A GENERATION ALGORITHM OF LATIN HYPERCUBE SAMPLING BASED ON ORIGINAL SAMPLING POINTS

Zhizhao LIU [(a)], Wei LI [(b)], Ming YANG [(c)], Jun XU [(d)], Duzheng QING [(e)]

[(a), (b), (c)]Control and Simulation Center, Harbin Institute of Technology, Harbin 150080, China
[(d), (e)]Science and Technology on Special System Simulation Laboratory, Beijing Simulation Center, Beijing 100854, China

[(a)]liuzhizhao2007@hotmail.com, [(b)]fleehit@163.com, [(c)]myang@hit.edu.cn
[(d)]xujun711@sina.com, [(e)]qing_dz@163.com

ABSTRACT

In order to utilize original sampling points as many as possible to construct a Latin Hypercube Sampling (LHS), a generation algorithm of LHS is proposed. The original sampling can be an arbitrary sampling and the new sampling is a LHS strictly. As the high structure of LHS, only part of original sampling points can be used for LHS. There are several problems including how to select the reserved original sampling points and generate the new ones if necessary. Matrices are used to describe the original sampling points and the original sampling points are reserved based on an adjacent matrix. The new sampling points are generated by a subtraction rule. Numerical experiments show that the generation algorithm is effective and saves a lot of running time for simulation systems.

Keywords: Latin Hypercube Sampling; generation algorithm; adjacent matrix; subtraction rule.

1. INTRODUCTION

Design and analysis of experiments is an important method of science and technology. For simulation systems, we need to apply different experimental design methods for different analysis goals such as sensitivity analysis (Borgonovo Castaings Tarantola 2005, Storlie Swiler Helton and et al. 2009), variance analysis (Huang Lo and Lu 2013), uncertainty analysis (Zouaoui and Wilson 2003, Helton Davis and Johnson 2005, Braton Nelson and Wei 2013) and so on. All these procedures need sampling, i.e., running simulation system. However, simulation experiments may be time-consuming to execute. There is a popular problem that one analysis cannot apply the sampling points of other analyses, which wastes a lot of original sampling points and needs too long time to run simulation system. For example, factorial design is used for sensitivity analysis at first and LHS is needed for uncertainty analysis. A new sample means a lot of simulation runs and LHS based on original points could reduce the running time. So this paper discusses how to utilize original sampling points for a new analysis.

In 1979, Mckay proposed a stratified sampling algorithm named Latin Hypercube Sampling (LHS) (Mckay Beckman and Conover 1979). It is one of the most popular sampling methods, which has some advantages such as space filling effect, robustness and good convergence character. There have been some researches about the extension of LHS. They start with a LHS and generate a new LHS of a larger size that remains the original sampling points. Some multiple extension algorithms were proposed during the past decade (Tong 2006, Sallaberry Helton and Hora 2008, Wu and Chen 2011). Under this circumstance, the new LHS can contain all the original LHS points. There were two general extension algorithms in 2003 and 2012 (Wang 2003, Wei 2012). But the general extension algorithm of LHS achieve a similar LHS finally, not a strict one. Therefore, recent researches focus on the generation algorithm of LHS based on a special sample, i.e. LHS. In this paper, we consider a general generation algorithm that achieves a strict LHS and the original sample can be any sample, such as a Monte Carlo sampling, a factorial design, an orthogonal array and so on. This algorithm can be applied for more circumstances, including normal analysis based on original sampling points and sequential simulation experiments.

The remainder of this paper is organized as follows. In section 2, the generation problem and the generation algorithm of LHS are described. The locations of original sampling points in LHS are expressed by several matrices and the relationship of original sampling points in LHS is expressed by an adjacent matrix. Section 3 gives two applications of generation algorithm, Section 3.1 for illustration of algorithm and Section 3.2 for metamodeling. Section 4 gets the conclusion and the future research.

