Enhancement of enhanced stochastic evolutionary algorithm for computer simulated experiment

Thanyadol Chantarawong,* Jaratsri Rungrattanaubol,* and Anamai Na-udom**

Abstract

Computer simulated experiments (CSE) are currently used to replace complex engineering systems and applications. The design of experiment plays the major roles in CSE, since the accurate surrogate model usually obtains from the best experimental design. The process of constructing the design in this context is a space filling proces, so search techniques along with different optimal criterion are applied. This paper purposes the enhancement of enhanced stochastic evolutionary (EESE), adapted from the enhanced stochastic evolutionary (ESE), to search or construct the optimal latin hypercube design (LHD) based on Φ_p criteria. The results show EESE performs better than ESE in any dimensional designs, especcially in a small dimension in terms of time and number of element-exchange.

Keywords: Computer simulated experiment, Stochastic evolutionary, Optimal design, Latin hypercube design.

1. Introduction

Recently computer simulated experiment (CSE) has been widely used in a complex engineering system to analyse the performance of a system. This is probably because some physical experiments have many limits such as very high cost of instruments or too advanced and high technology to run the experiments. CSE consists of 3 major parts; experimental design, simulation routine and approximation model as presented in Figure 1 [1].

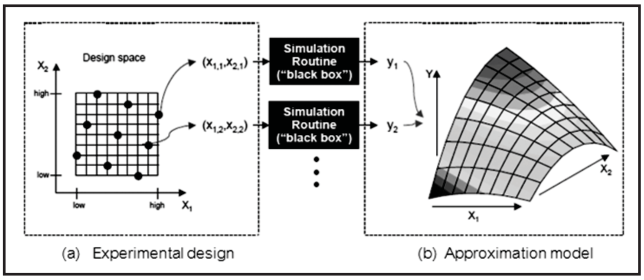

Figure 1 *Three parts of CSE*

The inspiration of CSE is to obtain an approximation model with high prediction accuracy. In order to achieve this, we require an optimal design, which is efficient enough to represent a design sample from a minimal sampling point (run), or the design with high space filling properties (a coverage of design space). The model built from the optimal design is guranteed to be highly accurate.

CSE have widely attracted many researchers. Sack et al. [2] introduced and demonstrated that CSE is correlated with the computational searches for an optimal design and the statistical model building. McKay et al. [3] proposed Latin hypercube design (LHD) for use in CSE. Morris and Mitchell [4] adopted a version of a simulated annealing (SA) along with Φ_p criterion for constructing an optimal design. Thamma et al. [5] presented a modified version of SA to improve the performance of an original SA by adding a tolerance level in a replacement design process. Jin et al. [6] developed Enhanced stochastic evolutionary (ESE) algorithm for searching an optimal design.

An experimental design or so called sampling is a set of sampled value of input variable from a design space. An objective of searching for an optimal design is to obtain a design that meets a space filling property so that the model built from this design will be very accurate. In Figure 1, the input variable X_1 and X_2 were sampled with 9 points of run. The 9 points of run were given as an input to a simulation routine (or "black box"). The output (Y) together with the 9 points of run are formed an approximation model (or a surrogate model).

Simulation routine is typically high fidelity, high cost and takes a long time, since the code is very complex and implemented by an expert. Thus, the design of experiment plays a crucial role here. The question is how we can search for optimal LHDs from the space with size $(n!)^d$, where n is the number of run and d is the number of input variables.

CSE is very complex and contains a lot of input variables. In nature, a design space is very large. To search for an optimal design from a large space, many search algorithms are applied along with various optimality criteria used as objective or fitness functions. Examples of LHD for CSE

* Department of Computer Science and Information Technology, Naresuan University

** Department of Mathematics, Naresuan University

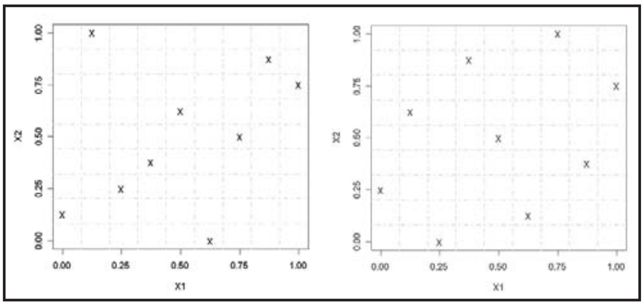

Figure 2 *9×2 LHD before and after applying search algorithm*

are shown in Figure 2, the LHD on the right hand side is an example of an optimal LHD constructed from a search algorithm based on Φ_p criterion, more details in section 2.5.

