

AN ENHANCED ALGORITHM FOR CONSTRUCTING OPTIMAL SPACE-FILLING DESIGNS USING HADAMARD MATRICES OF ORDERS 4λ AND 8λ

Kazeem A. Osuolale, Babatunde L. Adeleke, Waheed B. Yahya

Department of Statistics, University of Ilorin, P.M.B 1515, Ilorin, Nigeria

Corresponding Author: Kazeem A. Osuolale, whereisqosimadewale@gmail.com

ABSTRACT: Orthogonal array Latin hypercube designs have become popular in practice among strategies used for developing computer experiments. Hadamard matrices have been used to construct orthogonal arrays based on the connection between Hadamard matrices and orthogonal arrays (OAs). A Hadamard matrix is a square matrix of +1 and -1 whose rows are orthogonal. This study aimed at proposing an enhanced algorithm that employed the maximin criterion in the k -Nearest Neighbour with Euclidean distance for constructing optimal space-filling design called Orthogonal Array Latin Hypercube Design (OALHD). Orthogonal arrays (OAs) were constructed from Hadamard matrices of orders 4λ and 8λ which are subsequently used to construct the desired OALHD. The Orthogonal array (n, k) Latin hypercube designs were constructed at parameter values of $OA(n, k, s, t, \lambda) = (8, 7, 2, 2, 2)$ and $(16, 8, 2, 3, 2)$. The $OA(8, 7)$ LHD and $OA(16, 8)$ LHD constructed have better space-filling properties and they achieve uniformity in each dimension. MATLAB 2015 computer package was used for the development of the algorithm that constructs the OALHDs.

KEYWORDS: Computer experiments, Hadamard matrices, Latin hypercube designs, Orthogonal array, Space-filling designs.

1. INTRODUCTION

Computer experiments are becoming more commonly used in science and engineering. This is mainly because the underlying physical experiments are too time-consuming, expensive or even impossible to perform. Rapid growth in computer power has made it possible to perform experiments on a computer model or simulator. Since the emergence of the first computer experiment conducted by Enrico Fermi and colleagues ([Str03]) in Los Alamos in 1953, scientists in diverse areas such as engineering, cosmology, particle physics and aircraft design have turned to computer experiments as an effective tool to understand their respective processes. For instance, in the design of a vehicle, computer experiments are used to study the effect of a collision of the vehicle with a barrier before manufacturing the prototype of the vehicle ([B+02]). Space-filling design like OALHD is popularly used in designing computer experiments.

Space-filling designs are designs that spread points evenly throughout the experimental region.

Hadamard matrices which were named after the French Mathematician Jacques Hadamard (1865-1963) were used to generate orthogonal arrays based on the theorem in Hedayat et al. ([HSS99]). A Hadamard matrix of order n is an $n \times n$ matrix H in which every entry is ± 1 such that $HH^T = n I_n$ ([Sti03]). They are equivalent to the study of two-level orthogonal arrays of strengths 2 and 3.

Theorem 1 ([HSS99])

Orthogonal arrays $OA(4\lambda, 4\lambda-1, 2, 2)$ and $OA(8\lambda, 4\lambda, 2, 3)$ exist if and only if there exists a Hadamard matrix of order 4λ .

Orthogonal arrays (OAs) were initially introduced by Rao (1946) followed by Bose and Bush ([BB52]). Orthogonal arrays (OAs) are essential in statistics and they are primarily used in designing experiments which simply means that they are important in all areas of human investigation. Orthogonal arrays are used in medicine, agriculture and manufacturing. Your automobile lasts longer today because of orthogonal arrays ([HSS99]). Pharmaceutical companies use orthogonal arrays to investigate the stability and shelf life of drugs which usually involve many different factors. The use of orthogonal arrays can lead to more economical tests and provide better statistical information.

