Research Article

Two General Extension Algorithms of Latin Hypercube Sampling

Zhi-zhao Liu, Wei Li, and Ming Yang

Control and Simulation Center, Harbin Institute of Technology, Harbin 150080, China

Correspondence should be addressed to Ming Yang; myang@hit.edu.cn

Received 5 April 2015; Revised 30 June 2015; Accepted 8 July 2015

Academic Editor: Jose J. Muñoz

Copyright © 2015 Zhi-zhao Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For reserving original sampling points to reduce the simulation runs, two general extension algorithms of Latin Hypercube Sampling (LHS) are proposed. The extension algorithms start with an original LHS of size m and construct a new LHS of size $m + n$ that contains the original points as many as possible. In order to get a strict LHS of larger size, some original points might be deleted. The relationship of original sampling points in the new LHS structure is shown by a simple undirected acyclic graph. The basic general extension algorithm is proposed to reserve the most original points, but it costs too much time. Therefore, a general extension algorithm based on greedy algorithm is proposed to reduce the extension time, which cannot guarantee to contain the most original points. These algorithms are illustrated by an example and applied to evaluating the sample means to demonstrate the effectiveness.

1. Introduction

Latin Hypercube Sampling (LHS) is one of the most popular sampling approaches, which is widely used in the fields of simulation experiment design [1], uncertainty analysis [2], adaptive metamodeling [3], reliability analysis [4], and probabilistic load flow analysis [5]. Compared with other random or stratified sampling algorithms, LHS has a better space filling effect, better robustness, and better convergence character. The extension of LHS is to obtain a LHS of a larger size that reserves the preexisting LHS (or the original LHS). There are at least two situations that need the extension of sampling, especially for time consuming simulation systems. One is sequential sampling for sequential analysis, adaptive metamodeling, and so on. The other is to consider the extension of LHS when the original LHS was subsequently determined to be too small and a new LHS of a larger size without original sampling points might be time consuming. But the LHS structure makes it difficult to increase the size based on an original LHS while simultaneously keeping the stratification properties of LHS.

A special extension case is the integral-multiple extension where the new LHS is integral times the size of the original sampling. Tong [6] proposed integral-multiple extension algorithms for stratified sampling methods including LHS. Sallaberry et al. [7] gave a two-multiple extension algorithm

of LHS with correlated variables. Later, two related techniques appeared in the papers named “nested Latin hypercube design” [8] and “nested orthogonal array-based Latin hypercube design” [9]. A nested Latin hypercube design with two layers is defined to be a Latin hypercube design that contains a smaller Latin hypercube design as a subset. A special integral-multiple extension method called k -extended LHS method was illustrated, where the new LHS contains k smaller LHSs [10]. Some related papers were produced by Vorechovsky [11–13], where the new sampling size is multiple times more than the original sampling size. The integral-multiple extension algorithms have a good feature that can obtain a strict LHS of larger size and simultaneously preserve all the original sampling points.

In this study, we consider the general extension algorithm of LHS where the new sample size is more controllable and the algorithm can be applied more widely. Wang [14] and Blatman and Sudret [15] obtained an approximate LHS of a larger size, which might have two or more original points falling into the same variable interval. Wei [16] also proposed a general extension algorithm to get an approximate LHS, which might have no point falling into a variable interval. However, a sample is a LHS if (and only if) there is only one point in each variable interval. The approximate LHS does not satisfy the definition and is harmful to the extension

with some criteria, such as correlated variables, maximizing minimum distance, orthogonal array, and so on.

In this paper, we would like to obtain a strict extension of LHS (ELHS) rather than an approximate one and the new LHS contains original sampling points as many as possible. The extension algorithm includes two parts: the reservation of original sampling points and the generation of new ones. As the generation of new sampling points is almost the same as integral-multiple extension, the reservation of original sampling points is the main problem to discuss. The relationship of original sampling points in new LHS is expressed by a graph. Then, the reservation problem can be solved by graph theory.

In Section 2, the procedure of LHS and the mathematical description for extension problem of LHS are given, which is the basis of extension algorithm. In Section 3, a basic general extension algorithm of LHS (BGELHS) is proposed based on the graph theory, which reserves the most original sampling points. A general extension algorithm of LHS based on greedy algorithm (GGELHS) is intended to balance the simulation runs and the generation time of ELHS. In Section 4, the proposed extension algorithms are illustrated by an application, which is performed to evaluate sample means. Finally, the conclusion and some thoughts for the future research are given.

