Computer Simulation-Based Designs for Industrial Engineering Experiments

Dongyu Zhou, Weihua Guo, Hengzhen Huang[*]

College of Mathematics and Statistics, Guangxi Normal University, Guilin, China

Email address:

hhzgxnu@126.com (Hengzhen Huang)
[*]Corresponding author

Abstract: Computer simulations have been receiving a lot of attention in industrial engineering as the rapid growth in computer power and numerical techniques. In contrast to physical experiments which are usually carried out in factories, laboratories or fields, computer simulations can save considerable time and cost. From the statistical perspective, the current research work about computer simulations is mostly focusing on modeling the relationship between the output variable from the simulator and the input variables set by the experimenter. However, an experimental design with careful selection of the values of the input variables can significantly affect the quality of the statistical model. Specifically, prediction on the edge area of the experimental domain, which is extremely critical for an industrial engineering experiment often suffers from inadequate data information because the design points usually do not well cover the edge area of the experimental domain. To address this issue, a new type of design, called semi-LHD is proposed in this paper. Such a design type has the following appealing properties: (1) it encompasses a Latin hypercube design as a sub-design so that the design points are uniformly scattered over the interior of the design region; and (2) it possesses some extra marginal design points which are close to the edge so that the prediction accuracy on the edge area of the experimental domain is fully taken into account. Detailed algorithms for finding the marginal design points and how to construct the proposed semi-LHDs are given. Numerical comparisons between the proposed semi-LHDs with the commonly-used Latin hypercube designs, in terms of prediction accuracy, are illustrated through simulation studies. It turns out that the proposed semi-LHDs yield desirable prediction accuracy not only in the interior but also on the edge area of the experimental domain, so they are recommended as the experimental designs for simulation-based industrial engineering experiments.

Keywords: Computer Experiments, Design of Experiments, Gaussian Process Model, Kriging, Latin Hypercube Design

1. Introduction

Computer experiments are becoming widely used in industrial engineering where computational methods are used to simulate the real-world phenomena (see, for example, references [1-7]). Most computer experiments for industrial engineering are unique in that the response has no random-error component. That is, replicates of the same inputs to the computer code will yield the same response. Since a computer experiment is typically treated as a black-box and is time demanding, the experimental design should be chosen judiciously and explore the experimental domain as thoroughly as possible. To this end, Mckay, Beckman and Conover [8] first proposed the Latin hypercube designs (LHDs) which possess the property that when projected onto any one dimension, they achieve maximum stratification. LHDs are the most popular designs for computer-based industrial engineering experiments and have enjoyed lots of development (see, for example, references [6, 7, 9-12]).

The issue of building prediction models for computer experiments has drawn significant attention of researchers. Currin, Mitchell, Morris and Ylvisaker [13] applied Bayesian prediction to analyze the outputs of computer experiments; Joseph, Hung and Sudjianto [14] proposed blind kriging for developing surrogate models; Kennedy and O'Hagan [15] gave a method for predicting the output from a computer experiment when fast approximations are available. For a

detailed treatment of this topic, please refer to references [6, 7]. Most of the existing work are paying attentions on modeling methods, while a design with careful planning can significantly affect the prediction accuracy of the model. When predicting the response at an unobserved site located in the interior of the design region, information is often sufficiently received from the design points around it. However, when the unobserved site is located on the edge of the design region, inadequate information may be received because the relatively small number of design points around it. As a result, the prediction accuracy on the edge of the design region may be poor, especially for industrial engineering experiments. The engine block and head joint sealing assembly experiment, the robot arm experiment and the integrated circuit experiment described by [1] are some good examples of this and very few experimental designs have been proposed to address this issue.

Motivated by the aforementioned issue, a new design type called semi-LHD is proposed for computer-based industrial engineering experiments which pays extra attention to the edge area of the experimental domain. A semi-LHD adds extra points to the marginal points of an LHD so that the proposed design inherits the space-filling property of the LHD. Furthermore, the proposed semi-LHD helps to increase the prediction accuracy on the edge while not deteriorating the overall prediction accuracy. Methods for finding marginal points and choosing the added points are systematically proposed, which are the central contributions of this work.