An orthogonal array of n runs, k factors, s levels and strength $t \geq 2$ is an $n \times k$ matrix with entries from a set of s levels, usually taken as $0, \dots, s-1$, such that for every $n \times t$ submatrix, each of the s^t level combinations occurs the same number of times. Such an array is denoted by $OA(n, k, s, t)$. The number $\lambda(n/s^t)$ is called the index of the array. The rows of the array represent the experiments or tests to be performed, the columns correspond to different variables whose effects are being analysed and the entries in the array specify the levels at which the variables are to be applied. The construction of OALHDs depends on the existence of orthogonal arrays. For the existence results of orthogonal arrays, Hedayat et al. ([HSS99]) and Mukerjee and Wu ([MW06]) can be consulted. A library of orthogonal arrays is freely available on the N.J.A. Sloane

website and the MktEx macro using the software SAS ([Kuh09]). Another important problem in the study of OAs is to determine the minimal number of rows n in any OA (n, k, s, t) for given values k, s and t . The celebrated inequalities found by Rao [Rao46] were employed to proffer solution to the construction of orthogonal arrays in order to determine the minimal or maximal number of rows and columns in any OA (n, k, s, t) .

Theorem 2 (Rao’s Inequalities)

$$(i) N \geq \sum_{i=0}^u \binom{k}{i} (s-1)^i,$$

if $t = 2u$

$$(ii) N \geq \sum_{i=0}^u \binom{k}{i} (s-1)^i + \binom{k-1}{u} (s-1)^{u+1},$$

if $t = 2u+1$ for $u \geq 0$

Latin hypercube designs (LHDs) were the first type of design proposed for computer experiments ([MBC79]). LHDs do not have replicated runs but have one-dimensional uniformity in that when projected on each dimension, each portion of the design region has a design point. However, random LHDs may not be a good choice with respect to some useful criteria such as maximin distance and orthogonality but can easily be constructed. The maximin distance criterion was introduced by Johnson et al. ([JMY90]) and it maximizes the smallest distance between any two design points so that no two design points are too close to each other. A maximin distance design spreads out its points evenly over the entire design region.

An $n \times m$ matrix $D = (d_{ij})$ is called a Latin hypercube of n runs for k factors if each column of D is a permutation of $\{1, \dots, n\}$. There are two natural ways of generating design points in the unit cube $[0, 1]^m$ based on a given Latin hypercube. The first is through $L_{ij} = (d_{ij} - 0.5)/n$, with the n points given by (l_{i1}, \dots, l_{ik}) with $i = 1, \dots, n, j = 1, \dots, k$. The other way is to generate design points through $L_{ij} = (d_{ij} - u_{ij})/n$, with the n points given by (l_{i1}, \dots, l_{ik}) with $i = 1, \dots, n, j = 1, \dots, k$ where u_{ij} are independent random variables with a uniform distribution on $[0, 1]$. The two methods differ as the first method gives the mid-points of the intervals defined by $[0, 1/n), [1/n, 2/n), \dots, [(n-1)/n, 1]$ while the second gives points that are uniformly distributed in their corresponding intervals. The first method is due to Patterson ([Pat54]) while the second is due to Mckay et al. ([MBC79]).

Orthogonal array Latin hypercube designs (OALHDs) were proposed by Tang ([Tan93]). These OALHDs achieve better multi-dimensional space-filling property. Orthogonal Array (OA) can generate a design with better space-filling property than LHDs in that the former tend to place points both in the interior and on the boundary of the

design space while the latter is more likely to have design points on the boundary. Orthogonal arrays are used to construct OALHDs in this study to achieve better space-filling properties. An OA (n, k, s, t) LHD has the t -dimensional projection property that when projected onto any t columns, it has exactly λ points in each of the s^t cells. Tang ([Tan94]) provided a way to obtain OA-based Latin hypercubes based on single replicated full factorial designs and he showed that if the underlying orthogonal array is optimal with respect to the maximin distance criterion, so is the corresponding OA-based Latin hypercube. Leary et al. ([LBK03]) also considered searching for optimal OA-based Latin hypercubes that minimize:

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

where d_{ij} is the Euclidean distance, defined as

$$d_{ij} = \frac{l_{ij}^{+(n-1)/2+u_{ij}}}{n}, i = 1, \dots, n, j = 1, \dots, m$$

where $t = 2$, between the i_{th} and j_{th} design points. The optimization was performed via the simulated annealing algorithm ([MM95]) and the columnwise-pairwise algorithm ([LW97]). Osulale et al. ([OYA14a]) presented a paper on the algorithm for constructing spacefilling designs for Hadamard matrices of Orders 4λ and 8λ . Osulale et al. ([OYA14b]) proposed a technique for the construction of space-filling designs for three input variables computer experiments. Their intention is to utilize the design constructed to implement a simple pendulum experiment involving three input variables; length, displacement angle and mass as a demonstrative example for computer experiment. Similarly, OALHDs were constructed by Osulale et al. ([OYA15]) for developing borehole computer experiment using one of the OALHDs constructed in the paper. Yahya and Osulale ([YA16]) also proposed a method of constructing OALHDs for computer experiments.

2. MATERIALS AND METHODS

A computer program via MATLAB 2012 package was written to construct OALHDs. An n -by- n Hadamard matrix with $n > 2$ exists only if $\text{rem}(n, 4) = 0$ where

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \tag{1}$$

Since the Hadamard matrix is a matrix of +1's and -1's, the program converted +1's and -1's to 0's and 1's, that is, it converted the Hadamard matrix in bipolar array to binary array by the following relation:

$$H_{2N}^* = \frac{(H_{2N} + 1)}{2} \quad (2)$$

Orthogonal arrays (OAs) were constructed using Hadamard matrices of orders 4λ and 8λ based on Theorem 1 and they are subsequently used to construct the desired OALHDs using $L_{ij} = (d_{ij} - u_{ij})/n$ with u_{ij} equals 0.5 to initialise the program. The program sorted and ranked the entries in the array column by column ignoring ties and then applied the formula to arrive at the desired OALHDs. The steps taken in the construction of orthogonal array-based Latin hypercube designs can be illustrated with the aid of the program flowchart as displayed below:

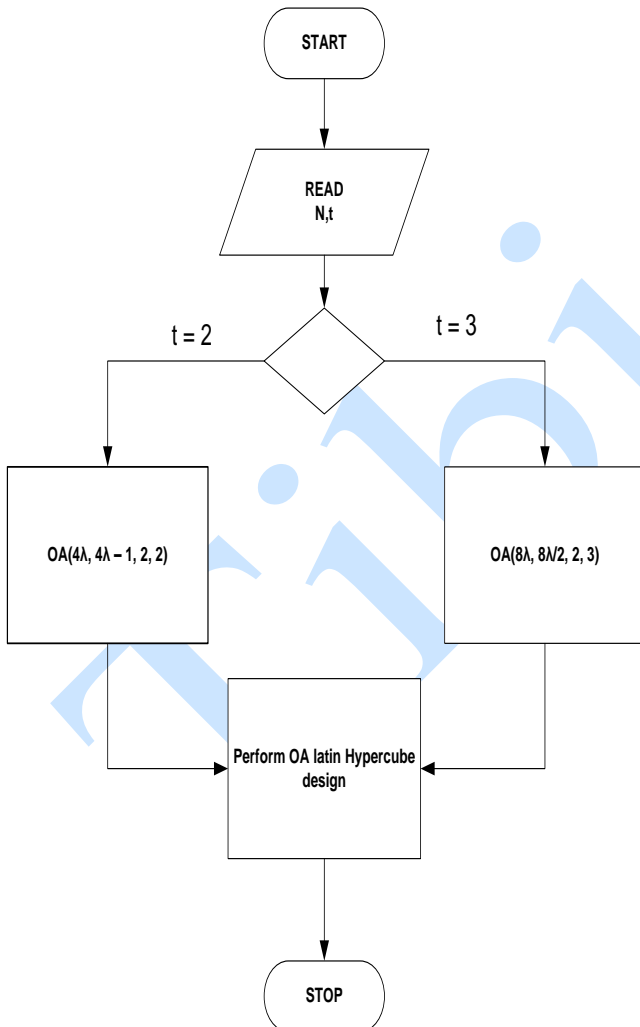


Figure 1: Program Flowchart for the Construction of Orthogonal Array Latin Hypercube Designs

