應用特定型態質群最佳化演算法於搜尋 *D*-Optimal 最佳設計之研究

Construct *D*-Optimal Designs Using Modified Particle Swarm Optimization

范書愷* 張克偉
Shu-Kai S. Fan*, Ko-Wei Chang

摘要

最佳化實驗設計在統計實務應用中，針對不對稱的實驗區域設計方法有其存在之必要性。本研究使用修正的質群演算法於一連串 *D* 最佳化設計問題，且在搜尋最佳解中找出適當方法平衡質群演算法於速度更新的搜尋能力。利用保留區域可行解重新設定不可行解在實驗區域邊界上的方式，進而處理混合實驗與複雜限制式的問題，實驗結果驗證本方法可獲得良好的績效。

關鍵詞：反應曲面法，最佳化設計準則，質群演算法，保留區域可行解法

Abstract

Alphabetic designs are applied extensively to the engineering design problems as the standard experimental designs cannot be directly used or some design variable is restricted due to practical limitation. In this situation, computer-generated designs are a widely accepted alternative to deal with an irregular region of experimentation, a non-standard model or specific design criteria. The most popular one is *D*-optimal designs which would minimize the volume of the joint confidence region on the vector of regression coefficients. This paper presents a new optimization technique to generate *D*-optimal designs using the modified particle swarm optimization (termed MPSO). The prominent merit of the proposed method is the ability to more likely reach a global optimal set of design points than using exiting techniques as a mixture design is solved. The experimental results obtained from running a set of test problems are used to illustrate MPSO.

Keywords: response surface methodology (RSM), design optimality criteria, particle swarm optimization (PSO), D-optimal design

I. INTRODUCTION & BACKGROUND

With progress and development in industry, alphabetic designs have been applied extensively to the statistical design situations in common practice. The settings of parameters are not easily to adjust subject to the constraints of design and execution of experiments. Asymmetrical design, factorial design, and mixture design could be frequently encountered in some realistic situations. For example, in mixture experiment, there are often constraints required to impose on the component proportions that prevent ones from extrapolating outside the simplex experimental region. These restricted conditions result in the form of lower and upper bounds on the component proportions. If the region of interest for the experiment is not a cube or a sphere, standard designs may not be a logical choice for such a circumstance. Furthermore, irregular experimental regions of interest still occur fairly often in practical applications. A mixture experiment is considered a special type of response surface methods in which the factors are the ingredients or components. The experimental region of mixture experiments is of simplex type. Typically, standard designs have assured degrees of precision with orthogonality and other optimal properties which are of primary importance for the exploratory nature of most experiments. In some instances, optimal designs are one of few alternatives an experimenter can count on if a regular design is not applicable. In other words, the traditional response surface methods are not effective enough to construct those nonstandard designs and also very difficult to deal with constraints. Optimal designs are beneficial for situations where standard designs cannot be readily

元智大學工業工程與管理學系
*Corresponding author. E-mail: simonfan@saturn.yzu.edu.tw
Department of Industrial Engineering and Management, Yuan-Ze University, Chung-Li, Taiwan, R.O.C.
Manuscript received 13 December 2008; revised 7 January 2009; accepted 27 February 2009

applied. Therefore, we consider using a particle swarm optimization (PSO) based algorithm to solve this problem. From the literature, most of research works have shown a wide body of successful applications of PSO in solving continuous, nonlinear problems. Certainly, PSO has advantage and attraction in solving continuous optimization problems. Furthermore, PSO-based algorithms are extremely easy-to-implement in practice as compared with other evolutionary algorithms. The proposal to be studied here is that PSO might be a competitive alternative to yield better optimization outcomes in faster and accurate convergence when applied to *D*-optimal design problems. To improve the computation efficiency of the basic PSO in the *D*-optimal problem, a modified particle swarm optimization (termed MPSO) is addressed by introducing an adaptive updating formula for balancing local and global versions in basic PSO (termed BPSO). To verify, a clear-cut design of experiment is conducted for investigating the parameters setting in MPSO, and then a set of suitable setting for general use in solving nonlinear, mixture, *D*-optimal design problems with constraints is suggested. At last, the performance of MPSO is evaluated and compared to other algorithms.

