Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2017) 663-670

ECCOMAS
Proceedia

UNCECOMP 2017
2nd ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15–17 June 2017

A NEW NESTED SPACE-FILLING DESIGNS ON EXPERIMENTS WITH DIFFERENT RANGES OF FACTORS

Jin Xu[1], Xiaojun Duan[1], Zhengming Wang[1] and Liang Yan[1]

[1]College of Science, National University of Defense Technology, Changsha, Hunan, HN 731, PRC
e-mail: xujin_nudt@163.com; xj_duan@163.com; wzm@nudt.edu.cn; yanliang@nudt.edu.cn

Keywords: computer experiment, Nested designs, Radar homing missile, multifidelity computer model, Latin hypercebe designs

Abstract. *Experiments with different levels of accuracy or fidelity are very common in statistics, applied mathematics and engineering. Nested space-filling designs are useful approach to deal with those questions. But in some special situations, e.g., the captive flight tests and missile tests in radar homing precision experiments, the rangers of factors are different in some accuracy experiments and sometimes they have overlapping portions. The existing methods deal almost exclusively with experiments with same ranges of factors. In this paper, we propose a new construction of nested space-filling designs named NDF(Nested Designs on Factor with different ranges) to solve these problems.The proposed construction are useful in the experiments with different ranges of factors and achieves uniformity in low dimensions.*

doi: 10.7712/120217.5401.17171

1 INTRODUCTION

Computer experiments, which simulate real world phenomena using some method in computer such as finite element analysis and computational fluid dynamics, have became widely used in statistics, applied mathematics and engineering[[1],[2],[7]]. A main goal in many computer experiments is to estimate the expected output of a computer model given a distribution of inputs. To address this issue, McKay, Conover, and Beckman[[8]] introduced Latin hypercube designs.Space-filling designs are important in the design and analysis for computer experiments [[1],[2]]. It is commonly believed that design points should be evenly spread in the experimental region. Nested Latin hypercube designs are popular researched in recently years[[3],[4],[5]], which mostly deal with the high-accuracy and low-accuracy experiments, where Sliced Latin hypercube designs are similar to the Nested one[[10],[11],[12]]. Meanwhile, a new method to construct SLHD is proposed in [9]. But in some special situations, e.g., the captive flight tests and missile tests in radar homing precision experiments, the ranges of factors are different in the original experiment and substitution experiment, NLHD could not deal with this situations and the existing methods deal almost exclusively with experiments which have same ranges of factors. In this paper, we propose a new construction of space-filling designs which first generates two small Latin hypercube designs and then arrange them together to form a new Latin hypercube design. This construction is similar with Sliced Latin hypercube designs(SLHD), however, the proposed construction is easy to implement, capable of accommodating any number of run size.

Firstly, We introduce a motivating example. Captive fight and missile physics experiments are two important methods to test the property of radar homing missile. But the ranges of factors are different in the two experiments when considering the CEP (Circular Error Probability) as the response. It is known that the error of radar homing is related to factors $\dot{R}$, $\ddot{R}$ and S/C which denote velocity, acceleration and signal-to-clutter ratio separately. The parameter interval for different experimental states is displayed in table 1. In general, the cost W of the four different experiments are not same, their relation is $W_1 < W_2 < W_3 < W_4$. The main difference factors which could be designed between captive fight and missile physics experiments are $\dot{R}$ and $\ddot{R}$.

Table 1: The Parameter Interval for different experimental states.

	State 1	State 2	State 3	State 4
$\dot{R}$	300−400	300−400	500−900	500−900
$\ddot{R}$	0−30	0−30	0−300	0−300
S/C	1.5−2.5	3−5	1.5−2.5	3−5

Then, we introduce LHD briefly. An $n \times m$ matrix $A = (a_{ij})$ is a called a Latin hypercube if each column of A is a permutation of $\{1, \cdots, n\}$. There are two methods to generate the design points in the unit cube $[0, 1]^n$ from A. The first is through $D_{ij} = (a_{ij} - 0.5)/n$, and the other is through $D_{ij} = (a_{ij} - u_{ij})/n$ where u_{ij} are independent random variables with a common uniform distribution on $[0, 1]$.The difference between the two methods can be seen as follows. When projected onto each of the m variables, both methods have the property that one and only one of the n design points fall within each of the n small intervals defined by$[0, 1/n), [1/n, 2/n), \cdots, [(n-1)/n, 1]$. The first method gives the mid-points of these intervals while the second gives points are uniformly distributed in their corresponding intervals.

