Algorithmic construction of optimal symmetric Latin hypercube designs

Kenny Q. Ye
Department of Applied Mathematics and Statistics
SUNY at Stony Brook
Stony Brook, NY 10021

William Li
Department of Operations and Management Science
University of Minnesota
Minneapolis, MN 55455

Agus Sudjianto
Reliability Methods Department
Ford Motor Company
Dearborn, MI 48121

Abstract

We propose symmetric Latin hypercubes for designs of computer exepriment. The goal is to offer a compromise between computing effort and design optimality. The proposed class of designs has some advantages over the regular Latin hypercube design with respect to criteria such as entropy and the minimum intersite distance. An exchange algorithm is proposed for constructing optimal symmetric Latin hypercube designs. This algorithm is compared with two existing algorithms by Park (1994) and Morris and Mitchell (1995). Some examples, including a real case study in the automotive industry, are used to illustrate the performance of the new designs and the algorithms.

Key Words: Computer experiment; Maximum Entropy Design; Maximin design

1 Introduction

One of our recent projects is concerned with the thermal analysis of multi-layer electrical traces at a major automotive company. As more and more electronic devices are installed in vehicles, the peak temperature of electrical traces becomes a major concern in designing the instrument panels. The temperature of an electrical trace is largely determined by its width, its passing current strength, and its position in a stack of traces. The goal of this project is to provide guidelines for design engineers for width and passing current strength of multi-layer electrical traces. Physical experiments are inevitably very expensive and time consuming since a set of electrical traces has to be assembled in certain configurations for each test and measuring the temperature of each trace is difficult. Therefore, finite element analysis

(FEA) models have been developed to simulate the thermal dynamics of electrical traces.

Using the computer model, the study starts from a simple case, in which there are two layers with three traces on each layer. One primary interest is the interaction between a center trace and an edge trace on two different layers since the heat coming off the center trace spreads out and affects the temperature at the edge. A center trace on layer 1 and an edge trace on layer 2 are selected in the study. The goal is to predict their peak temperatures (y_1 and y_2) based on four predictors: the width of the center trace (x_1), the applied current of the center trace (x_2), the width of the edge trace (x_3), and the applied current of the edge trace (x_4). Given a set of x_i-values, the computer model generates a deterministic peak temperature for each trace.

Though computer experiments are much cheaper and faster than physical experiments, each run is still time consuming and expensive. Thus, only a small number of combinations of the x_i can be tested. In this case, the experiment is to be conducted by another company which specializes in thermal dynamic computer models, and the budget only allows for 25 runs. A feasible approach is to establish a statistical model from the results of the 25 runs and then use it to predict peak temperatures for any given combinations of the x_i. An optimal Latin hypercube design was chosen for this experiment.

A Latin hypercube design (LHD) is an $n \times l$ matrix in which each column is a random permutation of $\{1, \ldots, n\}$ which can be mapped onto the actual range of the variables. It has good projection properties on any single dimension. Latin hypercube designs have been applied in many computer experiments since they were proposed by McKay *et al.* (1979). In practice, a LHD can be randomly generated, but a randomly selected LHD may have bad properties and act poorly in estimation and prediction. Another approach is to use optimal designs according to some criteria such as entropy (Shewry and Wynn 1987), Integrated Mean Squared Error (IMSE) (Sacks et al. 1989), and minimum intersite distance (Johnson *et al.* 1990). These designs have been shown to be efficient for certain models. However, the computational cost of obtaining these designs is high. In an attempt to offer a compromise between good projective properties of LHDs and a criterion, Park (1994) and Morris and Mitchell (1995) proposed optimal Latin hypercube designs. For an excellent review of design and analysis of computer experiments, see Koehler and Owen (1996).

One of the criteria considered in this paper is the entropy criterion, first proposed by Shewry and Wynn (1987) and then adopted by Currin *et al.* (1991). The response of a computer model is modeled by $Y(\mathbf{x}) = \sum_{j=1}^{k} \beta_j f_j(\mathbf{x}) + Z(\mathbf{x})$, in which $Z(\mathbf{x})$ is a Gaussian process with mean zero and covariance

$$R(\mathbf{s}, \mathbf{t}) = \sigma^2 \exp\left\{-\theta \sum_{j=1}^{l} |\mathbf{s}_j - \mathbf{t}_j|^q\right\}, \quad 0 < q \leq 2 \tag{1}$$

between two l-dimensional inputs $\mathbf{s}$ and $\mathbf{t}$. The entropy criterion is equivalent to the minimization of $-\log|R|$, where R is the covariance matrix of the design. The parameters θ and q determine the properties of $Z(x)$. Throughout this paper, we set $q = 2$ so that the correlation between two sites is a function of their L_2 distance.