II. LITERATURE REVIEW FOR THE CONSTRUCTION OF *D*-EFFICIENT DESIGNS

1. Optimal Design

To introduce the theory of optimal design, initially consider the standard linear model (Khuri and Cornell [1]) in *k* variables,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{Y}$ represents the response vector, $\mathbf{X}$ is an $N \times p$ ($k < p$) matrix of design variables in model form while estimating the elements of the $p \times 1$ parameter vector $\boldsymbol{\beta}$. It is noted that the elements of $\mathbf{(X'X)^{-1}}$ are proportional to the variance and covariance of the elements in $\hat{\boldsymbol{\beta}}$, the vector of least squares estimates of $\boldsymbol{\beta}$. ε is a random error term, usually assumed to be $N\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$. The parameters in $\boldsymbol{\beta}$ are estimated via least squares using the system of equations defined by

$$\mathbf{(X'X)}\hat{\boldsymbol{\beta}} = \mathbf{X'Y} \tag{2}$$

where $\mathbf{X'X}$ is a symmetric, invertible matrix. The solution to this system of equations is given by

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'Y} \tag{3}$$

with the variance-covariance structure

$$Var\left(\hat{\boldsymbol{\beta}}\right) = \sigma^2\left(\mathbf{X'X}\right)^{-1} \tag{4}$$

At a particular point $\boldsymbol{x}$ inside the experimental region, the predicted value of the response is

$$\hat{\boldsymbol{Y}}\left(\boldsymbol{x}\right) = f\left(\boldsymbol{x}\right)\hat{\boldsymbol{\beta}} \tag{5}$$

and the predicted value $\hat{Y}(x)$ has a variance of

$$Var\left(\hat{\boldsymbol{Y}}\left(\boldsymbol{x}\right)\right) = f'\left(\boldsymbol{x}\right)\left(\mathbf{X'X}\right)^{-1} f\left(\boldsymbol{x}\right)\sigma^2 \tag{6}$$

where $f'\left(\boldsymbol{x}\right)$ is a vector of p $\left(p \geq k+1\right)$ functions that model how the response depends on $\boldsymbol{x}$ and depends on both the response values and the settings of regressor within the design region. Under the assumption that the model errors are i.i.d. $N\left(0, \sigma^2\mathbf{I}\right)$, the estimators in $\hat{\boldsymbol{\beta}}$ are unbiased and have minimum variance. Some ideas associated with choosing a design are almost always centered on the selection of the particular settings of $\mathbf{X}$ that make the $p \times p$ matrix $\mathbf{(X'X)^{-1}}$ as "small" as possible, serving as a basic criterion for the *D*-optimal design. The *D*-optimal criterion is to select the design points so as to maximize the determinant $\left|\mathbf{X'X}\right|$. The problem is thus the one of selecting the available number of design points to maximize this determinant. In the moment matrix, it gives

$$\mathbf{M} = \frac{\mathbf{X'X}}{N} \tag{7}$$

Then, the inverse of $\mathbf{M}$ becomes

$$\mathbf{M^{-1}} = N\left(\mathbf{X'X}\right)^{-1} \tag{8}$$

the scaled dispersion matrix that contains the variance and covariance of regression coefficients, scaled by N/σ^2. It turns out that an important "norm" on the moment matrix is the determinant; that is,

$$\left|\mathbf{M}\right| = \frac{\left|\mathbf{X'X}\right|}{N^p} \tag{9}$$

where p is the number of parameters in the model. The determinant of $\mathbf{X'X}$ is inversely proportional to the square of the volume of the confidence region on the regression coefficients. The volume of the confidence region is of practical relevance since it reflects how well the set of coefficients were being estimated. So, a *D*-optimal design is the one in which $\left|\mathbf{M}\right| = \left|\mathbf{X'X}\right|/N^p$ is maximized; that is,

$$\underset{\zeta}{Max}\left|\mathbf{M}\left(\zeta\right)\right| \tag{10}$$

where *Max* implies that the maximum is taken over all possible designs ζ. As a result, it is natural to define the *D*-efficiency (Myers and Montgomery [2]) of a design ζ^* as

$$\mathrm{D}_{\text{efficiency}} = \left(\frac{\mathbf{M}\left(\boldsymbol{\zeta}^*\right)}{\underset{\boldsymbol{\zeta}}{\mathrm{Max}}\ \mathbf{M}\left(\boldsymbol{\zeta}\right)}\right)^{\frac{1}{p}} \tag{11}$$

the $1/p$ power takes account of the p parameter estimates being assessed when one computes the determinant of the variance-covariance matrix. The definition of *D*-efficiency in Equation (11) allows for comparing different designs or designs that use different sample sizes by comparing *D*-efficiencies. Optimal design methodologies, in general, attempt to select experiment runs that optimize some criterion under all applicable constraints. *D*-optimality criterion is the one that selects design points to minimize the volume of the joint confidence ellipsoid for the estimated regression coefficients.

2. Algorithms for Construction of *D*-optimal Designs

Welch [3] developed a branch-and-bound approach that constructs a catalog of all *D*-optimal *n*-point designs for specified design, linear model, and number of experimental runs, *n*. It guarantees that global *D*-optimal designs will be found. Nonetheless, the number of nodes that needs to be examined quickly grows beyond current computing capabilities for at least modest-sized problems. As experimental runs and variables increase so the number of possible designs increases exponentially and the execution time rapidly becomes prohibitive.

