การศึกษาอัลกอริทึมการสืบค้น สำหรับการจำลองการทดลองด้วยคอมพิวเตอร์

Exploratory search algorithms for computer simulated experiments

อนามัย นาอุดม

ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร จ.พิษณุโลก

Anamai Na-udom*

Department of Mathematics, Faculty of Science, Naresuan University, Phitsanulok

บทคัดย่อ

เทคนิคการจำลองการทดลองด้วยคอมพิวเตอร์ถูกนำมาใช้อย่างแพร่หลายในการศึกษาระบบที่มีความซับซ้อนและไม่สามารถทำการทดลองทางกายภาพได้ การจำลองการทดลองประเภทนี้มักจะใช้เวลายาวนานมากในการประมวลผล โดยทั่วไปผลลัพธ์ที่ได้จากการจำลองการทดลองด้วยคอมพิวเตอร์จะมีลักษณะตรงแบบ กล่าวคือเมื่อทำการทดลองภายใต้เงื่อนไขเดิมของตัวแปรเข้าจะทำให้ได้ค่าผลลัพธ์คงที่เสมอดังนั้นแผนการทดลองที่เน้นการกระจายจุดทดลองเพื่อเติมเต็มปริภูมิของการทดลองให้มากที่สุด ซึ่งมีชื่อเรียกว่า แผนการทดลองแบบลาตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) จึงถูกนำมาใช้อย่างแพร่หลายในทางปฏิบัติ การสร้างแผนการทดลองแบบ LHD ที่เหมาะสมที่สุดสามารถทำได้โดยใช้อัลกอริทึมการสืบค้นควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม ซึ่งกระบวนสืบค้นดังกล่าวนี้มักจะใช้เวลานานมากโดยเฉพาะอย่างยิ่งเมื่อมิติปัญหาของแผนการทดลองมีขนาดใหญ่ขึ้น ดังนั้นจึงมีงานวิจัยจำนวนมากที่ได้นำเสนออัลกอริทึมการสืบค้นแบบต่าง ๆ เพื่อการค้นหาแผนการทดลองแบบ LHD ที่มีประสิทธิภาพมากที่สุด บทความนี้จึงได้ทำการรวบรวมและทบทวนคุณลักษณะของอัลกอริทึมการสืบค้นเหล่านี้ รวมไปถึงเกณฑ์ที่ใช้ในการตัดสินใจเลือกค่าที่เหมาะสมประเภทต่าง ๆ เพื่อเป็นแนวทางในการเลือกใช้อัลกอริทึมการสืบค้นที่เหมาะสมสำหรับแต่ละมิติปัญหาของการทดลอง ส่วนทิศทางการวิจัยในงานด้านนี้ได้นำเสนอไว้ในตอนท้ายของบทความ

คำสำคัญ: การจำลองการทดลองด้วยคอมพิวเตอร์, แผนการทดลองแบบลาตินไฮเปอร์คิวบ์, อัลกอริทึมการสืบค้น, เกณฑ์การเลือกค่าที่เหมาะสม

Abstract

Computer simulated experiments are often used to explore complex physical phenomena. They are usually time consuming and computationally expensive to run. Normally, the output responses from computer simulated experiments are deterministic. Consequently the space filling designs, which focus on spreading design points over a design space, are necessary. Latin hypercube designs (LHD) are normally practiced in the context of computer simulated experiments. The best LHD for a given dimensional problem is obtained by using a search algorithm under a pre-specified optimality criterion. Usually this searching process takes a long time to terminate, especially when the dimension of the problem is large. A number of search algorithms have been practiced along with the optimality criteria to search for the best LHD. In this paper we review the popular search algorithms and the optimality criteria that have been extensively used in the context of computer simulated experiments. The guidelines for the choice of best search algorithm and further study are also presented.

