Biometrics 000, 000–000 DOI: 000
000 0000

Particle Swarm Optimization Techniques for Finding Optimal Mixture Designs

Weichung Wang[1], Ray-Bing Chen[2], Chien-Chih Huang[1], and Weng Kee Wong[3]

[1]Department of Mathematics, National Taiwan University, Taiwan

[2]Department of Statistics, National Cheng-Kung University, Taiwan

[3]Department of Biostatistics, Fielding School of Public Health, UCLA, U.S.A.

Summary: Particle Swarm Optimization (PSO) is a meta-heuristic algorithm that has been shown to be successful in finding the optimum solution or close to the optimum for a wide variety of real and complicated optimization problems in engineering and computer science. This paper adapts PSO methodology by first solving an optimization problem on the hypercube and then projecting the solution onto the q-simplex optimization space to find different optimal designs for mixture models commonly used in agronomy, food science and pharmaceutical science. We show that PSO is also flexible in that it can be modified straightforwardly to find optimal designs for log contrast models and constrained mixture models. We conclude with a list of advantages of this simple and novel way for finding optimal mixture designs over current methods.

Key words: A-, D- and L-optimal design; Efficiency; Equivalence theorem; Projection; Regression model.

This paper has been submitted for consideration for publication in *Biometrics*

1. Introduction

Mixture experiments are widely used in food processing, chemical, manufacturing, agricultural, cosmetics and pharmaceutical industries. This list is growing as interest in mixture models expands and optimal designs are becoming more available thorough computer codes, software packages and interactive online websites. Cornell (1990) provides an excellent introduction and broad coverage in mixture experiments. A comprehensive review of mixture models and their optimal designs can be found in Chan (2000) and a more recent review of research work in mixture experiments over the last 50 years is the work of Piepel in Chapter 12 of an edited volume by Khuri (2009).

The aim of this paper is to introduce a modified version of a popular optimization technique already widely used in engineering and computer science research for finding optimal designs for mixture models. The technique has been around for more than ten years, but interestingly hasn't had much impact in statistical methodology to date. Our experience reinforces the widely held findings that PSO is a very simple and powerful optimization tool. PSO requires no assumption of the objective function and has few and easy to work with parameters. PSO is intriguing because despite its lack of a firm theoretical basis, its repeated successes and increasing widespread use in various disciplines have even resulted in new journals that simply track its applications in various fields.

Section 2 provides statistical setup and briefly reviews general theory for optimal experimental designs before discussing mixture models and their optimal designs. In Section 3, we propose a meta-heuristic method for finding a variety of optimal designs for different types of mixture experiments. The stochastic search is based on a modified version of the particle swarm optimization first proposed by Kennedy and Eberhart (1995). In Section 4 we demonstrate this procedure is efficient for finding various optimal designs for different types of mixture models, including new models for which optimal designs are not known

analytically. We conclude in Section 5 with a summary of the advantages of the proposed PSO method over current methods for generating optimal designs.

2. Background

Our interest is in the general linear model given by

$$y = \beta' f(\mathbf{x}) + \epsilon. \tag{1}$$

Here y is the response variable, β is a t-dimensional column vector of unknown coefficients, $f(\mathbf{x})$ is a given vector of regression functions defined on a user-defined compact multi-dimensional design space. The error ϵ is random noise with zero mean and constant variance and we assume all errors are normally and independently distributed. An approximate design ξ is defined by its design points ($\mathbf{x}_i$'s) and the proportions (p_i's) of observations to be taken at these points. Once the sample size n is fixed, either by cost or time considerations, and an optimality criterion is given, the optimization problem is to determine the number (k) of points needed and the values of $\mathbf{x}_i, p_i, i = 1, \ldots, k$ subject to $p_1 + \ldots + p_k = 1$. The implemented design takes roughly np_i observations at $\mathbf{x}_i$, $i = 1 \ldots, k$ subject to $np_1 + \ldots + np_k = n$.