Haines [4] employed simulated annealing to generate *D*-optimal designs. In an annealing process a melting point, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. Therefore, the process can be thought of as an adiabatic approach to the lowest energy state. If the initial temperature of the system is too low or cooling is done insufficiently slowly the system may become quenched forming defects or freezing out in meta-stable states. The computation time grows exponentially with the number of variables; therefore, a complex problem requires longer time to obtain the best solution. Exchange algorithms (EA) have long been the standard approach for the construction of *D*-optimal designs, and some of the most popular, commercially available software packages make use of them in their design generation routines.

The first EA is attributed to Fedorov's exchange algorithm. Afterwards, modified Fedorov exchange algorithm (MFEA) was proposed. It was introduced by Cook and Nachtsheim [5] which will be used for the comparison purpose in every example shown later. This algorithm selects an experimental run in the current design and finds a run in the candidate set. The exchange is carried out, the design is updated, and the procedure continues until convergence is reached. Genetic algorithms (GAs) are iterative optimization procedures that repeatedly apply mating, selection, and mutation operations to a group of suitably encoded solutions until some criterion of convergence has been met.

Recently, Heredia-Langner [6] *et al.* expand GA for the construction of *D*-optimal designs. In their study, GA eliminates the need to explicitly consider a candidate set of experimental points and it can be used in highly constrained experimental regions while maintaining a level of performance comparable to more traditional design construction techniques. The addition of GAs to the current design building techniques gives practitioners valuable alternatives in problems of practical importance. However, it must take a great deal of computation time in function evaluation.

Ho [7] *et al*. enhanced the global searching ability of the available PSOs. A novel formula for updating the particles' velocity and positions, as well as the introduction of craziness parameter, is developed. Typically, the BPSO algorithm had difficulties in striking a balance between exploration and exploitation. Hence, the global search ability of BPSO algorithm is restricted. They suggested some adjustments to address this problem. The attention is first paid on the velocity and position updating. Since the "random numbers" corresponding to c_1 and c_2 are independently and randomly generated, there are cases in which the two random parameters are both too large or to small. Both the personal and social experiences accumulated thus far might be overused and the particle is driven too far away form the local optimum. Both the personal and social experiences are not used fully, and the convergence speed of the algorithm is reduced. The two random weighting parameters are not completely independent. By modeling this reasoning ability into the updating formula, the sum of the two interrelated weighting parameters is set equal to 1. Only one random parameter is used to include the collective experiences of the individual particle and his neighbors when updating the velocity.

To control the balance of global and local searches, another random parameter r_2 as below, is introduced in their proposed algorithm. After including all the aforementioned improvement perspectives, the new formula for velocity updating becomes

$$v_{i,d}(k+1) = r_2 v_{i,d}(k) + (1-r_2)\left[c_1 r_1 \left(p_{i,d} - x_{i,d}(k)\right) + c_2 (1-r_1)\left(g_{i,d} - x_{i,d}(k)\right)\right] \tag{12}$$

where r_1 and r_2 are two random parameters which are chosen uniformly within the interval [0, 1]. Owing to birds flocking for food, there could be some rare cases that after the position of the particle is changed according to position update, a bird may not, due to inertia, fly toward a region at which it is thought most promising for food. Instead, it may be bound for a region which is in the opposite direction of what it should fly in order to reach the expected promising regions. As a consequence, in the step that follows, the direction of the bird's velocity should be reversed in order for it to fly back into the promising region. By modeling this fact in the proposed PSO algorithm, the former formula is further modified into

$$v_{i,d}(k+1) = r_2 sign(r_3) v_{i,d}(k) + (1-r_2)\left[c_1 r_1 \left(p_{i,d} - x_{i,d}(k)\right) + c_2 (1-r_1)\left(g_{i,d} - x_{i,d}(k)\right)\right] \tag{13}$$

where r_3 is a random parameter uniformly taken from the interval [0, 1], and the sign is defined by

$$sign(r_3) = \begin{cases} -1 & (r_3 \le 0.5) \\ 1 & (r_3 > 0.5) \end{cases} \tag{14}$$

Then, the position update formula is

$$x_{i,d}(k+1) = x_{i,d}(k) + v_{i,d}(k+1) \tag{15}$$

Michalewicz [8] *et al*. (1996) stated that industrial engineering problems usually are quite hard to solve due to a high complexity of the objective functions and a significant number of problem-specific constraints; often an algorithm to solve such problem requires incorporation of some heuristic methods. They discussed several methods for handling feasible and infeasible solutions in a population. The different problems require different treatments.

Some similar methods also are addressed by Coath and Halgamuge [9]. Hu and Eberhart [10] illustrated preserving feasibility strategy employed to deal with constraints. Many scholars proposed some different method to handle constrained problems.

