Mathematical Modelling and Applications
2021; 6(4): 92-100
http://www.sciencepublishinggroup.com/j/mma
doi: 10.11648/j.mma.20210604.12
ISSN: 2575-1786 (Print); ISSN: 2575-1794 (Online)

Orthogonal Array-Based Latin Hypercube Designs for Computer Experiments

Kazeem Adewale Osuolale

Biostatistics Sub-Unit, Grant, Monitoring and Evaluation Unit, Nigerian Institute of Medical Research Yaba, Lagos, Nigeria

Email address:
whereisqosimadewale@gmail.com

To cite this article:
Kazeem Adewale Osuolale. Orthogonal Array-Based Latin Hypercube Designs for Computer Experiments. *Mathematical Modelling and Applications*. Vol. 6, No. 4, 2021, pp. 92-100. doi: 10.11648/j.mma.20210604.12

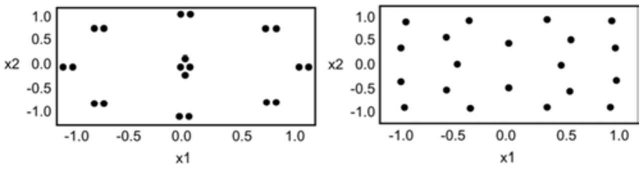
Received: October 24, 2021; **Accepted**: November 9, 2021; **Published**: December 24, 2021

Abstract: Orthogonal Array-based Latin Hypercube Designs (OALHDs) have not only become popular in practice among strategies used in the development of computer experiments but also useful whenever interest is focused on performing some physical experiments. Design construction for computer experiments is a new issue in this part of the world since it is more about experimental planning rather than modelling aspect in which some progress has been made. The Bush Construction Type II method was presented in this paper to construct a strong Orthogonal Array (OA) of strength three, using Galois Fields (GF) of order s which gave rise to the constructed Orthogonal Array-Based Latin Hypercube Designs (OALHD) for computer experiments. Orthogonal Array-based Latin Hypercube Design was used in this paper as a Latin hypercube design constructed based on orthogonal array in order to achieve better space-filling properties that would otherwise not be possessed by a random Latin hypercube design (LHD). Orthogonal Array (N, k) LHD were constructed at parameter values of OA (N, k)=(64, 6) and (125, 7). This study is an improvement on the early paper which adopted the Bush Construction Type I technique and it therefore aimed at proposing a novel approach that employed the maximin criterion in the k-Nearest Neighbour with Euclidean distance for constructing strong orthogonal arrays along with the Orthogonal Array-Based Latin Hypercube Designs (OALHDs). The OA (64, 6) LHD and OA (125, 7) LHD constructed have better space-filling properties and they achieve uniformity in each dimension. This study concludes that the constructed OALHDs can be used whenever interest is focused on performing either a conventional or computer experiment on real life situations. A program implementation for the construction of OALHDs was done using MATLAB 2016 computer package.

Keywords: Computer Experiments, Bush Construction Type II Method, Galois Fields, Latin Hypercube Designs, Orthogonal Array

1. Introduction

Experimentation through computer modelling has been widely accepted in many areas of engineering, modern industry, science and technology due to a rapid increase in power of computers. Mathematical models are used as computer models to describe some physical phenomena that are either complex or difficult to investigate using conventional statistical experiments. Computer models simply refer to simulators that are used to mimic the real-life experiment. A *computer experiment* is conducted using data obtained from a computer model in place of the physical process. Computer experiment may be performed to serve as a prototype before a physical experiment is conducted. Osuolale et al. [12] quoted Strogatz [24] to have reported that the first computer experiment was conducted by Enrico Fermi and colleagues at the Los Alamos Scientific Laboratory in 1953. Since the emergence of the first computer experiment, scientists in diverse fields have embraced computer experiments as an efficient tool to understand their respective processes. The values of the input variables in the computer code or model can be varied in order to determine the effect of various inputs on the output (s). The conventional experimental designs are well embraced when conducting physical experiments while space-filling designs are used in developing computer experiments. Space-filling designs are designs that spread design points evenly throughout the experimental region. Space-filling designs prevent replicate points by spreading the design points out to the maximum distance possible between any two points and distribute the points uniformly. The classical and space-filling designs are presented in Figure 1.



(a) "Classical" Design (b) "Space Filling" Design

***Figure 1.** Classical and Space-Filling Designs [1].*

It could be observed in Figure 1 that classical designs spread the sample points around the boundaries and leave a few at the centre of the design space but space-filling design fills the design space. A good experimental design tends to fill the design space.

In an early paper, Osuolale et al. [14] constructed Orthogonal arrays (OAs) using Bush Construction Type I and the orthogonal arrays (OAs) constructed were subsequently used to construct the desired OALHDs. The OALHDs constructed were from OA of strength two. The OAs with strength 3 or more tend to have very large run sizes. The Bush Construction Type II method using a mathematical theorem discussed in Hedayat et al. [4] was therefore adopted in this paper to construct strong Orthogonal Arrays (OAs) of strength three, using Galois Fields (GF) of order *s* which gave rise to the constructed Orthogonal Array-Based Latin Hypercube Designs (OALHDs) for computer experiments. This approach is somewhat different from the existing techniques as it constucts OAs of both prime and even numbers in terms of the levels of the orthogonal arrays used to construct OALHDs.

2. Orthogonal Arrays

The subject of Orthogonal Arrays (OAs) was first introduced by Rao [21] and later by Bose and Bush [2]. Orthogonal arrays (OAs) are used in the design of experiments, especially in fractional factorial experiment for their attractive statistical properties. Orthogonal arrays are not only useful in statistics but also useful in computer science and cryptography, medicine, agriculture and manufacturing. Owen [16] recommended the use of orthogonal arrays as suitable designs for computer experiments, numerical integration and visualization since OA ensures orthogonality which guarantees that the input variables are uncorrelated. The applications of orthogonal arrays to statistical design of experiments discussed in Hedayat et al. [4] are well known. Orthogonal arrays are greatly important in all areas of human investigation.