The aim of this paper is to develop a search technique named Enhanced stochastic evolutionary (ESE) introduced by Jin et al. [6] to construct an optimal LHD based on Φ_p criterion and to improve its performance. The comparative study will be conducted to present how enhancement of ESE (EESE) performs better, especially with a large dimension of design, e.g. 801×20, in terms of computational time and number of exchange elements.

2. Research methodology

We first develop the original ESE and later adapt the EESE using R language. The class of design used here is Latin hypercube design (LHD) along with Φ_p criterion.

2.1 Latin hypercube design (LHD)

LHD is widely used in CSE in a part of experimental design. LHD is $n \times d$ matrix (n runs and d input variables), each column contains $1, 2, 3, \ldots, n$ run, randomly permutation from possible values of input variable, which is divided into an equal interval and takes a form of a unit interval [0,1] as shown in Figure 3.

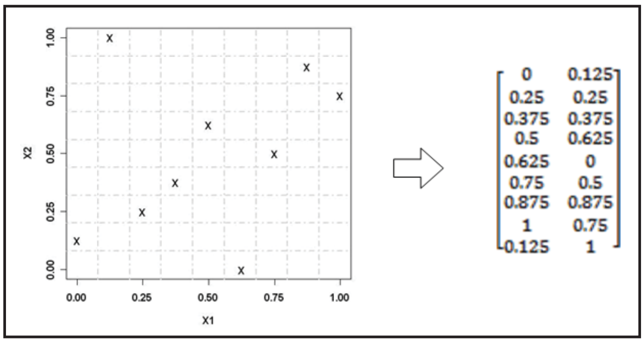

Figure 3 *9×2 Random LHD*

In Figure 3, it is 9×2 LHD, which the 1st column represents the input value for 9 runs of input variable X_1 and the 2nd column represents X_2. The value of each input variable is in the range of [0, 1] with 9 equal interval values. Each row represents a point of run. LHD is flexible since it can apply for a design with any number of input variables and any number of run. This characteristic can ensure the stratified sampling technique [2].

2.2 Element-exchange operation

The element exchange operation to construct a new LHD design is developed from a concept of column-wise operation purposed by Li and Wu [7]. The process is randomly interchange two distinct elements in a column as shown in Figure 4. After element-exchange, the design maintains LHD properties.

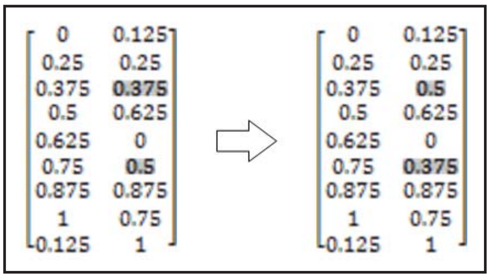

Figure 4 *9×2 Random LHD*

2.3 Enhanced stochastic evolutionary (ESE)

ESE is developed from Stochastic evolutionary (SE) algorithm proposed by Sabb and Rao [8], contains 2 nested loop called inner and outer loop. The inner loop performs a local search process by constructing a new design and decides for an acceptance of this new design. The outer loop over the inner loop performs a global search controlled by adjusting the threshold (T_h) base on acceptance ratio and improved ratio from the inner loop.

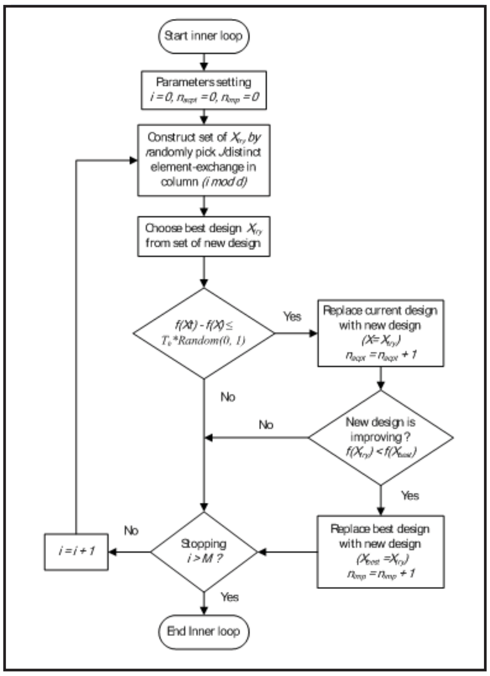

Figure 5 *Flowchart of ESE inner loop* [6]