2. Latin Hypercube Sampling and Extension Problem

2.1. Procedure of LHS. Suppose that the input variables are $x_1, x_2, \dots, x_r$ and the range of x_i is $u_i = [a_i \ b_i]$, $i = 1, 2, \dots, r$. Then, LHS can be obtained as follows, which can ensure that each input variable has all portions among its range.

- Divide the range u_i into n equiprobable intervals $u_{i1}, u_{i2}, \dots, u_{in}$, $i = 1, 2, \dots, r$. So the intervals satisfy $u_{i1} \cup u_{i2} \cup \dots \cup u_{in} = u_i$, $u_{ij} \cap u_{ik} = \emptyset$, and $P(x \in u_{ij}) = 1/n$, where $j, k = 1, 2, \dots, n$.
- For the j th interval of variable x_i , the cumulative probability can be obtained as

$$\text{Prob}_{ji} = \frac{(j-1)}{n} + \frac{r_{ji}}{n}, \quad (1)$$

where r_{ji} is a uniform random number ranging from 0 to 1. So all the probability values can be noted as $\text{Prob} = (\text{Prob}_{ji})_{n \times r}$.

- Transform the probability into the sample value x_{ji} by the inverse of the cumulative distribution function $F(\cdot)$:

$$x_{ji} = F_i^{-1}(\text{Prob}_{ji}). \quad (2)$$

Then, the sample matrix is

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1r} \\ x_{21} & x_{22} & \cdots & x_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} \end{bmatrix}. \quad (3)$$

FIGURE 1: An example of LHS.

- The n values of each variable are paired randomly or in some prescribed order with the n values of the other variables. Then the sample matrix of LHS can be written as

$$X' = \begin{bmatrix} x'_{11} & x'_{12} & \cdots & x'_{1r} \\ x'_{21} & x'_{22} & \cdots & x'_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{n1} & x'_{n2} & \cdots & x'_{nr} \end{bmatrix}, \quad (4)$$

where each row is a sampling point.

Figure 1 shows an example of LHS in size of 5, where each interval of each input variable has one sampling point.

2.2. Extension Problem of LHS. In this paper, the extension problem of LHS has two purposes. One is to construct a strict LHS of a larger size than original LHS. The other is to reserve the most original sampling points in the new LHS. Assume that there is an original LHS and the sample set is $A = \{X_1, X_2, \dots, X_n\}$, where X_i is a sampling point, $i = 1, 2, \dots, n$. The objective of extension is to get ELHS of size $m + n$ noted as $B = \{X'_1, X'_2, \dots, X'_{n+m}\}$, where $m \geq 1$ and m is an integer. B satisfies the constraint condition as shown in

$$\max \text{card}(A \cap B), \quad (5)$$

where $\text{card}(\cdot)$ denotes the number of elements in set. It means that the objective is to obtain ELHS in size of $n + m$ that contains the original LHS points as many as possible. Figure 2 shows the original LHS points in Figure 1 and the structure of ELHS in size of 7, which can be seen that the original sampling points do not accord with the new structure. So we need to delete original sampling points as little as possible and then create some new sampling points. The reserved points and the new ones are ELHS in size of 7.

FIGURE 2: Distribution of generated sampling points in new structure.

3. General Extension Algorithms of LHS

The general extension problem can be described as (5). In this section, two general extension algorithms of LHS are demonstrated to get ELHS. The problem of selecting reserved points is transformed into a problem to get the maximum independent set, which is a graph theory problem. The generation procedure is based on a subtraction rule [14]. These two stages construct BGELHS in Section 3.1. As BGELHS costs too much time to obtain ELHS, GGELHS is proposed in Section 3.2.

3.1. Basic General Extension Algorithm of LHS. Some original sampling points might be deleted before generating new sample. At the beginning of BGELHS, the original sample is transformed into a graph. Here the definition of simple graph is introduced.

Definition A (see [17]). A simple graph $G = (V, E)$ is a graph where there is no loop or multiple edges, where $V(G)$ is a vertex set and $E(G)$ is an edge set. No loop means that there is no edge from a vertex to itself. No multiple edges means that there is 0 or 1 edge between two vertices.

Let the original sample matrix be $X_{n \times r}^{(old)}$ and note the sampling points as $1, 2, \dots, n$ by the row orders. Transform the sampling points into vertices of a graph as follows: Every point is a vertex and links two vertices by a line if the corresponding two sampling points are in the same interval of the ELHS structure. Then we get a simple undirected graph. Order the sampling points in Figure 2 from left to right and the corresponding simple undirected graph is obtained as shown in Figure 3. The original problem in (5) is to delete the least vertices and their corresponding lines so that the remaining vertices have no line to link, which can be proved by reduction. Independent set, maximal independent set,

FIGURE 3: The simple undirected acyclic graph of Figure 2.

and maximum independent set are introduced to solve this problem.