3. RESULTS

3.1. Construction of OA (8, 7) LHD for Hadamard Matrix of Order 4λ when $\lambda = 2$

Hadamard (8) =

1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1
1	1	1	1	-1	-1	-1	-1
1	-1	1	-1	-1	1	-1	1
1	1	-1	-1	-1	-1	1	1
1	-1	-1	1	-1	1	1	-1

[D,L]=oa_hadamard (8,2)

D =

1	1	1	1	1	1	1
0	1	0	1	0	1	0
1	0	0	1	1	0	0
0	0	1	1	0	0	1
1	1	1	0	0	0	0
0	1	0	0	1	0	1
1	0	0	0	0	1	1
0	0	1	0	1	1	0

L =

0.5625	0.5625	0.5625	0.5625	0.5625	0.5625	0.5625
0.0625	0.6875	0.0625	0.6875	0.0625	0.6875	0.0625
0.6875	0.0625	0.1875	0.8125	0.6875	0.0625	0.1875
0.1875	0.1875	0.6875	0.9375	0.1875	0.1875	0.6875
0.8125	0.8125	0.8125	0.0625	0.3125	0.3125	0.3125
0.3125	0.9375	0.3125	0.1875	0.8125	0.4375	0.8125
0.9375	0.3125	0.4375	0.3125	0.4375	0.8125	0.9375
0.4375	0.4375	0.9375	0.4375	0.9375	0.9375	0.4375