Inner loop process (Figure 5)

Step 1: after initializing parameters and design X_0 in the outer loop, $X = X_0$, $i = 0$, $n_{acpt} = 0$ and $n_{imp} = 0$

Step 2: construct a set of new design X_{try}

Step 3: select the best design X_{try} from this set

Step 4: decide to accept the best design X_{try} and replace the current best design X from as shown in Figure 5.

Step 5: if X_{try} is better than the global best design X_{best}, replace it with X_{try} and increase n_{imp} by 1 ($n_{imp} = n_{imp} + 1$).

Step 6: end the inner loop if $i > M$ else go to step 2. The parameters J and M in this study is $j = (n_2)/5$ but not larger than 50 and $M = 2(n_2)d/J$ but not larger than 100.

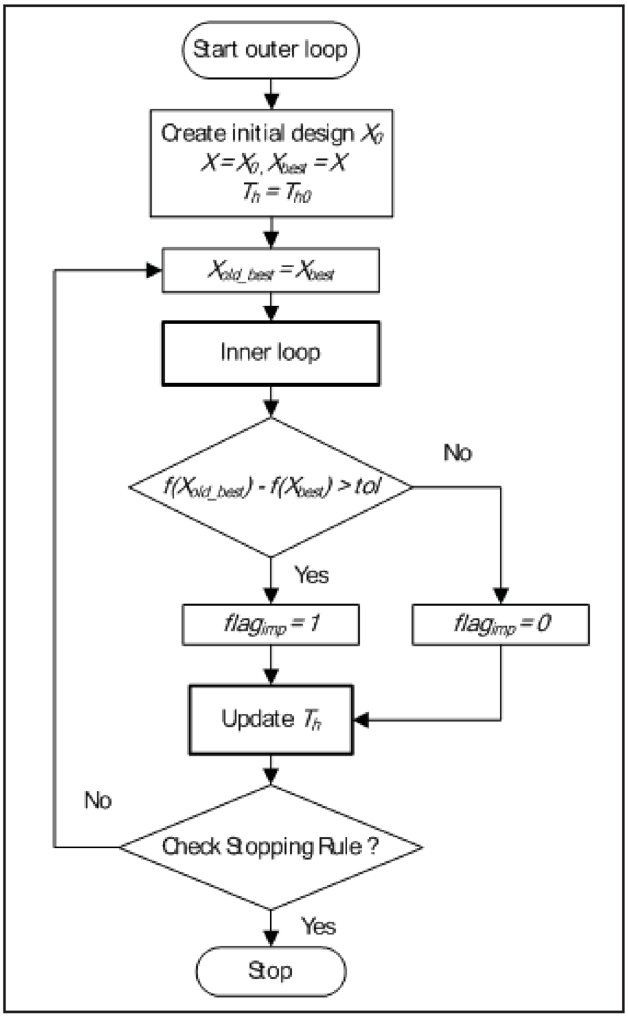

***Figure 6** Flowchart of ESE outer loop [6]*

Outer loop process (Figure 6)

Step 1: randomly create an initial design X_0 and set $X = X_0$, $X_{best} = X$, initialize $Th_0 = 0.005 * \Phi_p(X_0)$ and $Th = Th_0$.

Step 2: let $X_{old\_best} = X_{best}$.

Step 3: go to the inner loop process.

Step 4: select a method to update Th, by setting flagimp.

Step 5: update Th, more details later.

Step 6: terminated by a stopping rule else goto step 2.

The tolerance tol in this paper is set to 0.0001; from the empirical study the smaller value does not improve the search process. The process of updating the value of Th in step 5 is divided into 2 processes named improving process and exploration process. The search process is in the improving process when $flag_{imp} = 1$, if the best design X_{best} is improved in the inner loop. If not, the search process will be in the exploration process ($flag_{imp} = 0$).