2. LHS AND GENERATION ALGORITHM

Before discussing the generation problem, the procedure of classical LHS is described in Section 2.1. The generation problem and correlation studies are in Section 2.2. Then, we give the steps of generation algorithm of LHS based on original sampling points in Section 2.3.

2.1 Procedure of LHS

The input variables are noted as $x_1, x_2, ..., x_r$ and the range of x_i is $u_i = [\alpha_i \ \beta_i]$, $i = 1, 2, \ldots, r$. Then, LHS can be got as the steps below, which can ensure that each input variable has all intervals among its range.

Step 1. divide the range u_i into n equiprobable intervals $u_{i1}, u_{i2}, \ldots, u_{in}$, $i = 1, 2, \ldots, r$. So the intervals satisfy $u_{i1} \cup u_{i2} \ldots \cup u_{in} = u_i$, $u_{ij} \cap u_{ik} = \varnothing$ and $P(x \in u_{ij}) = 1/n$, where $j, k = 1, 2, \ldots, n$。

Step 2. for the j th interval of variable x_i, the cumulative probability can be obtained as:

$$Prob_{ji} = (j-1)/n + r_{ji}/n \tag{1}$$

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

Step 3. transform the probability into the sample value x_{ji} by the inverse of the distribution function $F(\cdot)$:

$$x_{ji} = F_i^{-1}(Prob_{ji}) \tag{2}$$

Then, the sample matrix is

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

Step 4. the n values of each variables 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:

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

where each row is a sample point.

2.2 Research on key problems

Assume that there is an original sampling noted as $A = \{\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_m\}$, where $\vec{a}_i$ is a sampling point, $i = 1, 2, ..., m$. The objective is to get LHS of size n noted as $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\}$, where $n \geq 1$ and n is an integer. B satisfies the constraint condition as shown in Eq. (3).

$$\max \operatorname{card}(A \cap B) \tag{3}$$

where $\operatorname{card}(\cdot)$ denotes the number of elements in set. It means that the new sampling is a LHS strictly and maintains the most original sampling points.

The first problem is how to indicate the location of original sampling points in the expected LHS, which is the base of selecting original points and adding new ones. In this paper, we use some matrices including sampling matrix $X_{m \times r}$, cumulative probability matrix $U_{m \times r}$ (Every element of $U_{m \times r}$ is the cumulative probability of the corresponding element of $X_{m \times r}$, that is the $Prob_{ji}$ in Eq. (1)), order matrix $L_{m \times r}$ (Every element of $L_{m \times r}$ is the interval number of the corresponding element of $X_{m \times r}$), random matrix $R_{m \times r}$ (Every element of $R_{m \times r}$ is the random location among interval of the corresponding element of $X_{m \times r}$).

Another problem is how to remain the most original points or delete the least ones. It is difficult because the relation of original sampling points is hard to express. We use an adjacent matrix $A_{m \times m}$ to solve the problem. The element a_{ij} of $A_{m \times m}$ notes whether a variable value of the i th original point is in the same interval as the one of the j th original point, 1 for the existence and 0 for the other circumstance. Therefore, how to select original points is based on $A_{m \times m}$. If $A_{m' \times m'} = \vec{0}$, there is no need to delete any remaining sampling points.

The finally problem is how to add new sampling points if the remaining original ones are not enough. In this paper, we follow the subtraction rule (Wu and Chen 2011). At the beginning, we generate a LHS randomly. And then, delete the interval number which is the same as remaining original sampling points. The rest of interval number is randomly combined and the location among the corresponding interval is also achieved randomly, which can ensure that the remaining points and the new points can construct a LHS that meets the requirement.

2.3 Generation algorithm

In Section 2.2, we solve the main problems of generation algorithm. This section mainly shows the step of generation algorithm of LHS based on original sampling points.