Definition B (see [17]). It is supposed that there is a simple graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. If a set of vertices $I \subseteq V(G)$ and there is no edge between any two vertices of I , the set I is called an independent set.

Definition C (see [17]). It is supposed that there is a simple graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. I and I' are independent sets. If $\forall v_i \in V$ and $v_i \notin I$, $I' = I \cup \{v_i\}$ is not an independent set. Then, the set I is the maximal independent set. If $\forall I'$, $\text{card}(I) \geq \text{card}(I')$. Then the set I is the maximum independent set. The number of elements in I is the independent number of graph G noted as $\alpha(G)$ or α .

From the definitions, reserving the most original sampling points can be transformed into computing the maximum independent set; that is, delete the least vertices so that there is no edge between any two vertices. In graph theory, the simple graph can be expressed as an adjacent matrix $A_{n \times n} = (a_{ij})$, which shows the lines between vertices. If there is a line between vertex v_i and vertex v_j , $a_{ij} = 1$. Otherwise $a_{ij} = 0$. Obviously, the properties are given as follows:

- (a) symmetrical characteristic $A_{n \times n} = A_{n \times n}^T$;
- (b) $a_{ii} = 0, i = 1, 2, \dots, n$;
- (c) if $A_{n \times n} = \vec{0}$, the vertex set V is an independent set.

So the reservation problem is transformed into a problem to get the maximum independent set; that is, delete the least vertices so that the corresponding adjacent matrix of remaining vertices is a zero matrix. Suppose that there is a LHS of size n , the sample matrix is $X_{n \times r}^{(old)} = \{x_{ij}^{(old)}\}$, the order matrix is $L_{n \times r}^{(old)} = \{l_{ij}^{(old)}\}$ ($l_{ij}^{(old)}$ is the interval order number of $x_{ij}^{(old)}$ in the j th variable), the cumulative probability matrix is $U_{n \times r}^{(old)} = \{u_{ij}^{(old)}\}$ ($u_{ij}^{(old)}$ is the cumulative probability of $x_{ij}^{(old)}$, which is the Prob_{ji} in (1)), and the random matrix is $R_{n \times r}^{(old)} = \{r_{ij}^{(old)}\}$ ($r_{ij}^{(old)}$ is the random value of $x_{ij}^{(old)}$, which is the r_{ji} in (1)). The objective is to get ELHS of size $n + m$ noted

as $X_{(n+m) \times r}^{(\text{new})}$. The detailed steps of BGELHS algorithm are as follows.

Step 1. Compute the positions of original sampling points in ELHS structure.

- (a) Transform $L_{n \times r}^{(\text{old})}$ into the new sample structure and get the order matrix $L_{n \times r}^{(1)}$ as shown in the following:

$$l_{i,j}^{(1)} = \lfloor u_{i,j}^{(\text{old})} \cdot (n+m) \rfloor + 1, \quad (6)$$

where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, r$, and $\lfloor \cdot \rfloor$ is the floor function ($\text{floor}(x) = \lfloor x \rfloor$ is the largest integer not greater than x).

- (b) Transform $R_{n \times r}^{(1)}$ into the new sample structure and get the random matrix $R_{n \times r}^{(1)}$ as shown in the following:

$$r_{i,j}^{(1)} = u_{i,j}^{(\text{old})} \cdot (n+m) - (l_{i,j}^{(1)} - 1), \quad (7)$$

where $i = 1, 2, \dots, n$, $j = 1, 2, \dots, r$.

Step 2. Reserve the most original sampling points, noted as the sample matrix $X_{n' \times r}^{(2)}$.

- (a) Compute the adjacent matrix $A_{n \times n}$ of the original LHS based on $L_{n \times r}^{(1)}$ as shown in the following:

$$a_{ij} = \begin{cases} 1, & \text{If } \exists p \in [1 \ n], p \in Z, l_{ip}^{(1)} = l_{jp}^{(1)} \\ 0, & \text{If } \forall p \in [1 \ n], p \in Z, l_{ip}^{(1)} \neq l_{jp}^{(1)}. \end{cases} \quad (8)$$

