Computers and Chemical Engineering 96 (2017) 103–114

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Smart Sampling Algorithm for Surrogate Model Development

Sushant Suhas Garud[a,b], I.A. Karimi[a,b,*], Markus Kraft[a,c,d]

[a] Cambridge Centre for Carbon Reduction in Chemical Technology (C4T), 05-05 CARES, CREATE Tower, 1 CREATE Way, 138602, Singapore
[b] Department of Chemical and Biomolecular Engineering, 4 Engineering Drive 4, National Univeristy of Singapore, 117585, Singapore
[c] Department of Chemical Engineering and Biotechnology, New Museums Site, Pembroke Street, Cambridge, CB2 3RA, United Kingdom
[d] School of Chemical and Biomedical Engineering, Nanyang Technological University, 62 Nanyang Drive, 637459, Singapore

ARTICLE INFO

Article history:
Received 4 May 2016
Received in revised form 4 September 2016
Accepted 12 October 2016
Available online 19 October 2016

Keywords:
Smart sampling
Adaptive surrogate modelling
Point placement

ABSTRACT

Surrogate modelling aims to reduce computational costs by avoiding the solution of rigorous models for complex physicochemical systems. However, it requires extensive sampling to attain acceptable accuracy over the entire domain. The well-known space-filling techniques use sampling based on uniform, quasi-random, or stochastic distributions, and are typically non-adaptive. We present a novel technique to select sample points systematically in an adaptive and optimized manner, assuring that the points are placed in regions of complex behaviour and poor representation. Our proposed smart sampling algorithm (SSA) solves a series of surrogate-based nonlinear programming problems for point placement to enhance the overall accuracy and reduce computational burden. Our extensive numerical evaluations using 1-variable test problems suggest that our SSA performs the best, when its initial sample points are generated using uniform sampling. For now, this conclusion is valid for 1-variable functions only, and we are testing our algorithm for n-variable functions.

1. Introduction

Naturally, we grasp the understanding of a complex phenomenon by opting for a simpler format. Similarly, we use process simulators to model, study, and analyze complex and nonlinear physicochemical processes. However, such simulations can be compute-intensive, and running them repeatedly in an optimization/analysis procedure can be computationally prohibitive. Furthermore, numerical models can pose significant hurdles within a continuous optimization algorithm. Therefore, it helps to convert a high-fidelity simulation model into a computationally inexpensive surrogate model that captures its essential features with prescribed numerical accuracy.

Surrogate modelling, also known as metamodeling, is a technique to generate a mathematical or numerical representation of a complex system based on some sampled input-output data. Many surrogate modelling techniques have been developed over the past few decades such as Polynomial Surface Response Models (PRSM) (Forrester and Keane, 2009; Myers and Montgomery, 2002; Queipo et al., 2005), Kriging (Cressie, 1990; Forrester et al., 2008a; Martin and Simpson, 2005; Sakata et al., 2003; Simpson, 1998), Radial Basis Functions (RBF) (Hardy, 1971; Hussain et al., 2002), Support Vector Regression (SVR) (Clarke et al., 2005), and Artificial Neural Networks (ANNs) (Yegnanarayana, 2004). The literature (Forrester and Keane, 2009; Henao and Maravelias, 2010, 2011; Queipo et al., 2005; Shan and Wang, 2010; Wang and Shan, 2007) has compared them and discussed their applications to various systems.

Irrespective of the technique, building a surrogate model requires sample points. The process of generating such points is known as sampling. We can classify the existing sampling methods into two broad categories: non-adaptive and adaptive. The four types of non-adaptive methods are grid-based, pattern/geometry-based, stochastic, and quasi-random. The grid-based method simply distributes sample points to form a uniform grid (Cartesian grid). The second type employs statistics-driven methods such as the design of experiments to fill space. These include full/half factorial designs (Fisher, 1935), central composite (CC) designs (Box and Wilson, 1951), Box-Behnken (Box and Behnken, 1960), Plackett-Burman (Plackett and Burman, 1946), Delaunay triangulations and their dual structures (Delaunay, 1934), and Voronoi tessellations (Voronoï, 1908). These methods work well for low dimensions ($N \leq 3$) (Davis and Ierapetritou, 2010); (Crombecq et al., 2009), but become extremely costly for large N as in the case of geometry and factorial designs, or inaccurate due to the lack of spatial coverage as in the case of CC, Plackett-Burman, and Box-Beheken designs. Moreover, these methods rapidly face the *curse of dimensionality*

* Corresponding author at: Department of Chemical and Biomolecular Engineering, 4 Engineering Drive 4, National Univeristy of Singapore, 117585, Singapore.
E-mail address: cheiak@nus.edu.sg (I.A. Karimi).

http://dx.doi.org/10.1016/j.compchemeng.2016.10.006
0098-1354/

(Forrester et al., 2008c). The third type of sampling methods relies on stochastic sampling with the aim to fill space. For instance, sampling based on random distribution is the most straightforward. The more sophisticated methods include Monte Carlo sampling (Metropolis and Ulam, 1949) and variations (Koehler and Owen, 1996; Niederreiter, 2010) that combine random sampling and probabilistic filtering; Latin hypercube sampling (McKay et al., 1979) that fills hypercube bins via random placements but subject to projection filters; and orthogonal arrays (Hedayat et al., 2012; Rao, 1946, 1947) that generalize Latin hypercube sampling (Giunta et al., 2003). Finally, the fourth type generates sample points quasi-randomly using low-discrepancy sequences such as Hammersley (Hammersley and Handscomb, 1964), Sobol (Sobol', 1967), and Halton (Halton and Smith, 1964). While these methods manage the sample size much better (Mascagni and Hongmei, 2004; Queipo et al., 2005), they may fail to capture the characteristics of the space properly at higher N. Typically, the system output is computed from the high-fidelity model at these sample points to generate the required input-output data for surrogate development. If the resulting surrogate does not meet expectations, then the model is reconstructed by adding more sample points.

The adaptive sampling methods attempt to address the drawbacks of the non-adaptive techniques discussed above. An adaptive method starts with a small set of sample points, and then adds points sequentially using some criteria or procedures. Provost et al. (1999) have shown that such methods normally require fewer sample points than the non-adaptive ones for the same surrogate accuracy. The approaches and objectives behind the adaptive techniques are varied. The most commonly used grid sampling technique, namely the Cartesian grid, can be made adaptive by evolving the entire grid rather than individual sample points. If a grid at hand is inadequate, then midpoints are inserted as additional sample points to make a finer grid. Thus, if a 5×5 grid (25 sample points) is inadequate for a $2D$ surrogate, then 56 additional points are added to get a 9×9 grid of 81 points. It is clear that the sample size increases exponentially with N in this method. Therefore, it is important to explore the idea of strategic sampling that not only fills the space, but also exploits the system knowledge for smart placements (Forrester et al., 2008b). Crombecq et al. (2009) proposed a novel sequential strategy involving both exploration and exploitation. They used a combination of derivative-based local linear approaximations and Voronoi tesselations to place new sample points. But, in most cases, the derivatives of a system to be modelled are not available a priori, and estimating them accurately and efficiently can be an arduous task due to the black-box and compute-intensive nature of the system. Moreover, as discussed earlier, the Voronoi tesselations can rapidly become computationally expensive at higher dimensions. A recent work by Eason and Cremaschi (2014) proposes an adaptive sampling strategy for ANN surrogates. Instead of generating all sample points in one shot, they generate them sequentially and randomly in a piecemeal manner. They use a *score* to select new sample points from the random points generated at each iteration. 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 (Quenouille, 1956). Though their selection of sample points is systematic, it is still from randomly generated points. Cozad et al. (2014, 2015) propose adaptive sampling for their surrogate modelling tool called ALAMO. They add sample points one at a time to an initial DoE-based set. For each 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. Jin et al. (2016) essentially made two modifications to the work of (Eason and Cremaschi, 2014). One, they improved ANN modelling by introducing auto-node selection, and second, they used maximum predicted error instead of expected variance.

In this work, we develop a smart sampling algorithm (SSA) that differs from the current approaches described above in four novel aspects. First, unlike Cremaschi and coworkers, it does not use any random sample points, potential or otherwise. Second, unlike Cozad et al., it employs surrogate-based optimization using derivative information rather than a black-box-based, derivative-free optimization. Third, it exploits all sample points as one set instead of dividing them into small subsets. Lastly, it integrates both spatial and quality considerations in a single objective to place new sample points.

In the rest of this article, we first define our problem and the key concepts behind our algorithm. Then, we present our algorithm, and illustrate it with a simple example and a practical case study. Finally, we evaluate its performance numerically using a variety of single-variable test problems from the literature.

2. Problem Statement

Let $y = f(\boldsymbol{x}); f : \Re^N \rightarrow \Re$ for $\boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U$ describe the behaviour of a unit/process/system whose experimental or computational quantification is complex and compute-intensive. We need an analytical or numerical surrogate model $\tilde{y} = S(\boldsymbol{x})$ to replace $f(\boldsymbol{x})$ in an optimization and/or analysis task. Hence, our problem is as follows.

Given:

- $y = f(\boldsymbol{x}); f : \Re^N \rightarrow \Re$ for $\boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U$.
- A mathematical form for $S(\boldsymbol{x})$.
- Upper limit (K_{max}) on the number of sample points at which $f(\boldsymbol{x})$ may be evaluated to obtain $S(\boldsymbol{x})$, or a desired accuracy for $S(\boldsymbol{x})$.

Obtain:

- K_{max} sampling points $\boldsymbol{x}_k$ ($k = 1, 2, \ldots, K_{max}$) that give the best $S(\boldsymbol{x})$ for approximating $f(\boldsymbol{x})$
- Or, the sampling points that give $S(x)$ with a prescribed accuracy for approximating $f(\boldsymbol{x})$.