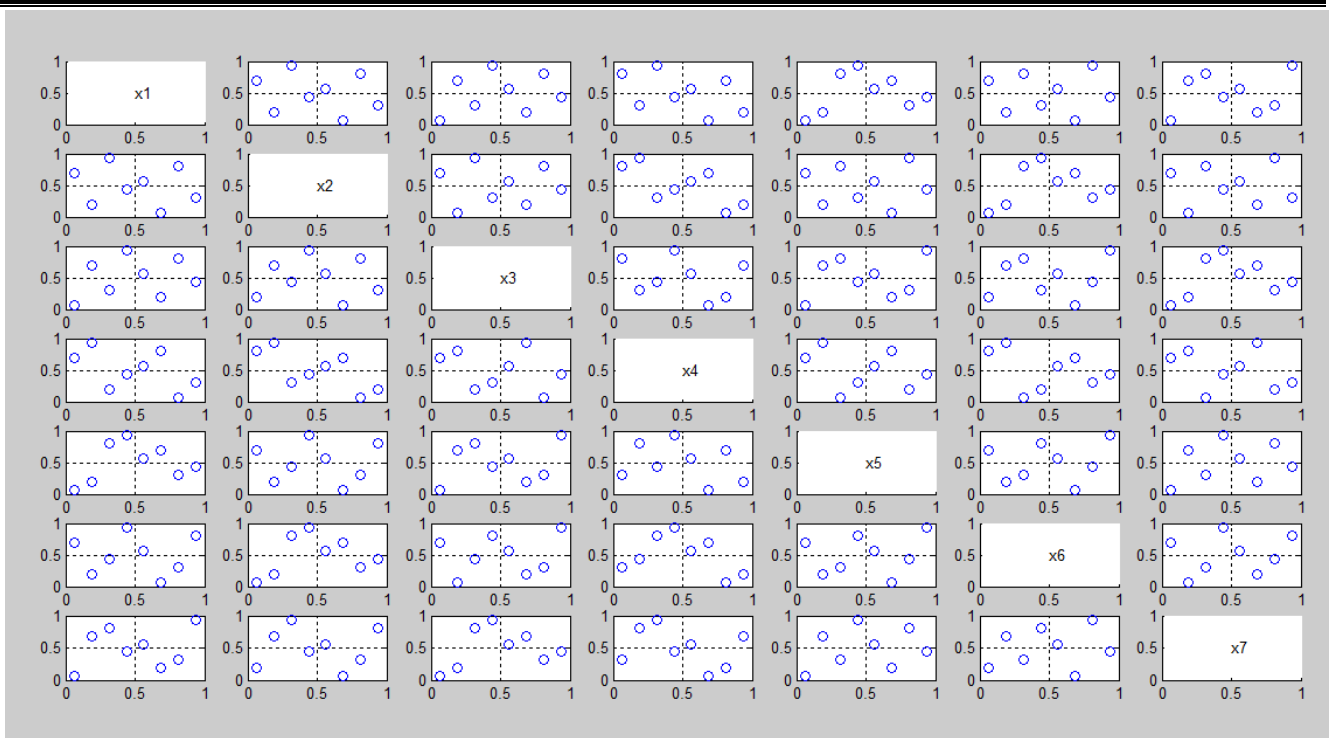


Figure 2: Projection Properties of OA (8, 7) LHD

3.2. Construction of OA (16, 8) LHD for Hadamard Matrix of Order 8λ when $\lambda = 2$

Hadamard (16) =

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	1	1	1	-1	-1	-1	-1	1	1	1	-1	-1	-1	1	1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
1	1	-1	-1	1	1	-1	-1	-1	1	1	-1	-1	1	1	1
1	-1	-1	1	1	-1	-1	1	1	1	-1	-1	1	1	1	-1
1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	-1	-1	1	1

[D, L]=oa_hadamard (16,3)

D =

1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0
1	1	0	0	1	1	0	0
1	0	0	1	1	0	0	1
1	1	1	1	0	0	0	0
1	0	1	0	0	1	0	1
1	1	0	0	0	0	1	1
1	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	0
0	0	0	0	1	1	1	1
0	1	0	1	1	0	1	0
0	0	1	1	1	1	0	0
0	1	1	0	1	0	0	1

L =

0.5313	0.5313	0.5313	0.5313	0.5313	0.5313	0.5313	0.5313
0.5938	0.0313	0.5938	0.0313	0.5938	0.0313	0.5938	0.0313
0.6563	0.5938	0.0313	0.0938	0.6563	0.5938	0.0313	0.0938
0.7188	0.0938	0.0938	0.5938	0.7188	0.0938	0.0938	0.5938
0.7813	0.6563	0.6563	0.6563	0.0313	0.1563	0.1563	0.1563
0.8438	0.1563	0.7188	0.1563	0.0938	0.6563	0.2188	0.6563
0.9063	0.7188	0.1563	0.2188	0.1563	0.2188	0.6563	0.7188
0.9688	0.2188	0.2188	0.7188	0.2188	0.7188	0.7188	0.2188
0.0313	0.2813	0.2813	0.2813	0.2813	0.2813	0.2813	0.2813
0.0938	0.7813	0.3438	0.7813	0.3438	0.7813	0.3438	0.7813
0.1563	0.3438	0.7813	0.8438	0.4063	0.3438	0.7813	0.8438
0.2188	0.8438	0.8438	0.3438	0.4688	0.8438	0.8438	0.3438
0.2813	0.4063	0.4063	0.4063	0.7813	0.9063	0.9063	0.9063
0.3438	0.9063	0.4688	0.9063	0.8438	0.4063	0.9688	0.4063
0.4063	0.4688	0.9063	0.9688	0.9063	0.9688	0.4063	0.4688
0.4688	0.9688	0.9688	0.4688	0.9688	0.4688	0.4688	0.9688