- (b) Choose k sampling points of original LHS to delete, $k = 0, 1, 2, \dots, n$. Then we can get a combination matrix $B_{q \times n} = [B_1^T \ B_2^T \ \dots \ B_q^T]^T$, where $q = C_n^0 + C_n^1 + C_n^2 + \dots + C_n^n = 2^n$, B_i is a combination, and the number of elements in B_i is not more than that in B_j , $1 \leq i < j \leq q$ (e.g., let $r = 2$ and $n = 3$, $B_{4 \times 2} = [\text{null} \ 1 \ 2 \ 1]^T$, where null means there is no element). Let $p = 1$ and delete the point(s) corresponding with the element(s) of B_p ; then judge whether the remaining adjacent matrix A' is equal to $\vec{0}$. If $A' = \vec{0}$, record the order number of deleted point(s) as B'_p and the dimension number of A' is noted as n' , that is, the number of remaining points. Otherwise, set $p = p + 1$, delete the points corresponding with the elements of B_p , and repeat the Step 2-(b).

- (c) Delete the rows of $L_{n \times r}^{(1)}$, $R_{n \times r}^{(1)}$, $U_{n \times r}^{(\text{old})}$, and $X_{n \times r}^{(\text{old})}$ according to the element of B'_p . The remaining matrices are noted as $L_{n' \times r}^{(2)}$, $R_{n' \times r}^{(2)}$, $U_{n' \times r}^{(2)}$, and $X_{n' \times r}^{(2)}$.

Step 3. Generate the new sample matrix $X_{(n+m-n') \times r}^{(3)}$.

- (a) Generate a new order matrix $L'_{(n+m-n') \times r}$ of size $(n+m-n') \times r$ and every column of $L'_{(n+m-n') \times r}$ is a random permutation of the integer number from 1 to $n+m$.

- (b) For every column, delete the element that is the same as the corresponding column in $L_{n' \times r}^{(2)}$ and get the matrix $L_{(n+m-n') \times r}^{(3)}$.

- (c) Generate a random matrix $R_{(n+m-n') \times r}^{(3)}$ where r_{ij} is a uniform random number ranging from 0 to 1.

- (d) Compute the cumulative probability matrix $U_{(n+m-n') \times r}^{(3)}$ as shown in the following:

$$u_{i,j}^{(3)} = \frac{l_{i,j}^{(3)} - 1 + r_{i,j}^{(3)}}{n+m}. \quad (9)$$

- (e) Compute the sample matrix $X_{(n+m-n') \times r}^{(3)}$ as shown in (2).

Step 4. Get the order matrix $L_{(n+m) \times r}^{(\text{new})}$, the random matrix

$R_{(n+m) \times r}^{(\text{new})}$, the cumulative probability matrix $U_{(n+m) \times r}^{(\text{new})}$, and the sample matrix $X_{(n+m) \times r}^{(\text{new})}$ of ELHS:

$$\begin{aligned} L_{(n+m) \times r}^{(\text{new})} &= \begin{bmatrix} L_{n' \times r}^{(2)} \\ L_{(n+m-n') \times r}^{(3)} \end{bmatrix}, \\ R_{(n+m) \times r}^{(\text{new})} &= \begin{bmatrix} R_{n' \times r}^{(2)} \\ R_{(n+m-n') \times r}^{(3)} \end{bmatrix}, \\ U_{(n+m) \times r}^{(\text{new})} &= \begin{bmatrix} U_{n' \times r}^{(2)} \\ U_{(n+m-n') \times r}^{(3)} \end{bmatrix}, \\ X_{(n+m) \times r}^{(\text{new})} &= \begin{bmatrix} X_{n' \times r}^{(2)} \\ X_{(n+m-n') \times r}^{(3)} \end{bmatrix}. \end{aligned} \quad (10)$$

As taking (7) into (1) as (11), we can prove that the matrix $X_{(n+m) \times r}^{(\text{new})}$ is a sample matrix of LHS. Equation (11) shows that the remaining original sampling points meet the basic principle of LHS. As the new sampling points have the same feature, $X_{(n+m) \times r}^{(\text{new})}$ is a strict LHS:

$$\begin{aligned} u_{i,j}^{(1)} &= \frac{l_{i,j}^{(1)} - 1 + r_{i,j}^{(1)}}{n+m} \\ &= \frac{l_{i,j}^{(1)} - 1 + u_{i,j}^{(\text{old})} \cdot (n+m) - (l_{i,j}^{(1)} - 1)}{n+m} = u_{i,j}^{(\text{old})}. \end{aligned} \quad (11)$$