The construction of an optimal LHD can still be time consuming. For example, to generate a maximum entropy 25×4 LHD using a columnwise-pairwise (CP) algorithm (discussed in Section 3), the whole procedure takes 3.3 hours on a Sun SPARC 20 workstation, which appears to be quite long as the size of the design is moderate. The search for a larger design would take even longer, and may be computationally prohibitive. This situation motivated us to look for alternatives that require less computing time. The easiest method is to generate a large number of random LHDs and then choose the best one according to the criterion. For example, the generation of 1000 random LHDs takes only 14.7 seconds on the same machine. However, the best design obtained from these random designs is usually significantly inferior to that produced by the algorithmic search. In our example, the entropy value at $\theta = 2$ of the former is 25.26, compared with the latter's 20.48. To reduce the searching time and still generate competitive designs, our approach is to restrict the search within a subset of the general LHD. If this subset of designs has some desirable properties with respect to a criterion, then selecting a design from this group of designs may be more efficient.

In Section 2, we introduce a new class of LHD, the *symmetric Latin hypercube design*, whose geometric property enables us to find optimal LHDs more efficiently. Section 3 considers a simple exchange algorithm for constructing optimal symmetric LHDs. Its performance is compared with the existing algorithmic approaches of Park (1994), and Morris and Mitchell (1995). Section 4 demonstrates the performance of the new design with an example. A summary is given in Section 5.

2 Symmetric Latin hypercubes

Our goal is to find a special type of LHD that has some good "built-in" properties with respect to the optimality of a design. In our definition, a

LHD is called a *Symmetric Latin hypercube design* (SLHD) if it has the following property: in an $n \times l$ LHD with levels from 1 to n, if $(a_1, a_2, \cdots, a_l)$ is one of the rows, then the vector $(n+1-a_1, n+1-a_2, \cdots, n+1-a_l)$ must be another row in the design matrix. In other words, if $\mathbf{t}_i$ is a design point in a SLHD, then there exists another point $\mathbf{t}_j$ in the design that is the reflection of $\mathbf{t}_i$ through the center. An example of a 10×5 SLHD is given in Table 1, in which the i^{th} row is the symmetric point of the $(n+1-i)^{th}$ row.

1	2	3	4	5
1	6	6	5	9
2	2	3	2	4
3	1	9	7	5
4	3	4	10	3
5	7	1	8	10
6	4	10	3	1
7	8	7	1	8
8	10	2	4	6
9	9	8	9	7
10	5	5	6	2

Table 1: A 10×5 symmetric Latin hypercube design

The symmetry of a SLHD provides some orthogonal properties. That is, the estimation of the linear effect of each variable in a SLHD is uncorrelated with all quadratic effects and bi-linear interactions. The proof of such properties can be found in Ye (1998), in which Orthogonal Latin Hypercube

Designs (OLHD) are constructed and proposed. OLHDs have the same symmetric properties but also process additional orthogonality which insures the zero correlation among estimation of linear effects. Therefore, one can view the SLHD as a generalization of the OLHD. However, the number of runs in an OLHD has to be a power of two, which increases dramatically as the number of factors increases. In the cases that an appropriate OLHD can not be found under the constraint of run size, one can consider using SLHDs which have the flexibility of the run size, yet retain some of the orthogonality of an OLHD.

Are SLHDs better than regular LHDs with respect to design criteria? Many optimal LHDs reported by Park (1994) and Morris and Mitchell (1995) have some symmetric properties. In particular, Morris and Mitchell (1995) noticed that a large number of the optimal LHDs they obtained are SLHDs and referred them as "foldover designs". This is the first time the SLHD is mentioned in the literature. They called for a thorough investigation on this phenomenon. Intuitively, the optimal designs are considered to have good space filling properties, and a good space filling design probably has some degree of symmetry. To verify this, we undertook a simulation study to compare random SLHDs to random LHDs with respect to both entropy and minimum intersite distance criteria. Table 2 compares the best design among the 1000 random SLHDs of size 25 with that of the 1000 random regular LHDs. The former has a smaller entropy criterion value of 23.60 at $\theta = 2$, compared with the latter's 25.26. In the table, we also list the minimum distances of three designs, a criterion first proposed by Johnson *et al.* (1990) and then used by Morris and Mitchell (1995) in constructing optimal LHDs. For a design S, the minimum distance $d^*(S) = \min_{\mathbf{s},\mathbf{t} \in S} d(\mathbf{s}, \mathbf{t})$, where $\mathbf{s}$ and $\mathbf{t}$ are two design points (*i.e.,* two rows in the design). Both the L_1 (rectangular) distance $L_1(\mathbf{s}, \mathbf{t}) = \sum_{j=1}^{l} |\mathbf{s}_j - \mathbf{t}_j|$ and L_2 (Euclidean) distance $L_2(\mathbf{s}, \mathbf{t}) = [\sum_{j=1}^{l} (\mathbf{s}_j - \mathbf{t}_j)^2]^{\frac{1}{2}}$ of three designs are listed in Table 2. A design S_1 is said to be better than design S_2 if $d^*(S_1) > d^*(S_2)$. The number of pairs separated by this distance, denoted J, is shown in parentheses in the table. If two designs have the same d^* values, then the design with smaller J value is better. Throughout this paper, entropy and distances are computed after LHDs are scaled into $[0, 1]^l$. The levels of scaled LHDs are $\{0, \frac{1}{n-1}, \frac{2}{n-1}, \cdots, 1\}$, which are the same used by Morris and Mitchell (1995). Note that Park (1994) used a different scale, $\{\frac{1}{n}, \frac{2}{n}, \cdots, 1\}$.