In improving process ($flag_{imp} = 1$). Th is adjusted to help in searching for the best in a local optimal design based on an accept ratio (n_{acpt}/M) and improve ratio (n_{imp}/M).

- If $n_{acpt}/M > \beta_1$ and $n_{imp}/M < n_{acpt}/M$, then Th is decreased by equation $Th = \alpha_1 * Th$.
- If $n_{acpt}/M > \beta 1$ and $n_{imp}/M = n_{acpt}/M$, then Th is unchanged.
- Otherwise, Th is increased by equation $Th = Th/\alpha_1$, where $0 < \alpha_1 < 1$and $0 < \beta_1 < 1$,we suggest $\alpha_1 = 0.8$ and β_1 should be small we suggest $\beta_1 = 0.1$ [6].

In exploration process ($flag_{imp} = 0$). Th will be adjusted to help algorithm move far away from a local optimal design based on the range of accept ratio.

- If $n_{acpt}/M < \beta_2$, then Th is increased till $n_{acpt}/M > \beta_3$ by equation $Th = Th/\alpha_3$.
- If $n_{acpt}/M > \beta_3$, then Th is decreased till $n_{acpt}/M < \beta_2$ by equation $Th = \alpha_2 * Th$,

where $0 < \alpha 2 < \alpha_3 < 1$ and $0 < \beta_2 < \beta_3 < 1$, we suggest $\alpha_2 = 0.9$, $\alpha_3 = 0.7$. While β_2 should be small, we set $\beta_2 = 0.1$ and β_3 should be large enough, $\beta_3 = 0.8$ [6].

2.4 Enhancement of ESE (EESE)

EESE is modified from ESE by combining the advantage of SA (i.e. local search process) and the advantage of ESE (i.e. global search process). EESE contains 2 nested loop similarly to ESE as displayed in Figure 7. The outer loop is almost the same as in ESE, there is only a chage in a stopping rule in step 6. The maximum number of cycles used is replaced by the following condition. If a local best design after the inner loop X_{best} is not improved from the global best design $X_{globalbest}$ δ consecutive times, then the search process will terminate. In this study we set δ to 10.

In the inner loop, there are many changes. Step 2, step 5 and step 6 are changed. In step 2, the process of constructing a new design X_{try} is changed to element-exchange in column (i mod d) for all J iterations while the original ESE randomly picks J distinct element-exchange in column (i mod d). Then the computational complexity decreases from $O(n^2)$ to $O(n)$.

