Proceedings of DETC'03
ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference
Chicago, Illinois USA, September 2-6, 2003

DETC2003/DAC-48760

AN EFFICIENT ALGORITHM FOR CONSTRUCTING OPTIMAL DESIGN OF COMPUTER EXPERIMENTS

Ruichen Jin, Wei Chen*
Integrated DEsign Automation Laboratory (IDEAL)
University of Illinois at Chicago

Agus Sudjianto
V-Engine Engineering Analytical Powertrain
Ford Motor Company

ABSTRACT

Metamodeling approach has been widely used due to the high computational cost of using high-fidelity simulations in engineering design. The accuracy of metamodels is directly related to the experimental designs used. Optimal experimental designs have been shown to have good "space filling" and projective properties. However, the high cost in constructing them limits their use. In this paper, a new algorithm for constructing optimal experimental designs is developed. There are two major developments involved in this work. One is on developing an efficient global optimal search algorithm, named as enhanced stochastic evolutionary (ESE) algorithm. The other is on developing efficient algorithms for evaluating optimality criteria. The proposed algorithm is compared to two existing algorithms and is found to be much more efficient in terms of the computation time, the number of exchanges needed for generating new designs, and the achieved optimality criteria. The algorithm is also very flexible to construct various classes of optimal designs to retain certain structural properties.
Key words: metamodeling, optimal design, computer experiments, stochastic evolutionary algorithm

1. INTRODUCTION

Metamodeling approach has been widely used due to the high computational cost of using high-fidelity simulations in engineering design. While the accuracy of an approximation is directly related to the metamodeling approach itself, design of computer experiments, or called sampling (for simulations) also has a considerable effect on the accuracy of a metamodel. As more details will be provided later, design of computer experiments could be formulated as an optimization problem for which finding the globally optimal design (locations of multiple samples) involves combinatorial exhaustive search and is computationally prohibitive even for a small dimensional problem. Developing an efficient sample construction algorithm for optimizing design of computer experiments is the focus of this paper.

It is generally believed that a good design for computer experiments should be "space-filling**"** which means that the sample points should spread out over the entire design space as evenly as possible to capture the design behavior. Because in most of problems only a small group of factors are virtually significant, it is also desired that there are no replicates or significant point-clustering in the projection of the design onto the subspace of significant (or called effective) factors. Tradeoff often has to be made between the aforementioned *space filling property* and the *projective property* in low-dimensional subspaces.

Various designs (or sampling techniques) have been used for computer experiments. Koehler and Owen (1996) provided a good review on design and analysis of computer experiments. Simpson, et al. (2001) compared five different experimental designs and four metamodeling approaches in terms of their capability to generate accurate approximations. Existing methods can be roughly put into two categories. One category of designs are constructed by combinatorial, geometrical or algebraic approaches, such as Latin hypercube designs (LHD) (McKay, et al., 1979), orthogonal arrays (OA) (Owen, 1992), orthogonal array-based Latin hypercube designs (Tang, 1993), etc. Those designs often have good projective property in low-dimensional subspaces; however, their sample points in high or full-dimensional space are scattered randomly. The other category of designs are constructed by algorithmic approaches under certain optimality criteria, such as minimax and maximin designs (Johnson, et. al., 1990), maximum entropy designs (Currin, et al., 1991), integrated mean squared-error (IMSE) designs (Sacks, et. al., 1989), and uniform designs (Fang and Wang, 1994). Those designs usually have good space-filling properties. However, obtaining those designs can be either difficult or computationally intractable. Some optimal designs may not have good projective properties in low-dimensional subspaces. For instance, Morris and Mitchell (1995) found that

* Corresponding Author, weichen1@uic.edu. After June 1, 2003, Mechanical Engineering, Northwestern University, Evanston, IL 60208-3111, weichen@northwestern.edu.

maximin distance designs are often concentrated in the corners of the design space when the number of points is relatively small compared to the number of variables.

To improve the low-dimensional projective property as well as to maintain a good computational efficiency in sampling, some researchers propose to search an optimal design within the class of LHDs, which have good projective properties in each dimension and still have much freedom in generating distinct candidate designs. Morris and Mitchell (1995) introduced optimal LHDs based the ϕ_p criterion (a variant of the maximin distance criterion); Parks (1994) introduced optimal LHDs based on either the maximum entropy criterion or the IMSE criterion; Fang, et al (2002) introduced optimal LHDs based on the Centered L_2 discrepancy criterion. Other classes of designs that have good projective properties in two-dimensional (or higher) subspace, e.g., OA-based LHDs, are also promising.

Searching the optimal design of experiments within a class of designs (e.g., LHD), even though more tractable than searching in the entire sample space without any restrictions, is still difficult to solve exactly. Exhaustive search method is computationally prohibitive even for a small problem. For example, for optimizing 10×4 LHDs (10 runs, 4 factors), the number of distinct designs is more than 10^{22}. It is more practical to solve optimal design (of experiments) problems approximately. Toward this effort, Morris and Mitchell (1995) adapted a version of simulated annealing (SA) algorithm for constructing optimal LHDs; Park (1994) developed a rowwise element exchange algorithm for constructing optimal LHDs; Ye, et al (2000) used the columnwise-pairwise (CP) algorithm (Li and Wu, 1997) for constructing optimal symmetrical LHDs; Fang, et al (2002) adapted the threshold accepting (TA) algorithm (essentially a variant of SA) in constructing optimal LHD. The optimal designs constructed by these algorithms have been shown to have a good space filling property. However, the computational cost of these existing algorithms is generally high. For example, Ye, et al (2000) reported that generating an optimal 25×4 LHDs using CP could take several hours on a Sun SPARC 20 workstation. For a design as large as 100×10, the computational cost could be formidable; thus, search processes often stopped before finding a good design.

In this paper, we propose an algorithm that is able to quickly construct a good design of experiments given a limited computational resource but also has the capability of moving away from a locally optimal design. There are two major developments involved. One is on developing an efficient global optimal search algorithm, named as the enhanced stochastic evolutionary (ESE) algorithm. The other is on developing efficient algorithms for evaluating optimality criteria (such as the entropy criterion, the ϕ_p criterion, and the CL_2 criterion) to facilitate the search of optimal experimental designs. The proposed method is especially useful for constructing median to large-sized design of experiments. For example, for a 100×10 LHD, the proposed algorithm is able to find a good design within minutes, if not within seconds. Furthermore, the algorithm is able to work on different classes of designs and retain certain special structural properties, e.g., the balance property of LHDs and the orthogonality of OA and OA-based LHDs. Due to the limited space, in this paper, we show only how it is used to optimize LHDs.

2. THE TECHNOLOGICAL BASE

An experimental design with n runs and m factors is usually written as an $n \times m$ matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_{2,...,}\mathbf{x}_n]^T$, where each row $\mathbf{x}_i^T = [x_{i1}, x_{i2}, ..., x_{im}]$ stands for an experimental run and each column stands for a factor or a variable. The optimal experimental design problem we are interested is to search a design $\mathbf{X}^*$ in a given design class $\boldsymbol{Z}$, which optimizes (for simplicity, minimization is considered) a given optimality criterion f, i.e,

$$\min_{\mathbf{X} \in \boldsymbol{Z}} f(\mathbf{X}) . \quad (1)$$

In Sections 2.1 and 2.2, descriptions of experimental designs with special structural properties and different optimality criteria will be given, respectively. Section 2.3 is an introduction of the existing sample construction algorithms.