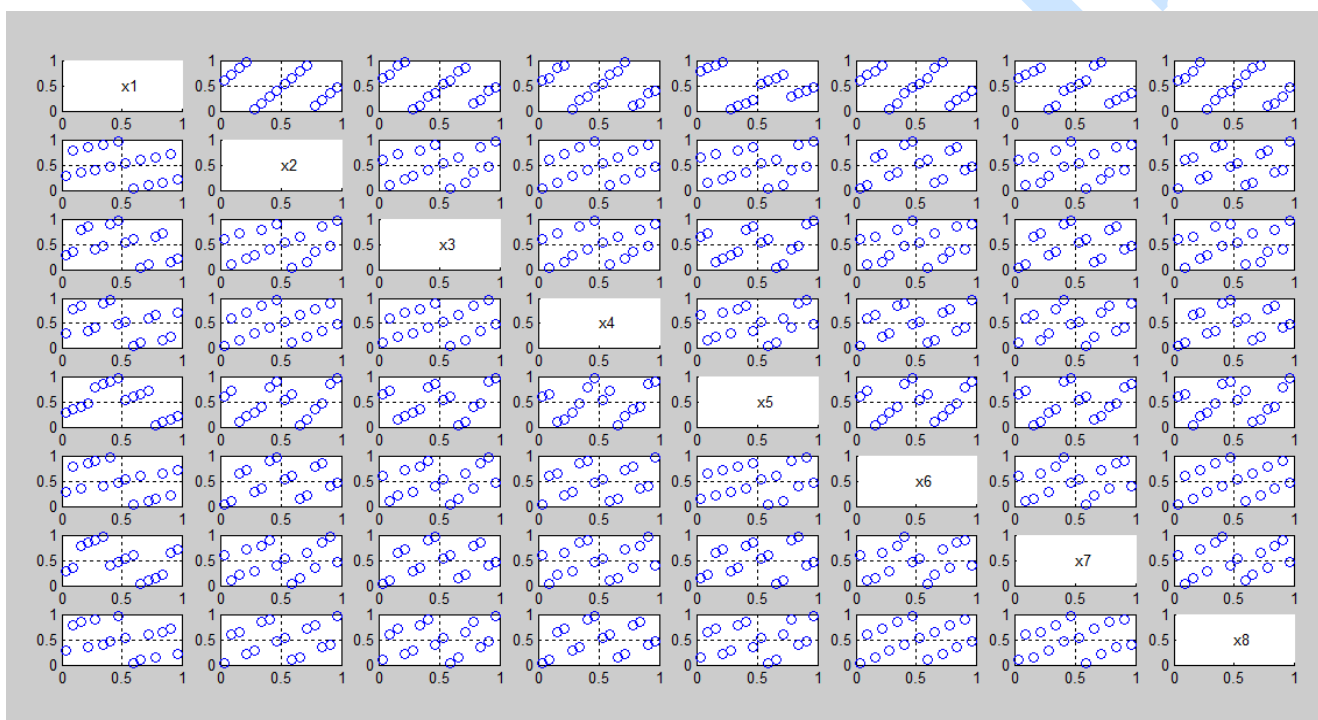


Figure 3: Projection Properties of OA (16, 8) LHD

4. DISCUSSION OF CONSTRUCTED OALHDs

The results of OALHDs in Section 3.1 shows the construction of OA (8, 7) LHD for Hadamard matrix of order 4λ when λ equals 2. The OA (8, 7) LHD has 8 runs with 7 factors. The first column of the Hadamard matrix of order 4λ ($H_{4\lambda}$) is deleted by the program to achieve the desired orthogonal array Latin hypercube designs. In this case, this study has constructed OALHD for the strength of 2 only at 2 levels.

The OA (16, 8) LHD in Section 3.2 has 16 runs (rows) with 8 factors (columns). The program halves the Hadamard matrix of order 8λ ($H_{8\lambda}$) to achieve the desired orthogonal array Latin hypercube designs. This method has constructed OALHDs for the

strength of 3 only at 2 levels. Table 1 summarises the results.

Table 1: Some orthogonal array based Latin hypercube designs (n, m) constructed from orthogonal array (n, m, s, t) obtained using Hadamard matrices of Orders 4λ and 8λ

OA (n, m, s, t)	OALHD(n,m)
(8, 7, 2, 2)	(8, 7)
(16, 8, 2, 3)	(16, 8)

5. CONCLUSION

Many authors and researchers have paid particular attention to the construction of Orthogonal Array Based Latin Hypercube Designs (OALHDs). There

are several techniques available for the construction of space-filling designs. These include the use of special permutations, difference matrices, Galois fields and orthogonal arrays among others.