An orthogonal array of N runs, k factors, s levels, strength $t \geq 2$ and index λ is an n-by- k matrix with entries from a set of s levels, usually taken as 0... s -1 such that for every n-by-k matrix of s symbols, every subset of t columns from among the k columns, when considered alone must contain each of the possible s^t ordered rows the same number of times. The variables n, k, s, t and λ are the parameters of the OA and such an array is denoted by OA (n, k, s, t). The parameter λ (n/s^t) is referred to as the index parameter of the orthogonal array and is determined by the other four parameters. The most familiar examples of orthogonal arrays are regular fractional factorial designs discussed in Wu and Hamada [26]. The OA with $s_1 = s_2 = \cdots s_n = s$ is symmetric while the OA $s_1 \neq s_2 \neq \cdots \neq s_n$ is said to be asymmetric. The rows of the array represent the experiments to be performed and the columns of the orthogonal array correspond to the different variables whose effects are being analyzed. The construction of OALHDs in this study is largely dependent on the existence of orthogonal arrays. Another problem to consider with OAs is to determine either the minimum number of rows N in any OA (n, k, s, t) for given values of k, s and t or the maximum number of columns k for given values n, s and t. The celebrated inequalities found by Rao [22] for the construction of OAs had proferred a solution to this problem and one of the inequalities was adopted in this study.

Theorem 1: Rao's Inequalities

The Rao's inequalities are given by

i. $n \geq \sum_{i=0}^{u}\binom{k}{i}(s-1)^{i}$, if t=2u and

ii. $n \geq \sum_{i=0}^{u}\binom{k}{i}(s-1)^{i} + \binom{k-1}{u}(s-1)^{u+1}$, if t=2u+1 for u ≥ 0

Theorem I provides a scheme for determining either a lower bound on the number of rows, n, in any OA (n, k, s, t) design for given values, k, s and t, or an upper bound on the number of columns, k, for given values, n, s and t as stated in Osuolale [2017]. The proof of this theorem is given in Hedayat et al. [4]. The use of these inequalities depends on whether t is even or odd.

3. Orthogonal Array-Based Latin Hypercube Designs (Oalhds)

Orthogonal Array-Based Latin Hypercube Design (OALHD) is a Latin hypercube design constructed based on orthogonal arrays in order to maintain a univariate stratification and achieve better space-filling properties that would otherwise not be possessed by a random Latin hypercube design. It is known that not all Latin hypercube designs are good and this study attempts to find good LHDs by hybridizing OAs and LHDs to have OALHDs with optimal space-filling properties. Although, several experimental space-filling criteria have been proposed in the literature to construct space-filling designs which give good coverage of the design space and also offer low correlation among the design points chosen. Orthogonal designs offer uncorrelated input variables that help to independently assess effects of individual input variable on the response. These two properties are important characteristics of good experimental designs for computer experiments. Maximin and Minimax Criteria are also useful criteria that guarantee space-filling properties. These criteria were originally proposed by Johnson et al. [8] for use in the development of computer experiments. Shewry and Wynn [23] and Currin et al. [3] use the maximum entropy principle to develop designs for computer experiments. An optimal Latin hypercube design for computer experiments which either minimizes Integrated Mean Squared Error (IMSE) or maximizes entropy has also been discussed by Park [18]. For details of LHDs that are good based on some optimal design criteria, Iman and Conover [6], Owen [17], Morris and Mitchell [11], Ye [28], Ye et al. [29], Jin et al. [7], Joseph et al. [9] and Hernandez et al. [5] can be consulted.

Tang [25] proposed OALHDs that are more suitable for computer experiments than general Latin hypercube designs. He started his construction with an OA (*s^2, k, s*), and then replaced the *s* positions with symbol *t* by a random permutation of $(t - 1)s + 1,..., ts$, for all *t=1,..., s*. After the replacement procedure was done for all the *k* columns, the resulting matrix was denoted by $D = \left(d_{ij}\right), i = 1,..., s^2, j = 1, ..., k$ which forms an s^2 x *k* OALHD with s^2 levels. Leary et al. [10] considered searching for optimal OA-based Latin hypercubes using an alternative distance metric that minimizes

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{d_{ij}^{2}}$$

where '*n*' is the number of sampled points and d_{ij} is the distance between points *i* and *j* and is defined as:

$$d_{ij} = \frac{l_{ij} + {(n-1)}/{2} + u_{ij}}{n}, i = 1,...n, j = 1,...k \tag{1}$$

This criterion is used to search for a restricted subspace of the set of all OALHDs. Leary et al. [10] adapted strategies found in Morris and Mitchell [11] and Ye et al. [29] by performing optimization using the simulated annealing and the columnwise-pairwise algorithms.

Qian et al. [19] also proposed a method for constructing OALHDs as nested space-filling designs for multiple experiments with different levels of accuracy. They considered two experiments called low-accuracy experiment (LE) and high-accuracy experiment (HE). Their construction used a 2-step procedure. The first step constructs an OALHD for D_l with size n_1 and the second step chooses a subset of D_l with size n_2 as D_h based on the maximin distance criterion using

$$D_h = \arg\max_D \left[\min_{x_i, x_j \in D} \mathrm{d}(\mathrm{x}_i, \mathrm{x}_j) \right] \tag{2}$$

where *D* is any subset of D_l with size n_2. OALHD appeared to be a good choice for D_l but D_h is far from being space-filling in this procedure. Qian et al. [20] used nested orthogonal arrays and nested difference matrices to achieve space-filling and maximum stratification for both D_h and D_l to mitigate the drawbacks in the non-space-filling property of D_h earlier constructed by Qian et al. [19].

Osuolale et al. [13] proposed a technique for the construction of space-filling designs for three input variables computer experiments. Their method was meant for computer experiments with only three input variables and the technique limits the number of input variables to three with different number of runs. Yahya and Osuolale [27] also proposed a method of constructing OALHDs for computer experiments. Yahya and Osuolale [27] made use of an improved technique to construct OALHDs. The maximin distance criterion employed in this study to obtain the optimal designs made use of KNN search with Euclidean distance which finds the nearest neighbour in L for each point in L.