Keywords : Computer simulated experiments, Latin hypercube designs, Search algorithms, Optimality criteria

* E-mail: anamain@nu.ac.th

1. Introduction

Recently computer simulated experiments (CSE) have replaced classical experiments to investigate a physical complex phenomena, especially when classical (physical) experiments are not feasible. For example, the use of reservoir simulator to predict ultimate recovery of oil, the use of finite element codes to predict behaviour of metal structure under stress, and so on. The nature of computer simulated experiments is deterministic; hence identical settings of input variables always produce an identical set of output response. Therefore, space filling designs that aim to spread the design points over a region of interest are necessary. The most popular class of space filling design in the context of computer simulated experiments is Latin hypercube design (LHD). LHD deign, was originally proposed by Mckay and co-workers (Mckay et al., 1979), is a matrix (X), of n rows and d columns where n is the number of runs and d is the number of input variables. LHD can be constructed based on the idea of stratified sampling (Mckay et al, 1979) to ensure that all subregions in the divided input variable space will be sampled with equally probability. A Latin hypercube sampling has

$$X_{ij} = \frac{\pi_{ij} - U_{ij}}{n}, \qquad (1)$$

where π_{ij} are the elements of an n x d matrix comprising of columns π_j (j = 1, 2,..., d). Each column π_j (j = 1, 2,..., d) is independent random permutation of number 1 through n and U_{ij} are n x d values of independent U[0,1] random variables independent of the π_{ij}. An example of LHD is presented in Table 1.

Table 1 The 5 x 4 random LHD

X_1	X_2	X_3	X_4
1	4	3	1
2	3	4	4
3	2	5	3
4	1	2	2
5	5	1	5

The ultimate goal of selecting the settings of input variables is to attain the coverage of all design regions of interest. As mentioned before that the space filling designs (LHD) are preferred in the context of computer simulated experiments. Space filling designs can be constructed through combinatorial methods (non-search algorithm) or searching for a design through search algorithms. The former method generates design with good design properties but it is restricted in terms of a design size. For example methods proposed by Butler (Butler, 2001) are limited to a design size of a prime number. The latter method is based largely on improving design by exchanging between the pairs of design points. Exchange algorithms can be time consuming to implement, however, the design generated are flexible in run sizes. This paper is concerned with a search based construction of a design using some kinds of exchange algorithms.

The computer simulated experiments are usually complex and consist of many input variables to investigate. In this case a large number of runs are required to estimate the parameters corresponding to the factors of interest in the model. For example, if the problem of interest consists of d input variable and n number of runs, the total number of LHD is $(n!)^d$. Obviously this number explodes exponentially as the values of n and d increase; hence the full space of LHD cannot be explored. In this case we need the search algorithms to lead us to a good design with respect to an optimality criterion. The key idea of all existing search algorithms is to use some kinds of exchange procedures to move towards the better designs. The search based approach for selecting a design.

The search based approach for selecting a desing is implemented by combining search algorithms and the optimality criterion. For example, Morris and Mitchell (Morris & Mitchell, 1995) adopted a version of Simulated Annealing algorithms (SA) to search for optimal LHDs with respect to ϕ_p criterion. Li and Wu proposed a columnwise-pairwise

algorithm (CP) with respect to the *D* efficiency criteria (Li & Wu, 1997). Ye and his co-workers adapted CP algorithm to search for symmetric LHD under various optimality criteria such as entropy and ϕ_p criteria (Ye et al., 2000). Park proposed a row-wise element exchange algorithm along with IMSE and entropy criteria (Park, 1994). Jin et al. developed an enhanced stochastic evolutionary algorithm to search for the best design considering various optimality criteria such as a maximin distance criterion, ϕ_p criterion and entropy criterion (Jin et al., 2005). Liefvendahl and Stocki applied a version of Genetic algorithm (GA) to search for the optimal LHD considering ϕ_p and a maximin distance criterion (Liefvendahl & Stocki, 2006). In the following sections we present details of thees search algorithms and optimality criteria. In this paper we explore popular search algorithms with a view of providing practical guidelines for choice of algorithms for searching designs (LHD) for any dimensional problems.

2. Search algorithms

Usually the best design for a given problem is selected using a pre-specified search algorithm and optimality criteria. There are a large number of search algorithms and optimality criteria (OC) proposed in the context of computer simulated experiments. In this section we will explore three of extensively used algorithms namely, CP, SA and GA. Details of each algorithm are presented below.

2.1 Columnwise-pairwise search algorithm (CP)

The columnwise-pairwise search algorithm (CP) is developed to use in the context of computer simulated experiments by Li and Wu in 1997 (Li & Wu, 1997). CP aims to search for a better design by exchanging any pairs within the columns. Main steps of algorithm are presented below

Step 1 : Start with a random Latin hypercube design of order n x d.