Step 1. Compute the location of original sample points in LHS.

a. Compute the cumulative probability matrix $U_{m \times r}^{(1)}$ and the order matrix $L_{m \times r}^{(1)}$ as shown in Eq. (4) and Eq. (5).

$$u_{i,j}^{(1)} = \frac{x_{i,j}^{(old)} - \beta_j}{\alpha_j - \beta_j} \tag{4}$$

$$l_{i,j}^{(1)} = \begin{cases} 1 & , \ x_{i,j}^{(old)} = \beta_j \\ \left\lceil \dfrac{x_{i,j}^{(old)} - \beta_j}{\alpha_j - \beta_j} \cdot n \right\rceil, & \beta_j < x_{i,j}^{(old)} \leq \alpha_j \end{cases} \tag{5}$$

where $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, r$, $\lceil \cdot \rceil$ is a bracket function to get the maximum integer number which is more than the element inside.

b. Get the random matrix $R_{m \times r}^{(1)}$ as shown in Eq. (6).

$$r_{i,j}^{(1)} = u_{i,j}^{(1)} \cdot n - l_{i,j}^{(1)} + 1 \qquad (6)$$

where $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, r$ 。

Step 2. Get the maximum remaining sample matrix $X_{m'\times r}^{(2)}$ of the original Sampling.

a. Get the adjacent matrix $A_{m\times m}$ of the original sampling based on $L_{m\times r}^{(1)}$ as shown in the equation bellow. If $A_{m\times m} = \vec{0}$, there is no need to delete more sample points and go to *step* 3. Otherwise, go to *step* 2-b.

$$a_{ij} = \begin{cases} 1, & \exists q \in [1\ \mathrm{m}], q \in Z, l_{iq}^{(1)} = l_{jq}^{(1)} \\ 0, & \text{otherwise} \end{cases}$$

b. Get all the combinations of the original points, which is noted as $D = [D_1 \quad D_2 \quad \cdots \quad D_q]^{\mathrm{T}}$ where every row of D is a rank (If the original point is deleted, the element value is 1. Otherwise, the element value is 0) and the number of 0 in B_i is not less than B_j where $1 \le i < j \le m$. Set $p = 1$ and delete the point(s) corresponded with the element(s) of D_p , then judge whether the remain adjacent matrix A' is equal to $\vec{0}$. If $A' = \vec{0}$, record the order of delete point(s) as D'_p that is the non-zero element of D_p and the dimension number of A' is noted as m' which is the number of remain points. Otherwise, set $p = p + 1$, delete the points corresponded with the elements of D_p and follow the deleting procedure of *step* 2-b.

c. Delete the rows of $L_{m\times r}^{(1)}$, $R_{m\times r}^{(1)}$, $U_{m\times r}^{(old)}$ and $X_{m\times r}^{(old)}$ according to the element of D'_p . The remain matrices are noted as $L_{(m-m')\times r}^{(2)}$, $R_{(m-m')\times r}^{(2)}$, $U_{(m-m')\times r}^{(2)}$ and $X_{(m-m')\times r}^{(2)}$. If $m' \ge n$, select n sampling points randomly note as $L_{n\times r}^{(new)}$, $R_{n\times r}^{(new)}$, $U_{n\times r}^{(new)}$ and $X_{n\times r}^{(new)}$. $X_{n\times r}^{(new)}$ is the sampling matrix of LHS. If $m' < n$, go to *step* 3.

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

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

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

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

d. Compute the cumulative probability matrix $U_{(n-m')\times r}^{(3)}$ as shown in Eq. (7).

$$u_{i,j}^{(3)} = \frac{l_{i,j}^{(3)} - 1 + r_{i,j}^{(3)}}{n} \qquad (7)$$

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

Step 4. Get the order matrix $L_{n\times r}^{(new)}$, the random matrix $R_{n\times r}^{(new)}$, the cumulative probability matrix $U_{n\times r}^{(new)}$ and the sample matrix $X_{n\times r}^{(new)}$ of LHS.