3. Motivation

The most common sampling methods employed in surrogate construction are uniform (US), random (RS), statistical (LHS, e.g. Latin Hypercube), central composite design (CCD), and quasi-random low-discrepancy (QS, e.g. Sobol sequence). Each generates a set of sample points to achieve a spatially uniform coverage of $\Re^N$, and evaluates $f(\boldsymbol{x})$ at these points to obtain the required input-output data to construct $S(\boldsymbol{x})$. This surrogate development paradigm involves several key issues: How many sample points should we use? How (one-shot or adaptively) should we generate them? How do they affect the quality of surrogate approximation? Which sampling method gives the best approximation for a given form of $S(\boldsymbol{x})$?

Let us look at these questions with the help of an example. Consider using US and RS to obtain $S(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$ for $f(x) = x \times \sin(2 \times \pi \times x)$ on $0 \leq x \leq 1$. For US, we compute $f(x)$ at $x = 0$, $x = 1$, and nine equidistant points between them. We get $S(x) = -0.02 + 0.75x + 6.65x^2 - 25.27x^3 + 17.94x^4$. Now, let us say that we have a new sampling method that gives us the seven points shown by diamonds in Fig. 1, which give us a surrogate as good as the one from US. In other words, US amounted to oversampling. Now, let us assume that seven points are sufficient to get an acceptable $S(x)$, so we generate seven random points between 0 and 1. Fig. 1 shows that $S(x)$ does well only for $x \in [0.08, 0.80]$. In other words, RS can mean poor sample placement. Hence, the

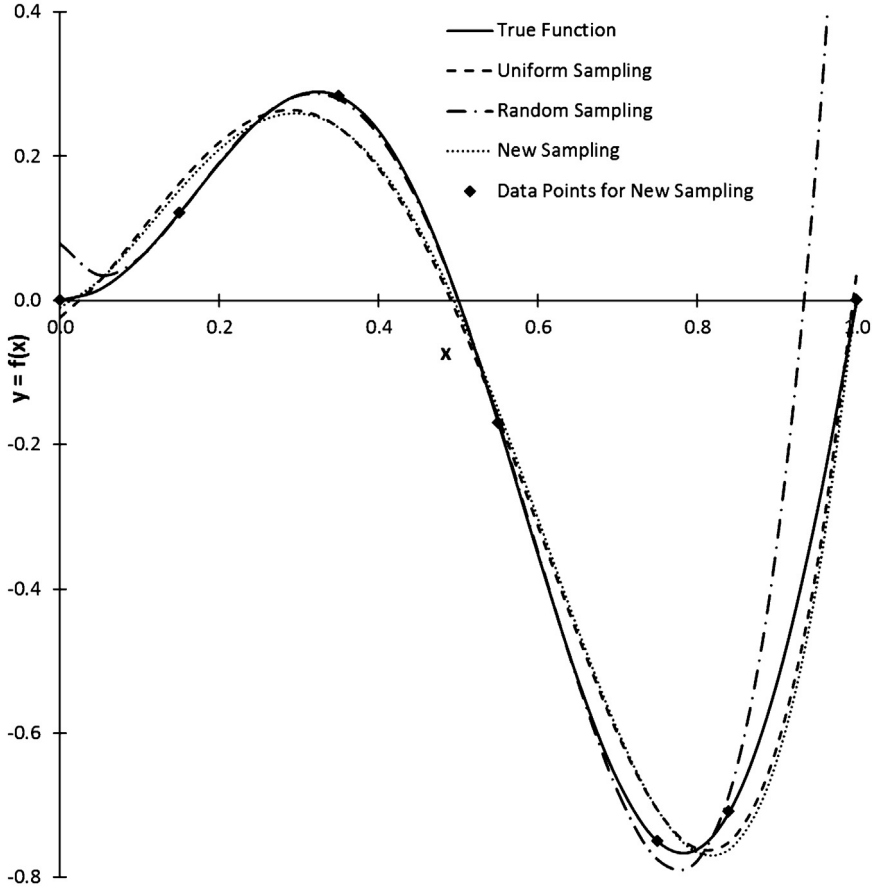

Fig. 1. Uniform sampling as a case of over-sampling and random sampling as a case of poor sample placement.

question is how we can place sample points smartly and avoid over-sampling and/or poor placement. This clearly needs an adaptive sampling strategy. We now present a few key concepts that underlie our proposed adaptive sampling strategy.

4. Key Concepts

First, we assume with no loss of generality that$0 \leq \boldsymbol{x} \leq 1$. This, as recommended by (Forrester et al., 2008c), is essential for reducing numerical problems from scaling. Thus, we normalize as follows.

$$\boldsymbol{z} = \frac{\boldsymbol{x} - \boldsymbol{x}^L}{\boldsymbol{x}^U - \boldsymbol{x}^L}$$

It is obvious that both sample points and surrogate model form together dictate the quality of a surrogate approximation. As discussed earlier, the literature provides several techniques for (1) surrogate form selection, and (2) fitting a surrogate to the sample points. In contrast to Cozad et al. whose primary aim was to develop a tool for selecting the best surrogate form, we wish to focus exclusively on sampling methodology for a given surrogate form. Therefore, we analyze the performance of our sampling method by averaging over different forms of$S(\boldsymbol{x})$.

4.1. Crowding Sistance Metric (CDM)

In any sampling method, distributing sample points spatially is a desirable feature. We judge this distribution by measuring distances between sample points. For instance, consider I sample points $(\boldsymbol{x}_i, i = 1, 2, \ldots, I)$. Then, the following crowding distance metric (Zhang et al., 2012) of a point $\boldsymbol{x}$ is a measure of its relative isolation from the remaining points.

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

where, $\|\boldsymbol{x} - \boldsymbol{x}_i\|$ is the Euclidean norm. The greater the CDM, the greater its isolation. We can use $CDM(\boldsymbol{x})$ to place new points in relatively unexplored regions and as far away from the existing points as possible. This helps us ensure a full and uniform coverage of$\Re^N$.

4.2. Departure Function

For measuring surrogate quality, we define a departure function. Let$S^I(\boldsymbol{x})$ denote a surrogate derived from I sample points (i=1,2,...,I). Let $\boldsymbol{x_j}$ be one of the I points, and let $S_j^I(\boldsymbol{x})$ be the surrogate derived from all points except$\boldsymbol{x}_j$. Then, we define the departure function as:

$$\Delta_j^I(\boldsymbol{x}) = S^I(\boldsymbol{x}) - S_j^I(\boldsymbol{x}) \quad j = 1, 2, \ldots, I \tag{2}$$

It measures the impact of locating a sample point in the neighborhood of $\boldsymbol{x}_j$ on$S(\boldsymbol{x})$.

4.3. Optimal Point Placement

We now combine the spatial (CDM) and quality (departure function) considerations discussed above to place a new sample point. Given I sample points $(i = 1, 2, \ldots, I)$, we should place the new point as far away from the existing points as possible to achieve space filling. Second, we should place it where the impact on $S(\boldsymbol{x})$ would be the highest. In other words, we should maximize both

$CDM(\boldsymbol{x})$ and the departure function. Therefore, we define a series of NLPs:

$$NLP(j): \max_{0\le \boldsymbol{x}\le 1} \left(\Delta_j^I(\boldsymbol{x})\right)^2 \times CDM(\boldsymbol{x}) \quad j = 1, 2, \ldots, I \tag{3}$$

An optimal solution of the above NLP can be a good candidate for the new sample point. Using the above three key concepts, we can now develop our algorithm (Fig. 2).

5. Smart Sampling Algorithm (SSA)

Let P denote the number of independent model parameters in $S(x)$. Then, at least P sample points are necessary to construct an $S(x)$. For SSA, we need an initial surrogate that is gradually improved by placing new sample points. Hence, our proposed algorithm proceeds as follows.

1. Set $K = P + 1 < K_{max}$. A higher $K < K_{max}$ is also acceptable.
2. Generate K sample points $\{\boldsymbol{x}_i, i = 1, 2, \ldots, K\}$. For this, one may use any existing sampling method (e.g. RS, US, QS, LHS, etc.). For 1-variable functions, our extensive numerical evaluation shows US to be the best.
3. Compute $f_i = f(\boldsymbol{x}_i)$, $i = 1, 2, \ldots, K$. Set $k = K$.
4. Construct $S^k(\boldsymbol{x})$ using $(\boldsymbol{x}_i, f_i)$, $i = 1, 2, \ldots, k$.
5. If $k = K_{max}$, then $S(\boldsymbol{x}) = S^k(\boldsymbol{x})$ and STOP. Otherwise, proceed next.
6. Compute $CDM_j = CDM(\boldsymbol{x}_j)$ using Eq. (1). Arrange CDM_j $(j = 1, 2, \ldots, k)$ in the descending order. Define the order as $p = 1, 2, \ldots, k$. Set $p = 1$.
7. Construct $S_p^k(\boldsymbol{x})$ using $(\boldsymbol{x}_i, f_i)$, $i = 1, 2, \ldots, k$, but $i \neq p$.
8. Construct (Eq. (3)) and solve $NLP(p)$. Let $\boldsymbol{x}^*$ be its optimal solution. If $\|\boldsymbol{x}^* - \boldsymbol{x}_i\| \le \varepsilon$ for any $i = 1, 2, \ldots, k$, then set $p = p + 1$ and go to Step 6. Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}^*$, and go to Step 4.