Following convention, the worth of a design is measured by its Fisher information matrix which is obtained by taking the negative of the expectation of the second derivative of the logarithm of the likelihood function with respect to β. Given the above statistical model, this matrix is $\mathbf{M}(\xi) = E_\xi(f(\mathbf{x})f(\mathbf{x})')$, which is inversely proportional to the variance-covariance matrix of the estimated parameters β. Following convention, the design criterion is formulated as a convex function of the information matrix. For example, D-optimality for estimating model parameters seeks to minimize the generalized variance using the convex functional $\Phi(\xi) = -ln\ |M(\xi)|$. Another popular and useful criterion is L-optimality defined by $\Phi(\xi) = tr\ L\ M(\xi)^{-1}$ and L is a user-selected matrix; if the goal is to minimize the average of the variances of the estimated parameters, we set $L = I$, the identity matrix whereupon

L-optimality reduces to A-optimality. Alternatively, if we like to estimate some average of the response over a user-selected region R, one chooses $L = \int_R f(\mathbf{x})f'(\mathbf{x})\mu(d\mathbf{x})$ and μ is a a selected measure over for R. This corresponds to estimating the response surface over R with weights specified by the measure μ with more important parts of R receiving a higher weight assignment from μ. If there is equal interest over the region R, one chooses μ to be the uniform measure on R. We obtain I-optimality when μ is the uniform measure and R is the same as the design space.

When the design criterion is a convex function of the information matrix, as it is for all the above criteria, Kiefer and Wolfowitz (1960), Kiefer (1974, 1975) showed an effective way to check whether a design is optimal for any model. In particular if Model (1) holds and each of the following inequalities holds for all $\mathbf{x}$ in the design space, then (a) a design ξ is D-optimal if and only if $f(\mathbf{x})'M^{-1}(\xi)f(\mathbf{x}) \leqslant t$; (b) a design ξ is L-optimal if and only if $f(\mathbf{x})'M^{-1}(\xi)\ L\ M^{-1}(\xi)f(\mathbf{x}) \leqslant \mathrm{tr}L\ M^{-1}(\xi)$.

The above results are frequently referred to as equivalence theorems or more informally as checking conditions and are derived from considerations of the Frechet derivatives of the convex functionals. When the regression model has one or two independent variables, the equivalence theorem can be easily applied to check the optimality of any design graphically. For instance, to check for D-optimality, we plot the function on the left hand side of the inequality (a) over the design space and determine whether the inequality is satisfied throughout the design space. If it is, the design ξ is D-optimal; otherwise it is not. In the latter case, one can also obtain a lower bound on the efficiency of the design by examining the plot (without knowing the optimum). Details are in standard design monograph such as Pazman (1986).

2.1 *Mixture Models*

Clearly optimal designs are dependent on model assumptions and for nonlinear models they depend on the nominal values of the model parameters as well, see for example, Cobby et al. (1986), where they considered using a D-optimal design to estimate parameters in an inverse quadratic with 4 parameters in agronomy. In mixture problems, it appears much of the recent design work for mixture models focuses on constructing designs robust to model mis-specification. For example, Huang et al. (2009) is one of the recent papers to study robustness properties of optimal designs for mixture models when there is uncertainty in the model assumptions. Specifically, they investigated how to find A-optimal designs for mixture experiments that are robust to the linear and quadratic models proposed by Scheffé (1958). In addition, Huang and Huang (2009a) and, Huang and Huang (2009b) found D- and A-optimal designs for linear log contrast and quadratic log contrast models for experiments with mixtures, respectively. Some new criteria were proposed in Mandal and Pal (2008) and Pal and Mandal (2008). The former advocated a trace criterion to estimate the best proportions for the ingredients or components and the latter explored a minimax criterion to estimate the response surface in a mixture experiment, including using a deficiency criterion to measure the goodness of a mixture experiment. In both papers, the model was a quadratic polynomial in several factors over the simplex region.