The remainder of this paper is organized as follows. Section 2 provides useful definitions and notations. Section 3 presents the design construction method and gives some numerical examples that simulate industrial engineering experiments. Concluding remarks in Section 4 close this paper.

2. Definitions and Notations

In this section, some useful definitions and notation will be given. A Latin hypercube design (LHD) is a $n \times p$ matrix, denoted by LHD (n, p) in which the number of levels in each column equals to the sample size and the levels are equally spaced. Without loss of generality, suppose that the design region is the unit cube $[0,1]^p$. That is, when the design points of an n-run LHD are projected on each dimension, there is exactly one point falling into each of the n equally spaced intervals on [0, 1]. In general, a random LHD can be constructed as follows. Let $A = (a_{ij})_{n \times p}$ be an $n \times p$ matrix each column of which is a random permutation of (1, 2,..., n), and the columns are generated independently.

Via A, construct an $n \times p$ matrix $D = (d_{ij})_{n \times p}$ where

$$d_{ij} = \frac{a_{ij} - u_{ij}}{n}, \qquad (1)$$

and u_{ij} is a random number drawn from U(0, 1) and a_{ij} and

u_{ij} are independent, i=1,...n, j=1,...p. Then D is a random LHD.

One primary goal of a computer experiment is to establish an *emulator* (also called a *metamodel*) which can capture the relationship between the inputs and the output of the computer code. In this paper the most commonly-used model, the *Gaussian process model* (see references [6, 7, 16]) is considered. Given an input $x \in [0,1]^p$, the response $y(x)$ is modeled as

$$y(x) = f(x)'\beta + Z(x,\Theta), \qquad (2)$$

where $f(x) = (f_1(x),...,f_q(x))'$ are some known functions defined on the design region, $\beta = (\beta_1,...,\beta_q)'$ is a vector of unknown parameters to be estimated and $Z(x,\Theta)$ assumed to be a realization of a stationary Gaussian process with zero mean, variance σ^2 and a vector of correlation parameters $\Theta = (\theta_1,...,\theta_p)'$. Given two different inputs $x_1 = (x_{11},...,x_{1p})$

$$Cov[y(x_1), y(x_2)] = \sigma^2 r(x_1 - x_2 \mid \Theta),$$

where $r(\cdot \mid \Theta)$ is the correlation function defined by the parameter Θ. The most commonly-used correlation function is the *Gaussian correlation function*, which is given by

$$r(x_1 - x_2 \mid \Theta) = corr[y(x_1), y(x_2)] = \exp\{-\sum_{i=1}^{p} \theta_i (x_{1i} - x_{2i})^2\}, \qquad (3)$$

where $\theta_i's$ are required to be positive. Throughout this paper, this correlation function is used since it is infinitely differentiable which is favorable in applications for industrial engineering [3].

A popular method for analyzing the model in (2) is through kriging [7]. Given a design of n inputs $D = (x_1',...,x_n')$ on $[0,1]^p$ with the corresponding response vector $y = (y(x_1),...,y(x_n))'$, according to the Gaussian process assumption, y follows the following multivariate normal distribution

$$y \sim MN(F\beta, \sigma^2 R),$$

where $F = (f(x_1),...,f(x_n))', R = (r_{ij})_{n \times n}$ is the $n \times n$ correlation matrix with r_{ij} being $r(x_i - x_j \mid \Theta)$ defined in (3). The negative log-likelihood of y is proportional to

$$n\log(\sigma^2) + \log(\mid R \mid) + (y - F\beta)'R^{-1}(y - F\beta)/\sigma^2, \qquad (4)$$

where $\mid A \mid$ denotes the determinant of matrix A. Following the maximum likelihood approach, the maximum likelihood estimates (MLEs) of β and σ^2 are given by

$$\hat{\beta} = (F'R^{-1}F)^{-1}F'R^{-1}y, \qquad (5)$$

$$\hat{\sigma^2} = (y - F\beta)'R^{-1}(y - F\beta)/n, \qquad (6)$$

respectively. It is easy to see that $\hat{\beta}$, $\hat{\sigma^2}$ and R are functions of the parameter Θ. Plug the MLEs of β and σ^2 into formula (4), the MLE of Θ is then given by

$$\hat{\Theta} = \arg\min_{\Theta}(n\log(\hat{\sigma^2}) + \log(|R|)). \qquad (7)$$