In this study, a MATLAB program was written to construct orthogonal array based Latin hypercube designs using Hadamard matrices of orders 4λ and 8λ . A technique described by Hedayat et al. ([HSS99]) was thereafter used, via the MATLAB program to obtain their corresponding orthogonal arrays which gave rise to OA (n, m, t) LHD where $t=1$ in this case and the advantage is that the OALHDs constructed achieved stratification.

The OALHDs examples given in this study achieved better space-filling properties as depicted in Figures 2 and 3. This study has made the construction of OALHDs easier by writing a computer program that runs in a few seconds to get a desired design. The program is optimised using the maximin distance criterion that employed K- Nearest Neighbour (KNN) search with Euclidean distance. The algorithm proposed can produce numerous results of construction of OALHDs beyond the two examples given in this study. This study concludes that OALHD constructed in this study can be used whenever interest is focused on performing either a traditional or computer experiment on real life situations.

REFERENCES

- [BB52] **Bose R. C., Bush K. A.** - *Orthogonal arrays of strength two and three.* Annals of Mathematical Statistics 23, 508–524, 1952.
- [B+02] **Bayarri M., Berger J. O., Higdon D., Kennedy M., Kottas A., Paulo R., Sacks J., Cafeo J., Cavendish J., Tu J.** - *A framework for the validation of computer models.* Proceedings of the Workshop on Foundations for V&V in the 21st Century, D. Pace and S. Stevenson (eds.), Society for Modelling and Simulation International, 2002.
- [Cha11] **Challenor P.** - *Designing a computer experiment that involves switches.* Journal of Statistical Theory and Practice 5, 47-57, 2011.
- [HSS99] **Hedayat A. S., Sloane N. J. A., Stufken J.** - *Orthogonal Arrays: Theory and Applications.* Springer-Verlag, New York, 1999.
- [JMY90] **Johnson M., Moore L., Ylvisaker D.** - *Minimax and maximin distance design.* J. Statist. Plann. Inference 26, 131-148, 1990.
- [Kuh09] **Kuhfeld W. F.** - *Orthogonal arrays.* Website courtesy of SAS Institute, 2009.
- [KT04] **Kharaghani H., Tayfeh-Rezaie B.** - *A Hadamard matrix of order 428,* Journal of Combinatorial Designs 13 , 435–440, 2004.
- [LW97] **Li W. W., Wu C. F. J.** - *Columnwise-pairwise algorithms with applications to the construction of supersaturated designs.* Technometrics 39, 171–179, 1997.
- [LBK03] **Leary S., Bhaskar A., Keane A.** - *Optimal orthogonal array-based Latin hypercubes.* J. Applied Statist. 30 585-598, 2003.
- [MM95] **Morris M. D., Mitchell T. J.** - *Exploratory designs for computer experiments.* Journal of Statistical Planning and Inference, 43, 381-402.
- [MW06] **Mukerjee R., Wu C. F.** - *A Modern Theory of Factorial Designs.* Springer Verlag, 2006.
- [MBC79] **McKay M. D., Beckman R. J., Conover W. J.** - *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,* Technometrics, 21, 239 -245, 1979.
- [OYA14a] **Osuolale K. A., Yahya W. B., Adeleke B. L.** - *An algorithm for constructing space-filling designs for Hadamard matrices of orders 4λ and 8λ .* Book of Abstract of the 49th Annual Conference of the Science Association of Nigeria, University of Ilorin, 2014.
- [OYA14b] **Osuolale K. A., Yahya W. B., Adeleke B. L.** - *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, 2014.