It is proven by; in ESE a random element-exchange process

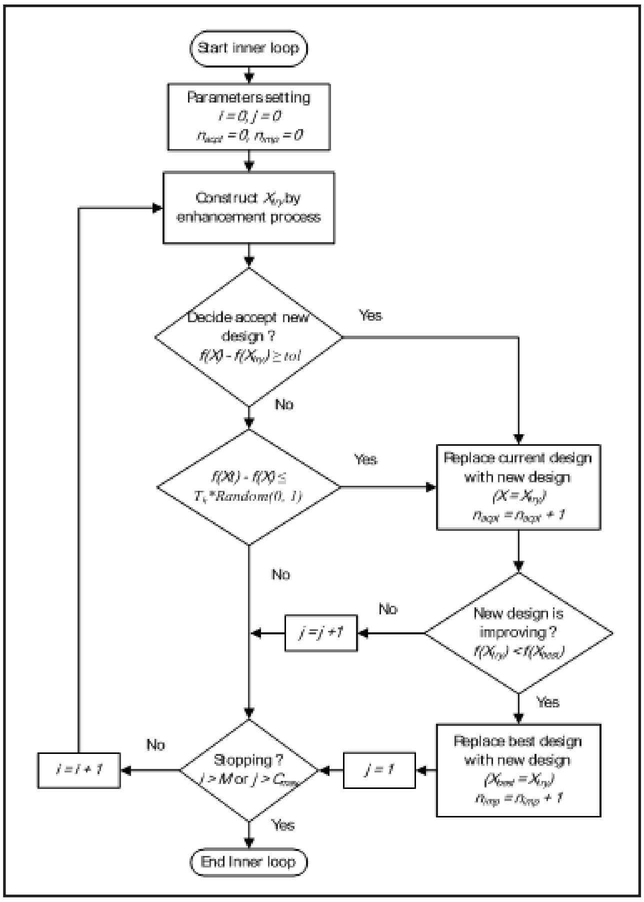

Figure 7 *Flowchart of EESE inner loop*

for all J iteration, then in every iteration i musts check for a distinct *i-1* loop, hence the complexity is $O(n*(n-1)) = O(n^2)$. In EESE, we adapted the process of element-exchange from SA shown in Figure 8. J iteration element-exchange of a current design X in column *i mod d* is independent. It is no need to perform all J iterations. The computation complexity becomes $O(n)$.

In step 5, if a new design X_{try} is improved to be better than the best design X_{best}, let $j = 0$ otherwise increase j by 1 ($j = j + 1$). Finally, in step 6 of the inner loop a stopping rule is modified to if $i > M$ or $j > C_{max}$. In this study, we use $C_{max} = 10$.

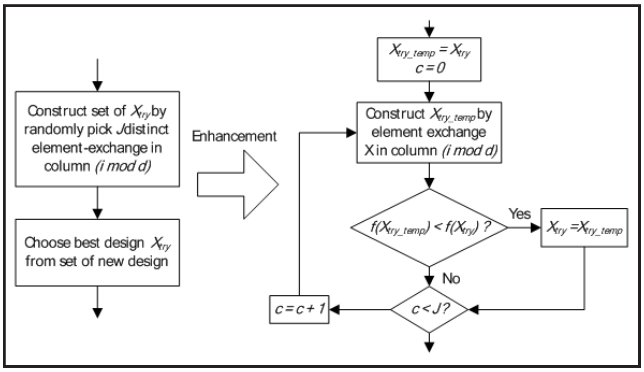

Figure 8 *A new design construction using SA (Step 2)*

2.5 Φ_p Criterion

In this study, the Φ_p criterion, proposed by Morris and Mitchell [4], is used to evaluate the design. It is developed from a maximin distance criterion. Given design X, the Euclidean inter-site distance between the sample points is calculated from equation (1).

$$d(x_j, x_k) = \left[\sum_{x=1}^{k} (x_{ij}, x_{ki})^2\right]^{1/2} \quad (1)$$

From Euclidean distance for X, we get a symmetric matrix $D = [d_{j,k}]n \times n$. Then Φ_p is calculated by equation (2).

$$\Phi_p = \left[\sum_{i=1}^{n}\sum_{j=i+1}^{n} (1/d_{ij})^p\right]^{1/p} \; ; p \in I^+ \quad (2)$$

Equation (2) is a direct method to compute Φ_p, which takes a long time. Jin et al [6] noticed that in the process of element-exchange in ESE, only some values in matrix D are changed, so it is no need to recalculate the complete matrix D. In element-exchange operation, after exchanging elements of a design X between rows i_1 and i_2 within column k ($x_{i1k} \leftrightarrow x_{i2k}$), there are changes only elements in row i_1 and i_2 and column i_1 and i_2 in matrix D [6]. For any $1 \le j \le n$ and $j \ne i_1, i_2$ let

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

Then $$d'_{i_1j} = d'_{j_i1} = \left[d^t_{i_1j} + s(i_1, i_2, k, j)\right]^{1/t}$$

and $$d'_{i_2j} = d'_{j_i2} = \left[d^t_{i_2j} - s(i_1, i_2, k, j)\right]^{1/t}$$

The simplified version of Φp calculation is:

$$\Phi'_p = \left[\Phi^p_p + \sum_{j-1}^{n}\left[(d'_{i_1j})^{-p} - (d_{i_1j})^{-p}\right] + \sum_{j-1}^{n}\left[(d'_{i_2j})^{-p} - (d'_{i_2j})^{-p}\right]\right]^{1/p} \quad (3)$$

where $j \ne i_1, i_2$, and $p = 5$ and $t = 2$. This version of Φ_p has a significant reduction of computational time comparing to Φ_p calculation in equation (2) [6].

2.6 Data quality

We compared EESE with ESE to construct optimal LHDs for a given dimension of design shown in Table 1.