Step 2 : Each iteration has d steps. At the i^{th} where i = 1, 2,..., d step, the best exchange (with respect to an optimality criterion) of points in column i is selected. Then, the design matrix is updated accordingly.

Step 3 : If the design is better, with respect to the specified criterion, repeat Step 2. Otherwise it is considered as an optimal design and the search process is terminated. It was reported that CP is very simple and easy to implement. The only parameter that is needed to be set a priori is the tolerance level (in Step 3). Further, CP is able to generate a good supersaturated design and it can be used along with various optimality criteria (Li & Wu, 1997). In order to avoid the problem of convergence and the search being stuck at a local optimum value, usually multiple search with different starting points are performed. The best result, among different trials, is selected as optimal design. It should be noted that for large dimensional problems, CP algorithm can be time consuming to implement.

2.2 Simulated annealing algorithm (SA)

Morris and Mitchell (1995) developed a simulated annealing algorithm to search for an optimal LHD using ϕ_p optimality criterion. The design that minimizes ϕ_p value is considered as the best design in the class. The steps of SA are presented as follows.

Step 1 : Generate a random LHD of order n by n. Let the LHD be denoted by X. Calculate the value of optimality criteria (OC) for X. Let it be denoted by OC(X).

Step 2 : Randomly select a column j of X, randomly choose two elements (X_{aj}) and X_{bj} within column j and interchange them. Let the new LHD thus generated be called X_{new}.

Step 3 : Calculate the value of optimality criteria (ϕ_p criterion) for X_{new}, call it OC(X_{new}). If OC(X) > OC(X_{new}), replace X by X_{new} and go to Step 2. If OC(X) < OC(X_{new}), replace X by X_{new} with probability π where,

$$\pi = \exp\{-[OC(X_{new}) - OC(X)]/t\} \qquad (2)$$

where t is an algorithmic parameter, known as the "temperature". After a given number of perturbations are tried at the given temperature without improving the best

design, the temperature is lowered by the standard factor and the search is continued. When a large fixed number of iterations are made with no improvement, the search is ended and the best design is reported. An example of the design generated from ϕ_p criterion is shown in the Figure 1

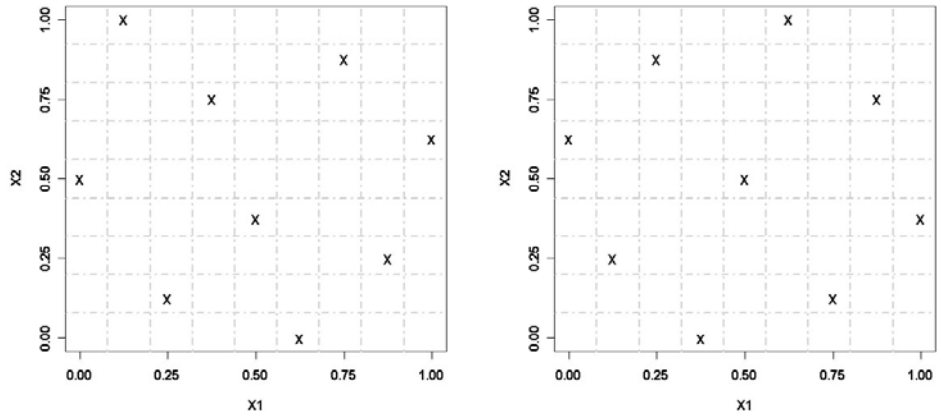


Figure 1 Examples of 9 x 2 optimal LHD based on ϕ_p optimality criterion

The choice of initial parameters for SA can be found in (Morris & Mitchell, 1995). It was also reported in the paper that SA performed very well in terms of moving away from the local optimum value of ϕ_p criterion.

2.3 Genetic algorithm (GA)

Genetic Algorithm (GA) is very popular and is extensively used in the optimization field. Use of GA to search for an optimal design for a computer simulated experiment is proposed by Liefvendahl and Stocki (Liefvendahl & Stocki 2006). They use a random mechanism to generate offspring columns from the randomly selected population of LHD's. The steps of GA can be summarized as follows.