In the above, we assumed termination after K_{max} sample points. Other termination criteria can also be used. For instance, we could compute the changes in the surrogates between successive iterations as follows, and terminate, when (1) the maximum absolute change is minimal, or (2) the root mean squared change (RMSC) is minimal.

$$\Delta S_{max} = \max_{1\le \boldsymbol{q}\le \boldsymbol{Q}} |\boldsymbol{S^k}(\boldsymbol{x_q}) - \boldsymbol{S^{k-1}}(\boldsymbol{x_q})| \tag{4}$$

$$RMSC = \sqrt{\sum_{q=1}^{Q} [\boldsymbol{S^k}(\boldsymbol{x_q}) - \boldsymbol{S^{k-1}}(\boldsymbol{x_q})]^2 / (Q - P)} \tag{5}$$

where, $(\boldsymbol{x}_q, q = 1, 2, \ldots, Q)$ are some selected points in the domain. Note that ΔS_{max} in Eq. (4) may not be the maximum absolute error of the surrogate model over the entire domain, as it is based on only Q discrete points.

6. Surrogate Quality

We can measure quality in terms of the accuracy of surrogate predictions. This can be done at two sets of points. While the simplest set is $\boldsymbol{x}_k$, $k = 1, 2, \ldots, K_{max}$; the real test of a surrogate is at points away from these points. Hence, we select Q additional points $(\boldsymbol{x}_q, q = 1, 2, \ldots, Q)$ as the second set. Then, we take the average errors in surrogate predictions at these two sets of points as two separate measures of surrogate quality. The first set depicts the ability of a surrogate to emulate $f(x)$ at the K_{max} sample points, while the second describes the same at some points other than the sample points. Together, they can give us a better measure of emulation quality. As discussed later with an illustrative example, the two separate sets are useful in comparing sampling methods for the same surrogate model. The average error can be either root mean square or absolute deviation. Thus, we define the following six measures of surrogate quality.

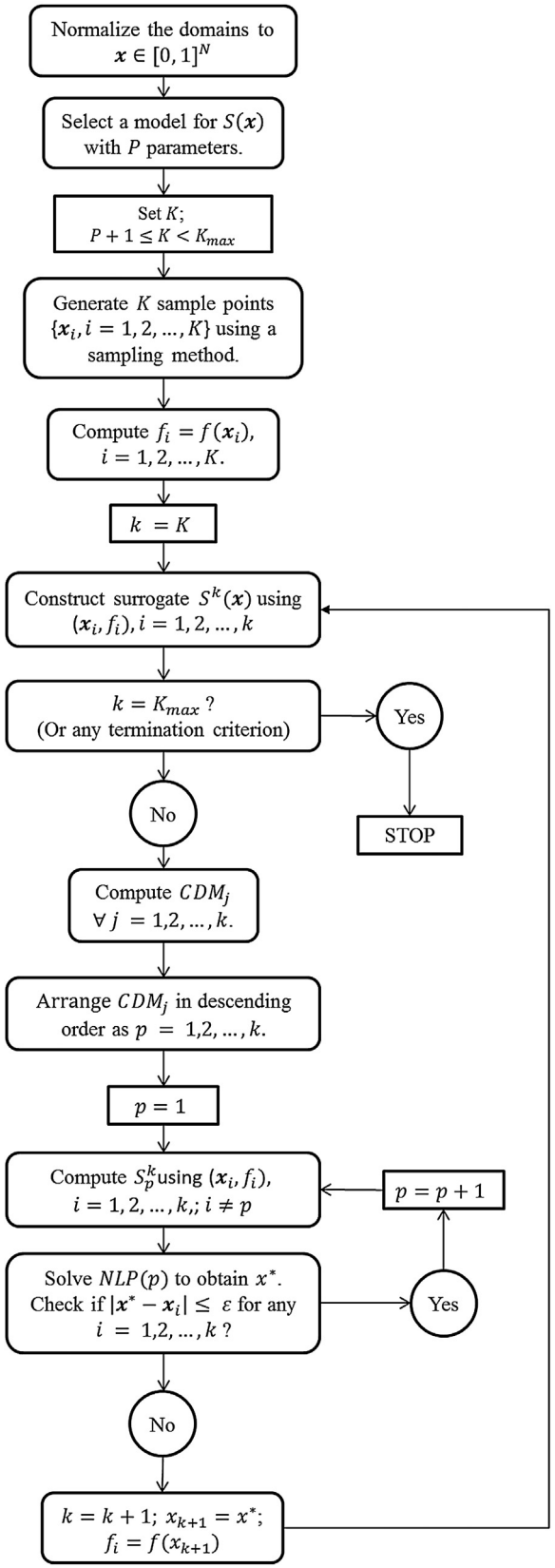

Fig. 2. Flow chart describing SSA.

RMS (Root Mean Square) Errors:

$$RMSE_1 = \sqrt{\frac{\sum_{k=1}^{K_{max}} [f(\boldsymbol{x_k}) - S(\boldsymbol{x_k})]^2}{K_{max} - P}} \tag{6a}$$

$$RMSE_2 = \sqrt{\frac{\sum_{q=1}^{Q} \left[f\left(\boldsymbol{x_q}\right) - S\left(\boldsymbol{x_q}\right)\right]^2}{Q - P}} \tag{6b}$$

$$RMSE = \sqrt{\frac{\sum_{q=1}^{Q} \left[f\left(\boldsymbol{x_q}\right) - S\left(\boldsymbol{x_q}\right)\right]^2 + \sum_{k=1}^{K_{max}} [f(\boldsymbol{x_k}) - S(\boldsymbol{x_k})]^2}{Q + K_{max} - P}} \tag{6c}$$

Absolute Errors:

$$AE_1 = \frac{1}{(K_{max} - P)} \sum_{k=1}^{K_{max}} |f(\boldsymbol{x_k}) - S(\boldsymbol{x_k})| \tag{7a}$$

$$AE_2 = \frac{1}{(Q - P)} \sum_{q=1}^{Q} |f\left(\boldsymbol{x_q}\right) - S\left(\boldsymbol{x_q}\right)| \tag{7b}$$

$$AE = \frac{1}{(Q + K_{max} - P)} \left\{ \sum_{k=1}^{K_{max}} |f(\boldsymbol{x_k}) - S(\boldsymbol{x_k})| + \sum_{q=1}^{Q} |f\left(\boldsymbol{x_q}\right) - S\left(\boldsymbol{x_q}\right)| \right\} \tag{7c}$$

Representative Error:

$$\psi = \sqrt{AE \times RMSE} \tag{8}$$

The lower the above errors for a surrogate, the better its quality. The quality measures in Eqs. (6c) and (7c) are essentially pooled estimates of their two constituents. Eq. (8) combines the two errors, *AE* and *RMSE*, into one single comprehensive or representative error. These errors will help us compare various surrogate models and sampling methods. To this end, we define relative (normalized) errors (Bhushan and Karimi, 2004) as follows.

Consider comparing M surrogate models ($m = 1, 2, \ldots, M$) for a given function. Given the errors of all models individually, we compute the relative error for model m by normalizing as follows,

$$\hat{AE}_m = AE_m / \min [AE_1, AE_2, \ldots, AE_M] \tag{9a}$$

$$\hat{RMSE}_m = RMSE_m / \min [RMSE_1, RMSE_2, \ldots, RMSE_M] \tag{9b}$$

$$\hat{\psi}_m = \psi_m / \min \left[\psi_1, \psi_2, \ldots, \psi_M\right] \tag{9c}$$

The lower the relative error, the better the model. The model with the relative error closest to 1.0 is the best.

Similarly, for comparing sampling methods (US, QS, RS, and SSA), we compute relative errors as,

$$\hat{AE}_{US} = AE_{US} / \min \left[AE_{US}, AE_{QS}, AE_{RS}, AE_{SSA}\right] \tag{10a}$$

$$\hat{AE}_{RS} = AE_{RS} / \min \left[AE_{US}, AE_{QS}, AE_{RS}, AE_{SSA}\right] \tag{10b}$$

$$\hat{AE}_{QS} = AE_{QS} / \min \left[AE_{US}, AE_{QS}, AE_{RS}, AE_{SSA}\right] \tag{10c}$$

$$\hat{AE}_{SSA} = AE_{SSA} / \min \left[AE_{US}, AE_{QS}, AE_{RS}, AE_{SSA}\right] \tag{10d}$$

$$\hat{RMSE}_{US} = RMSE_{US} / \min \left[RMSE_{US}, RMSE_{QS}, RMSE_{RS}, RMSE_{SSA}\right] \tag{11a}$$

$$\hat{RMSE}_{RS} = RMSE_{RS} / \min \left[RMSE_{US}, RMSE_{QS}, RMSE_{RS}, RMSE_{SSA}\right] \tag{11b}$$

$$\hat{RMSE}_{QS} = RMSE_{QS} / \min \left[RMSE_{US}, RMSE_{QS}, RMSE_{RS}, RMSE_{SSA}\right] \tag{11c}$$

$$\hat{RMSE}_{SSA} = RMSE_{SSA} / \min \left[RMSE_{US}, RMSE_{QS}, RMSE_{RS}, RMSE_{SSA}\right] \tag{11d}$$

$$\hat{\psi_{US}} = \psi_{US} / \min \left[\psi_{US}, \psi_{QS}, \psi_{RS}, \psi_{SSA}\right] \tag{12a}$$

$$\hat{\psi_{RS}} = \psi_{RS} / \min \left[\psi_{US}, \psi_{QS}, \psi_{RS}, \psi_{SSA}\right] \tag{12b}$$

$$\hat{\psi_{QS}} = \psi_{QS} / \min \left[\psi_{US}, \psi_{QS}, \psi_{RS}, \psi_{SSA}\right] \tag{12c}$$

$$\hat{\psi_{SSA}} = \psi_{SSA} / \min \left[\psi_{US}, \psi_{QS}, \psi_{RS}, \psi_{SSA}\right] \tag{12d}$$

The foregoing represent just one way of measuring surrogate quality. It is useful for simulation and analysis purposes. However, for optimization purposes, the ability of a surrogate to locate the global optima successfully is a useful measure of quality. The closer a surrogate predicts the global optima, the better its quality. Let us now illustrate these various measures of surrogate quality with a simple example.