There are several optimization algorithms available to solve the problem in (7), see Fang, Li and Sudjianto [1] for details. Plug $\hat{\Theta}$ back into Eqs. (5) and (6), the best linear unbiased predictor (BLUP) of the response y at an untried point x^* is given by

$$\hat{y}(x^*) = f(x^*)'\hat{\beta} + r'\hat{R}^{-1}(y - F\hat{\beta}), \qquad (8)$$

where $r = (r(x^* - x_1)|\hat{\Theta}),...,r(x^* - x_n)|\hat{\Theta}))'$ and $\hat{R}$ is the $n \times n$ correlation matrix with the (i, j) th entry being $r(x_i - x_j)|\hat{\Theta})$. It is easy to verify that predictor (8) interpolates the response at any observed point x_i, which is a desired property for deterministic computer experiments.

3. The Methodology

3.1. The Design Construction Method

In this section an easy-to-implement method is presented to construct a new type of design called semi Latin hypercube design (semi-LHD). This design type is useful for predicting the responses at untried points which are specially located on or close to the edge of the experimental region. As for the points far away from the edge, the proposed design can also yield good prediction accuracy. The main idea of the proposed design is to add points to an existing LHD, thus the resulting design contains an LHD as a subdesign.

Suppose an n-run semi-LHD is to be constructed, the number of runs of the built-in LHD $n_0 (< n)$ is determined according to the needs of the experiment. Determine the parameter m which is the number of "marginal points" of the built-in LHD. Given n, n_0 and m, the general framework of the proposed design construction method is presented as follows.

Display 1 (Framework of the construction of semi-LHDs).

Step 1. Construct an LHD (n_0, p), denoted as D_1. Without loss of generality, let $D_1 = (x_1,...,x_{n_0})'$, where $x_1,...,x_{n_0}$ are rows (points) of D_1.

Step 2. For $i = 1,...,n_0$, calculate the average distance from x_i to the remaining points in D_1, denoted by a_i. Let $a = (a_1,...,a_{n_0})'$. Sort the elements in vector a in descending order and denote the resulting vector by $a^* = (a_1^*,...,a_{n_0}^*)'$. Rearrange the rows in D_1 by a^* and denote the resulting design by D_1^*.

Step 3. Let the first m rows of D_1^* form a subdesign d_1^*. For each point in d_1^*, add l points around it following a pre-specified regulation, where l is the smallest integer satisfying $ml \geq n - n_0$.

Step 4. Combine the ml points added in the previous step to form a design E^*. Let $D^* = (D_1^*{}', E^*{}')'$. If $ml = n - n_0$, $D = D^*$ is the desired design. If $ml > n - n_0$, delete $ml - (n - n_0)$ points from E^* following some rule and still denote the resulting array by E^*. Then $D = (D_1^*{}', E^*{}')'$ is the desired design.

The steps in Display 1 are given in a general manner, hereafter the details surrounding the framework are presented. As alluded to earlier, n_0 is determined according to the experimental needs. If n is not large, say $n \leq 50$, $n/2 \leq n_0 \leq 2n/3$ is recommended as a standard choice so that at least a small number of design points can be added; if n is large, n_0 can be a relative large portion of n. In *Step 1* of Display 1, a random LHD or a maximin distance LHD with sample size n_0 can be used. Other optimal LHDs could also be considered at this step. For simplicity, a random LHD is used throughout this paper. In *Step 2*, the average distance a_i from x_i of D_1 to the remaining points in D_1 is calculated by

$$a_i = \frac{1}{n_0 - 1}\sum_{k=1, k \neq i}^{n_0}\sqrt{\sum_{j=1}^{p}(x_{ij} - x_{kj})^2}. \qquad (9)$$

The design points in D_1^* in *Step 2* is ordered so that the first row has the largest average distance from the others, the second row has the second largest average distance and so on. It is clear that points with relatively large average distance are located near the edge of the design region and far from the other points.