Tables 3 and 4 provide more comparisons between regular LHDs and SLHDs. We study Latin hypercubes with six different sizes, 25×4, 20×6,

design	entropy	min. L_1 distance	min. L_2 distance
best random LHD	25.26	0.50 (1)	0.29 (1)
best random SLHD	23.60	0.63 (2)	0.35 (2)
optimal LHD	20.48	0.75(1)	0.50(1)

Table 2: Comparison of three 25×4 LHDs by (i) entropy criterion($\theta = 2$); (ii) smallest L_1 distance and the number of pairs separated by that distance (in parentheses); and (iii) smallest L_2 distance and the number of pairs separated by that distance. The "best" is in terms of the entropy criterion.

	LHD		SLHD	
Size	Mean	Max	Mean	Max
25×4	0.3478	0.5417	0.3944	0.625
20×6	0.8205	1.211	0.8968	1.316
50×10	1.316	1.735	1.407	1.816
200×20	2.9457	3.4925	3.0884	3.5678
500×30	4.8737	5.3567	5.0240	5.4188
1000×50	9.4014	9.9339	9.6016	10.2002

Table 3: Minimum L_1 distances of random SLHDs and random LHDs

50×10 , 200×20, 500×30, 1000×50. The sample sizes are 1000 for the first three designs and 100 for the last two designs. SLHDs are consistently superior to the corresponding LHDs with respect to both L_1 and L_2 distances. We also compare the means of minimum distance of LHDs and SLHDs using t-tests. The p-values are all smaller than 0.0001. Therefore, SLHDs are statistically significantly better than LHDs with respect to the minimum distance criteria.

	LHD		SLHD	
Size	Mean	Max	Mean	Max
25×4	0.1943	0.3200	0.2230	0.3727
20×6	0.3875	0.5642	0.4270	0.6316
50×10	0.5163	0.7061	0.5492	0.7210
200×20	0.8492	1.0568	0.8879	1.0281
500×30	1.1383	1.2812	1.1812	1.2960
1000×50	1.7327	1.8328	1.7658	1.8689

Table 4: Minimum L_2 distances of random SLHDs and random LHDs

These simulation studies have shown the advantages of "picking the winner" from SLHDs instead of regular LHDs. However, the best SLHD obtained by the "picking the winner" approach is usually inferior to the corresponding optimal design obtained by a searching algorithm, as shown in Table 2. In the next section, a simple exchange algorithm is presented to search optimal SLHDs.

3 An algorithm and examples

In this section, we review the two existing algorithms proposed by Park (1994) and Morris and Mitchell (1995), respectively, for constructing optimal LHDs. Then a columnwise-pairwise exchange algorithm (CP) is introduced in the context of the construction of optimal SLHDs. The similarities and differences between the CP and the other two algorithms are discussed.

Through examples, we compare

(1) the performance of the CP and other two algorithms;

(2) the optimal SLHDs and the optimal regular LHDs with respect to the design criteria and the searching time.

3.1 Existing algorithms

To construct optimal LHDs, Park (1994) presented an approach based on the the exchanges of several pairs of the elements in two rows. His algorithm first selects some active pairs which minimize the objective criterion value by excluding that pair from the design. Then for each chosen pair of two rows i_1 and i_2, the algorithm considers all the possible exchanges $x_{i_1j_1} \leftrightarrow x_{i_2j_1}, \ldots, x_{i_1j_k} \leftrightarrow x_{i_2j_k}$ for $k \leq l$ and finds the best exchange among them.