2.1. Designs With Special Structural Properties

2.1.1. Balanced Design

A column in a design is *balanced* if the number of runs assigned to each level of the column is the same. A design is called a *balanced* design if all columns of the design are balanced (c.f., Li, 1997). The concept of balanced designs covers a wide range of designs of interest. For example, OAs and LHDs are special cases of balanced designs. A balanced design with n runs and m factors is denoted as: $U_n(q_1 q_2 ... q_m)$, where q_i is the number of levels for the i^{th} column. If all the q_i's in a balanced design are equal, the design is said to be *symmetrical* and denoted as $U_n(q^m)$; otherwise, it is said to be *mixed-level* or *asymmetrical*.

2.1.2. Latin Hypercube Design

A LHD (McKay et al., 1979) is an $n \times m$ matrix in which each column is a random permutation of {1, 2, …, n}. It has good projective properties on any single dimension. LHDs are special cases of symmetrical balanced designs with its level numbers equal to run numbers. LHDs have been applied in many computer experiments where all the factors (variables) are continuous. However, in the case that some factors are discrete or have to be fixed at certain given values, asymmetrical balanced designs are more appropriate: for continuous factors, the number of levels could be set to be equal to the number of runs; for other factors, the number of levels could be set based on the discrete levels of the factors.

2.1.3. Orthogonal Array (OA) and OA-based LHD (OL)

A design is called a strength-r orthogonal array and denoted as $OA_n(q_1 q_2 ... q_m; r)$, if *all possible* level combinations for any r factors appear equally often (Owen, 1992; Hedayat et al., 1999). OAs are special cases of balanced designs with orthogonality between columns. Geometrically, the projection of a strength-r OA onto a r-dimensional subspace of factors i_1, i_2,...,i_r will be a $q_{i1} \times q_{i2} \times .. q_{ir}$ grid. Strength-2 OAs have been extensively used for planning experiments in industry. However, when a large number of factors are studied but only a few of them are virtually significant, OA projected onto the subspace of the significant factors can result in replication of points. For example, each one-dimensional projection of an $OA_{16}(4^5;2)$ has only 4 distinct points. To avoid replications in projections, Tang (1993) proposed to use OA to construct an

improved LHD, called OA-based LHD (OL), which to some degree inherits both the *r*-dimensional uniformity of a strength-*r* OA and one-dimensional uniformity of LHD.

2.2. Optimality Criteria

Optimal criteria are used to achieve the space-filling property in design of computer experiments. Three widely used optimality criteria are considered in this work.

2.2.1. Maximin Distance Criterion and ϕ_p Criterion

A design is called a maximin distance design (Johnson, et al, 1990) if it maximizes the minimum inter-site distance:

$$\min_{1\le i,j\le n, i\ne j} d(\mathbf{x}_i,\mathbf{x}_j), \tag{2}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between two sample points $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$d(\mathbf{x}_i,\mathbf{x}_j)=d_{ij}=\left[\sum_{k=1}^{m}\left|x_{ik}-x_{jk}\right|^t\right]^{1/t}, t=1 \text{ or } 2. \tag{3}$$

Morris and Mitchell (1995) proposed an intuitively appealing extension of the maximin distance criterion. For a given design, by sorting all the inter-sited distance d_{ij} ($1\le i, j \le n$, $i \ne j$), a distance list (d_1, d_2, ..., d_s) and an index list (J_1, J_2,..., J_s) can be obtained, where d_i's are distinct distance values with $d_1<d_2<...<d_s$, J_i is the number of pairs of sites in the design separated by d_i, s is the number of distinct distance values. A design is called a ϕ_p-optimal design if it minimizes:

$$\phi_p=\left[\sum_{i=1}^{s}J_i d_i^{-p}\right]^{1/p}, \tag{4}$$

where p is a positive integer. With a very large p, the minimum distance d_1 will dominate all subsequent items. In that case, the ϕ_p criterion is equivalent to the maximin distance criterion.

2.2.2. Entropy Criterion

Shannon (1948) used entropy to quantify the "amount of information": the lower the entropy, the more precise the knowledge is. From the Bayesian viewpoint, the lower the posterior entropy, the smaller is the uncertainty in the prediction of the response at unobserved sites. Minimizing the posterior entropy is equivalent to finding a set of design points on which we have the least knowledge. It has been further shown that the entropy criterion is equivalent to minimizing the following (see, e.g., Koehler and Owen, 1996):

$$-\log|\mathbf{R}|, \tag{5}$$

where $\mathbf{R}$ is the correlation matrix of the experimental design matrix $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n]^T$, whose elements are:

$$R_{ij}=\exp\left(-\sum_{k=1}^{m}\theta_k\left|x_{ik}-x_{jk}\right|^t\right), 1\le i,j\le n; 1\le t\le 2, \tag{6}$$

where θ_k (k=1,..,m) are correlation coefficients.

2.2.3. Centered L_2 Discrepancy Criterion

The L_p discrepancy is a measure of the difference between the empirical cumulative distribution function of an experimental design and the uniform cumulative distribution function. In other words, the L_p discrepancy is a measure of non-uniformity of a design. Among L_p discrepancy, L_2 discrepancy is used most frequently since it can be expressed analytically and is much easier to compute. Hickernell (1998) proposed three formulas of L_2 discrepancy, among which the centered L_2-discrepancy (CL_2) seems the most interesting.

$$CL_2(\mathbf{X})^2=\left(\frac{13}{12}\right)^2-\frac{2}{n}\sum_{i=1}^{n}\prod_{k=1}^{m}\left(1+\frac{1}{2}\left|x_{ik}-0.5\right|-\frac{1}{2}\left|x_{ik}-0.5\right|^2\right)$$
$$+\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\prod_{k=1}^{m}\left(1+\frac{1}{2}\left|x_{ik}-0.5\right|+\frac{1}{2}\left|x_{jk}-0.5\right|-\frac{1}{2}\left|x_{ik}-x_{jk}\right|\right). \tag{7}$$

A design is called uniform design if it minimizes the centered L_2 discrepancy (Fang, et al, 2000).

2.3. Existing Algorithms for Constructing Experiments

A typical experiment-constructing algorithm searches a good design of experiments, represented by $\mathbf{X}$, repeatedly in the following procedure:

1. Start from a randomly chosen starting design $\mathbf{X_0}$;
2. Construct a new design (or a set of new designs) by some kinds of *updating operations* on the current design;
3. Compute the criterion value of the new design and decide whether to replace the current design with the new one.

Updating operations are critical in sample construction algorithms. There are two major types of operations, i.e., *rowwise* operations and *columnwise* operations. A rowwise operation changes a row of a design $\mathbf{X}$, while a columnwise operation changes a column. A review on columnwise and rowwise algorithms can be found in Li and Wu's (1997). We are interested in columnwise operations since they are particularly easier to keep the structure properties of a design in relation to columns, such as the balance and orthogonality properties introduced earlier in Section 2.1.

For LHDs or balanced designs in general, permuting (changing the values of) individual elements in a column may still retain the balance structure of the design but could largely change the 'current' design and therefore some good features of the current design will not be inherited and the optimization process may dramatically slow down. In this study, we focus on a particular type of columnwise operation, called *element-exchange*, which interchanges two distinct elements in a column and guarantee to retain the balance property. Take a 5×4 LHD for example:

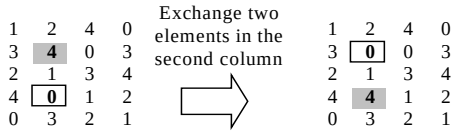

Figure 1. Element-exchange in a 5×4 LHD

Obviously, after the element-exchange, the balance property of 2nd column is retained, i.e., exactly one run is assigned to each of the five levels in the column. Therefore, the design induced by exchanging two elements in the second column of the LHD is still a LHD. Another advantage of using element-exchange, as to be shown in Section 3.2, is that the evaluation of an optimal criterion of a new design induced by an element-exchange can be very efficient. In the rest of this section, we review three existing optimization algorithms.