Step 1 : Initial population generation. GA starts with the generation of the population where number of members in population is pre-specified. Each member of the population is an LHD of given dimension n and d.

Step 2 : Selection. A pair of members is randomly selected to be used in the next step.

Step 3 : Crossover and Mutation. An offspring LHD is generated by a random mechanism from parent LHD selected in previous steps. Random mutations are also applied to offspring columns.

Selection and crossover steps are repeated to generate a new generation. Each generation is evaluated with respect to an optimality criterion and the best design is sought. To achieve the best performance of GA, the best set of parameter is needed to be obtained (e.g. m : number of population, m_i : number of designs generated in each generation, ω : mutation rate). More details on parameter settings for GA can be found in (Li & Shigeru, 2001; Liefvendahl & Stocki, 2006).

3. Optimality criteria

For a specified number of run (n) and dimension of the problem (d), the best design in the class of LHD can be chosen considering the optimality criteria controlling the spread of points in the design region. There are many types of optimality criteria proposed to measure how well the spread of points over the design region is. The best design can be chosen by minimizing the integrated mean squared error (IMSE), maximization of entropy (log |R|) and minimization of maximin distance criteria and aggregate maximin (ϕ_p).

The concept of constructing an optimal design is based on the idea of searching for a design X^* that

optimizes a pre-specified optimality criterion. There have been several kinds of optimality criteria in the context of computer experiments as indicated in the previous section. These criteria could be classified into 2 main groups.

3.1 Model dependent criteria

The model dependent criteria require the knowledge of the statistical model to implement. These criteria are listed below.

- **Integrated mean square error (IMSE)**

This criterion was proposed by Sacks et al. (Sacks et. al., 1989). It is based on the idea of selecting X_{ij}, so as to minimize the MSE of prediction. The best design X is selected to minimize,

$$\int_x MSE[\hat{y}(x)]\psi(x)dx \tag{3}$$

where $\hat{y}(x)$ is the fitted value of the output response from Kriging approximation model (Sacks et al., 1989; Koehler & Owen, 1996).

For a given weight function $\psi(x)$ and the value of $MSE[\hat{y}(x)]$ defined as

$$MSE[\hat{y}(x)] = \sigma^2\left[1-(f^T(x) \quad r^T(x))\begin{pmatrix}0 & F^T\\ F & R\end{pmatrix}^{-1}\begin{pmatrix}f(x)\\ r(x)\end{pmatrix}\right], \tag{4}$$

where $f(x)$, $r(x)$, F and R are terms specified by the design matrix and the output response from Kriging approximation model (Bates et al., 1996; Morris & Mitchell, 1995; Sacks et al., 1989). The integral is usually calculated through the numerical approximations and parameters of the correlation matrix R (see Sacks et al., 1989 for more details) are pre-specified.

- **Entropy criterion**

Entropy criterion was first applied in designs for spatial models by Shewry and Wynn (Shewry & Wynn, 1987). The authors showed that minimizing the posterior entropy is equivalent to maximizing the prior entropy. The idea of this criterion is that the amount of information in a design matrix depends on the prior knowledge of distribution of $Z(x)$ and $R(x)$. Usually, the prior of $R(x)$ is unknown and finding the optimal designs is not easy. In practice, the design X that maximizes $|E(\log p_y)|$ is considered as the best design in the class, where p_y is the probability density function of Y. Therefore, the entropy criterion is related to the maximization of log $|R|$ in Gaussian field. The design is selected if it maximizes

$$Entropy = \log|R| \tag{5}$$

It can be clearly seen that in order to use the model dependent criteria. The knowledge of the statistical models is required. For example, if we treat the response model as a Kriging model (Bates et al., 1996; Sack et al., 1989; Welch et al., 1992), the estimation of all unknown parameters in the correlation function are required. Unfortunately, in the construction of the design for computer simulated experiments, the correlation parameters are not known a priori. There is no way of estimating these without performing the experiment itself. This makes the implementation of these criteria impossible unless we have pilot data or guess estimate the values of parameter of the correlation function.

3.2 Model independent criteria

In contrast to the model dependent criteria, these criteria do not require any knowledge of the statistical models. Therefore, these criteria are easier to implement in practice. In this section we will describe 2 types of optimality criteria which are commonly used in the context of computer simulated experiments.