Morris and Mitchell(1995) adopted a simulated annealing algorithm to search for optimal LHDs. They also defined a maximin distance criterion. For a given design, define a distance list $\{d_1, d_2, \ldots, d_m\}, d_1 < d_2 < \ldots < d_m$, in which the d_i's are the distinct values of intersite distances. Let J_i be the number of pairs of sites in the design separated by d_i. Then a design is a maximin distance design if and only if

(1a) d_1 is maximized, and among the designs for which this is true, (1b) J_1 is minimized, and among the designs for which this is true,

(2a) d_2 is maximized, and among the designs for which this is true,

(2b) J_2 is minimized,

and so forth. Morris and Mitchell (1995) also pointed out that although this extended definition is intuitively appealing, it would be better to use a scalar-valued criterion as the driving criterion. For this purpose, they proposed a family of functions

$$\phi_p = \left[\sum_{j=1}^{m} J_j d_j^{-p} \right]^{1/p}, \tag{2}$$

where p is a positive integer. Normally, different p values are tried to obtain a maximin distance LHD.

In Morris and Mitchell's algorithm, a search begins with a randomly chosen LHD, and proceeds through examination of a sequence of designs, each generated as a perturbation of the preceding one. A perturbation D_{try} of a design D is generated by interchanging two randomly chosen elements within a randomly chosen column in D. The perturbation D_{try} replaces D if it leads to an improvement. Otherwise, it will replace D with probability $\pi = \exp\{-[\phi(D_{\text{try}}) - \phi(D)]/t\}$, where t is a preset parameter known as the "temperature".

3.2 Our algorithm

Li and Wu (1997) considered a class of columnwise-pairwise algorithms in the context of the construction of optimal supersaturated designs. A columnwise algorithm makes exchanges on the columns in a design and can be particularly useful for designs that have structure requirements on the columns. Note that each column in an n-run LHD is a permutation of $\{1, \ldots, n\}$. At each step, another permutation of $\{1, \ldots, n\}$ is chosen to replace a column so that the Latin hypercube structure is retained. Therefore, we adopt the columnwise-pairwise idea in searching for optimal LHDs. However, one important change has to be made to accommodate the special structures of the SLHD. For a SLHD *two simultaneous* pair exchanges are made in each column to retain the symmetry. For example, suppose a column in a 6-row SLHD is $(1, 2, 3, 4, 5, 6)'$. If element 1 is exchanged with i, element 6 must be exchanged with $n + 1 - i$ (i.e. $7 - i$) to keep the design symmetric. The only exception is when element i is exchanged with element $n + 1 - i$, which does not require a second exchange. The exchange procedure for a SLHD with an odd number of rows is slightly different. The center point of the design does not participate in the exchange. For example, if a column in a 7-row SLHD is $(1, 2, 3, 4, 5, 6, 7)'$, then element 4 may not to be exchanged with any other element.

The algorithm for searching optimal SLHD is summarized as follows:

1. Start with a random SLHD.

2. Each iteration has l steps. At the ith step, the best two simultaneous exchanges within column i are found. The design matrix is updated accordingly.

3. If the resulting design is better with respect to the criterion, repeat Step 2. Otherwise, it is considered to be an "optimal design", and the search is terminated.

The resulting optimal designs depend largely on the starting designs used in the algorithm. Hence, one should repeat the algorithm with several different random starting designs. The best design among the generated optimal designs is chosen to be the final design.

3.3 Examples

Example 1 *CP vs. Simulated Annealing*

The simulated annealing algorithm proposed by Morris and Mitchell (1995) aims at constructing optimal regular LHDs. We modify their algorithm to search for SLHDs. Similarly, the CP algorithm discussed in the previous section is modified to construct optimal regular LHDs. Both algorithms are columnwise-pairwise procedures. The simulated annealing algorithm operates on a (randomly chosen) column and then considers a (randomly chosen) pair in each column. Our proposed CP algorithm resembles the former with a very low starting temperature (so that switches to inferior designs are never made). An important difference is that the simulated annealing algorithm perturbs the design in a random manner, and our CP algorithm perturbs the design in a deterministic manner.

To compare their performances, we use both algorithms to construct optimal regular LHDs and optimal SLHDs. Two examples are considered. Table 5 lists the 12×2 maximin distance LHDs and SLHDs generated by both algorithms. The simulated annealing algorithm uses 10 starting designs and the CP uses 100 starting designs. The driving criterion is ϕ_p with $p = 50$. To compare the efficiency of the algorithms, the number of searched LHSs is also recorded. Both algorithms obtain equally good optimal designs. But the CP algorithm searches far fewer LHDs than the simulated annealing algorithm does. When both algorithms are used to construct 25×4 designs, the simulated annealing algorithm produces better optimal designs than the CP.

design	algorithm	min. dist.	# of searched LHDs
LHD	Simulated Annealing	.4545 (16)	269520
LHD	CP	.4545 (16)	44220
SLHD	Simulated Annealing	.4545 (16)	240416
SLHD	CP	.4545 (16)	14652