2 A New Construction of Latin hypercube design

In this section, we present a simple method to construct a new nested Latin hypercube design, named NDF, which can be used in the experiments with factors that have different ranges. The proposed Latin hypercube design has different construction from the existing one in[3], where we first generates two small Latin hypercube designs and then arrange them together to form the NDF. And this construction . The proposed LHD and its two parts could achieve good uniformity in low dimensions, then this construction could deal with the above mentioned problems efficiently and directly. Meanwhile, its parameters could be more flexible parameters.

Let Z_n denotes the set $\{1, \cdots, n\}$, now we present a Z_n-to-Z_m projection where n, m are two positive integers with $n > m$. For any x in Z_n, define

$$\varphi(x) = \lceil x * m/n \rceil \tag{1}$$

We now introduce some useful notation. For $a \in R$, $\lceil a \rceil$ denotes the smallest integer no less than a and $\lfloor a \rfloor$ denotes the largest integer no greater than a. Similarly define $\lceil D \rceil$ and $\lfloor D \rfloor$ for a real matrix D. For a matrix A, let $A(:, j)$ be its jth column, $A(i, :)$ be its ith row, and $AA(i, j)$ be its (i, j)th element. For the projection $\varphi : Z_n \to Z_m$, let $y \in Z_m$, the set $\{\varphi^{-1}(y)\}$ denotes all the integers $x \in Z_n$ which satisfy $\varphi(x) = y$. For a matrix H based on Z_n, $\varphi(H)$ denotes the matrix obtained form H after the levels of its entries are collapsed according to φ. Obviously, we have some results as follows: If n, m are two positive integers with $n = k * m + b$, where $k > 0, 0 \leq b < m$ are integers, for the projection φ, we have that

Lemma 1 *If n, m are two positive integers with $n = km + b$, where $k > 0, 0 \leq b < m$ are integers, for the projection φ, we have that*

1. *φ is a monotonous increasing function;*
2. *For any integer $y \in Z_m$, the number of the entries $\{\varphi^{-1}(y)\}$ is k or $k+1$, and the number of $\{\varphi^{-1}(y)\}$ which have k and $k+1$ entries is $m - b$ and b respectively.*

Proof. Only part (ii) needs a proof. Consider $x \in Z_n, y \in Z_m$ and $x \in \{\varphi^{-1}(y)\}$, obviously, $y - 1 < xm/n \leq y$, so we have $n(y-1)/m < x \leq yn/m$. Thus,

$$x \in (\lceil n(y-1)/m \rceil, \lceil yn/m \rceil] \tag{2}$$

Note that $n(y-1)/m = k(y-1) + b(y-1)/m$ and $yn/m = ky + by/m$. For x is a integer, which yields that the number of the entries $\{\varphi^{-1}(y)\}$ is k or $k+1$. For there are m sets $\{\varphi^{-1}(y)\}$ and n entries of Z_n, the rest of part (ii) holds.

Lemma 2 *For n_1, n_2, n_3, p are positive integers with $n_1 = k_1 p, n_2 = k_2 p, n_3 = k_3 p$, and $k_1 \geq k_2 \geq k_3 > 0$ are integers, let $\varphi_1 : Z_{n_1} \to Z_{n_2}, \varphi_2 : Z_{n_2} \to Z_{n_3}$, and $\varphi : Z_{n_1} \to Z_{n_3}$. Then, for any $x \in Z_{n_1}$, the equation*

$$\varphi(x) = \varphi_2[\varphi_1(x)] \tag{3}$$

is satisfied.