- [OYA15] **Osuolale K. A., Yahya W. B., Adeleke B. L.** - *Construction of orthogonal array-based Latin hypercube designs for deterministic computer experiments.* Annals. Computer Science Series, 13(1), 24-29, 2015.
- [Pat54] **Patterson H. D.** - *The error of lattice sampling.* Journal of Royal Statistical Society. Series B 16,140-149, 1954.
- [Rao46] **Rao C. R.** - *Hypercube of strength 'd' leading to confounded designs in factorial experiments.* Bull. Calcutta Math. Soc., 38, 67-78, 1946.
- [Sti03] **Stinson D. R.** - *Combinatorial Designs: Construction and Analysis.* New York: Springer, 2003.
- [Str03] **Strogatz S.** - *The real scientific hero of 1953,* New York Times. March 4, Editorial/Op-Ed, 2003.
- [Tan93] **Tang B.** - *Orthogonal array-based Latin hypercubes.* Journal of the American Statistical Association 88, 1392–1397, 1993.
- [Tan94] **Tang B.** - *A Theorem for selecting OA-based Latin hypercubes using a distance criterion.* Comm. Statist.: A Theory and Methods 23 2047-2058, 1994.
- [YO16] **Yahya W. B., Osuolale K. A.** - *A method of constructing orthogonal array-based Latin hypercube designs for computer experiments.* International Journal of Hybrid Information Technology, 9(6), 253-262, 2016.

Abridged Version of MATLAB Code for the Construction of Orthogonal Array Latin Hypercube Designs

```
function [D,L] = oa_hadamard(n,t,varargin)
switch t
case 2
    H = hadamard(n);
    D = (H(:,2:end) + 1)/2;
    f = n-1;
case 3
    H = hadamard(n);
    D = (H(:,end-(n/2)-1):end) + 1)/2;
    f = n/2;
end
S = RandStream('mt19937ar','Seed',0); % Create random stream
criteria = 'maximin';
iter = 10;
plot_status = true;
plot_type = 'bivariate';

for ii = 1:2:length(varargin)
    switch varargin{ii}
    case 'criteria'
        criteria = varargin{ii+1};
    case 'maxiter'
        iter = varargin{ii+1};
    case 'max_factors'
        f = varargin{ii+1};
    case 'plot'
        plot_status = logical(varargin{ii+1});
    case 'plot_type'
        plot_type = varargin{ii+1};
    end
end

% -----Generate Latin Hypercube design-----
if isempty(D)
    return
end
for jj = 1:size(D,2)
    d(:,jj) = rank(D(:,jj));
end
N = n;
L = (d - 0.5)/N;
% -----Optimise Space filling-----
bestscore = score(L,criteria);
switch criteria
case 'maximin'
    for ii = 2:iter
        LL = (L - rand(S))/N;
        newscore = score(LL,criteria);
        if newscore > bestscore
            L = LL;
            bestscore = newscore;
        end
    end
end
end
```



```
% -----Plot latin hypercube-----  
if ~plot_status  
    return  
end  
if strcmpi(plot_type,'bivariate')  
    for ii = 1:f  
        for jj = 1:f  
            ind = (ii-1)*f + jj;  
            if ii ~= jj  
                subplot(f,f,ind)  
                plot(L(:,jj),L(:,ii),'o')  
                axis(gca,[0 1 0 1])  
                grid(gca,'on')  
            else  
                subplot(f,f,ind)  
                text(0.5,0.5,sprintf('x%d',ii))  
            end  
        end  
    end  
else % monovariate  
    for ii = 1:f  
        for jj = 1:f  
            ind = (ii-1)*f + jj;  
            if ii ~= jj  
                subplot(f,f,ind)  
                text(0.5,0.5,sprintf('x%d-x%d',ii,jj))  
            else  
                subplot(f,f,ind)  
                plot(L(:,jj),L(:,ii),'o')  
                axis(gca,[0 1 0 1])  
                grid(gca,'on')  
            end  
        end  
    end  
end  
end  
end  
end  
  
%% Score function  
function s = score(L,criteria)  
    switch lower(criteria)  
        case 'maximin'  
            [~,dist] = knnsearch(L,L,'k',2);  
            s = min(dist(:,2));  
  
%% Rank Function  
function r=rank(x)  
    % Similar to tiedrank, but no adjustment for ties here  
    [~, rowidx] = sort(x);  
    r(rowidx) = 1:length(x);  
    r = r(:);  
end  
  
%% Takeout Function  
end
```