Parameter m in *Step 3* is another value determined according to the practical needs. In applications, $m \approx n_0/4$ may be used as a standard choice, which represents the belief that about a quarter of the design points of the built-LHD are treated as the marginal points. The first m rows of D_1^* are thought to be marginal points since each of them has relatively large distance from the other points in D_1^*. With a little abuse of notation, denote $d_1^* = (x_1,...,x_m)'$ where $x_1',...,x_m'$ are rows (points) in d_1^* and

$x_i = (x_{i1},...,x_{ip})', i = 1,...,m.$ In *Step 3*, l points are added around each point in d_1^*. An algorithm for how to add the points is given as follows.

Algorithm 1.

Step 1. For $i = 1,...,m$, $j = 1,...,l$ and $k = 1,...,p$, let

$$e_{ij}^{(k)} = x_{ik} + \frac{2}{n_0} sign(\alpha)(0.5 + 0.5\gamma),$$

where $sign(x)$ returns 1 if $x \geq 0$ and -1 if $x < 0$, α is a random sample drawn from standard normal distribution and γ is a random sample drawn from uniform distribution on $(0,1)$.

Step 2. For $i = 1,...,m$, $j = 1,...,l$ and $k = 1,...,p$, if $e_{ij}^{(k)} > 1$, let $e_{ij}^{(k)} = 1 - 0.1(0.5 + 0.5\gamma)$ or if $e_{ij}^{(k)} < 1$, let $e_{ij}^{(k)} = 0.1(0.5 + 0.5\gamma)$.

Step 3. For $i = 1,...,m$ and $j = 1,...,l$, let $e_{ij} = (e_{ij}^{(k)},...,e_{ij}^{(p)})'$. For $i = 1,...,m$, $e_i = (e_{i1}',...,e_{il}')'$. Then let $E^* = (e_1',...,e_m')'$.

Step 4. Let $D^* = (D_1^{*'}, E_1^{*'})'$. Calculate the minimum distance between different points in D^*, denoted by $\bar{a}$. If $\bar{a} < a_0/3$, go to *Step 1*; otherwise, return D^*.

The main idea of Algorithm 1 is to add points in the neighborhood of the marginal design points. *Step 1* of Algorithm 1 defines the scope of the neighborhood which is neither too close to nor too far away from the marginal points and chooses points from this neighborhood. Fraction $2/n_0$ controls the degree of farness which could be altered to other values according to the practical needs and $0.5 + 0.5\gamma$ controls the degree of closeness which could also be changed. Since the coordinates of the points added according to *Step 1* of Algorithm 1 may exceeds the scope of the design region, *Step 2* is needed to deal with this issue. *Step 3* of Algorithm 1 assesses the combination of the added and existing points, i.e. D^*, so they are not too close from each other. If the points in D^* are too close from each other, during the analysis of the experiment some computational issues, such as the singularity problem, may be encountered. Moreover, Algorithm 1 still maintains the space-filling property of the resulting design.

After implementing Algorithm 1, *Step 3* of Display 1 is completed and one needs to go on to *Step 4* of Display 1. In this step, since ml may be larger than $n - n_0$, $ml - (n - n_0)$ points should be deleted from E^*. The way to delete points from E^* is described as follows. For each point e in E^*, calculate the average distance between e and points in $D^* \setminus \{e\}$ which is the subset of D^* lying outside $\{e\}$. Record the first $ml - (n - n_0)$ smallest distances and delete the corresponding points from E^*, then a semi-LHD with

sample size n is obtained. The proposed design is denoted by semi-LHD $(n, p; n_0)$.

The proposed semi-LHDs have appealing properties. First of all, semi-LHDs inherit the space-filling property of their built-in LHDs. Secondly, with extra points added to the marginal points of the built-in LHDs, semi-LHDs would yield good prediction on the edge area. In practice, the behavior of an industrial engineering experiment on (or near) the edge of the experimental domain is usually of great importance to practitioners since it may be unstable or variable to influence the statistical inference. The proposed designs not only spread points over the experimental region but also put effort on the edge area, which benefits predictions on both the edge area and interior of the experimental region.

An example illustrating the construction of semi-LHDs is given below.