III. PROPOSED PARTICLE SWARM OPTIMIZATION

1. MPSO Method

The initial swarm, consisting of N particles, is constructed using random generation inside the feasible region. To handle constraints, no matter the range of each design variable or mixture component variable, the positions of particles are produced in restricted space at the beginning. In this way, initially feasible solutions can be easily produced. Moreover, the initial solutions are all imposed on the edge of the constraints. It has long been manifest that the solutions of the *D*-optimal design problem are frequently discovered to be located at boundary. Here, the objective (*i.e.*, fitness) to be evaluated is the determinant of $\mathbf{(X'X)^{-1}}$. Hence, the smaller the determinant of $\mathbf{(X'X)^{-1}}$ is, the better fitness the obtained design has. As will be seen from the experimental results shown in a later section, PSO often selects points at the edge and produces a design with quite high efficiency. What follows explains how the velocity update influences the objective improvement. A better fitness can be anticipated if the step size of velocity update is adequately enlarged. As a *D*-optimal design is sought, it is important for a search procedure to be developed that can arrive at the boundary swiftly. Hence, the inflation factor "*K*" is introduced in the process of velocity update in MPSO. The factor "*K*" is used to stabilize the quality of search for solutions throughout the entire optimization process. Without considering the inflation factor, the solution is found not as stable as the one having it. Consequently, "*K*" can improve Equation (13) on finding better *D*-optimal designs. By integrating all the aforementioned concepts, the brand new formula for velocity update becomes

$$v_d^i(k+1) = K \times \left\{ r_2 sign(r_3) v_d^i(k) + (1-r_2)\left[c_1 r_1 \left(p_d^i - x_d^i(k) \right) + c_2 (1-r_1)(g_d^i - x_d^i(k)) \right] \right\} \quad (16)$$

where r_1 and r_2 are two random numbers chosen uniformly from the interval [0, 1], and r_3 is another random number from the interval [0, 1]. *sign* is a sign function defined as in Equation (14). To avoid overshooting the boundary of the constrained experimental region, the maximum velocity method is also adopted as below

$$v_d^i(k+1) = \frac{v_d^i(k+1) \cdot v_d^{\max}}{\left| v_d^i(k+1) \right|} \quad \text{if } \left| v_d^i(k+1) \right| > v_d^{\max} \quad (17)$$

Subsequently, particles' positions are updated through Equation (15). The formal PSO procedure is described by Figure 1.

2. Illustrating Examples

2-1. Case 1: A Series of Factorial Region

Meyer and Nachtsheim [11] measured the performance of their coordinate exchange method by comparing it to the traditional *k*-exchange method on a series of problems. In their examples, a full quadratic model as in Equation (18) on a number of variables ranging from 2 to 7 was considered. All variables take values in {-1, 0, 1}. Both saturated and non-saturated designs are constructed.

$$E(Y) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \sum_{i=1}^{k} \beta_{ii} x_i^2 \quad (18)$$