4. Material and Methods

A computer program via MATLAB 2016 was written to construct Orthogonal Array-Based Latin Hypercube Designs (OALHDs). A mathematical theorem was adopted in the construction of OALHDs and the desired OALHDs were optimized using a maximin distance criterion. The OALHD algorithm proposed employed the maximin distance criterion in a unique way using the *k*-Nearest Neighbour (KNN) search with Euclidean distance to maximize the minimum distance between any two design points from the possible design points of $(n!)^k$ in order to obtain the optimal design with space-filling properties. The technique searched only for OALHDs and consequently, optimal designs that are guaranteed to have some space-filling properties were constructed. The OALHD main function is given as:

$$\left[D / L\right] = f\left(level, strength\right) \tag{3}$$

An uppercase character *D* represents Orthogonal Array (OA), *L* is the desired OALHD, factors $\{x_1, x_2, \dots, x_n\}$ are the input (independent) variables and the level $\{0, 1, 2, \dots, s-1\}$ represents a set of entries used in the orthogonal array to construct the design. This method is based on the use of Galois fields. A field is composed of a set, *F*, and two binary operations that map *F X F* into *F*. A simple example is the set of non negative integers along with the operations of ordinary addition and multiplication. A Galois field is one for which the set, F, is finite.

Theorem 2: Orthogonal Arrays using Bush Construction Type II method [4]

If $s = 2^m$, $m \geq 1$ and t=3 then there exists an $OA(s^t$, $s + 2, s, t)$ of index unity. This technique works for OALHDs of large number of runs and different number of input variables and s levels with even and odd numbers.

Having constructed the OA, ranking was done such that:

i. Each column of the OA was sorted in ascending order
ii. The sorting order was used to create a set taken from $S = \{1,2, \dots, n\}$
iii. It returns a column vector of S as d_i.

Therefore:

$$R = [d_1 d_2 \dots d_m]. \tag{4}$$

With this, an initial OALHD was created as:

$$L = \frac{d - 0.1}{n} \tag{5}$$

where n is given as $n = s^3$. To add space filling properties, the design, L, is optimised using a choice of maximin distance criterion as follows:

i. A new design was created as follows:

$$L_{new} = \frac{d - u_{ij}}{n} \tag{6}$$

where u_{ij} is taken from a uniform distribution [0 1] and d from the OA earlier constructed.

ii. The new design was scored based on k-Nearest Neighbours using Euclidean distance which found the nearest neighbour in L for each point in L. The algorithm was based on:

$$d_{st}^2 = (L_s - L_t)(L_s - L_t)' \tag{7}$$

The d_{st} is the distance between L_s and L_t. Since two nearest neighbours are required, d_{st} is returned as a two-column matrix of which the second column was extracted as the maximum between the two columns since the first column contains zero all through. Then the extracted column was scored by minimizing the distance between the entries in the column using:

$$score = \min(d_{st_2}) \tag{8}$$

where d_{st_2} is the extracted second column.

iii. An iteration was performed to determine the best design based on maximin criterion as follows:

$$Score_{old} = score(L) \tag{9}$$

$$Score_{new} = score(L_{new}) \tag{10}$$

$$L = \begin{cases} L_{new} \ if \ Score_{new} > Score_{old} \\ L \ Otherwise \end{cases} \tag{11}$$

$$Score_{old} = \begin{cases} Score_{new} \, Score_{new} > Score_{old} \\ Score_{old} \ Otherwise \end{cases} \tag{12}$$

5. Results

The results of the orthogonal array-based Latin hypercube designs, OA (64, 6) LHD and OA (125, 7) LHD constructed from OA (64, 6, 4, 3) and OA (125, 7, 5, 3) are provided in the two cases in Table 1 and Table 2 with [D,L]=oa_test2 (4,3) and oa_test2 (5,3), respectively. The plots for the projections of design points among various input variables for OA (64, 6) LHD and OA (125, 7) LHD are also given in Figures 2 and 3, respectively. The proposed algorithm is capable of constructing OA (64, 6) LHD, OA (125, 7) LHD, OA (216, 8) LHD, OA (343, 9) LHD, OA (512, 10) LHD, OA (729, 11) LHD, OA (1000, 12) LHD, OA (1331, 13) LHD, OA (1728, 14) LHD, OA (2197, 15) LHD and OA (2744, 16) LHD.

Table 1. Construction of OA (64, 6, 4, 3) and OA (64, 6) LHD.

	Orthogonal Array (D)						Design Points (L)					
	x_1	x_2	x_3	x_4	x_5	x_6	x_1	x_2	x_3	x_4	x_5	x_6
1	0	0	0	0	0	0	0.014	0.014	0.014	0.014	0.014	0.014
2	0	1	3	2	1	0	0.030	0.264	0.764	0.514	0.264	0.030
3	0	2	1	3	2	0	0.045	0.514	0.264	0.764	0.514	0.045
4	0	3	2	1	3	0	0.061	0.764	0.514	0.264	0.764	0.061
5	0	1	2	3	0	1	0.077	0.280	0.530	0.780	0.030	0.264
6	0	0	1	1	1	1	0.092	0.030	0.280	0.280	0.280	0.280
7	0	3	3	0	2	1	0.108	0.780	0.780	0.030	0.530	0.295
8	0	2	0	2	3	1	0.123	0.530	0.030	0.530	0.780	0.311
9	0	2	3	1	0	2	0.139	0.545	0.795	0.295	0.045	0.514
10	0	3	0	3	1	2	0.155	0.795	0.045	0.795	0.295	0.530
11	0	0	2	2	2	2	0.170	0.045	0.545	0.545	0.545	0.545
12	0	1	1	0	3	2	0.186	0.295	0.295	0.045	0.795	0.561
13	0	3	1	2	0	3	0.202	0.811	0.311	0.561	0.061	0.764
14	0	2	2	0	1	3	0.217	0.561	0.561	0.061	0.311	0.780
15	0	1	0	1	2	3	0.233	0.311	0.061	0.311	0.561	0.795
16	0	0	3	3	3	3	0.248	0.061	0.811	0.811	0.811	0.811
17	1	1	1	1	0	0	0.264	0.327	0.327	0.327	0.077	0.077
18	1	0	2	3	1	0	0.280	0.077	0.577	0.827	0.327	0.092
19	1	3	0	2	2	0	0.295	0.827	0.077	0.577	0.577	0.108
20	1	2	3	0	3	0	0.311	0.577	0.827	0.077	0.827	0.123
21	1	0	3	2	0	1	0.327	0.092	0.842	0.592	0.092	0.327
22	1	1	0	0	1	1	0.342	0.342	0.092	0.092	0.342	0.342
23	1	2	2	1	2	1	0.358	0.592	0.592	0.342	0.592	0.358
24	1	3	1	3	3	1	0.373	0.842	0.342	0.842	0.842	0.373
25	1	3	2	0	0	2	0.389	0.858	0.608	0.108	0.108	0.577
26	1	2	1	2	1	2	0.405	0.608	0.358	0.608	0.358	0.592
27	1	1	3	3	2	2	0.420	0.358	0.858	0.858	0.608	0.608
28	1	0	0	1	3	2	0.436	0.108	0.108	0.358	0.858	0.623
29	1	2	0	3	0	3	0.452	0.623	0.123	0.873	0.123	0.827
30	1	3	3	1	1	3	0.467	0.873	0.873	0.373	0.373	0.842