2.3.1. CP algorithm

The CP algorithm (Li and Wu, 1997) starts from an $n \times m$ randomly chosen design $\mathbf{X}$. Each iteration in the algorithm is divided into m steps. At the i^{th} step, the CP algorithm compares all possible distinct designs induced by exchanges in the i^{th} column of the current design $\mathbf{X}$, and selects the best design $\mathbf{X}_{try}$ from all those designs. If after an iteration, $\mathbf{X}_{try}$ is better than $\mathbf{X}$, i.e., $f(\mathbf{X}_{try}) < f(\mathbf{X})$, the procedure will be repeated; if no improvement is achieved at an iteration, the search will be terminated. The CP algorithm could quickly find a locally optimal design. However, depending on the starting design, the optimal design obtained could be of low quality. In practice, with the CP algorithm the optimization process needs to repeat for N_s cycles from different starting designs and the best design among the optimal designs from all cycles is selected.

2.3.2. SA algorithm

The SA algorithm (Morris and Mitchell, 1995) begins with a randomly chosen design, and proceeds through examination of a sequence of new designs, each generated by a randomly chosen element-exchange within a randomly chosen column of the current design $\mathbf{X}$. A new design $\mathbf{X}_{try}$ replaces $\mathbf{X}$ if it leads to an improvement. Otherwise, it will replace $\mathbf{X}$ with probability of $\exp\{-[f(\mathbf{X}_{try}) - f(\mathbf{X})]/T\}$, where T is a parameter called "temperature" in the analogous physical process of annealing of solids. Initial set to T_0, T will be monotonically reduced by some *cooling schedule*. Morris and Mitchell used $T' = \alpha T$ as the cooling schedule, where α is a constant called cooling factor here. SA usually converges slowly to a high quality design.

2.3.3. TA algorithm

The TA algorithm (Winker and Fang, 1998) is essentially a variant of the SA. Instead of accepting a new design with some probability, TA determines whether to accept a new design $\mathbf{X}_{try}$ by using a simple deterministic acceptance criterion: $f(\mathbf{X}_{try}) - f(\mathbf{X}) \le T_h$, where T_h is called "threshold". T_h is monotonically reduced based on some cooling schedule. TA has been used for constructing uniform designs (c.f., Fang 2000; Fang, et al, 2002).

3. PROPOSED ALGORITHM FOR CONSTRUCTING OPTIMAL EXPERIMENTAL DESIGN

In this section, a new algorithm for constructing optimal experimental design is presented. Using the columnwise element-exchange as the basic operation, our proposed algorithm can be used to find efficiently an optimal design that maintains the special structural property of a particular class of design, e.g., to obtain an optimal LHD when randomly choosing LHD as the starting design. To overcome the difficulties associated with the existing methods and to achieve much improved efficiency, our proposed method adapts and enhances a global search algorithm, i.e., the stochastic evolutionary algorithm (Section 3.1), and utilizes efficient algorithms for evaluating different optimality criteria (Section 3.2) that significantly reduces the computational burden.

3.1. Enhanced Stochastic Evolutionary (ESE) Algorithm

The enhanced stochastic evolutionary (ESE) algorithm is used in this work to control the entire process of searching optimal designs. The method is adapted and enhanced from the stochastic evolutionary (SE) algorithm, which was developed by Saab and Rao (1991) for general combinatorial optimization applications. Similar to TA, SE decides whether to accept a new design by a threshold-based acceptance criterion. However, the strategy (or schedule) for SE to change the value of threshold is different from TA or SA. The threshold T_h is initially set to a small value T_{h0}. The value is incremented based on certain 'warming' schedules only if it seems that the algorithm is stuck at a local optimum; whenever a better solution is found in the process of warming up, T_h is set back to T_{h0}. It is shown (Saab and Rao, 1991) that SE can converge much faster than SA and be capable of moving away from low quality local optimum to find a high quality solution. However, practically, it is often difficult to decide the value of T_{h0} and the warming schedule for different problems. The ESE algorithm developed in this work uses a sophisticated combination of warming schedule and cooling schedule to control T_h so that the algorithm can be self-adjusted to suit different experimental design problems (i.e., different classes of designs, different optimality criteria and different sizes of designs).

The ESE algorithm, as shown in Fig 2, consists of double loops, i.e., the inner loop and the outer loop. While the inner loop constructs new designs by element-exchanges and decides whether to accept them based on an acceptance criterion, the outer loop controls the entire optimization process by adjusting the threshold T_h in the acceptance criterion. In the entire process, $\mathbf{X}_{best}$ is used to keep track of the updated best design.

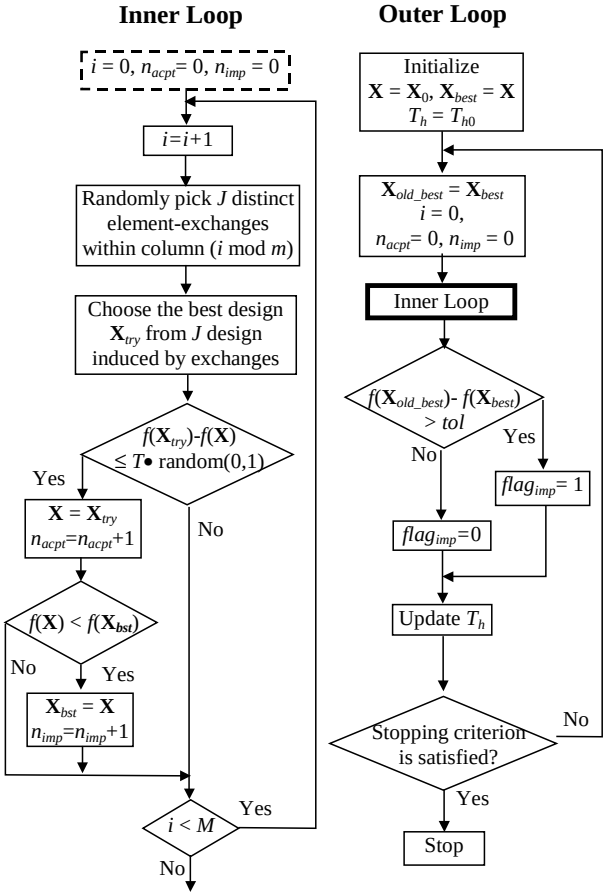

Figure 2. Flowchart of the ESE Algorithm

3.1.1. Inner Loop

The inner loop has M iterations. Generally, at iteration i, the algorithm randomly picks J distinct element-exchanges in i (mod m) column of the current design $\mathbf{X}$ and chooses the best design $\mathbf{X}_{try}$ based on the values of optimal criterion. If $\mathbf{X}_{try}$ is better than the current design $\mathbf{X}$, it will be accepted to replace $\mathbf{X}$; otherwise, $\mathbf{X}_{try}$ will be accepted to replace $\mathbf{X}$ if it satisfies the following acceptance criterion:

$$\Delta f \le T_h \cdot \text{random}(0,1), \quad (8)$$

where $\Delta f = f(\mathbf{X}_{try}) - f(\mathbf{X})$, random(0,1) is a function that generates uniform random numbers between 0 and 1 and $T_h > 0$ is a control parameter, which is called threshold here. If $\Delta f \ge T_h$, $\mathbf{X}_{try}$ will never be accepted and if $0 < \Delta f < T_h$, let S = random(0,1), then $\mathbf{X}_{try}$ will be accepted with probability:

$$P(S \ge \Delta f / T_h) = 1 - \Delta f / T_h . \quad (9)$$

With this acceptance criterion, a temporarily worse design could be accepted and a slightly worse design (i.e., a small Δf) is more likely to replace the current design than a significantly worse design (i.e., a large Δf). In addition, a given increase in criterion value is more likely to be accepted if T_h has a relatively high value. The setting of T_h will be discussed later. The values of parameters involved in the inner loop, i.e., J and M, are pre-specified.