Example 1. Suppose one would like to construct a semi-LHD (40, 2;24). Firstly, an LHD (24, 2), denoted by D_1 is generated as the built-in LHD. Next, eight marginal points are found from D_1. Following the steps of Algorithm 1, two extra points are added to each of the marginal points. Combine all the added points with points in D_1 to obtain the destined design which is denoted by $D^{(1)}$. Similarly, a semi-LHD (40, 2; 30) is constructed which is denoted by $D^{(2)}$. To show the difference between semi-LHDs and LHDs, an LHD (40, 2), denoted by $D^{(3)}$, is constructed. The bivariate projections for $D^{(1)}$, $D^{(2)}$ and $D^{(3)}$ are visualized in Figure 1. It is not difficult to observe that $D^{(1)}$ has more points near to the edge than $D^{(2)}$ since the built-in LHD of $D^{(1)}$ has less samples than that of $D^{(2)}$ while the sample sizes of $D^{(1)}$ and $D^{(2)}$ are the same. As alluded to earlier, it is up to the practical needs to determine the sample size of the built-in LHD. On the other hand, the LHD spreads design points more evenly over the design region.
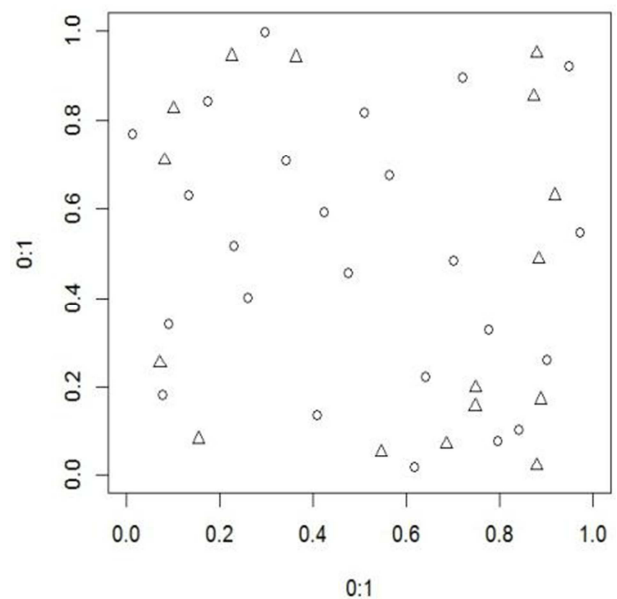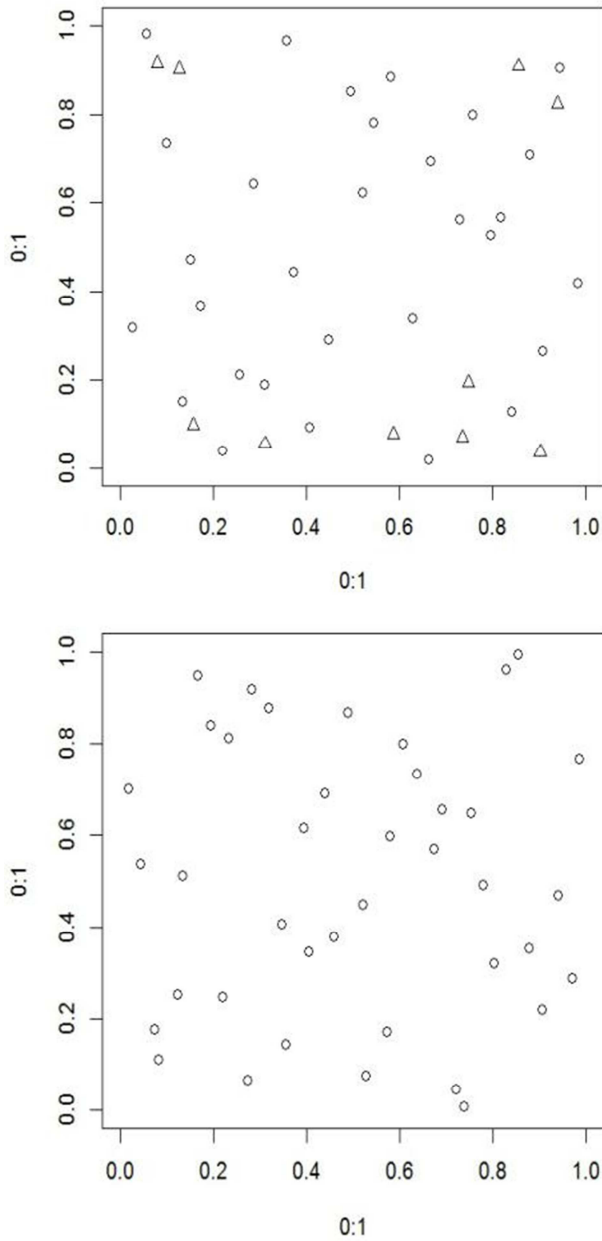
Figure 1. *Bivariate projetions for* $D^{(1)}$ *(panel on the left),* $D^{(2)}$ *(panel in the middle) and* $D^{(3)}$ *(panel on the right). Triangles represent the points in semi-LHDs apart from the built-in LHDs.*