Example 1. BGELHS is illustrated by the original LHS in Figure 1. Let x_1 have a uniform distribution on $[0, 10]$ and let

x_2 have a triangular distribution on $[5, 10]$ with mode at 7.5. The distribution function $F(\cdot)$ is

$$F_1(x_1) = 0.1 \times x_1,$$

$$F_2(x_2) = \begin{cases} \frac{(x_2 - 5)^2}{12.5}, & 5 \leq x_2 \leq 7.5 \\ 1 - \frac{(10 - x_2)^2}{12.5}, & 7.5 < x_2 \leq 10. \end{cases} \quad (12)$$

The inverse of the distribution function $F(\cdot)$ is

$$x_1 = 10 \times F_1(x_1),$$

$$x_2 = \begin{cases} 5 + \sqrt{12.5 \times F_2(x_2)}, & 0 \leq F_2(x_2) \leq 0.5 \\ 10 - \sqrt{12.5 \times (1 - F_2(x_2))}, & 0.5 < F_2(x_2) \leq 1. \end{cases} \quad (13)$$

The sample matrix and related matrices are

$$X_{5 \times 2}^{(old)} = \begin{bmatrix} 1.29 & 6.12 \\ 3.57 & 8.46 \\ 5.86 & 7.60 \\ 6.57 & 8.30 \\ 9.00 & 6.90 \end{bmatrix},$$

$$L_{5 \times 2}^{(old)} = \begin{bmatrix} 1 & 1 \\ 2 & 5 \\ 3 & 3 \\ 4 & 4 \\ 5 & 2 \end{bmatrix}, \quad (14)$$

$$R_{5 \times 2}^{(old)} = \begin{bmatrix} 0.645 & 0.500 \\ 0.785 & 0.070 \\ 0.930 & 0.710 \\ 0.285 & 0.860 \\ 0.500 & 0.430 \end{bmatrix},$$

$$U_{5 \times 2}^{(old)} = \begin{bmatrix} 0.129 & 0.100 \\ 0.357 & 0.814 \\ 0.586 & 0.542 \\ 0.657 & 0.772 \\ 0.900 & 0.286 \end{bmatrix}.$$

Then, the objective is to get ELHS of size 7, which contains the original sampling points as many as possible. So $n = 5$, $m = 2$.

From Step 1, the distribution of original sampling points in the new sample is

$$L_{5 \times 2}^{(1)} = \left[U_{5 \times 2}^{(old)} \times (5 + 2) \right] + I_{5 \times 2}$$

$$= \begin{bmatrix} 0.903 & 0.700 \\ 2.499 & 5.698 \\ 4.102 & 3.794 \\ 4.599 & 5.404 \\ 6.300 & 2.002 \end{bmatrix} + I_{5 \times 2} = \begin{bmatrix} 0 & 0 \\ 2 & 5 \\ 4 & 3 \\ 4 & 5 \\ 6 & 2 \end{bmatrix} + I_{5 \times 2}$$

$$= \begin{bmatrix} 1 & 1 \\ 3 & 6 \\ 5 & 4 \\ 5 & 6 \\ 7 & 3 \end{bmatrix}, \quad (15)$$

$$R_{5 \times 2}^{(1)} = U_{5 \times 2}^{(old)} \cdot (5 + 2) - (L_{5 \times 2}^{(1)} - I_{5 \times 2})$$

$$= \begin{bmatrix} 0.903 & 0.700 \\ 0.499 & 0.698 \\ 0.102 & 0.794 \\ 0.599 & 0.404 \\ 0.300 & 0.002 \end{bmatrix}.$$

From Step 2-(a), the adjacent matrix is computed:

$$A_{5 \times 5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (16)$$

From Step 2-(b), all the combinations are noted as B :

$$B = \begin{bmatrix} \text{null} & \text{null} & \text{null} & \text{null} & \text{null} \\ 1 & \text{null} & \text{null} & \text{null} & \text{null} \\ 2 & \text{null} & \text{null} & \text{null} & \text{null} \\ 3 & \text{null} & \text{null} & \text{null} & \text{null} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2 & 3 & 4 & 5 & \text{null} \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}_{32 \times 5}. \quad (17)$$

Delete the point(s) corresponding with the element(s) of B_p , $p = 1, 2, \dots, 32$. For the fifth iteration, $p = 5$ and $B_5 = [4 \text{ null null null null}]$. Delete the fourth point (i.e., delete the fourth row and fourth column of $A_{5 \times 5}$) and the

new adjacent matrix is $A'_{4 \times 4} = \vec{0}$. Then, $B'_4 = [1 \ 2 \ 3 \ 5]$ and $n' = 4$. From Step 2-(c), the addition sample is computed:

$$\begin{aligned}
 L_{4 \times 2}^{(2)} &= \begin{bmatrix} 1 & 1 \\ 3 & 6 \\ 5 & 4 \\ 7 & 3 \end{bmatrix}, \\
 R_{4 \times 2}^{(2)} &= \begin{bmatrix} 0.903 & 0.700 \\ 0.499 & 0.698 \\ 0.102 & 0.794 \\ 0.300 & 0.002 \end{bmatrix}, \\
 U_{4 \times 2}^{(2)} &= \begin{bmatrix} 0.129 & 0.100 \\ 0.357 & 0.814 \\ 0.586 & 0.542 \\ 0.900 & 0.286 \end{bmatrix}, \\
 X_{4 \times 2}^{(2)} &= \begin{bmatrix} 1.29 & 6.12 \\ 3.57 & 8.46 \\ 5.86 & 7.60 \\ 9.00 & 6.90 \end{bmatrix}.
 \end{aligned} \tag{18}$$

From Step 3, an order matrix is original:

$$L'_{7 \times 2} = \begin{bmatrix} 6 & 3 & 7 & 5 & 1 & 2 & 4 \\ 4 & 7 & 1 & 6 & 2 & 3 & 5 \end{bmatrix}^T. \tag{19}$$

Then

$$\begin{aligned}
 L_{3 \times 2}^{(3)} &= \begin{bmatrix} 6 & 7 \\ 2 & 2 \\ 4 & 5 \end{bmatrix}, \\
 R_{3 \times 2}^{(3)} &= \begin{bmatrix} 0.800 & 0.916 \\ 0.679 & 0.734 \\ 0.743 & 0.392 \end{bmatrix}, \\
 U_{3 \times 2}^{(3)} &= \frac{L_{3 \times 2}^{(3)} - \vec{I} + R_{3 \times 2}^{(3)}}{7} = \begin{bmatrix} 0.829 & 0.988 \\ 0.240 & 0.248 \\ 0.535 & 0.628 \end{bmatrix}, \\
 X_{3 \times 2}^{(3)} &= \begin{bmatrix} 8.29 & 9.61 \\ 2.40 & 6.76 \\ 5.35 & 7.84 \end{bmatrix}.
 \end{aligned} \tag{20}$$

FIGURE 4: ELHS of size 7.

Finally, the ELHS of size 7 is obtained, which is shown in Figure 4:

$$\begin{aligned}
 X_{7 \times 2}^{(\text{new})} &= \begin{bmatrix} X_{4 \times 2}^{(2)} \\ X_{3 \times 2}^{(3)} \end{bmatrix} \\
 &= \begin{bmatrix} 1.29 & 3.57 & 5.86 & 9.00 & 8.29 & 2.40 & 5.35 \\ 6.12 & 8.46 & 7.60 & 6.90 & 9.61 & 6.76 & 7.84 \end{bmatrix}^T.
 \end{aligned} \tag{21}$$

3.2. General Extension Algorithm of LHS Based on Greedy Algorithm. In graph theory, obtaining the maximum independent set is a NP-hard problem. Step 2-(b) of BGELHS needs the most time and the time complexity of BGELHS is $O(2^n)$. It means that, with the increase of the number of sampling points, the time of BGELHS will be increased exponential. In this section, GGELHS is proposed to reduce the time of extension algorithm. First of all, the definition of degree is introduced.

Definition D (see [17]). In a simple graph $G = (V, E)$, the number of edges associated with the vertex v is the degree of v noted as $d(v)$.

$d(v)$ indicates the number of vertices linked with the vertex v . The basic idea of greedy algorithm is to delete a point one time whose $d(v)$ is the maximum until $d(v)$ of any remaining vertex is 0. The steps of GGELHS are the same as those of BGELHS except for Step 2-(b). The details of Step 2-(b) are given as follows.

Step 2-(b): compute the degree of every vertex $d(v_i) = \sum_{j=1}^n a_{ij}$, $j = 1, 2, \dots, n$. Delete the vertex v_i whose degree is the maximum. Delete the corresponding row and column in the adjacent matrix A ($A_{n \times n}$ at the beginning) and note the remaining matrix as A' . Record the order of v_i in B_p . If

FIGURE 5: Change of standard deviation σ ((a) function 1, (b) function 2).

$A' \neq \vec{0}$, $A = A'$, and repeat Step 2-(b) to reconstruct A' . Otherwise, the element of B_p is the order of points to delete. The dimension of A' is noted as n' and the order of points to remain is recorded in B'_p .

In Step 2-(a) of GGELHS, when computing the adjacent matrix $A_{n \times n}$, the time required is $n(n-1)/2$. It is easy to know that the time complexities of other steps are not greater than $O(n)$. So the time complexity of GGELHS is $O(n^2)$, which is much less than BGELHS. But this algorithm gets the maximal independent set rather than the maximum independent set; that is, this algorithm cannot guarantee to reserve the most original sampling points.