Table 5: Comparison of optimal 12×2 LHDs and SLHDs using two algorithms, CP (100 starting designs) and simulated annealing (10 starting designs). The search criterion is ϕ_p with $p = 50$ and L_2 distance. Note: using the simulated annealing search in LHD, only two out of 10 starting designs result in 0.4545 (16), compared to all 10 in the SLHD case.

design	algorithm	min. L_1 dist.	# of searched LHDs
LHD	Simulated Annealing	.9177 (19)	1537663
LHD	CP	.8750 (6)	2241900
SLHD	Simulated Annealing	.9583 (36)	1426985
SLHD	CP	.9177 (6)	546480

Table 6: Comparison of optimal 25×4 LHDs and SLHDs using two algorithms, CP (100 starting designs) and simulated annealing (10 starting designs). The search criterion is ϕ_p with $p = 50$ and L_1 distance.

Therefore, we may conclude that the systematic search algorithm is better for small designs and the simulated annealing algorithm is better for larger designs.

Example 2 *CP vs. Park*

Park's algorithm (1994) cannot be easily modified to accommodate the property of symmetry. Thus, its comparison with the CP is done through construction of the optimal 9×2 regular LHDs, which is discussed in detail by Park(1994) to illustrate his exchange algorithms. The CP algorithm is also used to construct 9×2 SLHDs. Table 7 compares the optimal designs generated by two algorithms with respect to the entropy criterion ($\theta = 1, 5, 25$), along with the total number of searched LHDs for each algorithm. Three interesting observations are apparent in this example:

1. The CP seems to consistently produce better LHDs than Park's algorithm with respect to entropy. The former also reaches the final design slightly earlier since it searches fewer LHDs. In fact, exhaustive searches reveal that the CP produces the global optimum for each value of $\theta = 1, 5, 25$. Our study of constructing LHDs of different sizes shows the same patterns.

2. Comparisons between optimal LHDs and the corresponding SLHDs show that the former are slightly better than the latter but take approximately 4 times as much time to search. However, for such a small design, it takes so little time (6 to 7 seconds on a Sun Sparc 20 workstation) for an exhaustive search in the whole LHD class. Therefore, there is no need to restrict the search within SLHD class.

3. At $\theta = 25$, the global optimal LHD is symmetric. Moreover, it is also an orthogonal Latin hypercube as constructed algebraically by Ye (1998).

θ	design	algorithm	optimal	average	# of searched LHDs
1	SLHD	CP	20.38	22.16	5488
	LHD	CP	19.16	19.29	20592
	LHD	Park	20.01	20.79	24132
5	SLHD	CP	3.09	4.35	4768
	LHD	CP	2.95	3.06	20052
	LHD	Park	3.42	3.79	23970
25	SLHD	CP	0.49×10^{-2}	0.83×10^{-2}	4784
	LHD	CP	0.49×10^{-2}	1.11×10^{-2}	19080
	LHD	Park	1.03×10^{-2}	4.67×10^{-2}	23580

Table 7: Comparison of three algorithms for generating 9×2 optimal LHDs with 100 random starting designs

Example 3 *LHD vs. SLHD*

We now revisit the case study at the beginning of this paper: the construction of a 25×4 LHD for the thermal analysis of electrical traces. A primary motivation of using the SLHD is to reduce the searching time. Since the number of possible exchanges of each column in a SLHD is much less than that for a regular LHD, it is expected that the exchange algorithm for the SLHD will use much less CPU time. This is confirmed when the algorithm is applied to the construction of the optimal 25×4 SLHD. Without the restriction to SLHD, it takes 13.3 hours and 10.6 hours for the algorithm to terminate using the entropy and ϕ_p criteria, respectively. Using the same number of starting designs (100), the optimal SLHDs are found only after 1.6

hours and 1.3 hours. The results are summarized in Table 8. Theoretically, the global optimal SLHD cannot be better than the global optimal LHD since the SLHD is a subset of the LHD. It is seen in our Example 2 that the obtained optimal LHDs are globally optimal verified by exhaustive search, but they do not always have the symmetrical structure. Morris and Mitchell (1995) use an exhaustive search to find maximin distance LHDs for many small designs. Not all those global optimal designs are symmetric. In practice, a globally optimal design is rarely obtained when the exhaustive search is not feasible. In our case, with much less searching time, the optimal SLHDs found are actually better than the two optimal LHDs obtained previously with respect to both entropy and minimum distance criteria. The maximum entropy SLHD has the criterion value of 18.53 compared with 20.48 for the previously obtained optimal LHD. The former also has the better (i.e. larger) minimum distance of 0.83, with six pairs separated by this distance. Using ϕ_p as the driving criterion, a SLHD was obtained with four pairs separated by the minimum intersite distance of 0.92, which is considerably better than the optimal maximin distance LHD previously found (six pairs separated by 0.83).