7. Illustrative Example

Consider $f(x) = (6x - 2)^2 \times \sin(12x - 4)$ over $0 \le x \le 1$, which is known as the Forrester function (Forrester and Keane, 2009). Let $S(\boldsymbol{x}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4$. Hence, $N = 5, K = 6$, constructing $S(\boldsymbol{x})$ involves simple linear regression.

7.1. Algorithm Execution

We set $K = 6$, and randomly select the following initial sample points as per Step 2.

$$\text{ISS1 } x_1 = 0, \quad x_2 = 0.35, \quad x_3 = 0.47, \quad x_4 = 0.55, \quad x_5 = 0.69, \quad x_6 = 1, \tag{13}$$

Using the $f(x)$ values at the above points, we get the following as per Steps 4-6.

$$S^6(x) = 3.03 - 134.488x + 696.32x^2 - 1191.22x^3 + 642.18x^4$$

$$CDM_1 = 1.9011, CDM_2 = 0.7150, CDM_3 = 0.5710$$

$CDM_4 = 0.5710, CDM_5 = 0.7558, CDM_6 = 2.0020$ The descending order for the above crowding distances is $[i = 6, i = 1, i = 5, i = 2, i = 3 = 4]$. Hence, $p = 1$ refers to $i = 6$, and we obtain $S_1^6(x)$ by using x_1 to x_5.

$$S_1^6(x) = 155.02 - 1236.57x + 3554.21x^2 - 4332.49x^3 + 1875.66x^4$$

Then, $NLP(1)$ as per Eq. (4) is,

$$\max_{0 \le \boldsymbol{x} \le 1} [S^6(x) - S_1^6(x)]^2 \times CD(\boldsymbol{x})$$

$$\begin{aligned} &\max[151.09 - 1102.08x - 2857.89x^2 + 3141.27x^3 - 1233.48x^4]^2 \\ &\times [(x - 0.00)^2 + (x - 0.35)^2 + (x - 0.47)^2 + (x - 0.55)^2 \\ &+ (x - 0.69)^2 + (x - 1.00)^2] \end{aligned}$$

$x^* = 1$ is an optimal solution for the above NLP. Since this is already a sample point, we take $p = 2\,(i = 1)$. Solving $NLP(2)$ gives us $x^* = 0$. This being an existing sample point, we proceed to $p = 3$

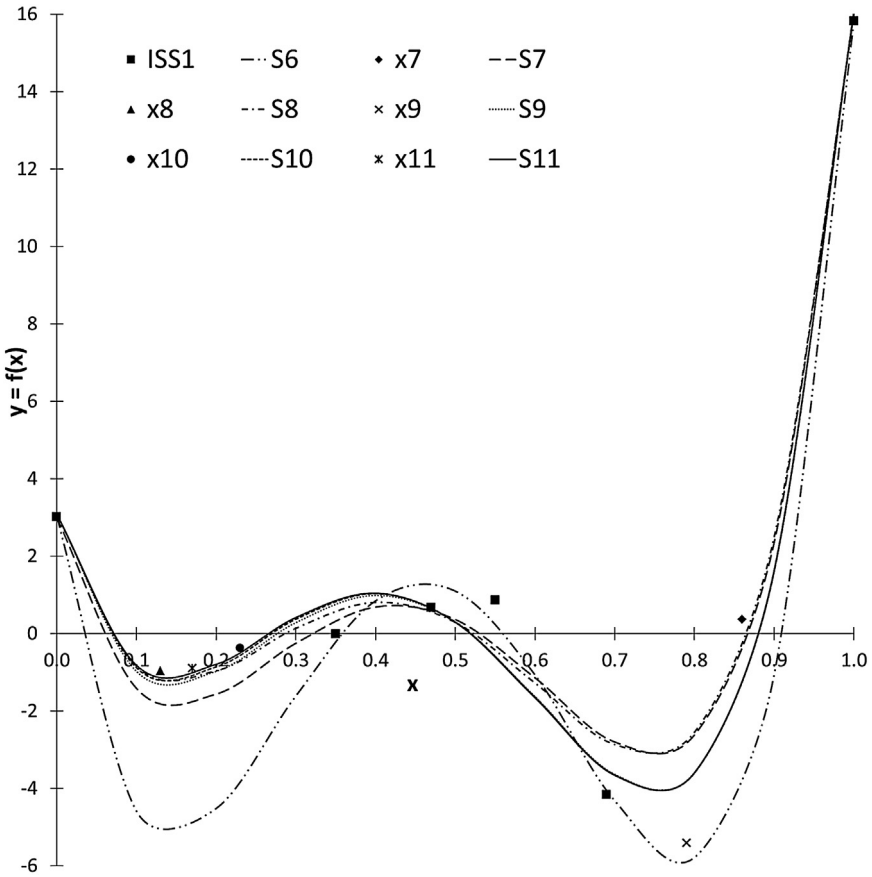

Fig. 3. Sample evolution and surrogate model progression during SSA for the illustrative example. (x7-x11 are the sample points generated by SSA and S7-S11 are the corresponding surrogates).

($i = 5$). The optimal solution of *NLP* (3) gives us a new sample point, namely$x_7 = 0.86$. The resulting surrogate is,

$$S^7(x) = 3.00 - 77.19x + 397.11x^2 - 706.20x^3 + 399.31x^4 \quad (14a)$$

Let us analyze the placement ofx_7. [0.00, 0.35] and [0.69, 1.00] are relatively sparse regions, hence they should be good for a new sample point. However, $f(x)$ is relatively linear in the former than the latter. A point in the latter region should improve the surrogate more than the former. $x_7 = 0.86$ is consistent with this argument.

Continuing further with our algorithm, we get,

$$S^8(x) = 3.07 - 68.59x + 357.10x^2 - 648.47x^3 + 372.87x^4 \quad (14b)$$

$$S^9(x) = 3.11 - 72.94x + 389.38x^2 - 715.42x^3 + 411.97x^4 \quad (14c)$$

$$S^{10}(x) = 3.09 - 71.22x + 382.45x^2 - 706.27x^3 + 408.05x^4 \quad (14d)$$

$$S(\boldsymbol{x}) = S^{11}(x) = 3.09 - 70.22x + 377.86x^2 - 699.69x^3 + 405.05x^4 \quad (14e)$$

Fig. 3 shows the evolutions of sample points and surrogates.

7.2. Surrogate Quality

For evaluating the quality of $S(\boldsymbol{x})$ (Eq. (14e)), we use$Q = 44$. This is called 4-fold validation, as it involves $4K_{max}$ points. We get$RMSE_1 = 1.58$, $RMSE_2 = 1.28$, $RMSE = 1.26$, $AE_1 = 0.64$, $AE_2 = 0.97$, and$AE = 0.83$ .AE_1, AE_2, and AE give an idea of the average errors in surrogate predictions, but do not give any insight into the variations of error magnitudes. Qualitatively, different sets of error values can have the same average, but their magnitudes may vary significantly. This is where$RMSE_1$, $RMSE_2$, and $RMSE$ along with the average errors can help. For comparable average absolute errors, one must compare the root mean squared errors. The lower the latter, the lower the variations in error. It is possible that one error set has a lower average but with higher variations, while another set has a higher average error but with lower variations. For instance, $AE_1 = 0.64$ is lower than$AE_2 = 0.97$. Thus, the error at the sampling points is lower than the validation points. But $RMSE_1 = 1.58$ is higher than$RMSE_2 = 1.28$. This means that the variation in errors at the sampling points is much higher than the validation points. Therefore, both measures of errors are useful, but combining their effects into a more comprehensive measure of ψ in Eq. (8) is helpful.

Since the errors estimated for $S(x)$ are based on a random initial sample set, we select four more sets to get an average performance of SSA with respect to the sample points. The errors listed below are in the order: $\{RMSE_1, RMSE_2, RMSE, AE_1, AE_2, AE\}$.

ISS1 : {0.00, 0.35, 0.47, 0.55, 0.69, 1.00};

Errors = {1.58, 1.28, 1.26, 0.64, 0.97, 0.83}

ISS2 : {0.00, 0.24, 0.35, 0.60, 0.85, 1.00};

Errors = {0.74, 1.73, 1.55, 0.30, 1.16, 0.94}

ISS3 : {0.00, 0.29, 0.38, 0.57, 0.76, 1.00};

Errors = {1.87, 1.29, 1.31, 0.76, 0.97, 0.85}

Table 1
Relative surrogate quality metrics from US, QS, and RS for initiating SSA in the illustrative example.

Relative Quality Metric	SSA-US	SSA-QS	SSA-RS
$A\hat{E}_1$	1.39	1.48	1.00-2.13
$A\hat{E}_2$	1.02	1.06	1.00-1.28
$\hat{AE}$	1.00	1.04	1.01-1.22
$RM\hat{S}E_1$	1.39	1.48	1.00-2.13
$RM\hat{S}E_2$	1.02	1.02	1.00-1.35
$\hat{RMSE}$	1.00	1.01	1.04-1.28
$\hat{\psi}$	1.00	1.02	1.02-1.21

$$\text{ISS4} : \{0.00, 0.16, 0.24, 0.40, 0.64, 1.00\};$$
$$\text{Errors} = \{1.10, 1.40, 1.29, 0.45, 1.06, 0.88\}$$

$$\text{ISS5} : \{0.00, 0.13, 0.35, 0.57, 0.82, 1.00\};$$
$$\text{Errors} = \{0.84, 1.47, 1.33, 0.34, 1.24, 1.01\}$$

Clearly, the quality metrics differ significantly from each other and vary much with the initial sample set. Since generating multiple initial sample sets is not a good idea, we must consider the options for selecting one initial set. US and QS are potential candidates, and it would be good to compare them with RS.