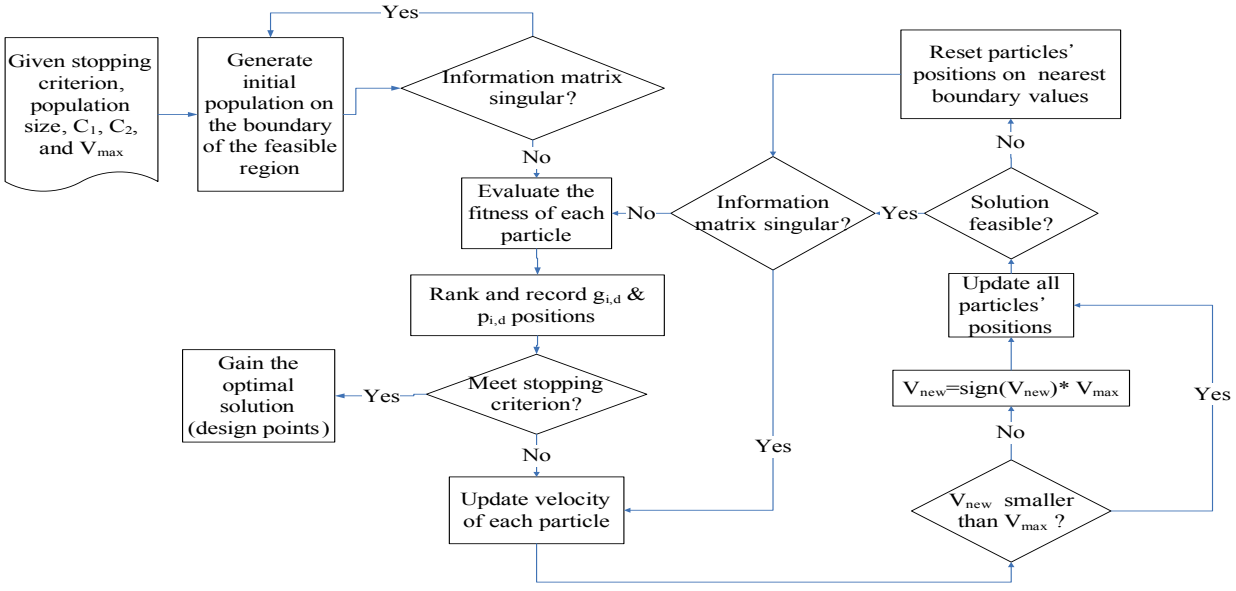

Fig. 1 The procedure of modified particle swarm optimization (MPSO)

MPSO is preliminarily tested in this same set of problems for the validation purpose. Considered, for instance, is the largest problem shown below. This design has 7 design variables in the $\{-1, 0, 1\}$ set and 54 runs are available. There are 3^7=2187 distinct candidate points in the design. For one MPSO solution in this example, there are $(3^7)^{54}$=2.2482e+180 combinations. One random initial design starts the algorithm which then proceeds to find the run in the candidate set that, when exchanged with a run in the current design, results in the largest improvement in the value of $|\mathbf{(X'X)}|$.

The MPSO works with a population of 50 particles that present 50 objective evaluations per iteration. Convergence was achieved after 1000 iterations elapsed. The number of variables is chosen and all initial solutions are produced within [-1, 1]. At first, positions of all particles are updated. Then, all particles are examined if their current locations are within [-1, 1]. If the new position falls outside the feasible region, the position is set to the extreme point set {-1, 1} according to the original search direction. Furthermore, singularity is checked for the currently obtained design. As convergence is achieved, the global best optimal design is reported with the minimum value of $|\mathbf{(X'X)^{-1}}|$. The design grid and region are shown as Figure 2.

2-2. Case 2: A Mixture Problem with One Processing Variable

This problem was the subject of analysis by Welch [12] who used an exchange algorithm to create *D*-efficient designs. Mixture problems are naturally constrained to simplex regions, and further restrictions often limit the region of the simplex that can be usefully applied. Another processing variable to this environment much complicates the construction of efficient designs by traditional approaches even further. With three mixture variables x_1, x_2, x_3, and processing variable x_4, the simplex space for the mixture components can be written as $x_1 + x_2 + x_3 = 1$. The processing variable is such that $x_4 \in \{-1, 0, 1\}$, although it is understood that, in practice, these values may be part of "continuous" region. The model under consideration is

$$E(Y) = \sum_{i=1}^{3} \beta_i x_i + \sum_{j=1}^{3} \sum_{k=j+1}^{4} \beta_{jk} x_j x_k + \beta_{44} x_4^2 \qquad (19)$$

This equation has ten unknown parameters. Weleh [12] developed experimental designs ranging from 10 to 15 runs. Each mixture variable is located at 13 equally spaced levels $\{0, 1/12, 2/12, \ldots, 1\}$. The design grid region in 3-variable simplex case is shown as Figure 3.

The update rules of new velocity and position updates are already described in MPSO. After position update, we must check whether all particles' positions are still in the feasible region and also on the grid. If the new position falls outside the feasible region, the position of "first variable" is reset on the nearest boundary. The design region for this problem is shown as Figure 4.

IV. COMPUTATIONAL EXPERIENCE

1. Design of Experiments (DOE) for Parameters Setting of MPSO

For each selected *K* in all design problems, five runs are independently performed and the iterations required for convergence are recorded. Some excerpts of computational results are listed in Tables 1-3. It can be seen from Table 1 that 1.15 seems a legitimate choice for *K* since MPSO with *K*=1.15 returns the highest *D*-efficiency within a reasonable number of iterations. Also, when *K* =1.15, the value of coefficient of variation (CV) of $|\mathbf{(X'X)^{-1}}|$ reaches the minimum. *D*-efficiency increases as *K* increases from 0.95 to 1.15. However, *D*-efficiency decreases monotonically from *K* =1.15 to *K* =1.35. Based on the computational experience learned from the literature, the stochastic factors c_1 and c_2 are the two most significant factors of all in the updating equation. A positive constant 2 is typically recommended for these two stochastic factors. The less dependent is on personal experience, the more capable of turning to leader direction is PSO. Weighting more social experience than personal experience helps PSO converge on