$$L_{n\times r}^{(new)} = \begin{bmatrix} L_{m'\times r}^{(2)} \\ L_{(n-m')\times r}^{(3)} \end{bmatrix},\ R_{n\times r}^{(new)} = \begin{bmatrix} R_{m'\times r}^{(2)} \\ R_{(n-m')\times r}^{(3)} \end{bmatrix},$$

$$U_{n\times r}^{(new)} = \begin{bmatrix} U_{m'\times r}^{(2)} \\ U_{(n-m')\times r}^{(3)} \end{bmatrix},\ X_{n\times r}^{(new)} = \begin{bmatrix} X_{m'\times r}^{(2)} \\ X_{(n-m')\times r}^{(3)} \end{bmatrix}$$

It is easy to be proved that the matrix $X_{n\times r}^{(new)}$ is a sample matrix of LHS.

3. NUMERICAL EXPERIMENTS AND RESULTS

The generation algorithm of LHS based on original sampling points is illustrated by a simple example in Section 3.1. The effectiveness is demonstrated by an application in Section 3.2.

3.1 Illustration of algorithm

Assume that we have a Monte Carlo sampling of size 7 and there are two variables noted as x_1 and x_2 . The sample matrix is

$$X_{7\times 2}^{(old)} = \begin{bmatrix} 18.5813 & 33.9815 \\ 14.1254 & 37.8181 \\ 15.4463 & 39.1858 \\ 13.3452 & 30.9443 \\ 16.4734 & 22.7725 \\ 13.5714 & 22.9859 \\ 15.0299 & 25.1502 \end{bmatrix}$$

x_1 has a triangular distribution on [10, 20] with mode at 15 and x_2 has a uniform distribution on [20, 40]. The distribution function $F(\cdot)$ is

$$F_1(x_1) = \begin{cases} \dfrac{(x_1 - 10)^2}{50} & ,10 \le x_1 \le 15 \\ 1 - \dfrac{(20 - x_1)^2}{50} & ,15 < x_1 \le 20 \end{cases}$$

$$F_2(x_2) = 0.05 \times (x_2 - 20)$$

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

$$x_1 = \begin{cases} 10 + \sqrt{50 \times F_1(x_1)} & ,0 \le F_1(x_1) \le 0.5 \\ 20 - \sqrt{50 \times (1 - F_1(x_1))} & ,0.5 < F_1(x_1) \le 1 \end{cases}$$

$$x_2 = 20 \times F_2(x_2) + 20$$

The objective is to get a LHS of size 5. The steps are as follows.

Step 1. Compute the distribution of original sample points in LHS.

a. Compute cumulative probability matrix $U_{m\times r}^{(1)}$ and the order matrix $L_{m\times r}^{(1)}$ as shown in Eq. (4) and Eq. (5). Fig. 1 shows the location of original sampling.

$$U_{7\times 2}^{(1)}=\begin{bmatrix}0.9597 & 0.6991\\ 0.3404 & 0.8909\\ 0.5853 & 0.9593\\ 0.2238 & 0.5472\\ 0.7513 & 0.1386\\ 0.2551 & 0.1493\\ 0.5060 & 0.2575\end{bmatrix},\ L_{7\times 2}^{(1)}=\begin{bmatrix}5 & 4\\ 2 & 5\\ 3 & 5\\ 2 & 3\\ 4 & 1\\ 2 & 1\\ 3 & 2\end{bmatrix}$$

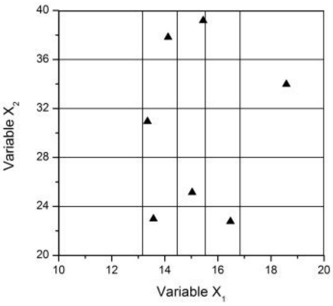


Fig. 1 Distribution of original sampling.

b. Get the random matrix $R_{m\times r}^{(1)}$ as shown in Eq. (6).

$$R_{7\times 2}^{(1)}=\begin{bmatrix}0.7987 & 0.4954\\ 0.7019 & 0.4545\\ 0.9263 & 0.7965\\ 0.1191 & 0.7361\\ 0.7563 & 0.6931\\ 0.2755 & 0.7465\\ 0.5298 & 0.2875\end{bmatrix}$$