It worthwhile to note that *Steps* 1-4 of Algorithm 1 may suffer from an endless loop since the condition $\bar{a} \geq a_0 / 3$ in *Step 4* of Algorithm 1 may be hard to achieve. It is thus recommended that the number of circulation should be preset.

3.2. Simulation Studies

Two simulative examples are given in this section to illustrate the usefulness of the proposed design. The prediction performances between random LHDs and semi-LHDs are compared. As for the criteria assessing the prediction performance, the leave-one-out cross-validation error (CVE) and the empirical mean squared prediction error

(MSPE) are adopted. Suppose $D = (x_1, ..., x_n)'$ is an n-run design and $y^* = (y_1, ..., y_n)'$ is the corresponding response vector, CVE is then defined as

$$CVE = \frac{1}{n} \sum_{x_i \in D} [\hat{y}_{-x_i}(x_i) - y(x_i)]^2, \qquad (10)$$

where $\hat{y}_{-x_i}$ is the predicted value of y based on data $D \setminus \{x_i\}$ and $y^* \setminus \{y_i\}$. The definition of MSPE is given by

$$MSPE = \frac{1}{N} \sum_{k=1}^{N} [\hat{y}(x_k) - y(x_k)]^2, \qquad (11)$$

where $x_1, ..., x_N$ are testing points selected from the design region. In the simulation, model (2) is established by setting $f(x) = 1$ and $\beta = \mu$ and the parameters are estimated through the maximum likelihood approach introduced in Section 2.

In the simulative examples, the following steps are carried out to assess the prediction performance.

Step 1. Create two data sets A and B. Let A be the collection of N_1 random points drawn from $[0,1]^p$ and B be the collection of N_2 points drawn from the area near the edge of $[0,1]^p$.

Step 2. Let $t = 1, ..., N_r$. Given t, this step proceeds as follows:

(1) Construct a semi-LHD ($n, p; n_0$), denoted by X_{1t}, and conduct the computer experiment based on X_{1t} to obtain the response vector y_{1t}. Fit model (2) based on $\{X_{1t}, y_{1t}\}$. Calculate CVE in (10) and denote the result by c_{1t}. Calculate MSPE in (11) based on A and B and denote the results by a_{1t} and b_{1t}, respectively.

(2) Construct an LHD (n, p), denoted by X_{2t}, and conduct the computer experiment based on X_{2t} to obtain the response vector y_{2t}. Fit model (2) based on $\{X_{2t}, y_{2t}\}$. Calculate CVE in (10) and denote the result by c_{2t}. Calculate MSPE in (11) based on A and B and denote the results by a_{2t} and b_{2t}, respectively.

Step 3. Let $a_j = (a_{j1}, ..., a_{jN_r})', b_j = (b_{j1}, ..., b_{jN_r})'$ and $c_j = (c_{j1}, ..., c_{jN_r})'$ for $j = 1, 2$.

Step 4. Calculate the mean and standard deviation of elements in the vectors obtained in *Step 3*.

Example 2. Consider a function from Currin, Mitchell, Morris and Ylvisaker (1991) which is given by

$$y = [1 - \exp(-\frac{1}{2x_2})] \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}. \quad (12)$$

The shape of (12) is visualized in Figure 2. This function is treated as a deterministic computer model for this example.