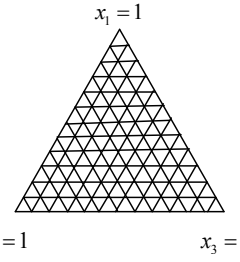

Fig. 3 The design grid region in 3-variable simplex case

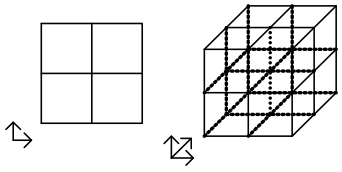

Fig. 2 Factorial regions and design points in 2-variable and 3-variable cases

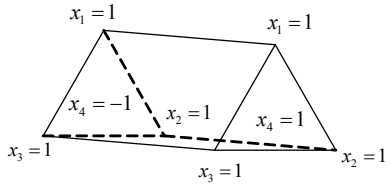

Fig. 4 The combined regions with x_4= -1 and 1

Table 1 The *D*-efficiency and its CV of the solutions obtained using different *K* levels

K level	0.95	1	1.05	1.1	1.15
D-efficiency	87.048	92.81	92.9	94.4	**97.32**
Average CV of *D*-efficiency	0.836	0.758	0.809	0.833	**0.386**
K level	1.2	1.25	1.3	1.35	
D-efficiency	94.99	94.1	93.098	91.16	
Average CV of *D*-efficiency	0.678	0.622	0.6049	0.651	

Table 2 The *D*-efficiency and its CV of solutions obtained in terms of 12 different combinations of stochastic factors

Stochastic factors	$C_1=1$ $C_2=1$	$C_1=1$ $C_2=2$	$C_1=1$ $C_2=3$
D-efficiency	74.2	93.2	**95.9**
Average CV of *D*-efficiency	1.22	0.92	0.58
Stochastic Factors	$C_1=1$ $C_2=4$	$C_1=2$ $C_2=1$	$C_1=2$ $C_2=2$
D-efficiency	81.7	80	94.8
Average CV of *D*-efficiency	0.56	1.12	0.6
Stochastic factors	$C_1=2$ $C_2=3$	$C_1=2$ $C_2=4$	$C_1=3$ $C_2=1$
D-efficiency	94.7	78.2	93.1
Average CV of *D*-efficiency	**0.44**	0.7	0.69
Stochastic factors	$C_1=3$ $C_2=2$	$C_1=3$ $C_2=3$	$C_1=3$ $C_2=4$
D-efficiency	93.2	91.9	76.7
Average CV of *D*-efficiency	0.69	0.53	0.72

a specific solution found, i.e., $c_2 > c_1$. Twelve different combinations of c_1 and c_2 were studied through a set of 16 test problems for the MPSO algorithm. The performances of *D*-efficiency and average CV of *D*-efficiency were recorded in Table 2. It was discovered that the maximum *D*-efficiency was achieved at the combination of $c_1 = 1$ and $c_2 = 3$. When c_1 was fixed at 1, a local refinement about c_2 was investigated within [2.5, 3.5]. The results of local refinement about c_2 revealed that the level of 2.8 together with $c_1 = 1$ yielded the best performance on *D*-efficiency with an acceptable CV value. Hence, the parameter setting of $K = 1.15$, $c_1 = 1$ and $c_2 = 2.8$ was suggested for MPSO as the *D*-optimal design problem is being solved.

2. Experimental Results

2-1. A Series of Factorial Region

In this section, MPSO is examined in the same set of problems previously used for the validation purpose. An initial design is randomly generated to start the algorithm which then proceeds to find the run in the candidate set that, when exchanged with a run in the current design, results in the largest improvement in the value of $|\mathbf{(X'X)}|$.

Table 3 Local refinement of c_2 when c_1 is fixed at 1

c_2	2.5	2.6	2.7	2.8	2.9	3.0
D-efficiency	94.6	95	94.1	**95.9**	94.5	93.7
Average CV of *D*-efficiency	0.76	0.97	0.84	0.68	0.64	0.82
c_2	3.1	3.2	3.3	3.4	3.5	
D-efficiency	93.9	93.8	94.3	92.9	94	
Average CV of *D*-efficiency	0.81	0.69	**0.59**	0.72	0.64	

This exchange is carried out, and the procedure is repeated until all the runs in the design have been considered for exchange. The MPSO works with a population of 50 particles. This produces 50 objective function evaluations per iteration. For this particular example, convergence was achieved after 12000 iterations. This produces a total of $50 \times 12000 = 600000$ objective function evaluations to achieve a design with the same efficiency as the final answer obtained the same number of decimal by the GA. It should be pointed out that MPSO eliminates the need to explicitly consider a candidate set of experimental points like EA, such as the modified Fedorov's algorithm. In BPSO, we must choose the number of variables in the factorial design. Number of variables multiply runs is the length of vector for the whole design matrix. While the fitness is evaluated in MPSO, every particle is a potential solution that corresponds to a particular design matrix. Initial solutions all were produced in [-1, 1]. The velocity update rules are followed as before in Equation (16-17).