Step 2. Get the maximum sample matrix $X_{m'\times r}^{(2)}$ of the original Sampling.

a. Compute the adjacent matrix $A_{7\times 7}$.

$$A_{7\times 7}=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}$$

b. Get D.

$$D=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}$$

Set $p=1$ and delete the first sampling point. Then,

$$A'=\begin{bmatrix}0 & 1 & 1 & 0 & 1 & 0\\ 1 & 0 & 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}\neq\vec{0}$$

Set $p=p+1=2$, delete the second sampling point. Then,

$$A'=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}\neq\vec{0}$$

Repeat the *step* 2-b. When $p=46$, $A'=\vec{0}$. At this time, $D_p=$[0 1 1 0 0 1 0], D_p' =[2 3 6] and m' =4.

c. Get the remain matrices are noted as $L_{4\times 2}^{(2)}$, $R_{4\times 2}^{(2)}$, $U_{4\times 2}^{(2)}$ and $X_{4\times 2}^{(2)}$.

$$L_{4\times 2}^{(2)}=\begin{bmatrix}5 & 4\\ 2 & 3\\ 4 & 1\\ 3 & 2\end{bmatrix},\ R_{4\times 2}^{(2)}=\begin{bmatrix}0.7987 & 0.4954\\ 0.1191 & 0.7361\\ 0.7563 & 0.6931\\ 0.5298 & 0.2875\end{bmatrix},$$

$$U_{4\times 2}^{(2)}=\begin{bmatrix}0.9597 & 0.6991\\ 0.2238 & 0.5472\\ 0.7513 & 0.1386\\ 0.5060 & 0.2575\end{bmatrix},$$

$$X_{4\times 2}^{(2)}=\begin{bmatrix}18.5813 & 33.9815\\ 13.3452 & 30.9443\\ 16.4734 & 22.7725\\ 15.0299 & 25.1502\end{bmatrix}$$

$m'=4<n=5$, so go to *step* 3.

Step 3. Generate the new sample matrix $X_{1\times 2}^{(3)}$.

a. Generate a new order matrix $L_{5\times 2}'$.

$$L_{5\times 2}'=\begin{bmatrix}4 & 1\\ 5 & 2\\ 3 & 3\\ 2 & 5\\ 1 & 4\end{bmatrix}$$

b. Get the matrix $L_{1\times 2}^{(3)}$.

$L_{1\times 2}^{(3)}$ =[1 5]

c. Generate a random matrix $R_{1\times 2}^{(3)}$.

$R_{1\times 2}^{(3)}$ =[0.3507 0.9390]

d. Compute the cumulative probability matrix $U_{1\times2}^{(3)}$ as shown in Eq. (7).

$$U_{1\times2}^{(3)} = [0.0701\ 0.9878]$$

e. Compute the sample matrix $X_{1\times2}^{(3)}$ as shown in Eq. (2).

$$X_{1\times2}^{(3)} = [11.8722\ 39.7560]$$

Step 4. Get the order matrix $L_{5\times2}^{(new)}$, the random matrix $R_{5\times2}^{(new)}$, the cumulative probability matrix $U_{5\times2}^{(new)}$ and the sample matrix $X_{5\times2}^{(new)}$ of LHS. The LHS is shown as Fig. 2.

$$L_{5\times2}^{(new)} = \begin{bmatrix} L_{4\times2}^{(2)} \\ L_{1\times2}^{(3)} \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 2 & 3 \\ 4 & 1 \\ 3 & 2 \\ 1 & 5 \end{bmatrix},$$

$$R_{5\times2}^{(new)} = \begin{bmatrix} R_{4\times2}^{(2)} \\ R_{1\times2}^{(3)} \end{bmatrix} = \begin{bmatrix} 0.7987 & 0.4954 \\ 0.1191 & 0.7361 \\ 0.7563 & 0.6931 \\ 0.5298 & 0.2875 \\ 0.3507 & 0.9390 \end{bmatrix},$$