Proof. For any $x \in Z_{n_1}$,

$$\varphi(x) = \lceil x n_3/n_1 \rceil = \lceil x k_3/k_1 \rceil$$

and

$$\varphi_2\left[\varphi_1\left(x\right)\right]=\lceil\lceil xn_2/n_1\rceil n_3/n_2\rceil=\lceil\lceil xk_2/k_1\rceil k_3/k_2\rceil$$

Let q be a integer and $\varphi\left(x\right)=q$, then, we must have that $k_1\left(q-1\right)/k_3<x\leq k_1q/k_3$. Thus

$$\begin{aligned}\varphi_2\left[\varphi_1\left(x\right)\right]&\leq\lceil\lceil qk_1k_2/\left(k_1k_3\right)\rceil k_3/k_2\rceil\\&<\lceil\left(qk_2/k_3+1\right)k_3/k_2\rceil\\&=\lceil q+k_3/k_2\rceil\\&=q+1\end{aligned}$$

Similarly,

$$\begin{aligned}\varphi_2\left[\varphi_1\left(x\right)\right]&>\lceil\lceil k_1k_2\left(q-1\right)/\left(k_1k_3\right)\rceil k_3/k_2\rceil\\&\geq\lceil\left(k_2\left(q-1\right)/k_3\right)k_3/k_2\rceil\\&=q-1\end{aligned}$$

For $\varphi_2\left[\varphi_1\left(x\right)\right]$ is a integer, $\varphi_2\left[\varphi_1\left(x\right)\right]=q$ holds, which completes the proof.

Let A_1, A_2 are two LHDs generated independently, where A_1 contains n_1 points in m factors with its factor levels $1,\cdots,n_1$ and A_2 contains n_2 points in m factors with its factor levels $1,\cdots,n_2$, without losing generality, let $n_1\leq n_2$. Stack A_1 and A_2 to form an $(n\times m)$ matrix $H=A_1\cup A_2$, where $n=n_1+n_2$, H_1 denotes the submatrix of H corresponding to A_1 and H_2 denotes the submatrix of H corresponding to A_2. Let φ_1 denotes $\varphi:Z_n\to Z_{n_1}$ and φ_2 denotes $\varphi:Z_n\to Z_{n_2}$. Let $t_1\left(i\right)$ denotes the numbers of entries in $\{\varphi_1^{-1}\left(i\right)\}\subset Z_n$ where $i\in Z_{n_1}$ and $t_2\left(j\right)$ denotes the numbers of entries in $\{\varphi_2^{-1}\left(j\right)\}\subset Z_n$ where $j\in Z_{n_2}$. Note that $t_2\left(i\right)=1$ or 2. For the arbitrary lth column of H, this construction method consists of there steps:

Step 1: Replace the level $i\in H_2\left(:,l\right)$ satisfies $t_2\left(i\right)=1$ with the single integer $c\in\{\varphi_2^{-1}\left(i\right)\}$, then let $t_2\left(i\right)=0$, $t_1\left[\varphi_1\left(c\right)\right]=t_1\left[\varphi_1\left(c\right)\right]-1$ and delete c from $\{\varphi_1^{-1}\left[\varphi_1\left(c\right)\right]\}$. Repeat step 1 until there is no $i\in H_2\left(:,l\right)$ satisfies the condition. If there exists $j\in H_1\left(:,l\right)$ satisfies $t_1\left(j\right)=1$,go to step 2, else go to step 3;

Step 2: Replace the level $j\in H_1\left(:,l\right)$ satisfies $t_1\left(j\right)=1$ with the single integer $c\in\{\varphi_1^{-1}\left(j\right)\}$, then let $t_1\left(j\right)j=0$, $t_2\left[\varphi_2\left(c\right)\right]=t_2\left[\varphi_2\left(c\right)\right]-1$ and delete c from $\{\varphi_2^{-1}\left[\varphi_2\left(c\right)\right]\}$. Repeat step 2 until there is no $j\in H_1\left(:,l\right)$ satisfies the condition. If there exists $i\in H_2\left(:,l\right)$ satisfies $t_2\left(i\right)=1$,return to step 1, else go to step 3;

Step 3: Choose any integer $i\in H_2\left(:,l\right)$ satisfies $t_2\left(i\right)=2$, and replace it with the any integer $c\in\{\varphi_2^{-1}\left(i\right)\}$. For another element $d\in\{\varphi_2^{-1}\left(i\right)\}$, let $t_1\left(\varphi\left(d\right)\right)=t_1\left(\varphi\left(d\right)\right)$ and delete d from $\{\varphi_1^{-1}\left(\varphi\left(d\right)\right)\}$. If $t_1\left(\varphi\left(d\right)\right)=1$, go to step 2. If for any $i\in H_2\left(:,l\right)$, the $t_2\left(i\right)=0$ is established, the achieved H is the wanted NDF.