- **Maximin distance criterion**

Maximin distance criterion was developed by Johnson et al. (Johnson et al., 1990). Any design X is called a maximin design if it maximizes the minimum intersite distance:

$$\text{maximin} = \min_{1\le i,j\le n} d(X_{i.}, X_{j.}); i \ne j \tag{6}$$

where $d(X_i, X_j)$ is the Euclidean distance between i^{th} and j^{th} design points:

$$d(X_{i.}, X_{j.}) = \left[\sum_{k=1}^{d}(X_{ik}-X_{jk})^2\right]^{1/2} \tag{7}$$

This criterion guarantees that the design points are not close to each other. It also reduces the problem of singularity of the correlation matrix (R) in a Kriging modelling method.

- **The ϕ_p criterion**

Morris and Mitchell proposed an extension to the maximin distance criterion to search for the optimal design (Morris & Mitchell, 1995). The method starts by calculating intersite distances $d(X_i, X_j)$ between all design points ($i, j = 1, 2,..., n$). Let ($d_1, d_2,..., d_m$) be the list of distinct distances arranged from smallest to the largest and ($J_1, J_2,..., J_m$) is the number of pairs of sites in the design separated by distance d_j, where $1 \leq m \leq \binom{n}{2}$.

The design for which d_1 is maximized is referred to as a maximin design. This idea is extended by inclusion of all elements in the distance and index list as follows:

- (1a) maximizes d_1, and among designs for which this is true;
- (1b) minimizes j_1, and among designs for which this is true;
- (2a) maximizes d_2, and among designs for which this is true;
- (2b) minimizes j_2, and among designs for which this is true;
- ⋮
- (ma) maximizes d_m, and among designs for which this is true;
- (mb) minimizes j_m, and among designs for which this is true;

By this idea, the criterion ϕ_p can be calculated from the following equation

$$\phi_p = \left[\sum_{j=1}^{m} J_j d_j^{-p} \right]^{1/p} \quad (8)$$

, where p is a large enough positive integer, J_j and d_j are specified from X. Morris and Mitchell suggested the choice of p as the smallest value for which ϕ_p and maximin distance criteria agree on the best design (Morris & Mitchell, 1995). In practice, a common robust value of p is 5. Later, the adaptive form of ϕ_p which is simpler than an equation (8) to implement is considered (Jin et al., 2005; Leary et al., 2003). This form can be expressed as follows.

$$\phi_p = \left[\sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{d_{ij}^p} \right]^{1/p} \quad (9)$$

The visualization of the design generated from ϕ_p criteria is already displayed in Figure 1.

4. Concluding remarks

In this paper we have explored the search algorithms, CP, SA and GA, commonly used in the context of computer simulated experiments. According to the results, published in the literature so far, including our own empirical studies. The following conclusions would be made.

1. CP algorithm is recommended for small dimensional problems. However, multiple searches with different starting points should be performed (for more details, see Leary et al., 2003; Li & Wu, 1997; Liefvandahl & Stocki, 1996).

2. If there is no restrictions on time constraint, SA would be the best choice since it is robust to different starting points. Further, SA is better than CP or GA in terms of moving away from a local optimum value of optimality criteria (Morris & Mitchell, 1995; Li & Shigeru, 2001).

3. For a large dimensional problem ($d \leq 10$), GA seems to be the best choice to use. However the parameter selection in the crossover and mutation step is needed to be made with care (Li & Shigeru, 2001).

4. The model independent criteria are more preferable than the model dependent criteria since the knowledge of model for computer simulated experiments is not known a priori (Johnson et al., 1990; Ye et al., 2000).

In addition to CP, SA and GA, other search algorithms like Tabu search, stochastic evolutionary algorithm etc. can be investigated. The technique of particle swarm optimization can also be developed in this area of research.

References

Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H.P. (1996). Experimental design and observation for large systems. Journal of the royal statistical society, Series B, 58, 77-94.

Butler, N.A. (2001). Optimal and orthogonal latin hypercube designs for computer experiments. *Biometrika, 88*(3),847-857.

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

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

Koehler, J. and Owen, A.B. (1996). Computer experiments. *Handbook of Statistics, volumn 13*, Elsevier Science, New York, 261-308.