Now we revisit Table 6 and focus on the difference between SLHD and

design	criterion	entropy	min. L_1 dist.	CPU time(hrs)
LHD	entropy	20.48	.75 (1)	13.34
LHD	ϕ_p	23.52	.83 (6)	10.63
SLHD	entropy	18.53	.83 (6)	1.6
SLHD	ϕ_p	19.48	.92 (4)	1.28

Table 8: Comparison of optimal 25×4 LHDs vs. SLHDs with respect to entropy and ϕ_p. The entropy is calculated for $\theta = 2$.

regular LHD. For the simulated annealing algorithm, restricting the search within the SLHD did not save much searching time, but the obtained design is significantly better with respect to the minimum distance criterion. For the CP algorithm, there is a dramatic reduction in searching time after we restrict the search within the SLHDs, yet the obtained design is much better. It is also interesting to observe that in far less time the CP found a better design within the SLHDs than the simulated annealing algorithm found within general LHDs.

4 A robust design simulation example

One of the goals in computer experiments is prediction. The Kriging method was developed in geostatistics and brought into computer experiment by Sacks *et al.* (1989) and Currin *et al.* (1991) to predict untested sites in the experimental regions. It models the response as a Gaussian process. Given a correlation function of the process, the best linear unbiased predictor of y at site x is given by

$$\hat{y}(x) = \hat{\mu} + \mathbf{r}^T R^{-1}(\mathbf{y} - \hat{\mu}\mathbf{1}),$$

where $\mathbf{r}$ is the vector of correlations between x and the design sites x_i, $\mathbf{R}$ is the correlation matrix among design sites, and $\mathbf{y}$ is the vector of the observed responses. In this section, an example is used to illustrate the advantages of using optimal SLHDs for Kriging methods.

The example presented here is taken from Mori (1985), which was originally presented as a robust design case study. It was later used by Li and Wu (1999) to illustrate an integrated approach to parameter design and tolerance design. The original study is concerned with the design of cyclones, which are used to separate solid mass and gaseous mass in chemical engineering. There are seven variables whose original values are given by $(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0.1, 0.3, 0.1, 0.11.5, 16.0, 0.75)$. The relation between the response, the diameter of a cyclone (y), and these seven variables is

$$y = 174.42(\frac{x_1}{x_5})(\frac{x_3}{x_2 - x_1})^{0.85} \times \sqrt{\frac{1 - 2.62\{1 - 0.36(\frac{x_4}{x_2})^{-0.56}\}^{3/2}(\frac{x_4}{x_2})^{1.16}}{x_6 x_7}}. \quad (3)$$

The range of the input variables in this example is taken to be $1 \pm 10\%$ of the original value (see Table 9).

Unscaled Input	Lower limit	Upper limit
x_1	0.09	0.11
x_2	0.27	0.33
x_3	0.09	0.11
x_4	0.09	0.11
x_5	1.35	1.65
x_6	14.4	17.6
x_7	0.675	0.825

Table 9: Range of the inputs in equation (3)

Experiments with 16 runs are performed using (1) 100 random LHDs; (2) 100 random SLHDs; (3) maximum entropy SLHDs generated by the CP with $\theta = 0.05, 0.5, 1$; and (4) a maximin L_2 distance SLHD generated by the CP with ϕ_{50}. In each experiment, using the Kriging method with the correlation function given in equation (1) with $\theta = 0.05, 0.1, 0.5$, we predict $Y(x)$ at the same 400 randomly selected sites. The mean squared error (MSE) of predictions at these 400 sites was calculated for each experiment. The results are summarized in Table 10. First, it can be seen that the MSE is sensitive to the θ used in the Kriging model but is insensitive to the optimal design criterion. Second, all of the optimal designs are better than the random designs. Third, in this case, SLHDs do not always outperform LHDs.

In practice, the choice of correlation function in the Kriging model is

Design	Correlation parameter in kriging model $\theta = 0.05$	$\theta = 0.1$	$\theta = 0.5$
random SLHD (mean of 100)	0.022	0.027	0.058
random LHD (mean of 100)	0.024	0.026	0.052
Max. Entropy($\theta = 0.05$) SLHD	0.016	0.017	0.025
Max. Entropy($\theta = 0.5$) SLHD	0.017	0.018	0.026
Max. Entropy($\theta = 1$) SLHD	0.019	0.020	0.027
Maximin Distance SLHD	0.020	0.020	0.028

Table 10: Square root of MSE for Maximum Entropy SLHD, Maximin distance SLHD, random LHD and SLHD over 400 randomly selected reference sites

complicated and crucial to prediction accuracy. Sacks *et al.* (1989) suggested using maximum likelihood estimate of θ. However, the modeling process should not be limited to Kriging. One advantage of using Latin hypercube designs is that they can facilitate almost any kind of model, parametric and non-parametric. Authors of this paper have used MARS (multivariate adaptive spline regression), GAM (generalized additive models) and second order polynomials to analyze computer experiments.