Let us define SSA-US as SSA with US for the initial sample, SSA-QS as SSA with QS for the initial sample, and SSA-RS as SSA with ISS1-ISS5 as the five initial sample sets. The initial sample sets from US and QS are as follows:

$$\text{ISS-US} : \{0.00, 0.20, 0.40, 0.60, 0.80, 1.00\}$$

$$\text{ISS-QS} : \{0.00, 0.05, 0.30, 0.55, 0.80, 1.00\}$$

Table 1 shows the relative errors for SSA-US, SSA-QS, and SSA-RS. Since SSA-RS uses ISS1-ISS5, we get a range of errors rather than one single error. Normalizing the errors with their least observed values, we see that SSA-US has the least relative error of 1.00. On other hand, SSA-RS can have high relative errors. In other words, US seems the best option for initializing SSA. This holds for other examples as well, as extensive numerical experiments confirm later in Section 8.

Having decided the method for generating the initial sample points for SSA, let us now compare SSA-US with the other sampling methods (US, QS, and RS) on this example. With $K_{max} = 11$, we generate eight surrogates: one for SSA-US, one for US, one for QS, and five for RS. As expected, RS surrogates have error ranges. Table 2 lists the average absolute and relative errors for the four methods. As we can see, RS does well (AE_1 = 0.10-0.41) at the sampling points, but poorly (AE_2 = 1.57-9.29) at the validation points. In contrast, US does poorly ($AE_1 = 1.83$) at the sample points, but well ($AE_2 = 0.77$) at the validation points. Surprisingly, SSA-US does well ($AE_2 = 0.76$) at not only the validation points, but also at the sampling points ($AE_1 = 0.99$). After normalizing the various errors, we get $\{\hat{AE}, \hat{RMSE}\}$ = {1.00, 1.00} for SSA-US, {1.15, 1.15} for US, {1.17, 1.16} for QS, and {(1.76-10.18), (2.26-20.15)} for RS. SSA-US achieves $\hat{\psi} = 1$, and performs the best with the selected model for this example.

After illustrating SSA and surrogate quality, we now study the effectiveness of the four sampling methods in yielding the global optima in an optimization problem. We solve two unconstrained optimization problems over $0 \le x \le 1$ using each of the eight surrogates (SSA-US, US, QS, and five RS). We then compare the optima from these surrogates with the true optima $f(x)$. The true global {minimum, maximum} for $f(x)$ is $\{0.76, 1.00\}$. In comparison, the SSA-US surrogate yields $\{0.75, 1.00\}$, the US surrogate yields $\{0.74, 1.00\}$, and the QS surrogate yields $\{0.76, 1.00\}$. While these surrogates do well in yielding the global optima, only two RS surrogates get close and yield {0.73, 1.00}. Clearly, RS performs poorly in this category as well.

This example demonstrates that SSA-US has the potential to outperform the other sampling methods. However, we clearly need more than just one small example to be certain. This brings us to the extensive numerical assessment of the next section.

8. Numerical Evaluation

We compare four sampling algorithms (SSA-US, US, QS, and RS) on seven test functions (TF1-TF7 in Table 3) using three surrogate models (M1-M3 in Table 4). As before, we use five sample sets for RS to get a range. We implemented SSA-US in Matlab R2012a (7.14.0.739) with "*fmincon*" as the NLP solver. For all the numerical experiments, we set $K = 6$ and $K_{max} = 11$. However, before we compare the four algorithms, we must first justify our choice of SSA-US over SSA-QS and SSA-RS.

Table 2
Quality metrics for the SSA-US, US, QS, and RS surrogates in the illustrative example.

Quality Metric	SSA-US	US	QS	RS
AE_1	0.99	1.83	0.88	0.10-0.41
$A\hat{E}_1$	10.21	18.84	9.03	1.00-4.20
AE_2	0.76	0.77	0.93	1.57-9.29
$A\hat{E}_2$	1.00	1.02	1.22	2.07-12.20
AE	0.71	0.82	0.83	1.26-7.26
$\hat{AE}$	1.00	1.15	1.17	1.76-10.18
$RMSE_1$	1.03	1.84	0.83	0.24-1.00
$RM\hat{S}E_1$	4.34	7.73	3.50	1.00-4.20
$RMSE_2$	1.09	1.13	1.29	2.61-23.36
$RM\hat{S}E_2$	1.00	1.04	1.19	2.40-21.51
$RMSE$	1.02	1.18	1.18	2.31-20.64
$\hat{RMSE}$	1.00	1.15	1.15	2.26-20.15
ψ	0.85	0.99	0.99	1.71-12.24
$\hat{\psi}$	1.00	1.15	1.16	2.00-14.32

Table 3
Test functions for the numerical evaluation of SSA.

Test Function	Name	Reference
$TF1 = (6x-2)^2 \sin(12x-4)$	Forrester	Forrester and Keane (2009)
$TF2 = 0.1x\sin(x)$	Holsclaw	Holsclaw et al. (2013)
$TF3 = \exp(-1.4x)\cos(3.5\pi x)$	Satner	Santner et al. (2003)
$TF4 = \sin(2\pi(x-0.1))$	Modified Currin[a]	Currin et al. (1988)
$TF5 = \frac{-(1+\cos(12x))}{2+0.5x^2}$	Modified Dropwave[a]	Surjanovic & Bingham
$TF6 = \frac{5}{\sqrt{2\pi}} \left\{ \begin{array}{l} exp(-0.5\left(x-1/3\right)^2 + \\ exp(-0.5\left(x-2/3\right)^2 \end{array} \right\}$	Modified Zhou[a]	Zhou (1998)
$TF7 = -94x\sin\left(\sqrt{94x}\right)$	Modified Eggholder[a]	Surjanovic & Bingham

[a] Modified to have single variable.

Table 4
Surrogate forms for the numerical evaluation of SSA.

Test Function	Form 1 (M1)	Form 2 (M2)	Form 3 (M3)
TF1	$\beta_0 + \beta_1 x + \beta_2 x^2$	$\beta_0 + \beta_1 \left(\frac{1}{1+x}\right) + \beta_2 x^3$	$\beta_0 + \beta_1 \left(\frac{1}{1+x}\right) + \beta_2 x^2$
	$+\beta_3 x^3 + \beta_4 x^4$	$+\beta_3 e^x + \beta_4 \tanh(x)$	$+\beta_3 x e^x + \beta_4 \tanh(x)$
TF2-TF7		$\beta_0 + \beta_1 x + \beta_2 x^2 +$	$\beta_0 + \beta_1 x + \beta_2 x \sin(x)$
		$\beta_3 \cos(x) + \beta_4 e^{-x^2}$	$+\beta_3 \cos(x) + \beta_4 e^{-x^2}$

Table 5
Relative surrogate quality metrics from US, QS, and RS for initiating SSA for various function-surrogate combinations.

Metric	TF1-M1			TF1-M2			TF1-M3		
	SSA-US	SSA-QS	SSA-RS	SSA-US	SSA-QS	SSA-RS	SSA-US	SSA-QS	SSA-RS
$\hat{AE}$	1.00	1.04	1.01-1.22	1.02	1.12	1.00-1.15	1.09	1.12	1.00-1.22
$\hat{RMSE}$	1.00	1.01	1.04-1.28	1.00	1.04	1.03-1.16	1.05	1.03	1.00-1.12
$\hat{\psi}$	1.00	1.02	1.02-1.21	1.00	1.07	1.01-1.10	1.05	1.06	1.00-1.14
		TF2-M1			TF2-M2			TF2-M3	
$\hat{AE}$	1.02	1.12	1.00-1.06	1.01	1.00	1.00-1.09	1.00	1.02	1.02-1.10
$\hat{RMSE}$	1.00	1.17	1.01-1.10	1.00	1.03	1.01-1.17	1.00	1.03	1.03-1.19
$\hat{\psi}$	1.00	1.14	1.00-1.07	1.00	1.01	1.00-1.12	1.00	1.02	1.02-1.14
		TF3-M1			TF3-M2			TF3-M3	
$\hat{AE}$	1.03	1.12	1.00-1.05	1.07	1.00	1.02-1.23	1.01	1.00	1.02-1.08
$\hat{RMSE}$	1.02	1.13	1.00-1.06	1.02	1.00	1.02-1.15	1.00	1.00	1.02-1.07
$\hat{\psi}$	1.03	1.14	1.00-1.05	1.04	1.00	1.02-1.19	1.01	1.00	1.02-1.08
		TF4-M1			TF4-M2			TF4-M3	
$\hat{AE}$	1.04	1.00	1.01-1.02	1.03	1.09	1.00-1.06	1.01	1.07	1.00-1.04
$\hat{RMSE}$	1.05	1.06	1.00-1.07	1.02	1.05	1.00-1.04	1.01	1.04	1.00-1.04
$\hat{\psi}$	1.04	1.00	1.01-1.04	1.03	1.07	1.00-1.05	1.01	1.06	1.00-1.04
		TF5-M1			TF5-M2			TF5-M3	
$\hat{AE}$	1.08	1.11	1.00-1.08	1.01	1.00	1.00-1.06	1.00	1.00	1.01-1.09
$\hat{RMSE}$	1.09	1.15	1.00-1.10	1.00	1.03	1.03-1.12	1.00	1.04	1.03-1.16
$\hat{\psi}$	1.08	1.13	1.00-1.09	1.00	1.01	1.01-1.08	1.00	1.02	1.02-1.12
		TF6-M1			TF6-M2			TF6-M3	
$\hat{AE}$	1.00	1.09	1.00-1.05	1.02	1.07	1.00-1.03	1.00	1.05	1.00-1.12
$\hat{RMSE}$	1.01	1.09	1.00-1.04	1.02	1.04	1.00-1.04	1.01	1.04	1.00-1.14
$\hat{\psi}$	1.00	1.09	1.01-1.03	1.02	1.05	1.00-1.03	1.01	1.05	1.00-1.13
		TF7-M1			TF7-M2			TF7-M3	
$\hat{AE}$	1.04	1.13	1.00-1.03	1.03	1.00	1.01-1.11	1.04	1.03	1.00-1.05
$\hat{RMSE}$	1.04	1.15	1.00-1.15	1.02	1.00	1.01-1.08	1.04	1.04	1.00-1.06
$\hat{\psi}$	1.04	1.14	1.01-1.03	1.02	1.00	1.01-1.10	1.02	1.02	1.00-1.03

For comparing SSA-US, SSA-QS, and SSA-RS, we follow the schema in Fig. 4. For each model (M1-M3), we use these algorithms to obtain a surrogate for each of TF1-TF7, i.e. 49 surrogates for each model. We then compute the relative errors for each function-model combination. Table 5 lists $\hat{AE}$, $\hat{RMSE}$, and $\hat{\psi}$ for each combination. Their averages over all the 21 combinations are {1.03, 1.02, 1.02} for SSA-US, {1.06, 1.06, 1.05} for SSA-QS, and {(1.00-1.10), (1.01-1.11), (1.01-1.09)} for SSA-RS. Clearly, SSA-US performs the best, and our choice of SSA-US is well justified.