We must have the conclusions shown as follows easily.:

Theorem 1 *Consider H, H_1 and H_2 as constructed above. Then we have that*

1. *the matrix H is a LHD(n,m);*

2. *the submatrix H_1 and H_2 of H become LHD(n_1,s) and LHD(n_1,s) when the n levels are collapsed into n_1 and n_2 levels according to the scheme: $i\to\varphi_1\left(i\right)$ and $i\to\varphi_1\left(i\right)$ respectively.*

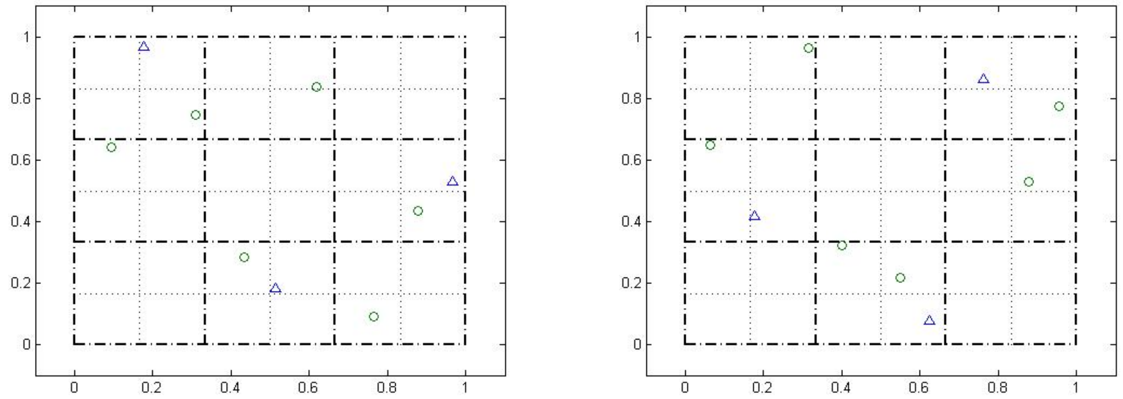

Figure 1: Bivariate projections of the NDF H with two parts H_1 and H_2 $(\circ, \triangle)$ in example 1 and NLHD proposed in [3]

Some examples are provided below to illustrate the proposed procedure.

Example 1 *We generate two LHD with $n_1 = 3$, $n_2 = 6$, and $m = 3$, then the LHD is as follows.*

$$H = \left[\begin{array}{ccc} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 3 & 2 & 3 \\ \hline 6 & 5 & 1 \\ 3 & 3 & 6 \\ 2 & 2 & 2 \\ 4 & 4 & 3 \\ 5 & 6 & 5 \\ 1 & 1 & 4 \end{array}\right] \rightarrow \left[\begin{array}{ccc} 6 & 2 & 3 \\ 2 & 8 & 5 \\ 9 & 5 & 8 \\ \hline 8 & 7 & 1 \\ 4 & 4 & 9 \\ 3 & 3 & 2 \\ 5 & 6 & 4 \\ 7 & 9 & 7 \\ 1 & 1 & 6 \end{array}\right]$$

Figure 1 is the NDF proposed above and the nested Latin hypercube design proposed in [3], the second part of Latin hypercube design H could achieve more uniformity in one dimension.

Meanwhile, the new nested Latin hypercube could deal with arbitrarily parameters which have not been considered in other paper and the two parts of LHD could achieve best uniformity in one dimension, which is shown in Figure 2.