The cyclone study was originally a case study in robust design. We choose this study to demonstrate the link between computer experiments and robust designs. Robust design studies can also be carried out using computer models as presented by Welch *et al.* (1990). Orthogonal Latin Hypercube design and Symmetric Latin Hypercube design can be used in a robust design study as well. One can follow the response model approach of robust designs as proposed in Welch *et al.* (1990) and Shoemaker, Tsui and Wu (1991). First, establish a prediction model for both control and noise factors. Then, given the distribution of noise variables, estimate the variation of Y for each combination of control variables using the model obtained at the first stage. If a computer experiment is not expensive, one can skip the first step and estimate the variation caused by the noise variable directly using the computer model. In that case, Latin hypercubes can serve as a sampling mechanism to obtain samples from noise variable, as it was first proposed by McKay *et al.* (1979).

5 Summary and Discussion

This paper proposes a class of symmetric Latin hypercube designs (SLHDs), referred previously by Morris and Mitchell(1995) as "foldover designs", and a new columnwise-pairwise (CP) algorithm for searching optimal design within the SLHD class as well as within the regular LHD.

We summarize the properties of SLHDs as follows.

1. They are a good subset of LHDs with respect to both entropy and

maximin distance criteria (see Tables 2-4).

2. As a generalization of Orthogonal Latin hypercube designs, SLHDs retain some orthogonality. The estimation of quadratic effects and bilinear interactions is uncorrelated with the estimation of linear effects.

3. The searching time of the CP algorithm is greatly reduced by restricting the search within the SLHDs (see Tables 5-8). The restriction does not significantly reduce the searching time of the simulated annealing algorithm, but it often leads to better designs (See Table 8).

4. The global optimal LHD is not always a SLHD. Morris and Mitchell (1995) did an exhaustive search to find the optimal LHDs of small sizes. Not all of the true optimal designs they found are symmetric.

Despite the fact that the true optimal LHDs do not necessarily fall into the symmetric class. We recommend using the SLHD in computer experiments for two reasons. First, users will benefit from the orthogonal properties of SLHD as summarized above when they try to fit the data with a polynomial model. A non-symmetric LHD does not have such orthogonality. Second, as shown in Tables 6 and 8, by restricting the search within SLHDs, one could obtain approximately optimal designs in a more efficient manner for moderate to large-size designs. In fact, in these cases, an exhaustive search is usually prohibitive and one should be less concerned about whether a search method has the potential to reach the global optimum and more about how it can obtain a good design with reasonable computing effort. Especially for computer experiments, extra computing power could be spent on additional runs rather than obtaining a slightly better design.

The performances of the three algorithms for searching optimal LHDs are summarized as follows:

1. The CP algorithm consistently outperforms the algorithm of Park (1994).

2. For smaller designs, the CP algorithm is more efficient than the simulated annealing algorithm of Morris and Mitchell (1995). However, the latter can generate better large designs.

One of the referees suggested that we briefly comment on the performance of optimal LHDs compare to other types of designs proposed for computer experiments in recent literature, such as orthogonal arrays, OA based Latin hypercubes and quasi-Monte Carlo lattices. The comparison of different kind of designs is one of the most important problems and deserves a thorough investigation that is beyond the scope of this paper. However, we would be glad to share some of our opinions. Unlike traditional designs for which the models are in known forms, the computer experimenter often has little idea which model in his/her statistical toolbox will best describe his/her complex computer model before an experiment is done and several kind of models are tried. Most of the proposed designs for computer experiments allow users to try many different models, linear or nonlinear, parametric or non-parametric. Among those, orthogonal arrays may not be appropriate for computer experiments since they do not take the advantage of flexibility of computer experiment in terms of changing levels. Their projections to low dimensions are only a few points so that they are not good for non-parametric regression methods. However, they are good for fitting low-order polynomial models.