Choice of J: This parameter is the number of distinct element exchanges generated at each iteration. The algorithm will compare the designs induced by those exchanges and find the best design. This treatment, as CP, could enable the algorithm rapidly find a locally optimal design. However, unlike CP, which compares all possible distinct designs induced by exchanges, our algorithm only randomly picks J distinct designs resulted from exchanges. This randomized behavior together with the acceptance criterion is intended to allow the search to escape from locally optimal designs. Based on our testing experience, too large of J may make it more possible to be stuck in a locally optimal design for small-sized designs and lead to low efficiency for large-sized designs. In our test, we set J to be n_e/5 but no large than 50, where n_e is the number of all possible distinct element-exchanges in a column ($\binom{n}{2}$ for a LHD and $\binom{q_i}{2} \times (n/q_i)^2$ for a balanced design). For mixed-level balanced designs, the values of J will be different for different columns.

Choice of M: The parameter is the number of iterations in the inner loop, i.e., the number of tries the algorithm will make before going on to the next threshold T_h. It seems reasonable that M should be larger for larger problems. In our test, we set M to be $2n_e m / J$ but no larger than 100.

3.1.2. Outer Loop

Depending on whether any improvement in criterion is made in a cycle (a run of the inner loop), the search process of ESE (and similarly that of the original SE) can be divided into two processes: the improving process and the exploration process. Once the criterion is improved after a cycle, i.e., $flag_{imp}$ = 1, the search process will be turned to the improving process; on the other hand, if no improvement is made in a criterion after a cycle, i.e., $flag_{imp}$ = 0, the search process will be turned to the exploration process. The improving process is intended to rapidly find a locally optimal design, while the exploration process is intended to help the algorithm escape from a locally optimal design. The maximum number of cycles is used as the stopping criterion.

The outer loop controls the optimization process by updating the value of the threshold T_h. At the beginning of the optimization process, T_h is set to be a small value, i.e., T_{h0} = 0.005×criterion value of the initial design. Unlike the original SE, in the improving process of ESE, the value of T_h will not be fixed to T_{h0}, rather it will be adjusted and maintained on a small value that is suitable to a specific problem based on n_{acp}, the number of accepted designs in the inner loop, and n_{imp}, the number of better designs found in the inner loop. In the exploration process, T_h is increased and decreased in a relatively large range based on n_{acp}. Based on our tests, the following proposed schedules for controlling T_h is found to work very well for different experimental design problems:

1. In the improving process, T_h is maintained on a small value so that only better design or slightly worse design will be accepted. Specifically, T_h will be decreased if the acceptance ratio n_{acpt} is larger than a small percentage (e.g., 0.1) of the number of total designs J and n_{imp} is less than n_{acpt}; T_h will be increased otherwise. The following equations are used in our algorithm to decrease and increase T_h, respectively, $T' = \alpha_1 T$ and $T' = T / \alpha_1$, where $0 < \alpha_1 < 1$. The setting of α_1 = 0.8 appears to work well in all tests..

2. In the exploration process, T_h will fluctuate within a range based on the value of n_{acpt}. If n_{acpt} is less than a small percentage (e.g., 0.1) of J, T_h will be rapidly increased until n_{acpt} is larger than a large percentage (e.g. 0.8) of J. If this happens, T_h will be slowly decreased until n_{acpt} is less than the small percentage. This process will be repeated until an improved design is found. The following equations are used to decrease and increase T_h, respectively, $T' = \alpha_2 T$ and $T' = T / \alpha_3$, where $0 < \alpha_3 < \alpha_2 < 1$. Based on our experience, we set α_2 = 0.9 and α_3 = 0.7. T_h is increased rapidly (so that more worse designs could be accepted) to help moving away from a locally optimal design. T_h is decreased slowly for searching better designs after moving away from the local optimal design.

3.2. Efficient Algorithms for Evaluating Optimality Criteria

As an optimality criterion is repeatedly evaluated whenever a new design of experiments is constructed, the efficiency of this evaluation becomes critical for optimizing the design of experiment within a reasonable time frame. In this work, we propose efficient evaluation methods that take into account the feature of our updating operation, i.e., when using columnwise element-exchanges for generating new designs, only two elements in the design matrix are involved each time. The evaluations of optimal criteria, such as ϕ_p criterion, the entropy criterion, and the CL_2 criterion, involve different types of matrices (e.g., the inter-distance matrix **D**, the correlation matrix **R**, and the discrepancy matrix **C**, respectively). Re-evaluating all the elements in the matrices each time is not affordable, especially if the matrix size is large (determined by the number of experiments and number of factors). Complete descriptions of our algorithms for the aforementioned criteria

can be found in Jin (2003). Illustration is only provided here for the algorithm associated with the ϕ_p criterion. The computational savings for all algorithms will be summarized.

The re-evaluation of ϕ_p based on Eq. 4 includes three parts, i.e., the evaluation of all the inter-site distances, the sorting of those inter-site distances to obtain a distance list and index list, and the evaluation of ϕ_p. The evaluation of all the inter-site distances will take $O(mn^2)$, the sorting will take $O(n^2\log_2(n))$ (c.f. Press, et al, 1997), and the evaluation of ϕ_p will take $O(s^2\log_2(p))$ (since p is an integer, p-powers can be computed by repeated multiplications). In total, the computational complexity will be $O(mn^2)+O(n^2\log_2(n))+O(s^2\log_2(p))$. Therefore, re-evaluating ϕ_p will be very time-consuming.

Before introducing the new algorithm, a new equation of ϕ_p is first provided, which helps develop an efficient evaluation algorithm by avoiding the sorting required by Eq. 4. Let $\mathbf{D}=[d_{ij}]_{n\times n}$ be a symmetric matrix, whose elements are the inter-site distances of the current design **X**, the new equation, called p-norm form here, is expressed by:

$$\phi_p=\left[\sum_{1\le i<j\le n}\left(1/d_{ij}\right)^p\right]^{1/p}=\left[\sum_{1\le i<j\le n}d_{ij}^{-p}\right]^{1/p}. \quad (10)$$

The equivalence between this form and Eq. 4 can be easily proved, which is omitted here.