$$U_{5\times2}^{(new)} = \begin{bmatrix} U_{4\times2}^{(2)} \\ U_{1\times2}^{(3)} \end{bmatrix} = \begin{bmatrix} 0.9597 & 0.6991 \\ 0.2238 & 0.5472 \\ 0.7513 & 0.1386 \\ 0.5060 & 0.2575 \\ 0.0701 & 0.9878 \end{bmatrix},$$

$$X_{5\times2}^{(new)} = \begin{bmatrix} X_{4\times2}^{(2)} \\ X_{1\times2}^{(3)} \end{bmatrix} = \begin{bmatrix} 18.5813 & 33.9815 \\ 13.3452 & 30.9443 \\ 16.4734 & 22.7725 \\ 15.0299 & 25.1502 \\ 11.8722 & 39.7560 \end{bmatrix}$$

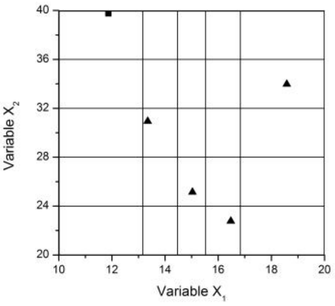


Fig. 2 Distribution of generated LHS

Key: ▲ Initial sample

■ New sample

3.2 Application for metamodeling

In this section, the generation algorithm is applied for metamodeling. There is a HLA-based electromagnetic gun simulation system, whose running time cannot be ignored. There is a sample of size 40, which is used for sensitivity analysis. We would like to use a metamodel-based simulation optimization method to optimize the system parameters for the best performance. In order to construct a metamodel, we want a LHS of size 90 and another LHS of size 40. LHS of 90 is the training set for constructing metamodel and evaluating the fitting effect. LHS of 40 is the predicting set for evaluating the predicting effect. For saving running time of simulation system, we choose the algorithm proposed in this paper to generate the training set.

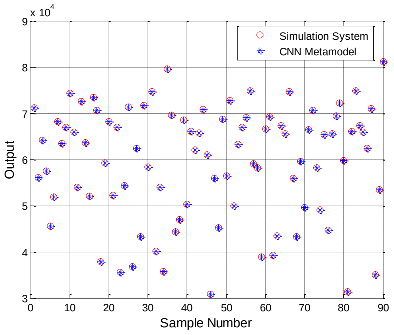
The algorithm is realized by MATLAB 2008a. There are 7 original sampling points to delete and 33 original sampling points to remain. Another 57 new sampling points are got by running the electromagnetic gun simulation system.

The evaluating index of metamodel is Multiple Correlation Coefficient (R^2) as shown in Eq. (8). R^2 is belonged in [0 1] and the more R^2 is, the better metamodel is.

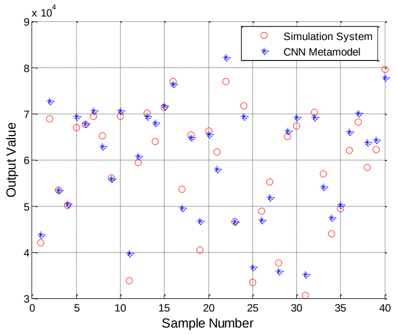
$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (8)$$

Artificial Neural Network (ANN) is the metamodel and the result is shown in Fig. 3. R^2=1 for training set and R^2=0.9487 for predicting set.