	Orthogonal Array (D)						**Design Points (L)**					
	x_1	**x_2**	**x_3**	**x_4**	**x_5**	**x_6**	**x_1**	**x_2**	**x_3**	**x_4**	**x_5**	**x_6**
31	1	0	1	0	2	3	0.483	0.123	0.373	0.123	0.623	0.858
32	1	1	2	2	3	3	0.498	0.373	0.623	0.623	0.873	0.873
33	2	2	2	2	0	0	0.514	0.639	0.639	0.639	0.139	0.139
34	2	3	1	0	1	0	0.530	0.889	0.389	0.139	0.389	0.155
35	2	0	3	1	2	0	0.545	0.139	0.889	0.389	0.639	0.170
36	2	1	0	3	3	0	0.561	0.389	0.139	0.889	0.889	0.186
37	2	3	0	1	0	1	0.577	0.905	0.155	0.405	0.155	0.389
38	2	2	3	3	1	1	0.592	0.655	0.905	0.905	0.405	0.405
39	2	1	1	2	2	1	0.608	0.405	0.405	0.655	0.655	0.420
40	2	0	2	0	3	1	0.623	0.155	0.655	0.155	0.905	0.436
41	2	0	1	3	0	2	0.639	0.170	0.420	0.920	0.170	0.639
42	2	1	2	1	1	2	0.655	0.420	0.670	0.420	0.420	0.655
43	2	2	0	0	2	2	0.670	0.670	0.170	0.170	0.670	0.670
44	2	3	3	2	3	2	0.686	0.920	0.920	0.670	0.920	0.686
45	2	1	3	0	0	3	0.702	0.436	0.936	0.186	0.186	0.889
46	2	0	0	2	1	3	0.717	0.186	0.186	0.686	0.436	0.905
47	2	3	2	3	2	3	0.733	0.936	0.686	0.936	0.686	0.920
48	2	2	1	1	3	3	0.748	0.686	0.436	0.436	0.936	0.936
49	3	3	3	3	0	0	0.764	0.952	0.952	0.952	0.202	0.202
50	3	2	0	1	1	0	0.780	0.702	0.202	0.452	0.452	0.217
51	3	1	2	0	2	0	0.795	0.452	0.702	0.202	0.702	0.233
52	3	0	1	2	3	0	0.811	0.202	0.452	0.702	0.952	0.248
53	3	2	1	0	0	1	0.827	0.717	0.467	0.217	0.217	0.452
54	3	3	2	2	1	1	0.842	0.967	0.717	0.717	0.467	0.467
55	3	0	0	3	2	1	0.858	0.217	0.217	0.967	0.717	0.483
56	3	1	3	1	3	1	0.873	0.467	0.967	0.467	0.967	0.498
57	3	1	0	2	0	2	0.889	0.483	0.233	0.733	0.233	0.702
58	3	0	3	0	1	2	0.905	0.233	0.983	0.233	0.483	0.717
59	3	3	1	1	2	2	0.920	0.983	0.483	0.483	0.733	0.733
60	3	2	2	3	3	2	0.936	0.733	0.733	0.983	0.983	0.748
61	3	0	2	1	0	3	0.952	0.248	0.748	0.498	0.248	0.952
62	3	1	1	3	1	3	0.967	0.498	0.498	0.998	0.498	0.967
63	3	2	3	2	2	3	0.983	0.748	0.998	0.748	0.748	0.983
64	3	3	0	0	3	3	0.998	0.998	0.248	0.248	0.998	0.998

[D,L]=oa_test2 (4, 3).