Our new algorithm takes into account the fact that after an exchange ($x_{i_1k}\leftrightarrow x_{i_2k}$), only elements in rows i_1 and i_2 and columns i_1 and i_2 are changed in **D** matrix. For any $1\le j\le n$ and $j\ne i_1, i_2$, let:

$$s(i_1,i_2,k,j)=\left|x_{i_2k}-x_{jk}\right|^t-\left|x_{i_1k}-x_{jk}\right|^t, \quad (11)$$

then:

$$d_{i_1j}'=d_{ji_1}'=\left[d_{i_1j}{}^t+s(i_1,i_2,k,j)\right]^{1/t} \quad (12)$$

and

$$d_{i_2j}'=d_{ji_2}'=\left[d_{i_2j}{}^t-s(i_1,i_2,k,j)\right]^{1/t}. \quad (13)$$

With the above representation, the computational complexity of updating the elements in **D** matrix is $O(n)$. The new ϕ_p can be computed by:

$$\phi_p'=\left[\phi_p{}^p+\sum_{1\le j\le n,\, j\ne i_1,i_2}\left[(d_{i_1j}')^{-p}-d_{i_1j}{}^{-p}\right]+\sum_{1\le j\le n,\, j\ne i_1,i_2}\left[(d_{i_2j}')^{-p}-d_{i_2j}{}^{-p}\right]\right]^{1/p}, \quad (14)$$

of which the computational complexity is $O(n\ \log_2(p))$. The total computational complexity of the new algorithm is $O(n)+O(n\ \log_2(p))$. This results in significant reduction of computation compared to re-evaluating ϕ_p.

The new algorithm for evaluating the entropy criterion involves a new Cholesky decomposition algorithm, while that for evaluating the CL_2 criterion employs a similar idea as that for the ϕ_p criterion. A comparison of the computational complexity of totally re-evaluating all elements in matrices and those of our new algorithms are summarized in Table 1. From the table, we find that for the ϕ_p criterion and the CL_2 criterion, with the new algorithms, the efficiency can be significantly improved. The new computational complexity is close to $O(n)$ in both cases. However, for the entropy criterion, because of the involvement of matrix determinant calculation, the efficiency is not improved dramatically (complexity larger than $O(n^2)$).

Table 1. Computational Complexity of Criterion Evaluation

	ϕ_p	CL_2	Entropy
Re-evaluating Algorithms	$O(mn^2)+O(n^2\log_2(n))$ $+O(s^2\log_2(p))$	$O(mn^2)$	$O(n^3)+O(mn^2)$
New Algorithms	$O(n)+O(n\log_2(p))$	$O(n)$	$O(n^2)+O(n)\sim O(n^3)+O(n)$

Table 2 provides illustrative results on the efficiency. The ratio between the time (T_r) needed to totally re-evaluating all matrix elements and the time (T_n) needed by our new algorithm shows the improvement. The empirical results match with our analytical examinations earlier. We also found that the larger the size of an experimental design, the more savings the algorithm will make. Compared to other two algorithms, the entropy criterion is much less efficient. It is also observed that with the new algorithms, the computing time for the ϕ_p criterion is 2.3~3.0 times as much as that for the CL_2 criterion.

Table 2. Computing Time (secs) of Criteria for 500,000 LHDs

T_r stands for the time needed to totally re-evaluating the criterion value of a LHD for 500,000 times. T_n stands for the time needed to construct 500,000 different LHDs by element-exchanges and compute their criterion values by the proposed criterion-evaluation algorithm in ESE.

	ϕ_p (p = 50, t =1)			CL_2			Entropy (θ = 5, t =2)		
	T_r	T_n	T_r/T_n	T_r	T_n	T_r/T_n	T_r	T_n	T_r/T_n
12×4 LHDs	12.2	5.5	2.2	10.7	2.4	4.5	16.6	14.2	1.2
25×4 LHDs	53.0	10.1	5.2	41.5	3.4	12.1	75.3	39.8	1.9
50×5 LHDs	239	19.8	12.1	197	6.5	30.3	347	167	2.1
100×10 LHDs	1378	45.2	30.5	1305	15.9	82.1	2116	1012	2.1

While the construction algorithm proposed is suitable for optimizing designs with the balance property, it can also be extended to optimizing designs with other special structural properties. For instance, it can be extended to obtain optimal OAs or OLs and maintain the orthogonality property. In that case, new updating operations will be used to replace simple element-exchanges in a column so as to retain the orthogonality property required by OAs or OLs. The description of this extension can be found in Jin, 2003.

4. TEST RESULTS AND COMPARATIVE STUDY

Our proposed algorithm can be used for optimizing various classes of designs of experiments, including but not limited to LHDs, general balanced designs, OAs, and OLs. Here we provide two examples of optimal designs based on the CL_2 criterion. The first one, as shown in Fig. 3, is an optimal balanced design, in which the factors 1~4 have 16 levels (equal to number of runs) and factors 5 and 6 have 4 levels. Fig. 5 shows its projection onto 4^{th} (16 levels) and 5^{th} (4 levels) factors. From the figure, we can find that the balance property is retained, i.e., 4^{th} factor is explored once in each of the 16 bins and 5^{th} factor is explored 4 times in each of the 4 bins. The second example, as shown in Fig. 4, is an optimal $OL_{16}(4^5; 2)$. Fig. 6 shows the projection of the optimal $OL_{16}(4^5)$ onto the subspace of 4^{th} and 5^{th} factors. Factors 4 and 5 get explored once in each of 4×4 squares and each of them individually gets explored once in each of 16 equal bins (not shown in the figure).

4	1	11	11	0	1
0	9	8	15	2	2
2	6	13	5	1	3
3	14	6	8	3	0
5	3	0	7	3	2
12	0	5	9	2	3
9	5	2	14	1	0
14	7	12	13	3	1
11	10	9	6	0	0
7	2	14	3	2	0
10	15	15	10	1	2
15	4	7	2	0	2
6	12	3	12	0	3
1	8	4	0	1	1
8	11	10	1	3	3
13	13	1	4	2	1

Figure 3. Optimal $U_{16}(16^4 4^2)$ Based on the CL_2 Criterion (CL_2 = 0.03652)

15	14	6	8	10	
12	2	9	13	14	
14	5	15	6	6	
13	9	1	2	3	
6	15	8	5	1	
4	3	7	0	5	
7	4	0	10	12	
5	8	14	15	9	
8	12	13	1	13	
10	0	3	4	8	
9	6	5	14	0	
11	11	11	11	4	
1	13	2	12	7	
2	1	12	9	2	
0	7	10	3	11	
3	10	4	7	15	

Figure 4. Optimal $OL_{16}(4^5; 2)$ Based on the CL_2 Criterion (CL_2 = 0.01364)

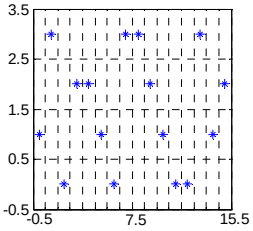

Figure 5. Projection of the Optimal $U_{16}(16^4 4^2)$ onto the Subspace of 4th and 5th Factors

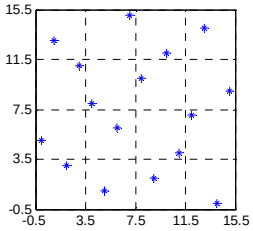

Figure 6. Projection of the Optimal $OL_{16}(4^5; 2)$onto the Subspace of 4th and 5th Factors

In the rest of this section, we will demonstrate the performance and efficiency of the proposed algorithm by comparing it with the existing algorithms. The comparative study below will focus on optimal LHDs, which have been widely applied and studied in the literature. In Section 4.1, we will compare the computing time of our method with the CP algorithm based on the results presented in the literature. This is followed in Section 4.2 with a more comprehensive comparison of performance and efficiency between our proposed ECE algorithm, and the existing CP and the SA algorithms. In Section 4.3, we verify the quality of the optimal designs of experiments obtained by our method by comparing the achieved optimality criterion with that from random designs.