Example 2. To compare BGELHS and GGELHS, we consider the same question of Example 1. Since GGELHS is almost the same as BGELHS except for Step 2-(b), we mainly give Step 2-(b) of GGELHS.

From Step 2-(b), the degrees of five points are $D_5 = [0 \ 1 \ 1 \ 2 \ 0]^T$. The fourth point has the maximum degree. So delete the fourth point, and the adjacent matrix is $A'_{4 \times 4} = \vec{0}$. So, $B'_p = [1 \ 2 \ 3 \ 5]$, $n' = 4$.

BGELHS run five times in Step 2-(b) and GGELHS run only one time. As computing the degree of original sampling points costs a little, GGELHS is less time consuming and more practical. From the result above, we can know that Step 2-(b) gets the maximum independent set in this example, which means that this algorithm deletes the least original sampling points. But the greedy algorithm cannot guarantee to have the same effect for every circumstance.

4. Application

BGELHS and GGELHS can be applied for many other applications, such as adaptive metamodel building and sequential experiments design and analysis. In Section 4, pick up two test examples to evaluate the sample means [6]. The functions

are shown in (22) where $x_i \in [0, 1]$, $i = 1, 2, 3, 4$, and x_i obeys uniform distribution:

$$\begin{aligned} y_1 &= \sum_{i=1}^4 x_i + \sum_{i=1}^4 \sum_{j=i+1}^4 x_i x_j, \\ y_2 &= e^{(\sum_{i=1}^4 x_i^2)} + \sum_{i=1}^4 \sin(16\pi x_i). \end{aligned} \quad (22)$$

Four algorithms are applied including Monte Carlo Sample (MCS), classical LHS (CLHS), BGELHS, and GGELHS. Set the maximum sample size noted as $N_{\max}$. The steps are as follows.

- Pick M samples (MCS or LHS) of size N .
- Compute the output Y_{ij} of samples, $i = 1, 2, \dots, M$, $j = 1, 2, \dots, N$.
- Compute the average of every sample outputs $\bar{Y}_i$ as shown in the following:

$$\bar{Y}_i = \frac{\sum_{j=1}^N Y_{ij}}{N}, \quad i = 1, 2, \dots, M. \quad (23)$$

- Compute the standard deviation σ of $\bar{Y}_i$ as shown in the following:

$$\sigma = \sqrt{\frac{\sum_{i=1}^M (\bar{Y}_i - \bar{Y})^2}{(M-1)}}, \quad (24)$$

where $\bar{Y}$ is the mean of $\bar{Y}_i$, $i = 1, 2, \dots, M$.

- Let $N = N + P$. If $N > N_{\max}$, stop. Otherwise, generate sample by the five algorithms and go to step b.

In this application, $M = 10$, $N = 10$, $N_{\max} = 100$, and $P = 10$ for MCS, CLHS, BGELHS, and GGELHS. Figure 5

FIGURE 6: Number of runs ((a) function 1, (b) function 2).

FIGURE 7: Time cost ((a) function 1, (b) function 2).

is the result of these four algorithms. It can be concluded that the convergence rates of two extension algorithms are the same as CLHS and better than MCS. Therefore, the extension algorithms reserve the good convergence of CLHS.

Compared with CLHS, an outstanding advantage of extension algorithms is the less number of runs. Figure 6 shows the number of function runs in this application. The run number of CLHS is $(10 + 20 + 30 + \dots + 100) \times 10 = 5500$. BGELHS saves 3302 runs for function 1 and saves 3325 runs for function 2. GGELHS saves 3263 runs for function 1 and saves 3259 runs for function 2. Therefore, BGELHS and GGELHS save more than half of the runs compared with CLHS.

The total times of these extension algorithms are shown in Figure 7. The times of MCS and CLHS are almost the same. The time of GGELHS is one and two orders of magnitude slower than that of MCS and CLHS because of Step 2. BGELHS cost the most time, which is a serious problem when applied. GGELHS is easy to apply in practice.

5. Conclusion

A drawback to LHS is the extension problem and this paper mainly discusses how to obtain a new strict LHS that contains the original sampling points as many as possible. There are three problems to solve including how to express the relationship of original sampling points in the ELHS structure, how to reserve the most original sampling points and how to generate the new sampling points. In this paper, BGELHS and GGELHS are proposed to solve these problems. BGELHS can reserve the most original points in theory. However, it is difficult to apply in practice because of the problem of time consuming. GGELHS cannot guarantee to reserve the most original points, but the time complexity is much less than that of BGELHS. Therefore, BGELHS is suitable for the analysis of time consuming system when the running time of system is much more valuable than the generation time of ELHS. GGELHS can be applied for more situations. A simple example and application for the

evaluation of sample means show the effectiveness of these algorithms.