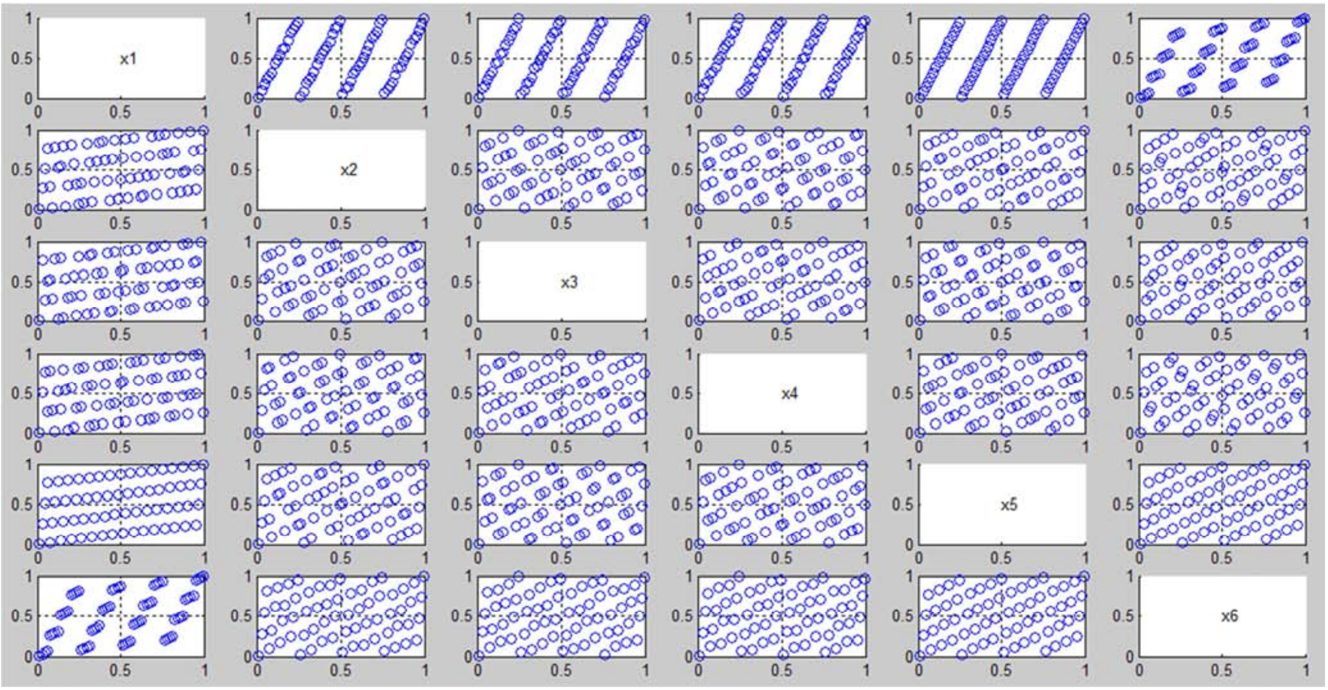


Figure 2. *Projection properties of OA (64, 6) LHD.*

Table 2. *Construction of OA (125, 7, 5, 3) and OA (125, 7) LHD.*

	Orthogonal Array (D)							Design Points (L)						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	0	0	0	0	0	0	0	0.007	0.007	0.007	0.007	0.007	0.007	0.007
2	0	1	4	5	6	1	0	0.015	0.159	0.495	0.615	0.767	0.207	0.015
3	0	2	3	1	7	2	0	0.023	0.311	0.375	0.127	0.887	0.407	0.023
4	0	3	7	4	1	3	0	0.031	0.463	0.879	0.495	0.127	0.607	0.031
5	0	4	6	2	5	4	0	0.039	0.615	0.751	0.247	0.639	0.807	0.039
6	0	1	2	3	4	0	1	0.047	0.167	0.255	0.367	0.519	0.015	0.207
7	0	0	6	6	2	1	1	0.055	0.015	0.759	0.743	0.255	0.215	0.215
8	0	3	1	2	3	2	1	0.063	0.471	0.135	0.255	0.391	0.415	0.223
9	0	2	5	7	5	3	1	0.071	0.319	0.631	0.879	0.647	0.615	0.231
10	0	5	4	1	1	4	1	0.079	0.719	0.503	0.135	0.135	0.815	0.239
11	0	2	4	6	3	0	2	0.087	0.327	0.511	0.751	0.399	0.023	0.407
12	0	3	0	3	5	1	2	0.095	0.479	0.015	0.375	0.655	0.223	0.415
13	0	0	7	7	4	2	2	0.103	0.023	0.887	0.887	0.527	0.423	0.423
14	0	1	3	2	2	3	2	0.111	0.175	0.383	0.263	0.263	0.623	0.431
15	0	6	2	4	6	4	2	0.119	0.815	0.263	0.503	0.775	0.823	0.439
16	0	3	6	5	7	0	3	0.127	0.487	0.767	0.623	0.895	0.031	0.607
17	0	2	2	0	1	1	3	0.135	0.335	0.271	0.015	0.143	0.231	0.615
18	0	1	5	4	0	2	3	0.143	0.183	0.639	0.511	0.015	0.431	0.623
19	0	0	1	1	6	3	3	0.151	0.031	0.143	0.143	0.783	0.631	0.631
20	0	7	0	7	2	4	3	0.159	0.911	0.023	0.895	0.271	0.831	0.639
21	0	4	3	7	6	0	4	0.167	0.623	0.391	0.903	0.791	0.039	0.807
22	0	5	7	2	0	1	4	0.175	0.727	0.895	0.271	0.023	0.239	0.815
23	0	6	0	6	1	2	4	0.183	0.823	0.031	0.759	0.151	0.439	0.823
24	0	7	4	3	7	3	4	0.191	0.919	0.519	0.383	0.903	0.639	0.831
25	0	0	5	5	3	4	4	0.199	0.039	0.647	0.631	0.407	0.839	0.839
26	1	1	1	1	1	0	0	0.207	0.191	0.151	0.151	0.159	0.047	0.047
27	1	0	5	4	7	1	0	0.215	0.047	0.655	0.519	0.911	0.247	0.055
28	1	3	2	0	6	2	0	0.223	0.495	0.279	0.023	0.799	0.447	0.063
29	1	2	6	5	0	3	0	0.231	0.343	0.775	0.639	0.031	0.647	0.071
30	1	5	7	3	4	4	0	0.239	0.735	0.903	0.391	0.535	0.847	0.079
31	1	0	3	2	5	0	1	0.247	0.055	0.399	0.279	0.663	0.055	0.247
32	1	1	7	7	3	1	1	0.255	0.199	0.911	0.911	0.415	0.255	0.255
33	1	2	0	3	2	2	1	0.263	0.351	0.039	0.399	0.279	0.455	0.263
34	1	3	4	6	4	3	1	0.271	0.503	0.527	0.767	0.543	0.655	0.271
35	1	4	5	0	0	4	1	0.279	0.631	0.663	0.031	0.039	0.855	0.279
36	1	3	5	7	2	0	2	0.287	0.511	0.671	0.919	0.287	0.063	0.447
37	1	2	1	2	4	1	2	0.295	0.359	0.159	0.287	0.551	0.263	0.455
38	1	1	6	6	5	2	2	0.303	0.207	0.783	0.775	0.671	0.463	0.463
39	1	0	2	3	3	3	2	0.311	0.063	0.287	0.407	0.423	0.663	0.471
40	1	7	3	5	7	4	2	0.319	0.927	0.407	0.647	0.919	0.863	0.479
41	1	2	7	4	6	0	3	0.327	0.367	0.919	0.527	0.807	0.071	0.647
42	1	3	3	1	0	1	3	0.335	0.519	0.415	0.159	0.047	0.271	0.655
43	1	0	4	5	1	2	3	0.343	0.071	0.535	0.655	0.167	0.471	0.663
44	1	1	0	0	7	3	3	0.351	0.215	0.047	0.039	0.927	0.671	0.671
45	1	6	1	6	3	4	3	0.359	0.831	0.167	0.783	0.431	0.871	0.679
46	1	5	2	6	7	0	4	0.367	0.743	0.295	0.791	0.935	0.079	0.847
47	1	4	6	3	1	1	4	0.375	0.639	0.791	0.415	0.175	0.279	0.855
48	1	7	1	7	0	2	4	0.383	0.935	0.175	0.927	0.055	0.479	0.863
49	1	6	5	2	6	3	4	0.391	0.839	0.679	0.295	0.815	0.679	0.871
50	1	1	4	4	2	4	4	0.399	0.223	0.543	0.535	0.295	0.879	0.879
51	2	2	2	2	2	0	0	0.407	0.375	0.303	0.303	0.303	0.087	0.087
52	2	3	6	7	4	1	0	0.415	0.527	0.799	0.935	0.559	0.287	0.095
53	2	0	1	3	5	2	0	0.423	0.079	0.183	0.423	0.679	0.487	0.103
54	2	1	5	6	3	3	0	0.431	0.231	0.687	0.799	0.439	0.687	0.111
55	2	6	4	0	7	4	0	0.439	0.847	0.551	0.047	0.943	0.887	0.119
56	2	3	0	1	6	0	1	0.447	0.535	0.055	0.167	0.823	0.095	0.287
57	2	2	4	4	0	1	1	0.455	0.383	0.559	0.543	0.063	0.295	0.295
58	2	1	3	0	1	2	1	0.463	0.239	0.423	0.055	0.183	0.495	0.303
59	2	0	7	5	7	3	1	0.471	0.087	0.927	0.663	0.951	0.695	0.311
60	2	7	6	3	3	4	1	0.479	0.943	0.807	0.431	0.447	0.895	0.319
61	2	0	6	4	1	0	2	0.487	0.095	0.815	0.551	0.191	0.103	0.487
62	2	1	2	1	7	1	2	0.495	0.247	0.311	0.175	0.959	0.303	0.495
63	2	2	5	5	6	2	2	0.503	0.391	0.695	0.671	0.831	0.503	0.503