Now, we can compare SSA-US with US, QS, and RS. Fig. 5 gives the schema for our numerical experiments. As before, we have 21 function-model combinations, and we generate eight surrogates for

Table 6
Relative quality metrics for surrogates from SSA-US, US, QS, and RS for various function-surrogate combinations.

Metric	TF1-M1				TF1-M2				TF1-M3			
	SSA-US	US	QS	RS	SSA-US	US	QS	RS	SSA-US	US	QS	RS
$\hat{AE}$	1.00	1.15	1.17	1.76-10.18	1.00	1.11	1.16	1.48-8.42	1.00	1.11	1.16	1.54-8.62
$\hat{RMSE}$	1.00	1.15	1.15	2.26-20.15	1.00	1.08	1.15	2.09-18.57	1.00	1.10	1.16	1.92-18.41
$\hat{\psi}$	1.00	1.15	1.16	2.00-14.32	1.00	1.10	1.16	1.76-2.50	1.00	1.11	1.16	1.72-12.59
			TF2-M1				TF2-M2				TF2-M3	
$\hat{AE}$	1.01	1.07	1.07	1.00-8.34	1.00	1.08	1.05	1.07-3.45	1.00	1.09	1.06	1.09-3.67
$\hat{RMSE}$	1.00	1.01	1.33	1.23-15.21	1.00	1.01	1.32	1.48-6.08	1.00	1.02	1.2	1.47-5.98
$\hat{\psi}$	1.00	1.04	1.19	1.10-1.22	1.00	1.01	1.22	1.26-4.41	1.00	1.05	1.14	1.27-4.62
			TF3-M1				TF3-M2				TF3-M3	
$\hat{AE}$	1.12	1.13	1.07	1.00-5.21	1.00	1.11	1.20	1.02-5.30	1.00	1.12	1.20	1.02-5.47
$\hat{RMSE}$	1.01	1.00	1.34	1.02-9.87	1.00	1.05	1.43	1.19-7.88	1.00	1.05	1.43	1.19-8.13
$\hat{\psi}$	1.05	1.05	1.18	1.00-7.10	1.00	1.08	1.31	1.10-6.46	1.00	1.08	1.31	1.10-6.67
			TF4-M1				TF4-M2				TF4-M3	
$\hat{AE}$	1.05	1.05	1.00	1.14-20.88	1.15	1.10	1.01	1.00-14.06	1.15	1.10	1.01	1.00-7.95
$\hat{RMSE}$	1.03	1.00	1.06	1.91-1.46	1.03	1.00	1.00	1.20-28.07	1.03	1.00	1.02	1.20-21.19
$\hat{\psi}$	1.02	1.00	1.00	1.50-24.97	1.07	1.03	1.00	1.10-19.57	1.07	1.03	1.00	1.11-12.78
			TF5-M1				TF5-M2				TF5-M3	
$\hat{AE}$	1.00	1.08	1.10	1.06-5.50	1.00	1.13	1.30	1.09-2.86	1.00	1.08	1.12	1.08-3.37
$\hat{RMSE}$	1.02	1.00	1.31	1.17-10.27	1.00	1.05	1.48	1.31-5.89	1.00	1.03	1.30	1.32-6.43
$\hat{\psi}$	1.00	1.03	1.19	1.10-7.45	1.00	1.09	1.39	1.20-4.11	1.00	1.05	1.21	1.19-4.65
			TF6-M1				TF6-M2				TF6-M3	
$\hat{AE}$	1.00	1.15	1.09	1.15-2.60	1.13	1.13	1.04	1.00-9.71	1.16	1.14	1.00	1.00-8.97
$\hat{RMSE}$	1.03	1.00	1.37	1.35-4.62	1.03	1.00	1.10	1.20-17.22	1.03	1.00	1.06	1.18-16.77
$\hat{\psi}$	1.00	1.06	1.20	1.22-3.41	1.01	1.00	1.00	1.08-12.15	1.06	1.03	1.00	1.09-11.89
			TF7-M1				TF7-M2				TF7-M3	
$\hat{AE}$	1.00	1.09	1.03	1.06-7.28	1.11	1.13	1.00	1.01-10.23	1.12	1.14	1.01	1.00-10.16
$\hat{RMSE}$	1.02	1.00	1.14	1.18-14.18	1.03	1.00	1.09	1.13-18.25	1.03	1.00	1.09	1.13-17.92
$\hat{\psi}$	1.00	1.03	1.07	1.11-10.04	1.02	1.02	1.00	1.07-13.12	1.02	1.01	1.00	1.06-12.83

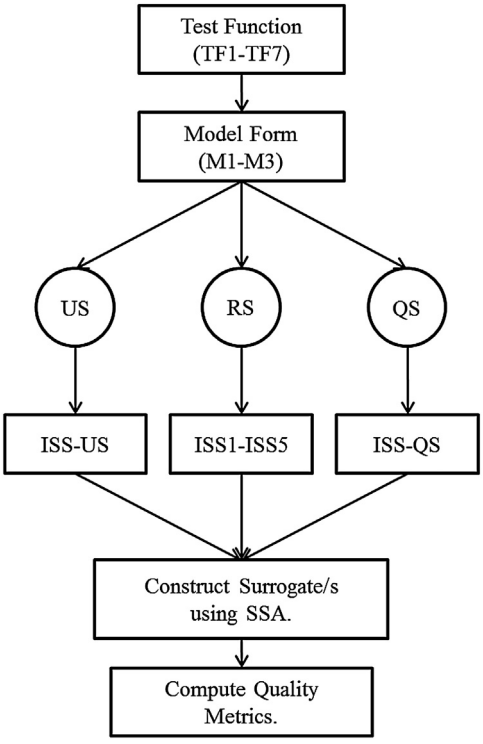

Fig. 4. Schema for evaluating the sampling methods for initiating SSA.

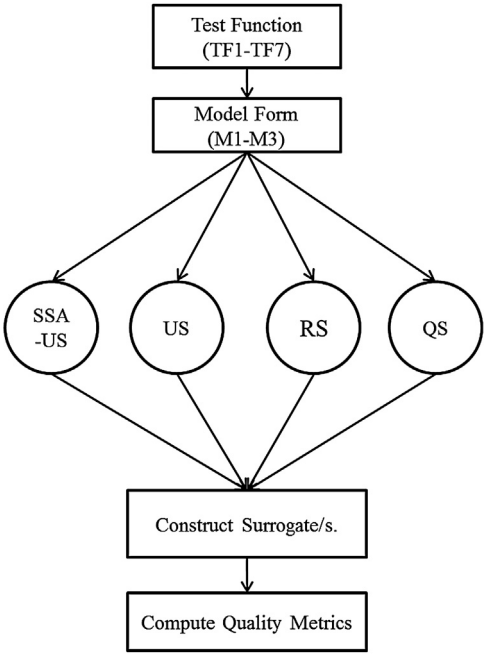

Fig. 5. Schema for the numerical evaluation of various sampling methods.

each. This results in 168 surrogates. The detailed results are in the supplementary, but Table 6 summarizes $\hat{AE}$, $\hat{RMSE}$, and $\hat{\psi}$. We can see that SSA-US outperforms all others in almost every case with $\hat{\psi}$ equal to or close to 1.00. In contrast, $\hat{\psi}$for the other methods are well above 1.00. After taking averages over the 21 combinations, $\hat{\psi}$is 1.02 for SSA-US, 1.05 for US, 1.13 for QS. and (1.25-10.33) for RS. Hence, SSA-US as a sampling method gives the best surrogates.

Table 7
Global optima from SSA-US, US, QS, and RS surrogates for various function-surrogate combinations.