This paper illustrates the relationship of ELHS and graph theory. Some conclusions of graph theory can be applied for the general extension of LHS directly, which is the main contribution of this paper. The relationship of original sampling points in ELHS is expressed in the form of adjacent matrix, but the reverse does not hold. The reason is that some information of original sampling points is lost including the dimensions of variables, the value of sampling points, and so on. Therefore, we can improve BGELHS based on the lost information and some conclusions of graph theory, which is under investigation. On the other hand, LHS is often applied with some criteria for a better distribute or space-filling, such as correlated variables, maximizing minimum distance, and minimizing discrepancy. The general extension of LHS with these criteria is also the future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the editor and two referees for helpful comments. This research is supported by the National Natural Science Foundation of China (Grant no. 61403097) and the “Fundamental Research Funds for the Central Universities” (Grant no. HIT. NSRIF. 2015035).

References

- [1] B. G. M. Husslage, G. Rennen, E. R. van Dam, and D. den Hertog, “Space-filling Latin hypercube designs for computer experiments,” *Optimization and Engineering*, vol. 12, no. 4, pp. 611–630, 2011.
- [2] J. C. Helton, F. J. Davis, and J. D. Johnson, “A comparison of uncertainty and sensitivity analysis results obtained with random and Latin hypercube sampling,” *Reliability Engineering and System Safety*, vol. 89, no. 3, pp. 305–330, 2005.
- [3] G. G. Wang and S. Shan, “Review of metamodeling techniques in support of engineering design optimization,” *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370–380, 2007.
- [4] A. Olsson, G. Sandberg, and O. Dahlblom, “On Latin hypercube sampling for structural reliability analysis,” *Structural Safety*, vol. 25, no. 1, pp. 47–68, 2003.
- [5] H. Yu, C. Chung, K. Wong, and J. Zhang, “A probabilistic load flow calculation method with latin hypercube sampling,” *Automation of Electric Power Systems*, vol. 33, no. 21, pp. 32–81, 2009.
- [6] C. Tong, “Refinement strategies for stratified sampling methods,” *Reliability Engineering and System Safety*, vol. 91, no. 10–11, pp. 1257–1265, 2006.
- [7] C. J. Sallaberry, J. C. Helton, and S. C. Hora, “Extension of Latin hypercube samples with correlated variables,” *Reliability Engineering and System Safety*, vol. 93, no. 7, pp. 1047–1059, 2008.
- [8] P. Z. G. Qian, “Nested Latin hypercube designs,” *Biometrika*, vol. 96, no. 4, pp. 957–970, 2009.
- [9] X. He and P. Z. Qian, “Nested orthogonal array-based Latin hypercube designs,” *Biometrika*, vol. 98, no. 3, pp. 721–731, 2011.
- [10] D. Williamson, “Exploratory ensemble designs for environmental models using k-extended Latin Hypercubes,” *Environmetrics*, vol. 26, no. 4, pp. 268–283, 2015.
- [11] M. Vorechovsky, “Hierarchical subset latin hypercube sampling,” in *Pravdepodobnost porusovani konstrukci (PPK '06)*, pp. 285–298, Brno, Czech Republic, 2006.
- [12] M. Vorechovsky, “Extension of sample size in Latin hypercube sampling—methodology and software,” in *Proceedings of the 3rd Symposium on Life-Cycle and Sustainability of Civil Infrastructure Systems*, pp. 2403–2410, Vienna, Austria, 2012.
- [13] M. Vorechovsky, “Hierarchical refinement of Latin hypercube samples,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 5, pp. 394–411, 2015.
- [14] G. G. Wang, “Adaptive response surface method using inherited Latin hypercube design points,” *Transactions of the ASME—Journal of Mechanical Design*, vol. 125, no. 2, pp. 210–220, 2003.
- [15] G. Blatman and B. Sudret, “An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis,” *Probabilistic Engineering Mechanics*, vol. 25, no. 2, pp. 183–197, 2010.
- [16] X. Wei, *Research of global optimization algorithm based on metamodel [Doctoral Dissertation]*, Huazhong University of Science and Technology, Wuhan, China, 2012.
- [17] D. B. West, *Introduction to Graph Theory*, China Machine Press, 2006.