	Orthogonal Array (D)							Design Points (L)						
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_1	x_2	x_3	x_4	x_5	x_6	x_7
64	2	3	1	0	0	3	2	0.511	0.543	0.191	0.063	0.071	0.703	0.511
65	2	4	0	6	4	4	2	0.519	0.647	0.063	0.807	0.567	0.903	0.519
66	2	1	4	7	5	0	3	0.527	0.255	0.567	0.943	0.687	0.111	0.687
67	2	0	0	2	3	1	3	0.535	0.103	0.071	0.311	0.455	0.311	0.695
68	2	3	7	6	2	2	3	0.543	0.551	0.935	0.815	0.311	0.511	0.703
69	2	2	3	3	4	3	3	0.551	0.399	0.431	0.439	0.575	0.711	0.711
70	2	5	2	5	0	4	3	0.559	0.751	0.319	0.679	0.079	0.911	0.719
71	2	6	1	5	4	0	4	0.567	0.855	0.199	0.687	0.583	0.119	0.887
72	2	7	5	0	2	1	4	0.575	0.951	0.703	0.071	0.319	0.319	0.895
73	2	4	2	4	3	2	4	0.583	0.655	0.327	0.559	0.463	0.519	0.903
74	2	5	6	1	5	3	4	0.591	0.759	0.823	0.183	0.695	0.719	0.911
75	2	2	7	7	1	4	4	0.599	0.407	0.943	0.951	0.199	0.919	0.919
76	3	3	3	3	3	0	0	0.607	0.559	0.439	0.447	0.471	0.127	0.127
77	3	2	7	6	5	1	0	0.615	0.415	0.951	0.823	0.703	0.327	0.135
78	3	1	0	2	4	2	0	0.623	0.263	0.079	0.319	0.591	0.527	0.143
79	3	0	4	7	2	3	0	0.631	0.111	0.575	0.959	0.327	0.727	0.151
80	3	7	5	1	6	4	0	0.639	0.959	0.711	0.191	0.839	0.927	0.159
81	3	2	1	0	7	0	1	0.647	0.423	0.207	0.079	0.967	0.135	0.327
82	3	3	5	5	1	1	1	0.655	0.567	0.719	0.695	0.207	0.335	0.335
83	3	0	2	1	0	2	1	0.663	0.119	0.335	0.199	0.087	0.535	0.343
84	3	1	6	4	6	3	1	0.671	0.271	0.831	0.567	0.847	0.735	0.351
85	3	6	7	2	2	4	1	0.679	0.863	0.959	0.327	0.335	0.935	0.359
86	3	1	7	5	0	0	2	0.687	0.279	0.967	0.703	0.095	0.143	0.527
87	3	0	3	0	6	1	2	0.695	0.127	0.447	0.087	0.855	0.343	0.535
88	3	3	4	4	7	2	2	0.703	0.575	0.583	0.575	0.975	0.543	0.543
89	3	2	0	1	1	3	2	0.711	0.431	0.087	0.207	0.215	0.743	0.551
90	3	5	1	7	5	4	2	0.719	0.767	0.215	0.967	0.711	0.943	0.559
91	3	0	5	6	4	0	3	0.727	0.135	0.727	0.831	0.599	0.151	0.727
92	3	1	1	3	2	1	3	0.735	0.287	0.223	0.455	0.343	0.351	0.735
93	3	2	6	7	3	2	3	0.743	0.439	0.839	0.975	0.479	0.551	0.743
94	3	3	2	2	5	3	3	0.751	0.583	0.343	0.335	0.719	0.751	0.751
95	3	4	3	4	1	4	3	0.759	0.663	0.455	0.583	0.223	0.951	0.759
96	3	7	0	4	5	0	4	0.767	0.967	0.095	0.591	0.727	0.159	0.927
97	3	6	4	1	3	1	4	0.775	0.871	0.591	0.215	0.487	0.359	0.935
98	3	5	3	5	2	2	4	0.783	0.775	0.463	0.711	0.351	0.559	0.943
99	3	4	7	0	4	3	4	0.791	0.671	0.975	0.095	0.607	0.759	0.951
100	3	3	6	6	0	4	4	0.799	0.591	0.847	0.839	0.103	0.959	0.959
101	4	4	4	4	4	0	0	0.807	0.679	0.599	0.599	0.615	0.167	0.167
102	4	5	0	1	2	1	0	0.815	0.783	0.103	0.223	0.359	0.367	0.175
103	4	6	7	5	3	2	0	0.823	0.879	0.983	0.719	0.495	0.567	0.183
104	4	7	3	0	5	3	0	0.831	0.975	0.471	0.103	0.735	0.767	0.191
`105	4	0	2	6	1	4	0	0.839	0.143	0.351	0.847	0.231	0.967	0.199
106	4	5	6	7	0	0	1	0.847	0.791	0.855	0.983	0.111	0.175	0.367
107	4	4	2	2	6	1	1	0.855	0.687	0.359	0.343	0.863	0.375	0.375
108	4	7	5	6	7	2	1	0.863	0.983	0.735	0.855	0.983	0.575	0.383
109	4	6	1	3	1	3	1	0.871	0.887	0.231	0.463	0.239	0.775	0.391
110	4	1	0	5	5	4	1	0.879	0.295	0.111	0.727	0.743	0.975	0.399
111	4	6	0	2	7	0	2	0.887	0.895	0.119	0.351	0.991	0.183	0.567
112	4	7	4	7	1	1	2	0.895	0.991	0.607	0.991	0.247	0.383	0.575
113	4	4	3	3	0	2	2	0.903	0.695	0.479	0.471	0.119	0.583	0.583
114	4	5	7	6	6	3	2	0.911	0.799	0.991	0.863	0.871	0.783	0.591
115	4	2	6	0	2	4	2	0.919	0.447	0.863	0.111	0.367	0.983	0.599
116	4	7	2	1	3	0	3	0.927	0.999	0.367	0.231	0.503	0.191	0.767
117	4	6	6	4	5	1	3	0.935	0.903	0.871	0.607	0.751	0.391	0.775
118	4	5	1	0	4	2	3	0.943	0.807	0.239	0.119	0.623	0.591	0.783
119	4	4	5	5	2	3	3	0.951	0.703	0.743	0.735	0.375	0.791	0.791
120	4	3	4	3	6	4	3	0.959	0.599	0.615	0.479	0.879	0.991	0.799
121	4	0	7	3	2	0	4	0.967	0.151	0.999	0.487	0.383	0.199	0.967
122	4	1	3	6	4	1	4	0.975	0.303	0.487	0.871	0.631	0.399	0.975
123	4	2	4	2	5	2	4	0.983	0.455	0.623	0.359	0.759	0.599	0.983
124	4	3	0	7	3	3	4	0.991	0.607	0.127	0.999	0.511	0.799	0.991
125	4	4	1	1	7	4	4	0.999	0.711	0.247	0.239	0.999	0.999	0.999