Positions of all particles are updated according to Equation (15). Then, we must check if all particles are still in [-1, 1]. If the new position falls outside the feasible region, its position is reset in edge (-1, 1). At last, the coordinate values of the global best matrix in the swarm need to be reset in order to ensure every values strictly fall on -1, 0, or 1. The adjusted global best matrix is gained and its minimal value of $|\mathbf{(X'X)^{-1}}|$ calculated. The computational results are shown in Table 4. As can be seen from Table 4, MPSO cannot produce as minimum *D*-efficiency as the existing exchange algorithms, but its average performance for this case study of 12 factorial design problems is about 98%. Its accuracy in *D*-efficiency is practically acceptable for the RSM practitioners.

2-2. A Mixture Problem with One Processing Variable

Weleh [12] employed a modified Mitchell's DETMAX algorithm to optimize the *D*-efficiency of the designs. Later, Heredia-Langner [6] *et al*. proposed using GA for solving the *D*-optimal design problems. The performance of this GA procedure is as good as the modified DETMAX. It involved a randomly generated parent population of 50 designs, from which 350 offspring designs were obtained and 80 generations were took. The MPSO uses 50 objective function evaluations per iteration and converges after 8000 iterations for total 50×8000=400000

Table 4 Performance comparison in terms of $|(X'X)^{-1}|$ and relative efficiencies among different algorithms

A series of factorial regions	Method								
	Coordinate exchange	k-exchange	Modified Fedorov	GA	SAS-DETMAX	MPSO			
						$\|(X'X)^{-1}\|$	Time (sec.)	Pop-size	Iteration
V2R6	3.91E-03	3.91E-03	3.91E-03	3.906E-03	3.900E-03	3.906E-03	0.967387	10	50
D-efficiency	100	100	100	100	100	100			
V2R9	1.93E-04	1.93E-04	1.93E-04	1.929E-04	1.930E-04	1.930E-04	10.7331	50	100
D-efficiency	100	100	100	100	100	100			
V3R10	3.82E-06	3.41E-06	7.54E-07	7.535E-07	7.540E-07	8.573E-07	220	50	1300
D-efficiency	85.02739455	85.98449674	99.99867296	100	99.99336672	98.71725641			
V3R15	4.15E-09	4.15E-09	4.13E-09	4.134E-09	4.130E-09	4.147E-09	328.25	50	1300
D-efficiency	99.95893069	99.96375275	99.99515867	99.99031993	100	99.95924887			
V4R15	2.87E-12	2.87E-12	8.55E-13	8.552E-13	9.090E-13	9.095E-13	268.8	50	800
D-efficiency	92.23704802	92.23704802	100	100	99.59409447	99.59048207			
V4R22	6.67E-16	6.67E-16	5.96E-16	5.953E-16	5.950E-16	6.833E-16	136.24	50	200
D-efficiency	99.24434445	99.23938446	99.99328092	99.99663956	100	99.08205888			
V5R21	2.17E-21	2.20E-21	2.17E-21	2.169E-21	2.170E-21	2.677E-21	2991.167	50	3700
D-efficiency	100	99.9324151	100	99.99780408	99.99560922	99.00070494			
V5R32	1.35E-25	1.36E-25	8.53E-26	1.001E-25	1.060E-25	1.430E-25	4906.866	50	4000
D-efficiency	97.82211052	97.81523519	100	99.23935058	98.96908219	97.56659901			
V6R28	1.15E-31	1.23E-31	1.18E-31	1.380E-31	1.530E-31	1.911E-31	18083.8	50	14000
D-efficiency	100	99.73901044	99.89261513	99.33860448	98.97320039	98.19092622			
V6R42	1.62E-37	1.60E-37	1.44E-37	1.336E-37	1.160E-37	2.311E-37	58029	50	30000
D-efficiency	98.82074912	98.85144375	99.23074861	99.49677022	100	93.33912568			
V7R36	7.76E-44	8.60E-44	8.66E-44	4.162E-44	1.410E-43	5.266E-43	77288.8	50	40000
D-efficiency	98.28542479	98.0032183	97.98618556	100	96.6674099	93.1931666			
V7R54	2.22E-52	2.19E-52	1.19E-52	1.304E-52	1.670E-52	3.934E-52	24234.6	50	12000
D-efficiency	98.29585089	98.32435545	100	99.75317921	99.07003868	96.74001998			

Table 5 Performance comparison in $|(X'X)^{-1}|$ and relative efficiencies among different algorithms

A mixture problem with one processing variable			$\|(X'X)^{-1}\|$	*D*-efficiency
Method	Modified DETMAX		0.3774	100
	Modified Fedorov		0.5464	96.3671734
	GA		**0.3774**	**100**
	SAS-DETMAX		0.503	97.1680285
	MPSO	$\|(X'X)^{-1}\|$	0.37926	99.9508486
		Time (sec.)	1919.517	
		Pop-size	50	
		Iteration	8000	

objective function evaluations. The magnitude of the determinant $|(X'X)^{-1}|$ for the quadratic model is 0.37926.