4.1. A Preliminary Comparison – Savings in Computing Time

In the following, an illustrative comparison between our proposed ESE algorithm and the CP algorithm presented in Ye, et al, 2000 is provided to show the significant savings achieved by our method. It should be noted that besides using the ESE algorithm, our method also employs efficient algorithms for evaluating optimality criteria; the savings in computing time is a combination of both. The comparison is for optimal 25×4 LHDs constructed based on the ϕ_p criterion (p = 50 and t = 1). It should be noted that even though Ye, et al used the ϕ_p criterion (with the same parameter settings as in our tests) as the optimality criterion in constructing optimal LHDs, their results were reported in the form of (maximizing) the minimum L_1 distance (the larger the better), which, as discussed before, is strongly related to but not totally in accord with the ϕ_p value (the smaller the better). To be consistent, the results of our proposed ESE are also in the form of minimum L_1 distance in Table 3.

Table 3. ESE vs. CP for Constructing Optimal 25×4 LHDs Based on ϕ_p Criterion (p = 50 and t = 1)

N_e stands for number of exchanges (shown **in thousands**). The results for CP are from Ye, et al (2000, based on Sun SPARC 20 Workstation). In the CP test, 100 cycles (shown in the parentheses following the numbers of exchanges) are used. ESE is tested on a PC with a Pentium III 650 MHZ CPU.

	N_e in Thous	Min L_1 Distance	Computing Time
CP	2242(100)	0.8750	10.63 hr
ESE	120	0.9167	2.5 sec

In the work of Ye, et al, the optimization process was repeated for 100 cycles starting from different random LHDs and the design with the largest minimum L_1 distance of the 100 constructed optimal designs was reported as the final optimal design. The number of exchanges or computing time of CP is the total number or time used in the 100 cycles. From the results, it is found that the designs constructed by our ESE with less than 2.5 seconds is better than those constructed by CP with around 10.63 hours. In fact, ESE is tested for many times and the minimum distances are consistently larger than or equal to 0.9167. The saving of computing time is dramatic even if the difference between the computing platforms is considered. As introduced earlier, such a good efficiency is achieved by:

- Improving the efficiency of criterion evaluation (5 times faster than totally re-evaluating for the example test case; more significant improvement for larger size designs, see Table 2);
- Using fewer exchanges with ESE to search an optimal design (120,000 with ESE Vs 2,241,900 with CP).

The test results match with our theoretical examinations of the efficiency of the algorithms for criterion evaluations shown in Section 3.2. The following comparative study is to further demonstrate the performance of ESE algorithm in saving the numbers of exchanges.

4.2. A Further Comparison of Performance and Efficiency

We tested two popular criteria, i.e., the ϕ_p criterion (p = 50 and t = 1) and the CL_2 criterion in our comparative study. For SA, the tests are limited to the ϕ_p criterion since the parameter settings of SA were originally provided by Morris and Mitchell (1995) to suit the ϕ_p criterion. For CP, both criteria are tested since there are no special parameters to be set. Due to the space limitation, we only discuss the results from using the ϕ_p criterion here.

Our implementations of CP and SA in the test are based on the algorithms proposed in Li and Wu (1997) and in Morris and Mitchell (1995), respectively (see the description in Section 2.3). One major change in our implementations of CP and SA is that our proposed algorithms for criterion evaluation described in Section 3.2 are used instead of reevaluating all matrix elements as currently being done in the literature. Thus, our implementations run much faster than the original implementations in the literature. The implementation of SA, including parameter settings, is the same as Morris and Mitchell's. Even though we attempt to reproduce the original implementations of CP and SA in the literature, the results from the two implementations may not be exactly the same as in the literature.

The tests are conducted on two sets of LHDs of relatively small sizes, i.e., 12×4 and 25×4, and two sets of LHDs of relatively large sizes, i.e., 50×5 and 100×10. As randomness is involved in all constructing algorithms, we repeat the same test

for 100 times starting from different initial LHDs. On each set of LHDs, two types of comparison are made, i.e.,

Type-I: Comparing the performance of ESE with that of SA and CP in terms of the average of criterion values of optimal designs with nearly the same numbers of exchanges.

Type-II: Comparing the efficiency of ESE with that of SA and CP in terms of numbers of exchanges needed for ESE to achieve optimal designs with the average of criterion values slightly better than that of SA or CP.

In both types of comparison, *t*-test is used to statistically compare the average criterion value of the optimal designs generated by ESE with those generated by SA or CP. The *p*-value is used to measure the level at which the observed difference (< 0) between the average criterion values is statistically significant. The smaller the *p*-values are, the more statistical significance it has. While the standard in scientific research is that the *p*-value should be below 5%, here we use a much tighter standard that the *p*-value should be smaller than 0.001%. For type-I comparison, this standard is not that critical since virtually all the *p*-values in the comparison are much smaller than 0.001%; for type-II comparison, however, this standard is used to judge whether optimal designs generated by ESE are close to but still statistically significantly better than those generated by SA or CP. Corresponding to two types of comparisons, two groups of tests for ESE are performed. For type-I comparison, ESE is terminated at a number of exchanges close to that of SA or CP and this group of tests for ESE is denoted as ESE (I); for type-II comparison, ESE is terminated when *p*-values are smaller than 0.001%, and this group of tests for ESE is denoted as ESE (II). We have also used permutation test (c.f., Good, 2000) for the comparison, which is a hypothesis test based on re-sampling from randomly permuted sample data. Unlike *t*-test, permutation test does not assume a normal distribution. The *p*-values achieved from permutation test (not presented here) are similar to those from *t*-test.

4.2.1. Results of Small Sizes of Designs

For small-sized LHDs, relatively large number of exchanges is affordable. For example, with 2,865,600 exchanges, it takes ESE about 57 seconds to construct an optimal 25×4 LHDs based on the ϕ_p criterion. The tests for small-sized problems are therefore focused on the capability of moving away from locally optimal designs and finding better experimental designs given a large number of exchanges.

The results of using the ϕ_p criterion are shown in Table 4. For each algorithm, two sets of tests with different numbers of exchanges are conducted. For SA, the two sets of tests correspond to two different values for cooling factor α suggested by Morris and Mitchell (1995), i.e., α = 0.90 (faster cooling) and 0.95 (slower cooling), respectively. In a particular set of tests, the numbers of exchanges of SA for constructing optimal designs will differ test by test. For instance, for 12×4 LHD and α = 0.95, the numbers of exchanges could be anywhere between 362,384 and 1,192,482. The numbers of exchanges of SA shown in the table are the average numbers. CP is terminated at a cycle number N_s, which is selected so that the average number of exchanges is close to that of SA. The numbers of exchanges shown are also the average of 100 tests. The results of SA are also used to determine when to stop ESE.

Table 4. Test Results of optimal 12×4 LHDs and 25×4 LHDs based on ϕ_p criterion (p =50 and t =1)

For SA, Sets 1 &2 correspond to α = 0.90 and α = 0.95, respectively. N_e (shown in thousands) stands for the average numbers of exchanges of 100 tests in each set of tests. For CP, cycle numbers N_s are given in the parentheses following the average numbers of exchanges.