[D,L]=oa_test2 (5, 3).

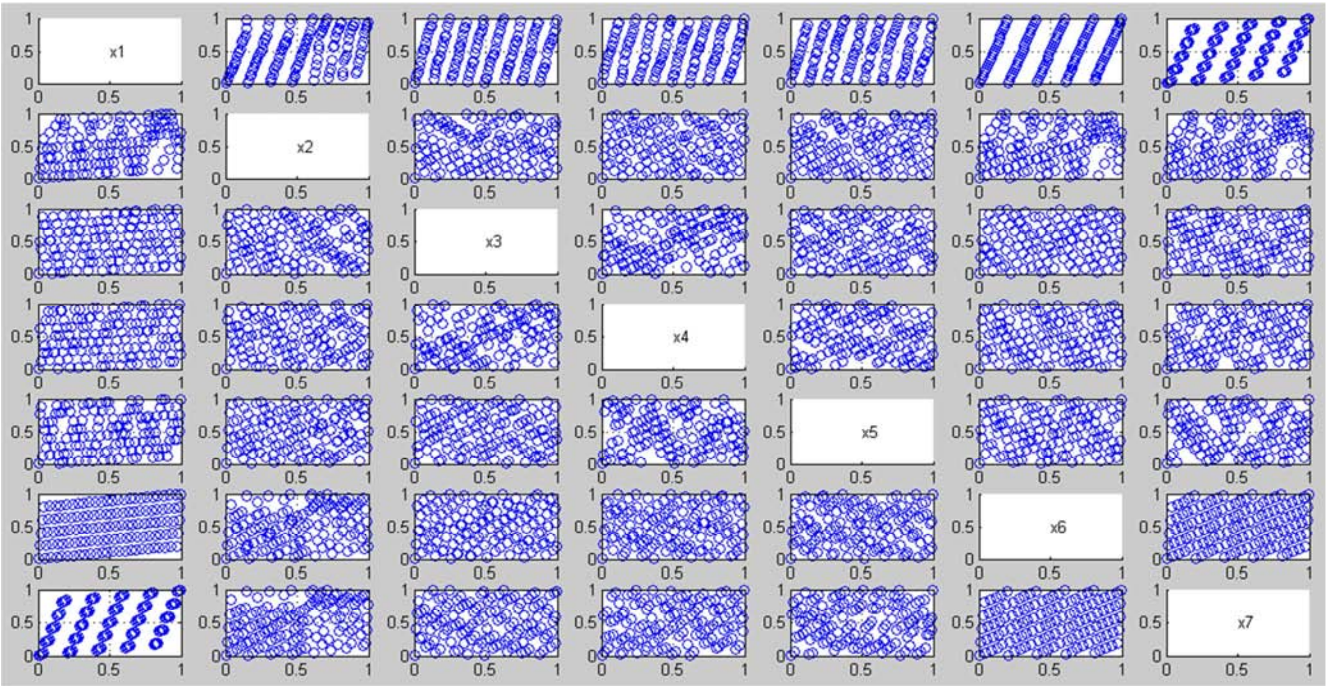


Figure 3. *Projection properties of OA (125, 7) LHD.*

5.1. Discussion of Constructed OA (64, 6) LHD

The technique has been applied to construct OALHDs using Galois fields of order s where s can be even or odd number. The two constructed OA (64, 6) LHD, and OA (125, 7) LHD using Galois fields of orders 4 and 5, respectively. The orthogonal array, *D*, constructed in the first case using a Galois field of order 4 revealed that each of the 64 possible rows 000, 013, 021, 032, 012,…., 330 appears a single time and this qualified it to be used to construct the desired OA (64, 6) LHD. The OA (64, 6) LHD contains 64 experimental runs (rows) and 6 factors (columns).