Table 1 *Dimension of LHDs*

d	2	5	10	15	20
n	9	51	201	451	801

For each dimension of LHDs, we run ESE and EESE for 10 times on a PC with AMD Athlon 64 X2 3.0 GHz CPU. The results in terms of efficiency are timed in second and number of element-exchange to construct optimal LHDs with the minimization of Φ_p criterion.

3. Results and comparative studies

The results in terms of Φ_p criterion values for each dimension of LHDs obtained from ESE and EESE are presented in Table 2. In Table 2, Φ_p values from ESE are slightly lower than EESE when the dimension is small whereas EESE is superior to ESE when the dimension becomes larger. For large dimension of designs, these two algorithms perform similar results of Φ_p values, indicates that either ESE or EESE can be applied.

The results based on the performance (efficiency) for ESE and EESE are presented in Table 3. When the dimension of design is small, EESE converges much faster than ESE and the number of exchanges required in the search process is less than the ESE. For a large dimension of design, the number of exchange is fixed at 100000 EESE converges slightly faster than ESE as shown in Figure 9. This indicates if time constraint is taken into account, EESE seems to be the better choice to use in the construction of the optimal LHD designs.

Table 2 *The values of Φ_p criterion from ESE and EESE*

LHDs	Algorithm	Φ_p (p = 5, t = 2)			
		Min	Max	Mean	SD
9 × 2	ESE	4.273	4.344	4.300	0.035
	EESE	4.273	4.364	4.317	0.038
51 × 5	ESE	5.424	5.439	5.432	0.005
	EESE	5.427	5.439	5.433	0.004
201 × 10	ESE	6.170	6.174	6.172	0.001
	EESE	6.171	6.173	6.172	0.000
451 × 15	ESE	6.760	6.762	6.761	0.000
	EESE	6.760	6.761	6.761	0.000
801 × 20	ESE	7.253	7.254	7.254	0.000
	EESE	7.253	7.254	7.254	0.000

Table 3 *Performance of ESE and EESE*

LHDs	Algorithm	*Performance (Average)*		*Time ESE/EESE*
		Time (sec.)	#Exchange	
9 × 2	ESE	1.829	2880	1.866
	EESE	0.98	2281	
51 × 5	ESE	358.976	100000	3.056
	EESE	117.428	44080	
201 × 10	ESE	1349.488	100000	1.119
	EESE	1205.082	96915	
451 × 15	ESE	3648.646	100000	1.044
	EESE	3492.132	100000	
801 × 20	ESE	7744.424	100000	1.009
	EESE	7672.487	100000	

4. Conclusions

This paper presents the method to enhance the ESE to construct the optimal LHD for use in CSE. The major enhancement appears in the inner loop as shown in Figure 7 and 8. As presented in the previous section, EESE perform better than ESE in terms of efficiency. Therefore EESE would be recommended in the construction of LHD for use CSE. In order to extend and complete the conclusion, the validation of the surrogate model accuracy could be further investigated.

5. References

[1] T.W. Simpson, D. K. J. Lin, and W. Chen, "Sampling Strategies for Computer Experiment: Design and Analysis," *International Journal of Reliability and Applications*, Vol. 2, No. 3, pp. 209-240, 2000.

[2] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp. 409-435, 1989.

[3] M. D. McKay, R. J. Beckman, and W. J. Conover," A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, pp. 239-245, 1979.

[4] M. D. Morris, and T. J. Mitchell, "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, Vol. 43, pp. 381-402, 1995.

[5] T. Thamma, J. Rungrattanaubol, and A. Na-udom, "Modification on Search Algorithm for Computer Simulated Experiment," *Proceedings of national operations research conference*, pp. 117 – 123, May 15th-24th, Thailand, 2008.

[6] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for construct optimal design of computer experiment," *Journal of statistical Planning and Inference*, Vol. 134, pp. 268-287, 2005.

[7] W. Li, and C. F. J. Wu, "Columnwise-Pairwise Algorithms with Applications to the Construction of Supersaturated Designs," *Technometrics*, Vol. 39, pp. 171-179, 1997.

[8] Y. G. Saab, and Y. B. Rao, "Combinatorial optimization by stochastic evolution," *IEEE Transaction on Computeraided Design*, Vol.10, pp. 525–535, 1991.