ϕ_p criterion		Set 1		Set 2	
		N_e in Thous	Mean(STD)	N_e in Thous	Mean(STD)
12×4 LHDs	SA	289	0.8569(0.0131)	523	0.8505(0.0133)
	CP	292(154)	0.8581(0.0082)	530(280)	0.8546(0.0096)
	ESE (I)	286	0.8384(0.0057)	520	0.8362(0.0041)
	ESE (II)	96	0.8483(0.0114)	174	0.8426(0.0084)
25×4 LHDs	SA	1416	1.1205(0.0101)	2724	1.1149(0.0103)
	CP	1442(65)	1.1495(0.0078)	2744(124)	1.1455(0.0070)
	ESE (I)	1416	1.1051(0.0060)	2724	1.0989(0.0051)
	ESE (II)	470	1.1150(0.0072)	840	1.1072(0.0072)

For type-I comparison, error-bar plots (Figs. 7 and 8) are used to display the mean and variability of achieved ϕ_p values from 100 tests. The error bars (thick vertical lines) are each drawn a distance of one STD above and below the mean value. For each algorithm, a mean-line links the middles (i.e., the means) of error-bars. The dash error-bars and mean-lines are for the results of SA. From the figures, it is found that with similar number of exchanges, on average the proposed ESE always achieves better designs than both SA and CP with respect to the ϕ_p criterion. This is also confirmed statistically by the *p*-values in *t*-tests, which are all smaller than 1.0e-15. Furthermore, ESE is more efficient than both SA and CP. Table 4 shows that to obtain a statistically significantly better design for both 12×4 and 25×4 LHDs, ESE needs less than 1/3 of exchanges used in SA and in CP.

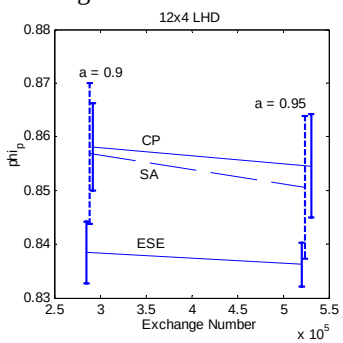

Figure 7. Type-I Comparison for 12 ×4 LHDs (ϕ_p criterion)

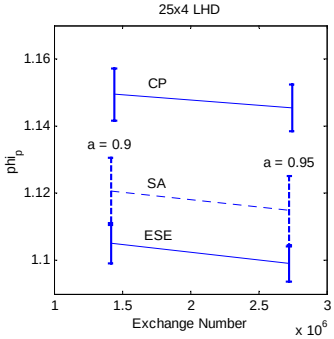

Figure 8. Type-I Comparison for 25 ×4 LHDs (ϕ_p criterion)

When using the CL_2 criterion, it is found that ESE uses around 1/3 ~ 1/2 exchanges used in CP for 12×4 LHDs and around 1/6 ~ 1/2 of exchanges used in CP for 25×4 LHDs to achieve statistically significantly better designs.

4.2.2. Results for Large Sizes of Designs

The computational cost of constructing an optimal design of large sizes is much larger than that of small sizes. For large-size designs, our goal is to find a good design using limited computational resource. Our comparison focuses on how efficient our algorithm is compared to others by using the same amount of reasonable numbers of exchanges, which are considered as small in relative to the size of the LHDs.

For large-sized designs, SA in general converges much more slowly than CP and ESE. Therefore with the numbers of exchanges that are small relative to the size of design, the SA search process will not be able converge before the maximum

number of exchanges is reached. As the result, the design generated by SA could be much inferior to those generated by CP and ESE. For instance, for 50×5 LHDs based on the ϕ_p criterion, with around 1,520,000 exchanges, the average criterion value of SA (α = 0.9) is 1.4658 in comparison with 0.9875 for ESE and 1.0322 for CP. Therefore for large problems, SA may not be suitable since it needs excessive numbers of exchanges. Our test for large-sized designs will only focus on CP and ESE.

CP provides baselines for determining when to stop ESE in both types of comparisons. For large-sized problems, the computational cost could be too high for CP to even finish a single cycle. For instance, a single cycle of CP for 100×10 LHD with ϕ_p criterion could take 31,482,000 exchanges (2,758 seconds). Therefore, the tests of CP for large-sized LHDs have been restricted to at most several cycles for 50×5 LHDs and one cycle for 100×10 LHDs. Table 5 shows the maximum numbers of exchanges and the computing time. From the table we can find that the computing time has been limited to merely several minutes (if not seconds).

Table 5. Maximum Number of exchanges and Computing Time for Constructing Optimal LHDs

N_e stands for number of exchanges. T_c stands for computing time. The data shown are for ESE. The data for CP are similar.

	ϕ_p		CL_2	
	Max N_e×1000	Max T_c	Max N_e×1000	Max T_c
50×5 LHDs	1945	77 sec	2960	35 sec
100×10 LHDs	2500	219 sec	7685	198 sec

As shown in Table 6, for each algorithm, three sets of tests with different numbers of exchanges are performed. For 50×5 LHDs, the numbers of exchanges of the first set of tests are not sufficient to finish one cycle; the second set of tests involves exactly one cycle and the numbers of exchanges are the average of the 100 tests; likewise, the third set of tests involves exactly 5 cycles. For 100×10 LHDs, even though large numbers of exchanges are used for CP in all three sets of tests, they are not sufficient to finish the first cycle.

Table 6. Test Results of optimal 50×5 LHDs and 100×10 LHDs based on ϕ_p criterion (p =50, t =1)

N_e stands for the average numbers of exchanges. For CP, the cycle numbers are provided in the parentheses following the numbers of exchanges. If there are no cycle numbers marked, it means that CP is stopped within the first cycle

ϕ_p criterion		Set 1		Set 2		Set 3	
		N_e in Thous	Mean (STD)	N_e in Thous	Mean (STD)	N_e in Thous	Mean (STD)
50×5 LHDs	**CP**	61	1.1564 (0.0121)	404 (1)	1.0420 (0.0097)	1948 (5)	1.0311 (0.0068)
	ESE (I)	60	1.0486 (0.0072)	400	1.0076 (0.0059)	1945	0.9850 (0.0038)
	ESE (II)	10	1.1264 (0.0099)	80	1.0348 (0.0069)	110	1.0248 (0.0063)
100×10 LHDs	**CP**	297	0.5381 (0.0044)	545	0.5059 (0.0024)	2525	0.4660 (0.0014)
	ESE (I)	280	0.4562 (0.0012)	500	0.4525 (0.0014)	2500	0.4440 (0.0010)
	ESE (II)	10	0.5214 (0.0031)	20	0.4996 (0.0025)	140	0.4634 (0.0015)

The means and variability of the achieved ϕ_p values for 50×5 LHDs and 100×10 are shown in Figs. 9 and 10 for types-I comparison. From the figures, it is found that ESE consistently outperforms CP, which is also confirmed by *t*-tests (*p*-values are all smaller than 1.0e-15). From Table 6, it is observed that ESE is much more efficient than CP. To reach statistically significantly better designs than CP, ESE needs only around 1/17~ 1/5 of exchanges used in CP for 50×5 LHDs and 1/29 ~ 1/18 of exchanges used in CP for 100×10 LHDs.

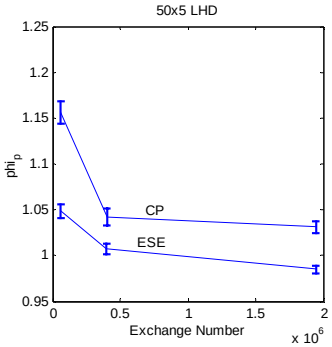

Figure 9. Type-I Comparison for 50 ×5 LHDs (ϕ_p criterion)

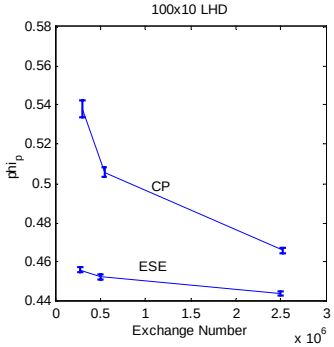

Figure 10. Type-I Comparison for 100 ×10 LHDs(ϕ_p criterion)

Similar tests to the above have been carried out for the CL_2 criterion. It is found that ESE consistently outperforms CP, which is confirmed by *t*-tests (*p*-values are all smaller than 1.0e-15). It is observed that ESE is much more efficient than CP. To reach statistically significantly better designs than CP, ESE needs only around 1/23 ~ 1/4 of exchanges used in CP for 50×5 LHDs and 1/33 ~ 1/10 of exchanges used in CP for 100×10 LHDs.