Surrogate	US	RS	QS	SSA-US
TF1: Minimum – Maximum = 0.76-1.00				
M1	0.74-1.00	0.73 (2)-1.00 (4)	0.76-1.00	0.75-1.00
M2	0.73-1.00	0.72 (2)-1.00 (4)	0.75-1.00	0.73-1.00
M3	0.73-1.00	0.76 (1)-1.00 (4)	0.76-1.00	0.74-1.00
TF2: Minimum – Maximum = 1.00-0.80				
M1	1.00-0.83	0.00 (3)-0.81 (3)	0.00-0.83	1.00-0.83
M2	1.00-0.84	1.00 (4)-0.83 (1)	1.00-0.85	1.00-0.85
M3	1.00-0.84	1.00 (4)-0.82 (2)	1.00-0.85	1.00-0.85
TF3: Minimum – Maximum = 0.28-0.00				
M1	0.22-0.00	0.25 (3)-0.00 (5)	0.23-0.00	0.21-0.00
M2	0.24-0.00	0.26 (3)-0.00 (5)	0.25-0.00	0.24-0.00
M3	0.24-0.00	0.26 (3)-0.00 (5)	0.25-0.00	0.24-0.00
TF4: Minimum – Maximum = 0.85-0.35				
M1	0.87-0.34	0.85 (1)-0.35 (2)	0.87-0.34	0.88-0.34
M2	0.88-0.33	0.85 (1)-0.35 (1)	0.87-0.34	0.88-0.33
M3	0.88-0.33	0.82 (1)-1.00 (3)	0.87-0.34	0.88-0.33
TF5: Minimum – Maximum = 0.00-0.80				
M1	0.00-0.85	0.00 (5)-0.81 (1)	0.00-0.86	0.00-0.85
M2	0.00-0.86	0.00 (5)-0.85 (1)	0.00-0.87	0.00-0.87
M3	0.00-0.86	0.00 (5)-0.83 (1)	0.00-0.87	0.00-0.87
TF6: Minimum – Maximum = 0.00/1.00-0.50				
M1	0.00/1.00-0.50	0.00/1.00 (5)-0.50 (5)	0.00/1.00-0.50	0.00/1.00-0.50
M2	0.00/1.00-0.50	0.00/1.00 (5)-0.50 (5)	0.00/1.00-0.50	0.00/1.00-0.50
M3	0.00/1.00-0.50	0.00/1.00 (5)-0.50 (5)	0.00/1.00-0.50	0.00/1.00-0.50
TF7: Minimum – Maximum = 0.70-1.00				
M1	0.74-1.00	0.00 (2)-1.00 (5)	0.74-1.00	0.74-1.00
M2	0.74-1.00	0.00 (2)-1.00 (5)	0.73-1.00	0.72-1.00
M3	0.74-1.00	0.00 (2)-1.00 (5)	0.73-1.00	0.73-1.00

After evaluating surrogate qualities, we use the surrogate for each function-model combination to find its global optima. However, note that our sampling method is for developing a good surrogate approximation that can subsequently be used for simulation/optimization. This is in contrast to some works (Boukouvala et al., 2016) that employ adaptive sampling for direct surrogate-based optimization. In case of US, QS, and SSA-US, we get one global minimum and one global maximum for each surrogate. However, in case of RS, we get five pairs of such optima. Table 7 lists these global optima for all 21 combinations and four methods. Since RS surrogates often fail to yield the true optima, we count the instances when they do get close. Thus, Table 7 also gives the number of samples in which RS surrogates yield optima close to the true ones. SSA-US, US, and QS perform well in locating the true optima, but RS can often be far away from them. Finally, Table 8 provides the number of NLPs solved to generate the sampling set via SSA for each function-model combination.

Table 8
Range of number of NLPs solved to place a new sample point in SSA for three model forms (M1-M3) and seven test functions (TF1-TF7).

Test Functions	M1	M2	M3
TF1	21	26	27
TF2	29	31	31
TF3	29	31	21
TF4	31	31	31
TF5	32	31	31
TF6	30	31	31
TF7	27	31	31

9. Case Study

Styrene is industrially produced by the dehydrogenation of ethylbenzene (EB). Typically, EB is vaporized and mixed with steam (generally 1:10 on the molar basis), and passed over a catalytic bed to produce styrene. Three competing reactions occur in the reactor, whose extents vary significantly with feed temperature.

$$EB \rightleftharpoons Styrene + H_2 \tag{15}$$

$$EB \rightarrow Benzene + C_2H_4 \tag{16}$$

$$EB \rightarrow Toluene + CH_4 \tag{17}$$

Thus, the reactor output contains hydrogen, benzene, toluene, methane, and ethylene. Our objective is to study styrene flow as a function of feed temperature, and find the temperature that maximizes its production. The styrene reactor can be modelled rigorously inside Aspen Plus using a Langmuir-Hinshelwood-Hougen-Watson kinetics (LHHW) model with kinetics parameters from (Dittmeyer et al., 1999). However, we apply SSA-US to develop a surrogate approximation for this rigorous simulation model. To this end, we first develop an Aspen Plus simulation model using an adiabatic Plug Flow Reactor (PFR). We take the PFR as a cylindrical vessel with spherical catalyst beads and use Ergun's equation to compute its pressure drop. We assume the feed (EB:steam = 1:10 mol/mol) at 602.7 kmol/h and 1.378 bar, and restrict its temperature to be between 650 K and 1350 K.

Consider the surrogate model form in Eq. (18) for approximating three main product flows, namely styrene, benzene, and toluene. For SSA-US, we set K = 4 and K_{max} = 6.

$$S(x) = \beta_0 + \frac{\beta_1}{\left(1 + x^{4.7}\right)} + \beta_2 \exp\left(-9.6\left(x - 0.52\right)\right) \tag{18}$$

SSA-US yields {0.00, 0.33, 0.66, 1.00, 0.49, 0.54} as the six sample points for styrene, {0.00, 0.33, 0.66, 1.00, 0.49, 0.52} as the six points for benzene, and {0.00, 0.33, 0.66, 1.00, 0.49, 0.54} for toluene flow. The sample points for styrene and benzene are close, because their profiles are similar. This shows the effect of surrogate form and departure on sampling. Fig. 6 shows the flows of styrene, benzene, and toluene vs. feed temperature from the Aspen Plus model (Fstyrene-AP, Fbenzene-AP, and Ftoluene-AP) and our SSA-US surrogates (Fstyrene-SSA, Fbenzene-SSA and Ftoluene-SSA). Fig. 6 clearly shows that the surrogate approximation for toluene is very poor, and highlights the importance of a good model form. To

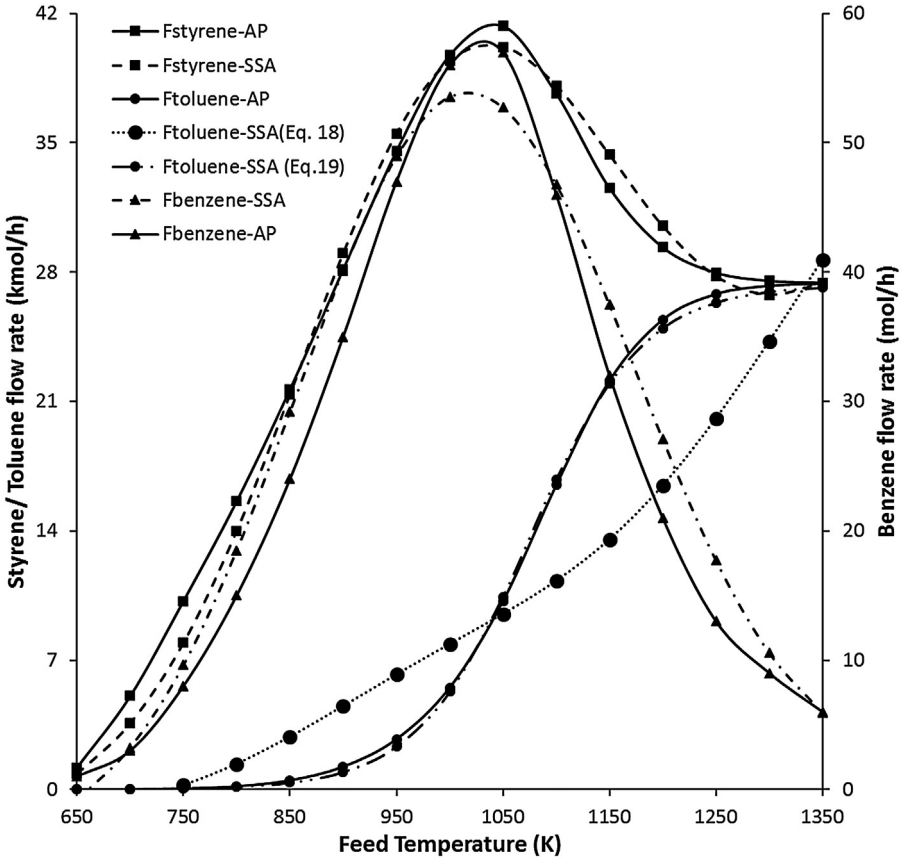

Fig. 6. Styrene, benzene and toluene flow rates as a function of feed temperature using rigorous Aspen Plus (AP) simulation and SSA based surrogate approximation.

improve the approximation for toluene, we try another surrogate form as follows.

$$S(x) = \beta_0 + \frac{\beta_1 \exp(13.2(x-0.61))}{(1+\exp(13.2(x-0.61)))} \tag{19}$$

With the above, we get {0.00, 0.33, 0.66, 1.00, 0.86, 0.47} as the sample points, which are quite different from those for toluene and benzene. As seen in Fig. 6, this surrogate works much better for toluene flow.

From Fig. 6, we see that the selectivity for styrene is higher at low temperatures, since the two side reactions in Eqs. (16) and (17) have low extents. However, the styrene conversion is also low at lower temperatures. As the temperature increases, the increased conversion of styrene outweighs the loss in selectivity for some temperatures. At higher temperatures, the effect of lower selectivity outweighs styrene conversion, which decreases the styrene flow. This is why we see a maximum in styrene flow.