5.2. Discussion of Constructed OA (125, 7) LHD

In this case, the orthogonal array, *D*, constructed using a Galois field of order 5 revealed that each of the 125 possible rows 000, 014, 023, 037, 046, …., 441 appears a single time and this qualified it to be used to construct the desired OA (125, 7) LHD. The OA (125**,** 7) LHD has 125 experimental runs (rows) and 7 factors (columns). The two cases of the Orthogonal array-based Latin hypercube designs (OALHDs) constructed in this study have space-filling properties as depicted in Figures 2 and 3 and they achieve univariate stratification.

6. Conclusion

The technique presented in this study is an improvement on the early paper which adopted the Bush Construction Type I technique and it therefore aimed at proposing a novel approach that employed the maximin criterion in the *k*-Nearest Neighbour with Euclidean distance for constructing strong orthogonal arrays along with the Orthogonal Array-Based Latin Hypercube Designs (OALHDs). The OA (64, 6) LHD and OA (125, 7) LHD constructed have better space-filling properties and they achieve uniformity in each dimension. This study concludes that the constructed OALHDs can be used whenever interest is focused on performing either a conventional physical experiment or computer experiment on real life situations. The construction of orthogonal array- based Latin hypercube designs has been simplified in this study using computer codes written in MATLAB that produce results before one could say Jack Robinson to get a desired result. The OALHDs constructed via this method will be adopted to improve study design in biomedical research for a future study.

References

[1] Booker, A. J. (1998). Design and analysis of computer experiments, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. St. Louis, MO, AIAA, 1, 118-128. AIAA-98-4757.

[2] Bose, R. C. & Bush, K. A. (1952). Orthogonal arrays of strength two and three. Annals of Mathematical Statistics, 23, 508–524.

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

[4] Hedayat, A. S., Sloane, N. J. A. & Stufken, J. (1999). Orthogonal arrays: Theory and applications. Springer-Verlag, New York.

[5] Hernandez, A., Lucas, T. W. & Carlyle, M. (2012). Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination. ACM Trans. Model. Computer Simul. 22 (4), Article 20, 1-17.

[6] Iman, R. L. & Conover, W. J. (1982). A distribution-free approach to inducingrank correlation among input variables. Communication in Statistics, Part B, 11, 311-334.

[7] Jin, R., Chen, W. & Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. Journal of Statistical Planning and Inference, 134, 268–287.

[8] Johnson, M., Moore, L. & Ylvisaker, D. (1990). Minimax and maximin distance design. Journal of Statistical Planning and Inference, 26, 131-148.

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

[10] Leary, S., Bhaskar, A. & Keane, A. (2003). Optimal orthogonal array-based Latin hypercubes. Journal of Applied Statistics, 30, 585–598. MR1969503.

[11] Morris, M. D. & Mitchell, T. J. (1995). Exploratory designs for computer experiments. Journal of Statistical Planning and Inference, 43, 381-402.

[12] Osuolale, K. A., Yahya, W. B. & Adeleke, B. L. (2014a). Performing a simple pendulum experiment as a demonstrative example for computer experiments. Annals. Computer Science Series, 12 (2), 33-38.

[13] Osuolale, K. A., Yahya, W. B. & Adeleke, B. L. (2014b). Construction of space-filling designs for three input variables computer experiments. WASET, International Journal of Computer, Control, Quantum and Information Engineering, 8 (9), 1624-1628.

[14] Osuolale, K. A., Yahya, W. B. & Adeleke, B. L. (2015). Construction of orthogonal array-based Latin hypercube designs for deterministic computer experiments. Annals. Computer Science Series, 13 (1), 24-29.

[15] Osuolale, K. A. (2017). Techniques for designing, modelling and analysing computer experiments (Ph.D Thesis). University of Ilorin, Nigeria.

[16] Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. Statistica Sinica, 2, 439-452.

[17] Owen, A. B. (1994). Controlling correlations in Latin hypercube samples. Journal of the American Statistical Association, 89, 1571–1522.

[18] Park, J. S. (1994). Optimal Latin hypercube designs for computer experiments. Journal of Statistical Planning and Inference, 39, 95–111.

[19] Qian, Z., Seepersad, C., Joseph, R., Allen, J. & Wu, C. F. J. (2006). Building surrogate models based on detailed and approximate simulations. ASME Transactions, Journal of Mechanical Design 128, 668–677.

[20] Qian, P. Z. G., Tang, B. & Wu, C. F. J. (2009). Nested space-filling designs for computer experiments with two levels of accuracy. Statistica Sinica, 19, 287-300.

[21] Rao, C. R. (1946). Hypercube of strength ‘d’ leading to confounded designs in factorial experiments. Bulletin of Calcutta Mathematical Society, 38, 67-78.

[22] Rao, C. R. (1947). Factorial experiments derivable from combinatorial arrangements of arrays. Journal of the Royal Statistical Society (Suppl.), 9, 128-139.

[23] Shewry, M. C. & Wynn, H. P. (1988). Maximum entropy sampling with application to simulation codes. Proceedings of the 12th World Congress on Scientific Computation, IMAC88.

[24] Strogatz, S. (2003). The real scientific hero of 1953. New York Times. March 4, Editorial/Op-Ed.

[25] Tang, B. (1993). Orthogonal array-based Latin hypercubes. Journal of the American Statistical Association, 88, 1392–1397.

[26] Wu, C. F. J. & Hamada, M. (2000). Experiments: Planning, analysis and parameter design optimization. Wiley, New York.

[27] Yahya, W. B. & Osuolale, K. A. (2016). A method of constructing orthogonal array-based Latin hypercube designs for computer experiments. International Journal of Hybrid Information Technology, 9 (6), 253-262.

[28] Ye, K. Q. (1998). Orthogonal column Latinhypercubes and their application in computer experiments. Journal of the American Statistical Association, 93 (444), 1430-1439. Retrieved from http://dx.doi.org/10.1080/01621459.1998.10473803.

[29] Ye, K. Q., Li, W. & Sudjianto, A. (2000). Algorithmic construction of optimal symmetric Latin hypercube deisgns. Journal of Statistical Planning and Inference, 90, 145-159.