A contrast experiment is carried out, which use a classical LHS instead. In this case, R^2=1 for training set and R^2=0.9443 for predicting set. Therefore, the algorithm proposed in this paper achieves a LHS that have the similar result of classical LHS and remains 33 original sampling points. It means LHS by proposed algorithm have a good space filling effect the same as classical LHS and shorten 36.7% time for getting training set, which is because that the algorithm gets a strict LHS and utilizes the most original sampling points.



a. training set



b. predicting set

Fig.3 output values of simulation system and metamodel

4. CONCLUSION

As many simulation systems are costly to execute, the original sampling points might be a precious resource of a new analysis. A generation algorithm of LHS based on original sampling points gives a method to construct LHS based on original sampling points. The original sampling can be a Monte Carlo, a LHS, an orthogonal array, a factorial design or any other sampling. Actually, the algorithm proposed in this paper is an expansion of LHS extension algorithm, which does not require the original sampling to be a LHS. Therefore, the generation algorithm can be applied in the same fields of the LHS extension algorithm such as adaptive metamodel building and sequential experiments design and analysis.

For the future research, there are at least two suggestions. One is the improvement of the proposed algorithm. *Step* 2-b is an ergodic method and need lots of time to select remaining sampling points. When there are many original sampling points and variables, the time cost cannot be ignored. So a fast generation algorithm of LHS based on original sampling points is worth to research. The other suggestion is the research on the generation algorithm of other sampling method. Many analysis methods need special sampling method and generation algorithm of LHS is just one of them.

ACKNOWLEDGMENTS

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

Borgonovo E., Castaings W., Tarantola S., Model emulation and moment-independent sensitivity analysis: An application to environmental modeling. Environmental Modelling and Software, 34(6), pp. 105-115(2005).

Storlie C. B., Swiler L. P., Helton J. C. and et al., Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. Reliability Engineering and System Safety, 94(11), pp. 1735-1763(2009).

Huang S. C., Lo Y. L., Lu C.N., Non-Technical Loss Detection Using State Estimation and Analysis of Variance. IEEE Transactions on Power Systems, 28(3), pp. 2959-2966(2013).

Zouaoui F., Wilson J. R., Accounting for parameter uncertainty in simulation input modeling. IIE Transactions, 35(9) pp. 781-792(2003).

Helton J. C., Davis F. J., Johnson J. D., A comparison of uncertainty and sensitivity analysis results obtained with random and Latin hypercube sampling. Reliability Engineering and System Safety, 89(3), pp. 305-330(2005).

Braton R. R., Nelson B. L., Wei X., Quantifying Input Uncertainty via Simulation Confidence Intervals. INFORMS Journal on Computing, 14(6), pp. 74-87 (2013).

Mckay M. D., Beckman R. J., Conover W. J., A comparison of three algorithms for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2), pp. 239-245(1979).

Tong C., Refinement strategies for stratified sampling algorithms. Reliability Engineering and System Safet, 91(11), pp. 1257-1265(2006).

Sallaberry C. J., Helton J. C., Hora S. C., Extension of Latin hypercube samples with correlated variables. Reliability Engineering and System Safety, 93(7), pp. 1047-1059(2008).

Wu Y. Z., Chen L. P., Simulation optimization algorithm of multi-domain physical system. Science Press, pp. 186-189(2011).

Wang G. G., Adaptive Response Surface Algorithm Using Inherited Latin Hypercube Design Points. Journal of Mechanical Design, 125(2), pp. 210-220(2003).

Wei X., Research of Global Optimization Algorithm Based on Metamodel. Doctoral Dissertation of Huazhong University of Science and Technology, pp. 70-75(2012).

AUTHORS BIOGRAPHY

ZHIZHAO LIU is a Ph.D. student at Harbin Institute of Technology (HIT), and received the B.S., M.E. from HIT in 2010 and 2012. His research interests include simulation optimization and operational effectiveness evaluation. His email is liuzhizhao2007@hotmail.com.

WEI LI is the corresponding author. He is an associate professor at HIT, and received the B.S., M.E. and Ph.D. from HIT in 2003, 2006 and 2009 respectively. His research interests include simulation evaluation, simulation data analysis, and distributed simulation. His email is fleehit@163.com.

MING YANG is a professor and the director of control and simulation center at HIT. Also he is the vice editor-in-chief for Journal of System Simulation and editor for International Journal of Modeling Simulation and Science Computing. His research interests include system simulation theory and VV&A. His email is myang@hit.edu.cn.