Once an experimental design is arranged, the response vector can be obtained from Eq.(12). As described at the beginning of this section, two testing data sets A and B are created where the former are collections of 40 points of a random LHD in $[0,1]^2$ and the latter 40 points drawn from area near the edge of $[0,1]^2$. The simulation is repeated 1000 times (i.e., $N_r = 1000$) and and in each time a semi-LHD (30, 2, 20) and an LHD (30, 2) are generated. The prediction performances of the semi- LHD (30, 2, 20) and the LHD (30, 2) are listed in Tables 1 and 2.

From Table 1 it is easy to see that when the testing data set is A, the prediction performance of semi-LHDs compares favorably to that of LHDs. On the other hand, when the testing data set is B, the mean and standard deviation of MSPE of semi-LHDs are considerably smaller than those of LHDs. This indicates that the proposed designs benefit the prediction capability on the area near to the edge while still behave well over the entire design region. Note that both of these design types have lower prediction accuracies on set B than set A. This is expected because predicting on the edge of the design region usually has more uncertainty, especially for industrial engineering experiments. Table 2 displays the CVE values from which the advantage of semi-LHDs over LHDs is also obvious.

Example 3. In this example, the following function

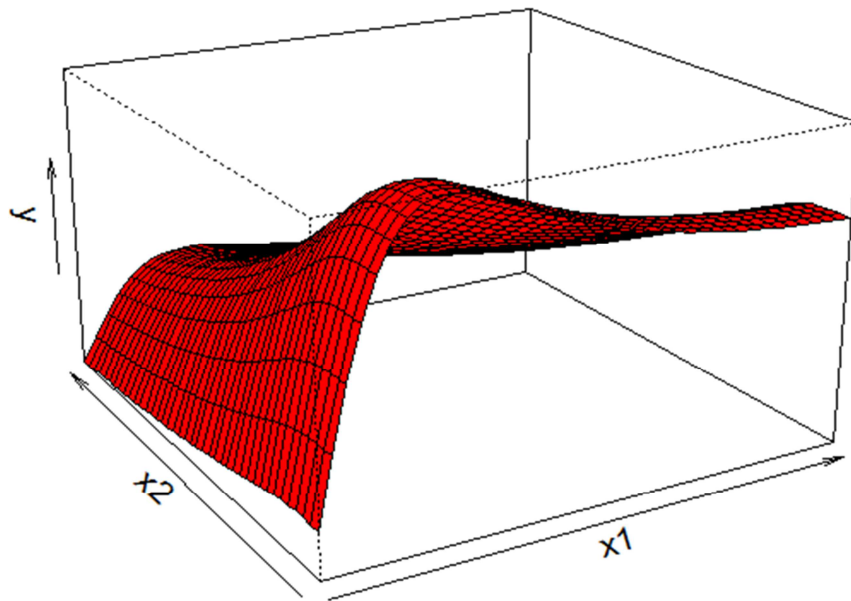$$y = -\log(0.01 + x_1) + 10\exp(x_2) + 5\sqrt{30x_3} \qquad (13)$$

Figure 2. *Plots of the function in (12).*

Table 1. *Means and standard deviations of the MPSE in Example 2, where the number in the parentheses is the standard deviation.*

	Data set A (LHD)	Data set B (edge)
semi-LHD	0.1235(0.1174)	0.1622(0.2726)
LHD	0.1196(0.1378)	0.2721(0.3916)

Table 2. *Means and standard deviations of the CVE in Example 2, where the number in the parentheses is the standard deviation.*

	CVE
Semi-LHD	0.1434(0.1462)
LHD	0.2665(0.3060)

is used as the deterministic computer model. Similar to Example 2, two testing data sets A and B are created where the former are collections of 40 points of a random LHD in $[0,1]^3$ and the latter 40 points drawn from area near the edge of $[0,1]^3$. The simulation is repeated 1000 times (i.e., $N_r = 1000$) and in each time a semi-LHD (30, 3, 20) and an LHD (30, 3) are generated. The prediction performances of the semi-LHD (30, 3, 20) and the LHD (30, 3) are listed in Tables 3 and 4.