Leary, S., Bhaskar, A., and Keane, A. (2003). Optimal orthogonal-array-based latin hypercubes. *Journal of Applied Statistics, 30*(5), 585-598.

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

Li, Z. and Shigeru, N. (2001). Maximin distance-lattice hypercube design for computer experiment based on genetic algorithm. *IEEE explore, 2*, 814-819.

Liefvandahl, M. and Stocki, R. (2006). Study on algorithms for optimization of latin hypercubes. *Journal of Statistical Planning and Inference, 136*, 3231-3247.

Mackay, 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*, 239-246.

Morris, M.D. and Mitchell, T.J. (1995). Exploratory design for computational experiments. *Journal of Statistical Planning and Inference, 43*, 381-402.

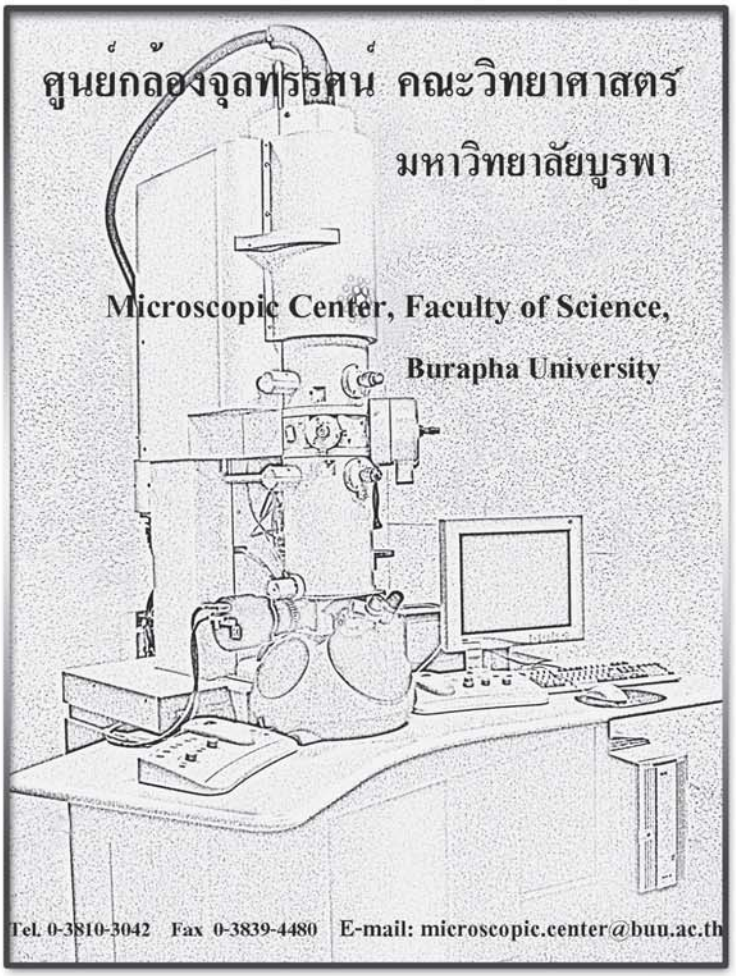
Park, J.S. (1994). Optimal latin hypercube designs for computer experiments. *Journal of Statistical Planning and Inference, 39*, 95-111.

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.

Shewry, M. and Wynn, H. P. (1987). Maximin entropy design. *Journal of applied statistics, 14*(2), 165-170.

Welch, W. J., Buck, R. J., Sacks, J, Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics, 34*, 15-25.

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



วารสารวิทยาศาสตร์บูรพา

ขอเชิญส่งผลงานทางวิชาการ ประเภทบทความทางวิชาการ / วิจัย

เพื่อพิจารณาลงตีพิมพ์ในวารสารวิทยาศาสตร์บูรพา

ติดต่อสอบถามโดยตรงที่

กองบรรณาธิการ วารสารวิทยาศาสตร์บูรพา

อาคารสิรินธร คณะวิทยาศาสตร์

อ.ดร.อาดูลย์ มีพูล หรือ คุณรังสิมา สูตรอนันต์

โทรศัพท์ 0-3810-3042 โทรสาร 0-3839-4480