The results of several methods are listed in Table 5. It is a model without intercept and only one square item x_4^2. Again, we must check if all particles are still in feasible region and fall in grid. The final solution is reported after 8000 iterations. Convergence of the best solution found in ten independent trials is that consistency in convergence exhibits among the ten independent trials. In this case study, MPSO returns a comparable performance on *D*-efficiency as compared to the Modified DETMAX and GA algorithms.

V. CONCLUDING REMARKS

This paper deals with the optimal design of a variety of non-standard experiments that could realistically occur in industry. By using the *D*-criterion of minimizing the generalized variance of the estimates, we proposed the algorithm of MPSO for different constrained optimal design problems. *D*-efficiency and the termination time are calculated for the evaluation purpose. In response surface modeling practice, the prediction variance is one of the most important concerns in parameter estimation. For this token, it is of particular importance to investigate the effects of these unknown parameters. The *D*-optimality criterion suffices to enquire how the unknown parameters are being estimated. The concluding remarks of this paper are addressed as follows:

1. Sensitivity analysis for the parameters K, c_1, and c_2 is conducted for MPSO. The part 1 of Section IV investigates the optimal *D*-efficiency under various combinations of (K, c_1, c_2). It indicates that the *D*-efficiency achieves the maximum level as the settings of (K, c_1, c_2) are (1.15, 1, 2.8). For a detailed account, K equals to 1.15 is chosen based upon the *D*-efficiency and its average CV. That implies that the determinant value can be largely improved as the inflation factor is chosen larger slightly than 1. Namely, it leads to particles arriving at boundary quickly. Further, the interrelationship of two stochastic factors, c_1 and c_2, is another importance concern. That stochastic factor c_2 is

larger than c_1 is proved beneficial. The comprehensive test shows that the *D*-efficiency culminates as stochastic factor c_2 is set equal to 2.8, greatly larger than stochastic factor c_1 set equal to 1. This parameter setting is considered the best one used for MPSO.

2. MPSO is compared to several exiting exchange algorithms for generating *D*-optimal design problems. In the other one of Section IV shows the experimental results of two case studies. MPSO can produce comparable performance on *D*-efficiency in a suite of factorial design problems and a special mixture problem.

REFERENCES

[1] A. I. Khuri and J. A. Cornell, *Response Surface: Design and Analyses,* 2nd ed., New York: Marcel Dekker, 1996.

[2] R. H. Myers and D. C. Montgomery, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd ed., New York: John Wiley and Sons, 2002.

[3] W. J. Welch, "Branch-and-bound search for experimental designs based on *D*-optimality and other criteria," *Technometrics*, vol. 24, no. 1, pp. 41-48, 1982.

[4] L. M. Haines, "The application of the annealing algorithm to the construction of exact optimal designs for linear-regression models," *Technometrics*, vol. 29, pp. 439-447, 1987.

[5] R. D. Cook and C. J. Nachtsheim, "A comparison of algorithms for constructing exact *D*-optimal designs," *Technometrics*, vol. 22, pp. 315-324, 1980.

[6] A. Heredia-Langner, W. M. Carlyle, D. C. Montgomery, C. M. Borror, and G. C. Runger, "Genetic algorithms for the construction of *D*-optimal designs," *Journal of Quality Technology*, vol. 35, pp. 28-46, 2003.

[7] S. L. Ho, S. Yang, G. Ni, W.C. Lo, and H. C. Wong, "A particle swarm optimization-based method for multiobjective design optimizations," *IEEE Transaction on Magnetics*, vol. 41, pp. 1756-1759, 2005.

[8] Z. Michalewicz, D. Dasgupta, R. G. L. Riche, and M. Schoenauer, "Evolutionary algorithms for constrained engineering problems," *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 851-870, 1996.

[9] G. Coath and S. K. Halgamuge, "A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems," *Proceedings of the Congress on Evolutionary Computation*, vol. 4, pp. 2419-2425, 2003.

[10] X. Hu and R. Eberhart, "Solving constrained nonlinear optimization problems with particle swarm optimization," *6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002. *(SCI 2002)*

[11] R. K. Meyer and C. J. Nachtsheim, "The coordinate-exchange algorithm for constructing exact optimal experimental designs," *Technometrics*, vol. 37, no. 1, pp. 60-69, 1995.

[12] W. J. Welch, "Computer-aided design of experiments for response estimation," *Technometrics*, vol. 26, pp. 217-224, 1984.