Table 3. *Means and standard deviations of the MPSE in Example 3, where the number in the parentheses is the standard deviation.*

	Data set A (LHD)	Data set B (edge)
semi-LHD	0.7178(0.6164)	0.8275(1.0019)
LHD	0.6961(0.4032)	1.0059(0.8824)

Table 4. *Means and standard deviations of the CVE in Example 3, where the number in the parentheses is the standard deviation.*

	CVE
Semi-LHD	0.2149(0.2395)
LHD	0.2315(0.2427)

From Tables 3 and 4, similar conclusions as in Example 2 can be drawn. That is, Semi-LHDs benefit the prediction accuracy on the edge area while still behave well over the entire design region.

4. Conclusion

With the advent of computing technology and numerical methods, industrial engineers frequently use computer simulations to study actual or theoretical physical systems. Typically, the simulation model is resource-intensive in terms of model preparation, computation, and output analysis. As a result, the availability of a "cheap-to-compute" statistical model as a surrogate to the original complex simulation model is particularly useful. To establish a high quality statistical model, choosing a good set of design points is an important issue. In this paper, a novel type of experimental design, namely semi-LHD is proposed and applied to simulation-based industrial engineering experiments. Compared to the most commonly-used Latin hypercube design [8], the proposed design has advantage in helping making predictions on the edge of the design region where the behavior of the response often exhibits substantial uncertainty for an industrial engineering process. An explicit design construction algorithm is given and simulation studies confirm that the proposed semi-LHDs are desirable for computer simulation-based industrial engineering experiments because they can yield good prediction accuracy not only in the interior but also on the edge area of the experimental domain.

Acknowledgements

References

[1]   Zhang, Y. and Notz, W. I. (2015). Computer experiments with qualitative and quantitative variables: a review and reexamination. Quality Engineering, 27, 2-13.

[2]   Huang, H. Z., Lin, D. K. J., Liu, M. Q. and Yang, J. F. (2016). Computer experiments with both qualitative and quantitative variables. Technometrics 58, 495-507.

[3]   Nie, X., Huling, J. and Qian, P. Z. G. (2017). Accelerating large-scale statistical computation with the GOEM algorithm. Technometrics 59, 416-425.

[4]   Pareek B., Ghosh, P., Wilson, H. N., Macdonald, E. K. and Baines, P. (2018). Tracking the Impact of Media on Voter Choice in Real Time: A Bayesian Dynamic Joint Model. Journal of the American Statistical Association, https://doi.org/10.1080/01621459.2017.1419134.

[5]   Chen, J. K., Chen, RB, Fujii, A., Suda, R. and Wang, W. (2018). Surrogate-assisted tuning for computer experiments with qualitative and quantitative parameters. Statistica Sinica 28, 761-789.

[6]   Santner, T. J., Williams, B. J. and Notz, W. I. (2003). The design and analysis of computer experiments. New York: Springer.

[7]   Fang, K. T., Li, R. and Sudjianto, A. (2006). Design and modeling for computer experiments. New York: CRC Press.

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

[9]   Wang, L., Yang, J. F., Lin, D. K. J. and Liu, M. Q. (2015). Nearly orthogonal Latin hypercube designs for many design columns. Statistica Sinica 25, 1599-1612.

[10] Yang, X., Yang, J. F., Lin, D. K. J. and Liu, M. Q. (2016). A new class of nested orthogonal Latin hypercube designs. Statistica Sinica 26, 1249-1267.

[11] Chen, H., Huang, H. Z., Lin, D. K. J. and Liu, M. Q. (2016). Uniform sliced Latin hypercube designs. Applied Stochastic Models in Business and Industry 32, 574-584.

[12] Wang, X. L., Zhao, Y. N., Yang, J. F. and Liu, M. Q. (2017). Construction of (nearly) orthogonal sliced Latin hypercube designs. Statistics and Probability Letters 125, 174-180.

[13] Currin, C., Mitchell, T. J., Morris, M. D. and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions with applications to the design and analysis of computer experiments. Journal of the American Statistical Association 86, 953-963.

[14] Joseph, V. R., Hung, Y. and Sudjianto, A. (2008). Blind kriging: A new method for developing metamodels. ASME Journal of Mechanical Design 130, 031102-1-8.

[15] Kennedy, M. C and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. Biometrika 87, 1-13.

[16] Fang, K. T., Liu, M. Q., Qin, H. and Zhou, Y. D. (2018). Theory and application of uniform experimental designs. New York: Springer.