An optimal SLHD actually takes three criteria into consideration: the discrepancy of one-dimension projection optimized by the Latin hypercube structure, desired orthogonality inherited from the symmetric structure, and a third criterion (entropy or minimum distance) optimized through an algorithmic search. Therefore, we expect that optimal SLHD should perform very well with many modeling methods. Quasi-Monte Carlo lattice designs (Fang and Wang, 1994) are generated by some sequence which are asymptotically optimal in discrepancy measure. Since it spreads the design point evenly in the design space, it should have robust performance with different modeling methods. In particular, a *glp (good lattice point) set* is a Latin Hypercube. Bates *et. al* (1996) compared Latin hypercubes designs with lattice designs and found the quasi-Monte Carlo lattice design performed surprisingly well. Tang(1993) and Owen(1993) proposed a special type of Latin Hypercubes which are constructed based on orthogonal arrays. Such Latin hypercubes spread points evenly on t-dimensional projections. The actual dimension t depends on the strength of the original orthogonal array. However, this approach only provides LHD at the sizes of which orthogonal arrays exist. In Table 11, we compare three LHDs. The first one is the fourth optimal SLHD listed in Table 6. The second one is a *glp* set of generating vector (25;11,29,6,13). The third one is a LHD constructed based on $OA(25; 5^{4-2})$ using the procedure described in Tang (1993). It can be seen that in terms of entropy and minimum intersite distance, the optimal SLH is better than the *glp* and OA-based LH. The *glp*, however, is surprisingly good given the fact that it is easy to generate. Therefore, it could be a good choice if quick solutions to design problems in computer experiments are needed. On the other hand, the OA-based LH is far inferior to the other two designs. Since

	Optimal SLH	glp	OA-based LH
Minimum L_1	0.92(6)	0.75(24)	0.54(13)
Minimum L_2	0.46(2)	0.41(12)	0.29(13)
Entropy $\theta = 0.05$	33.03	39.09	50.17
Entropy $\theta = 1$	21.68	26.99	37.00
Entropy $\theta = 2$	19.82	24.97	34.77
Entropy $\theta = 5$	15.96	20.72	30.02
Entropy $\theta = 10$	13.50	17.96	26.89

Table 11: Comparison of three types of LHDs.

a class of LHD can be constructed based on an orthogonal array, a similar algorithmic approach should be developed to find a better design within the class.

We would like to see more research effort to compare those designs with respect to different performance measure. We think that a good design will not necessarily score the highest for any particular criterion but will be reasonably high for all the criteria.

Acknowledgments

We thank Professors J. S. Park and M. Morris for providing the computer codes of their algorithms. We also thank a referee for pointing out our mistake in calculating the entropy criterion using Park's algorithm. We would also like to thank the associate editor and another referee for encouraging us to include comparison of different types of designs in our paper. We are

also indebted to Professor Morris for his valuable comments on our related research work.

References

Bates, R.A., Buck, R.J., Riccomagno, E., Wynn, H.P. (1996). Experimental Design and Observation for Large Systems, *Journal of Royal Statistical Society*, Series B. **58**, 77-94

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

Fang, K.-T., Wang, Y. (1994) *Number-theoretic Methods in Statistics*, Chapman & Hall, London.

Johnson, M. and Moore, L. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26**, 131-148.

Koehler J. and Owen, A. (1996). Computer experiments. In S. Ghosh and C.R. Rao, editor, *Handbook of Statistics, 13: Design and Analysis of Experiments*, 261-308, 1996, North-Holland.

Li, W. and Wu, C. F. J., (1997). Columnwise-pairwise algorithms with applications to the construction of supersaturated designs. *Technometrics* **39**, 171-179.

Li, W. and Wu, C. F. J., (1999) An intergrated method of parameter design and tolerance design. *Quality Engineering* **11**, 417-425

McKay, M., Beckman, R. and Conover, W. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239-246.

Mori, T. (1985), *Case studies in experimental design*, (in Japanese).

Morris, M. and Mitchell, T. (1995). Exploratory design for computer experiments. *Journal of Statistical Planing and Inference* **43**, 381-402.

Owen, A.(1992), Orthogonal Arrays for Computer Experiments, Integration, and Visualization, *Statistica Sinica*, **2**, 439-452.

Park, J.-S. (1994). Optimal Latin-hypercube designs for computer experiments. *Journal of Statistical Planning Inference* **39**, 95-111.

Sachs, J., Schiller, S. B. and Welch, W. J. (1989). Designs for computer experiments. *Technometrics* **34**, 15-25.

Shewry, M. and Wynn, H. (1987). Maximum entropy design. *Journal of Applied Statistics* **14**, 2, 165-170.

Shoemaker, A. C., Tsui, L. L., and Wu, C. F. J. (1991). Economical experimentation methods for robust design. *Technometrics*, **33**, 415-427.

Tang, B (1993). Orthogonal Array-Based Latin Hypercubes, *Journal of American Statistical Association*, **88**, 1392-1397.

Welch, W. J., Yu, T. K., Kang, S. M. and Sacks, J. (1990). Computer experiments for quality control by parameter design. *Journal of Quality Technology*, **22**, 15-22.

Ye, K. Q.(1998), Column Orthogonal Latin hypercubes and their application in computer experiments. *Journal of American Statistical Association*, **93** 1430-1439.