Example 2 *For $n_1 = 3$, $n_2 = 5$, and $m = 3$, then the NDF is as follows.*

$$H = \left[\begin{array}{ccc} 3 & 1 & 2 \\ 2 & 3 & 1 \\ 1 & 2 & 3 \\ \hline 5 & 1 & 1 \\ 2 & 2 & 4 \\ 4 & 5 & 3 \\ 3 & 3 & 2 \\ 1 & 4 & 5 \end{array}\right] \rightarrow \left[\begin{array}{ccc} 8 & 2 & 5 \\ 5 & 7 & 2 \\ 2 & 5 & 7 \\ \hline 7 & 1 & 1 \\ 3 & 3 & 6 \\ 6 & 8 & 4 \\ 4 & 4 & 3 \\ 1 & 6 & 8 \end{array}\right]$$

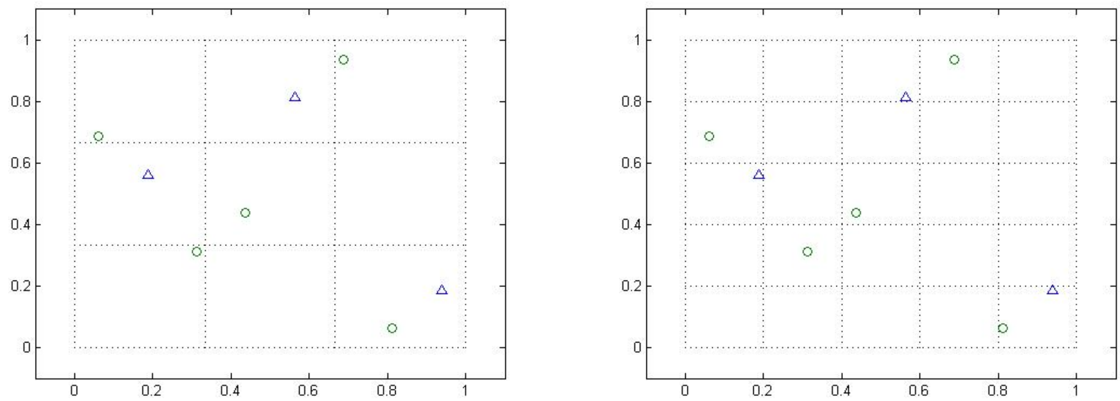

Figure 2: Bivariate projections of the NDF H with two parts H_1 and H_2 $(\circ, \triangle)$ in example 2

3 Numerical illustration

For the motivating problems described in Section 1, i.e., run a computer model in batches and multiple computer models based on similar mathematics where using different run size. Actually, the first problem can be viewed the experiments using multiple computer models based on the same mathematics.

More generally, the NDF. For experiments using t similar computer models, $f_1, \cdots, f_t$, where the inputs of each f_i are $X = (x_1, \cdots, x_m)$ whose distribution is the uniform measure on $(0,1]^m$. For $i = 1, \cdots, t$, define $\mu_i = E\left[f_1(X)\right]$. The goal here is to run $f_1, \cdots, f_t$, at $n_1, \cdots, n_t$ selected input values, to estimate $\mu_1, \cdots, \mu_t$. For $0 \leq \lambda_i \leq 1$, $i = 1, \cdots, t$, a linear combination of $\mu_1, \cdots, \mu_t$, given by

$$\eta = \sum_{i=1}^{t} \lambda_i \mu_i \tag{4}$$

can also be of interest in practice. Now, we introduce four different sampling schemes to achieve this goal as follows:

Definition Suppose that $n_1, \cdots, n_t, t, n$ are positive integers, with $n = \sum\limits_{i=1}^{t} n_i$.

1. Let IID denote a scheme that takes an independent and identically distributed sample of n_i runs for f_i, with the t samples generated independently;

2. Let SPLIT denote a scheme that randomly takes an Latin hypercube sample of n runs and splits it into t subdesigns which run sizes are $n_1, \cdots, n_t$, corresponding to $f_1, \cdots, f_t$.

3. Let COMBINE denote a scheme that obtains t independent Latin hypercube designs with $n_1, \cdots, n_t, t$ runs, each of which is associated with one f_i;

4. Let NFD denote a scheme that produces designs by using the method in Section 2, where each part, assigned to one f_i, is a smaller Latin hypercube design of n_i levels.