Then, to find the best feed temperature for maximizing the styrene flow, we can solve the following black-box optimization problem.

$$\max_{650K \le T \le 1350K} F_{Styrene}(T)$$

This requires us to simulate the reactor in Aspen Plus for many points in the range 650–1350 K, and we find 1042 K as the best temperature with 41.427 kmol/h as the maximum styrene flow. To avoid these Aspen simulations, we now optimize our surrogate, which gives us 1035 K as the optimal feed temperature and 40.3 kmol/h as the maximum predicted styrene flow. Thus, optimizing the surrogate in place of rigorous simulation gives us errors of −0.67% in the optimal feed temperature and −2.72% in maximum styrene flow. Evaluating the styrene flow accurately at T = 1035 K gives us a styrene flow of 41.40 kmol/h, which represents an error of −0.07% in actual styrene flow.

This simple but practical case study illustrates the application of our SSA-US to a realistic process. Such surrogate approximations become very important, when one tries to simulate or optimize a full plant or an eco-industrial park with many plants. A recent paper by Pan et al. (2016) discusses in detail the importance of surrogate approximation in modelling and optimizing eco-industrial parks and industrial networks.

10. Conclusion

We developed a novel adaptive method (SSA) for generating the sample points required to build a surrogate approximation of a given high-fidelity function. The method begins with a set of initial sample points to generate new sample points in an iterative manner. The key novelties of our method are: (1) it combines both exploration (spatial distribution) and exploitation (surrogate quality) into a single objective for placing points, (2) it solves an optimization problem involving surrogates rather than the true function, and (3) it avoids the use of empirical scores and stochastic placement.

Our extensive numerical evaluations using 1-variable test problems suggest that our SSA performs the best, when its initial sample points are generated using uniform sampling. For now, this conclusion is valid for 1-variable functions only, and we are testing our algorithm for n-variable functions. It results in surrogates whose qualities are better than US or QS, and whose use in optimization locates the global optima quite well. A case study on styrene reactor optimization confirms the utility of our SSA for both approximation and optimization.

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office of Singapore under its CREATE programme.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compchemeng.2016.10.006.

References

Bhushan, S., Karimi, I.A., 2004. Heuristic algorithms for scheduling an automated wet-etch station. Comput. Chem. Eng. 28, 363–379.
Boukouvala, F., Misener, R., Floudas, C.A., 2016. Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. Eur. J. Oper. Res. 252, 701–727.
Box, G.E., Behnken, D.W., 1960. Some new three level designs for the study of quantitative variables. Technometrics 2, 455–475.
Box, G., Wilson, K.B., 1951. On the experimental attainment of optimum condition. J. R. Stat. Soc. Series B 1 (13), 20.
Clarke, S.M., Griebsch, J.H., Simpson, T.W., 2005. Analysis of support vector regression for approximation of complex engineering analyses. J. Mech. Des. Trans. ASME 127, 1077–1087.
Cozad, A., Sahinidis, N.V., Miller, D.C., 2014. Learning surrogate models for simulation-based optimization. AIChE J. 60, 2211–2227.
Cozad, A., Sahinidis, N.V., Miller, D.C., 2015. A combined first-principles and data-driven approach to model building. Comput. Chem. Eng. 73, 116–127.
Cressie, N.A.C., 1990. Statistics for Spatial Data: A Wiley-Interscience Publication. John Wiley & Sons inc.
Crombecq, K., De Tommasi, L., Gorissen, D., Dhaene, T., 2009. A Novel Sequential Design Strategy for Global Surrogate Modeling. Proceedings – Winter Simulation Conference, 734–742.
Currin, C., Mitchell, T., Morris, M., Ylvisaker, D., 1988. A Bayesian Approach to the Design and Analysis of Computer Experiments. Oak Ridge National Laboratory.
Davis, E., Ierapetritou, M., 2010. A centroid-based sampling strategy for kriging global modeling and optimization. AIChE J. 56, 220–240.
Delaunay, B., 1934. Sur la sphere vide. izv. akad. nauk SSSR. Otdelenie Matematicheskii i Estestvennyka Nauk 7, 1–2.
Dittmeyer, R., Höllein, V., Quicker, P., Emig, G., Hausinger, G., Schmidt, F., 1999. Factors controlling the performance of catalytic dehydrogenation of ethylbenzene in palladium composite membrane reactors. Chem. Eng. Sci. 54, 1431–1439.
Eason, J., Cremaschi, S., 2014. Adaptive sequential sampling for surrogate model generation with artificial neural networks. Comput. Chem. Eng. 68, 220–232.
Fisher, R.A., 1935. The Design of Experiments. Oliver and Boyd, Edinburgh, pp. 1935.
Forrester, A.I.J., Keane, A.J., 2009. Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. 45, 50–79.
Forrester, A.I.J., Sóbester, A., Keane, A., 2008a. Constructing a Surrogate In Engineering Design via Surrogate Modelling. John Wiley & Sons, Ltd., pp. 33–76.
Forrester, A.I.J., Sóbester, A., Keane, A., 2008b. Exploring and exploiting a surrogate. In: Engineering Design via Surrogate Modelling. John Wiley & Sons, Ltd., pp. 77–107.
Forrester, A.I.J., Sóbester, A., Keane, A., 2008c. Sampling Plans In Engineering Design via Surrogate Modelling. John Wiley & Sons, Ltd., pp. 1–31.
Giunta, A.A., Wojtkiewicz, S.F., Eldred, M.S., 2003. Overview of modern design of experiments methods for computational simulations. Proceedings of the 41 St AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2003-0649.
Halton, J., Smith, G., 1964. Radical inverse quasi-random point sequence, Algorithm 247. Commun. ACM 7, 701.
Hammersley, J.M., Handscomb, D.C., 1964. The general nature of monte carlo methods. In: Monte Carlo Methods. Springer, pp. 1–9.
Hardy, R.L., 1971. Multiquadric equations of topography and other irregular surfaces. J. Geophys. Res. 76, 1905–1915.
Hedayat, A.S., Sloane, N.J.A., Stufken, J., 2012. Orthogonal arrays: theory and applications. Springer Sci. & Bus. Media.
Henao, C.A., Maravelias, C.T., 2010. Surrogate-based process synthesis. Comput. Aided Chem. Eng. 28, 1129–1134.
Henao, C.A., Maravelias, C.T., 2011. Surrogate-based superstructure optimization framework. AIChE J. 57, 1216–1232.
Holsclaw, T., Sansó, B., Lee, H.K.H., Heitmann, K., Habib, S., Higdon, D., Alam, U., 2013. Gaussian process modeling of derivative curves. Technometrics 55, 57–67.
Hussain, M.F., Barton, R.R., Joshi, S.B., 2002. Metamodeling: radial basis functions: versus polynomials. Eur. J. Oper. Res. 138, 142–154.
Jin, Y., Li, J., Du, W., Qian, F., 2016. Adaptive sampling for surrogate modelling with artificial neural network and its application in an industrial cracking furnace. Can. J. Chem. Eng. 94, 262–272.
Koehler, J., Owen, A., 1996. 9 Computer experiments. Handb. Stat. 13, 261–308.
Martin, J.D., Simpson, T.W., 2005. Use of kriging models to approximate deterministic computer models. AIAA J. 43, 853–863.
Mascagni, M., Hongmei, C., 2004. On the scrambled halton sequence. Monte Carlo Methods Appl. 10, 435–442.
McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42, 55–61.
Metropolis, N., Ulam, S., 1949. The monte carlo method. J. Am. Stat. Assoc. 44, 335–341.
Myers, R., Montgomery, D., 2002. Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 2nd edn. Wiley Interscience New York, New York.
Niederreiter, H., 2010. Quasi-Monte Carlo Methods. Wiley Online Library.
Pan, M., Sikorski, J., Akroyd, J., Mosbach, S., Lau, R., Kraft, M., 2016. Design technologies for eco-industrial parks: from unit operations to processes: plants and industrial networks. Appl. Energy 175, 305–323.
Plackett, R.L., Burman, J.P., 1946. The design of optimum multifactorial experiments. Biometrika 33, 305–325.
Provost, F., Jensen, D., Oates, T., 1999. Efficient progressive sampling. In: The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, USA : ACM (pp. 439).
Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R., Kevin Tucker, P., 2005. Surrogate-based analysis and optimization. Prog. Aerosp. Sci. 41, 1–28.
Quenouille, M.H., 1956. Notes on bias in estimation. Biometrika 43, 353–360.
Rao, C.R., 1946. Hypercubes of strength d leading to confounded designs in factorial experiments. Bull. Calcutta Math. Soc 38, 67–78.
Rao, C.R., 1947. Factorial experiments derivable from combinatorial arrangements of arrays. Suppl. J. R. Stat. Soc. 9, 128–139.
Sakata, S., Ashida, F., Zako, M., 2003. Structural optimization using Kriging approximation. Comput. Methods Appl. Mech. Eng. 192, 923–939.
Santner, T.J., Williams, B.J., Notz, W.I., 2003. The Design and Analysis of Computer Experiments. Springer-Verlag New York, Inc, New York.
Shan, S., Wang, G.G., 2010. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct. Multidiscip. Optim. 41, 219–241.
Simpson, T.W., 1998. A Concept Exploration Method for Product Family Design. Georgia Tech Institute, Atlanta, Georgia, US.
Sobol', I. y. M., 1967. On the distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 7, 784–802.
Surjanovic, S., Bingham, D., Virtual Library of Simulations Experiments: Test Functions and Datasets. In (Vol. 2016).
Voronoï, G., 1908. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. J. für die reine und angewandte Mathematik 134, 198–287.
Wang, G.G., Shan, S., 2007. Review of metamodeling techniques in support of engineering design optimization. J. Mech. Des. Trans. ASME 129, 370–380.
Yegnanarayana, B., 2004. PHI Learning Pvt Ltd, Artificial neural networks.
Zhang, J., Chowdhury, S., Messac, A., 2012. An adaptive hybrid surrogate model. Struct. Multidiscip. Optim. 46, 223–238.
Zhou, Y., 1998. Adaptive Importance Sampling for Integration. Stanford University.