4.3 Verifications by Comparing with Random Designs

For large-sized designs, we have used small number of exchanges in relative to the size of designs. As the global optimal design is never known, one way to assess the quality of these optimal designs are to estimate the probability of a random design being better than an optimal design, i.e.,

$$F(t_{opt}) = P(X \le t_{opt}), \quad (15)$$

where X, a random variable, stands for criterion values of random designs and t_{opt} is the criterion value of an optimal design (here the mean of the criterion values of a set of optimal designs are used for t_{opt}). If the probability is trivial, we could consider the optimal design has a significantly low criterion value. This evaluation depends on the cumulative distribution function (CDF) $F(x)$ of the criterion values of random designs, which generally can only be estimated by Monte-Carlo methods.

For 50×5 LHDs, 20,000,000 random designs are generated and their ϕ_p and CL_2 values are computed. It is found that the average criterion value (e.g., 0.9850 for ϕ_p, as shown in set 3 in Table 6) of the optimal designs by ESE is far beyond the left tails of the empirical cumulative distribution curves. To estimate the probability in Eq. 15, an approximation curve has been used to extend the left tail of the empirical cumulative distribution curve. It is obtained that for the ϕ_p criterion, the probability $P(\phi_p < t_{opt})$ is roughly in 10^{-15} magnitude; for the CL_2 criterion, the probability $P(CL_2 < t_{opt})$ is roughly in 10^{-19} magnitude. Similar observations can also be obtained for 100×10 LHDs. We then can conclude that for large-sized LHDs, those optimal designs constructed by ESE generally have significantly low ϕ_p values or CL_2 values.

5. SUMMARY

In this study, we develop a very efficient and flexible algorithm for constructing optimal experimental designs. Our method includes two major elements: the use of ESE algorithm for controlling the search process and the employment of efficient algorithms for evaluating the optimality criteria. Our proposed algorithm has shown great efficiency compared to some algorithms in the literature. Specifically, it has cut the computation time from hours to minutes and seconds. In comparison, we have the following observations:

1. With the same number of exchanges, the optimal designs generated by ESE is generally better than those generated by SA and CP.

2. To obtain a design statistically significantly better than those generated by SA and CP, ESE needs far less number of exchanges (typically around 1/6 ~ 1/2 of exchanges needed by SA or CP for small-sized designs and 1/33~ 1/4 of exchanges needed by CP for large-sized designs).

3. For small-size problems (a relatively large number of exchanges are affordable), SA often has better performance than CP. However, for large-size problems, SA may converge very slowly and require a tremendous number of exchanges.

While our focus in this paper is on optimizing LHDs, the ESE algorithm can be used to optimize other classes of designs such as OAs and OLs. Furthermore, while the algorithm works on the ϕ_p criterion, the entropy criterion, and the CL_2 criterion, it can be conveniently extended to other optimality criteria.

While there are many optimality criteria available in the literature, the comparison between different design criteria is certainly one of the most important problems in the field of design of computer experiments and deserves a thorough future investigation. The optimality criteria and the constructing-algorithm introduced in this paper are mainly focused on global metamodeling assuming anywhere in the design space is equally important. Another interesting issue is the comparison between different classes of designs. We compare the properties of optimal OLs (strength 2) and optimal LHDs. While detailed results are omitted, it is found that optimal OLs always have better two-dimensional projective properties than optimal LHDs. Such improvement, however, may come with small degradation of the full-dimensional projective properties.

ACKNOWLEDGMENTS

NSF grants 0099775 and 0217702 are acknowledged.

REFERENCES

Currin, C., Mitchell, T., Morris, M.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 (4), 953-963.

Fang, K.T., Lin, D. K., Winker, J. P., Zhang, Y., 2000, Uniform Design: Theory and Application, *Technometrics*, V42 n3, 237-248

Fang, K.T., Ma, C.X. and Winker, P., 2002, Centered L_2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs, *Mathematics of Computation*, 71, 275-296.

Fang, K.T. and Wang, Y., 1994, *Number-theoretic Methods in Statistics*, Chapman & Hall, New York.

Good, P. I., 2000, *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, Springer, New York.

Hickernell, F. J., 1998, "A generalized discrepancy and quadrature error bound", *Mathematics of Computation*, 67, 299-322.

Hedayat, A.S., Stufken, J., Sloane N. J., 1999, *Orthogonal Arrays: Theory and Applications*, Springer-Verlag New York.

Jin, R., 2003, *Investigations and Enhancements of Metamodeling Techniques in Engineering Design*, Ph.d Thesis, Univ. of Illinois at Chicago.

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.R. and Owen, A.B., 1996, Computer experiments, in Ghosh, S.and Rao, C.R., eds, *Handbook of Statistics*, 13, 261 –308, Elsevier Science, New York.

Li, W., 1997, Optimal Designs Using CP Algorithms, *Proceedings for 2nd World Conference of the International Association for statistical computing*, 130-139.

Li, W., and Wu, C. F. J., 1997, Columnwise-Pairwise Algorithms With Applications to the Construction of Supersaturated Designs, *Technometrics* , Vol. 39, 171-179.

McKay, M.D., Beckman, R.J. and Conover, W.J., 1979, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code", *Technometrics*, 21(2), 239-45.

Morris, M.D., Mitchell, T.J., 1995, Exploratory Designs for Computational Experiments, *Journal of statistical planning and inference*, 43, 381-402.

Owen, A.B., 1992, Orthogonal arrays for computer experiments, integration and visualization, *Statistical Sinica*, v2, n2, 439-452.

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

Press, W.H., Teukoisky, S.A., Vetterling, W.T., Flannery, B.P., 1997, Numerical Recipes in C: the Art of Scientific Computing, Cambridge U. Press, 332~338.

Saab, Y.G., Rao, Y.B., 1991, Combinatorial optimization by stochastic evolution, *IEEE Transactions on Computer-aided Design*, v10, n4, 525-535.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P., 1989, Design and analysis of computer experiments, *Statistical Science*, 4(4), 409-435.

Simpson, T.W., Lin, D.K.J. and Chen, W., 2001, "Sampling strategies for computer experiments: design and analysis", *Intl Journal of Reliability and Application*, 2(3), 209- 240.

Shannon, C.E., 1948, A mathematical theory of communication, *Bell System Technical Journal*, 27:379-423, 623-656.

Tang, Boxin, Orthogonal array-based Latin Hypercubes, 1993, Journal of the American Statistical Association, v88, n424, 1392~1397

Winker, P. and Fang, K. T., 1998, Optimal U-type design, in *Monte Carlo and Quasi-Monte Carlo Methods 1996*, eds. by H. Niederreiter, P. Zinterhof and P. Hellekalek, Springer, 436-448.

Ye, K. Q., Li, W., Sudjianto, A., 2000, Algorithmic construction of optimal symmetric Latin hypercube designs, *Journal of Statistical Planning and Inference*, 90, 145-159.