Example 3 *According to the physical background, we use the function*

$$f_1(X) = \frac{x_1}{2500} + \frac{x_2}{500} + \frac{1}{20\sqrt{x_3}}$$

Table 2: RMSEs of $\hat{\eta}$ of the four methods for Example 3.

$n_1 = 5$	IID	COMBINE	SPLIT	NDF
$n_2 = 5$	0.02200	0.01070	0.00486	0.00387
$n_2 = 6$	0.02133	0.01223	0.00472	0.00258
$n_2 = 7$	0.02131	0.01272	0.00471	0.00210
$n_2 = 8$	0.02137	0.01330	0.00465	0.00203
$n_2 = 9$	0.02095	0.01428	0.00465	0.00244
$n_2 = 10$	0.02088	0.01459	0.00453	0.00341

to act as the stage 1,2 where the distribution of x_1, x_2, x_3 is the uniform measure on $(300, 400]$, $(0, 30]$, $(1.5, 2.5] \bigcup (3, 5]$. Then, we regard the stage 3,4 as

$$f_2(X) = \frac{x_1}{5000} + \frac{x_2}{486} + \frac{1}{18\sqrt{x_3}}$$

where the distribution of x_1, x_2, x_3 is the uniform measure on $(500, 900], (0, 300], (1.5, 2.5] \bigcup (3, 5]$. For each of the four methods described above, we computer $\hat{\eta}$, with $\lambda_1 = \lambda_2 = 1/2$, 2000 times for $n_1 = 5$ and $n_2 = 5, 6, \cdots, 10$. Table 2 presents the RMSE (root mean square error) of $\hat{\eta}$ over the 2000 replicates for every method. The table clearly indicates that for every value of n_2, with $n_1 = 5$, the N method achieves greater degree of variance reduction than the other three methods for $\hat{\mu}$.

4 CONCLUSIONS

In this paper, we propose a new construction of Latin hypercube designs which first generates two small Latin hypercube and then arrange them together to form a new Latin hypercube design. The proposed construction is easy to implement, capable of accommodating any number of run size. For better uniformity in low dimension, the orthogonal array may be considered to constrict the new Latin hypercube designs for the experiments with different ranges of factors.

REFERENCES

[1] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*. 4(4), 205-206. , 1989.

[2] Santner, T. J., Williams, B. J., and Notz, W. I. The design and analysis of computer experiments. *Springer Science and Business Media.*, 2013.

[3] Qian, P. Z. Nested Latin hypercube designs. *Biometrika*, **96**, 4, 957-970, 2009. Statistica Sinica, 287-300.

[4] Qian, P. Z., Tang, B., and Wu, C. J. Nested space-filling designs for computer experiments with two levels of accuracy. *Statistica Sinica*,287-300, 2009

[5] Qian, P. Z. G., Ai, M., and Wu, C. F. J. Construction of nested space-filling designs. *The Annals of Statistics*,37(6A), 3616-3643, 2009

[6] Xiong, S., Qian, P. Z. G.,and Wu, C. F. J. Sequential Design and Analysis of High-Accuracy and Low-Accuracy Computer Codes. *Technometrics*,55(1), 37-46, 2013

[7] Fang, K. F., Li, R. Z., and Sudjianto, A., Design and Modeling for computer experiment. *New York: Chapman and Hall/CRC Press*, 393.

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

[9] Ba, S., Myers, W. R., and Brenneman, W. A. (2015). Optimal Sliced Latin Hypercube Designs. *Technometrics*, 57(4), 479-487.

[10] Qian, P. Z. G. Sliced Latin Hypercube Designs. *Journal of the American Statistical Association*, 107(497), 393-399, 2012.

[11] Qian, P. Z. G., and Wu, C. F. J. (2009). Sliced space-filling designs. *Biometrika*, 96(4), 945-956, 2009.

[12] Yin, Y., Lin, D. K. J., and Liu, M.-Q. . Sliced Latin hypercube designs via orthogonal arrays. *Journal of Statistical Planning and Inference*, 149, 162-17, 2014