear approximation for experimental design and statistical data fitting. *SIAM Journal on Scientific Computing*, 29(1):49–69, 2007. doi:10.1137/050639983.

[5] K. F. Butwell and C. R. Perry. Performance of gas purification systems utilizing dea solutions. In *Laurance Reid Gas Conditioning Conference*, 1975.

[6] J. A. Caballero and I. E. Grossmann. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal*, 54(10):2633–2650, 2008. doi:10.1002/aic.11579.

[7] A. Cozad, N. V. Sahinidis, and D. C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014. doi:10.1002/aic.14418.

[8] K. Crombecq, L. De Tommasi, D. Gorissen, and T. Dhaene. A novel sequential design strategy for global surrogate modeling. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 731–742. IEEE, 2009.

[9] K. Crombecq, D. Gorissen, D. Deschrijver, and T. Dhaene. A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. *SIAM Journal on Scientific Computing*, 33(4):1948–1974, 2011. doi:10.1137/090761811.

[10] K. Crombecq, E. Laermans, and T. Dhaene. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research*, 214(3):683–696, 2011. doi:10.1016/j.ejor.2011.05.032.

[11] V. Dhole and B. Linnhoff. Distillation column targets. *Computers & Chemical Engineering*, 17(5-6):549–560, 1993. doi:10.1016/0098-1354(93)80043-M.

[12] L. Dixon and G. Szegö. The global optimization problem: an introduction. *Towards Global Optimization*, 2:1–15, 1978.

[13] J. Eason and S. Cremaschi. Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers & Chemical Engineering*, 68: 220–232, 2014. doi:10.1016/j.compchemeng.2014.05.021.

[14] B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, 1982. doi:10.1137/1.9781611970319.

[15] F. P. Fernandes, M. F. Costa, and E. M. Fernandes. Assessment of a hybrid approach for nonconvex constrained minlp problems. *CMMSE 2011*, pages 484–495, 2011.

[16] J. H. Friedman, E. Grosse, and W. Stuetzle. Multidimensional additive spline approximation. *SIAM Journal on Scientific and Statistical Computing*, 4(2):291–301, 1983. doi:10.1137/0904023.

[17] S. S. Garud, I. Karimi, and M. Kraft. Smart sampling algorithm for surrogate model development. *Computers & Chemical Engineering*, 96:103–114, 2017. doi:10.1016/j.compchemeng.2016.10.006.

[18] S. S. Garud, I. A. Karimi, and M. Kraft. Design of computer experiments: A review. *Computers & Chemical Engineering*, 106:71 – 95, 2017. ISSN 0098-1354. doi:10.1016/j.compchemeng.2017.05.010.

[19] C. A. Henao and C. T. Maravelias. Surrogate-based process synthesis. *Computer Aided Chemical Engineering*, 28:1129–1134, 2010. doi:10.1016/S1570-7946(10)28189-0.

[20] C. A. Henao and C. T. Maravelias. Surrogate-based superstructure optimization framework. *AIChE Journal*, 57(5):1216–1232, 2011. doi:10.1002/aic.12341.

[21] M. Jamil and X.-S. Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013. doi:10.1504/IJMMNO.2013.055204.

[22] R. Jin, W. Chen, and A. Sudjianto. On sequential sampling for global metamodeling in engineering design. In *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 539–548. American Society of Mechanical Engineers, 2002. doi:10.1115/DETC2002/DAC-34092.

[23] F. Y. Jou, A. E. Mather, and F. D. Otto. Solubility of hydrogen sulfide and carbon dioxide in aqueous methyldiethanolamine solutions. *Industrial & Engineering Chemistry Process Design and Development*, 21(4):539–544, 1982. doi:10.1021/i200019a001.

[24] F. Y. Jou, J. J. Carroll, A. E. Mather, and F. D. Otto. Solubility of mixtures of hydrogen sulfide and carbon dioxide in aqueous n-methyldiethanolamine solutions. *Journal of Chemical and Engineering Data*, 38(1):75–77, 1993. doi:10.1021/je00009a018.

[25] J. P. Kleijnen. *Design and analysis of simulation experiments*, volume 20. Springer, 2008. doi:10.1007/978-0-387-71813-2.

[26] J. Koehler and A. Owen. 9 computer experiments. *Handbook of Statistics*, 13: 261–308, 1996. doi:10.1016/S0169-7161(96)13011-X.

[27] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8.

[28] M. Kraft and S. Mosbach. The future of computational modelling in reaction engineering. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1924):3633–3644, 2010. doi:10.1098/rsta.2010.0124.

[29] G. Li, V. Aute, and S. Azarm. An accumulative error based adaptive design of experiments for offline metamodeling. *Structural and Multidisciplinary Optimization*, 40(1-6):137–155, 2010. doi:10.1007/s00158-009-0395-z.

[30] B. Likozar and J. Levec. Effect of process conditions on equilibrium, reaction kinetics and mass transfer for triglyceride transesterification to biodiesel: Experimental and modeling based on fatty acid composition. *Fuel Processing Technology*, 122:30 – 41, 2014. ISSN 0378-3820. doi:http://doi.org/10.1016/j.fuproc.2014.01.017.

[31] MoDS. Description of the MoDS toolkit. `http://www.cmclinnovations.com/`, 2017. 2017-06-16.

[32] L. Pronzato and W. G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012. doi:10.1007/s11222-011-9242-3.

[33] E. B. Rinker, S. S. Ashour, and O. C. Sandall. Kinetics and modeling of carbon dioxide absorption into aqueous solutions of diethanolamine. *Industrial & Engineering Chemistry Research*, 35(4):1107–1114, 1996. doi:10.1021/ie950336v.

[34] H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[35] S. Shan and G. G. Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, 2010. doi:10.1007/s00158-009-0420-2.

[36] J. J. Sikorski, G. Brownbridge, S. S. Garud, S. Mosbach, I. A. Karimi, and M. Kraft. Parameterisation of a biodiesel plant process flow sheet model. *Computers & Chemical Engineering*, 95:108–122, 2016. doi:10.1016/j.compchemeng.2016.06.019.

[37] A. Torn and A. Zilinskas. *Global Optimization*. Springer-Verlag New York, Inc., New York, NY, USA, 1989. ISBN 0-387-50871-6.

[38] S. Xu, H. Liu, X. Wang, and X. Jiang. A robust error-pursuing sequential sampling approach for global metamodeling based on Voronoi diagram and cross validation. *Journal of Mechanical Design*, 136(7):071009, 2014. doi:10.1115/1.4027161.