We assume our mixture experiments have q factors $x_1, x_2, \ldots, x_q$ defined on the regular q simplex $S^{q-1} = \{\mathbf{x}' = (x_1, x_2, \ldots, x_q) \in [0,1]^q : \sum_{i=1}^{q} x_i = 1\}$. Some common mixture models used in practice are Scheffé's polynomials of degree n. If ϵ denotes random error, the simplest is an additive polynomial mixture model when $n = 1$ and $f(\mathbf{x})' = (x_1, x_2, \ldots, x_q)$ given by

$$y = \beta'\mathbf{x} = \sum_{i=1}^{q} \beta_i x_i + \epsilon, \tag{2}$$

When $f(\mathbf{x})' = (x_1, x_2, \ldots, x_q, x_1x_2, x_1x_3, \ldots, x_{q-1}x_q)$ and $n = 2$ the second degree Scheffé's polynomial mixture model is

$$y = \sum_{i=1}^{q} \beta_i x_i + \sum_{1 \leqslant i < j \leqslant q} \beta_{ij} x_i x_j + \epsilon. \tag{3}$$

This is an example of a Scheffé quadratic canonical polynomial models widely used in blending experiments in engineering, agriculture, biological and the medical sciences. In the notation of model (1), $t = q$ for model (2) and $t = q + q(q+1)/2$ for model (3). More generally, the Scheffé polynomial of order n for a q-component mixture model is

$$y = \sum_{i=1}^{q} \theta_i x_i + \sum_{1 \leqslant i < j \leqslant q} \phi_{ij} x_i x_j + \cdots + \sum_{1 \leqslant i_1 < \cdots < i_n \leqslant q} \phi_{i_1 \cdots i_n} x_{i_1} \cdots x_{i_n} + \epsilon. \tag{4}$$

Becker (1968, 1978) proposed a flexible modeling techniques for studying mixture experiments with additive effects using Becker polynomials. They are also useful for modeling the mean response when it also depends linearly on the total amount A used in the experiment and all components in the regression function are homogenous of degree 1. Becker's models include

$$y = \sum_{i=1}^{q} \theta_i x_i + \sum_{i<j} \phi_{ij} \min(x_i, x_j) + \cdots + \phi_{1,2,\cdots,q} \min(x_1, \cdots, x_q) + \epsilon, \tag{5}$$

$$y = \sum_{i=1}^{q} \theta_i x_i + \sum_{1 \leqslant i < j \leqslant q} \frac{\phi_{ij} x_i x_j}{x_i + x_j} + \cdots + \frac{\phi_{1,2,\cdots,q} x_1 x_2 \cdots, x_q}{(x_1 + \cdots + x_q)^{q-1}} + \epsilon, \tag{6}$$

$$y = \sum_{i=1}^{q} \theta_i x_i + \sum_{1 \leqslant i < j \leqslant q} \phi_{ij} (x_i x_j)^{1/2} + \cdots + \phi_{1,2,\cdots,q} (x_1 x_2 \cdots x_q)^{1/q} + \epsilon. \tag{7}$$

In metallurgy, when there are $q = 2$ ingredients in a mixture experiment, Kasatkin (1974) found a certain type of polynomials is useful for modeling the response and he called them Kasatkin's polynomials. Such a polynomial of n^{th} order has the form:

$$y = \theta_1 x_1 + \theta_2 x_2 + \sum_{i=0}^{n-2} \phi_i x_1 x_2 (x_1 - x_2)^i + \epsilon. \tag{8}$$

Among these models, Scheffé's models are the most widely used. Further details on rationale and specific applications of Becker's and Kasatkin's models can be found in Cornell (1990), Kasatkin (1974), Sobolev and Chemleva (1976), etc. Interestingly, these papers allude to D-optimal designs for Kasatkin's polynomial models but we were unable to find them in

English publications. We used our proposed method and generate D-optimal designs for some models in Table 4.

2.2 *Optimal Designs*

We first briefly review known optimal designs for mixture models from the literature. Analytical descriptions of D-optimal designs for different orders of Scheffé polynomial models were reported in Scheffé (1958), Kiefer (1961), Uranisi (1964), Galil and Kiefer (1977), He and Guan (1990), Yu and Guan (1993) and, Liu and Neudecker (1995). Formulae for A- and integrated or I-optimal designs are available for a much smaller class of models.

Scheffé (1958) found the theoretical A- and D-optimal designs for the first order linear models with q factors over S^{q-1}. Both A- and D-optimal designs coincide and are equally supported on the q vertices of the simplex given by $(1,0,\ldots,0),\ldots,(0,\ldots,0,1)$. The A-optimal design for the quadratic mixture model with $q \geqslant 4$ was found by Yu and Guan (1993) where they showed that the A-optimal design is the weighted $\{q,2\}$ simplex-centroid design. It has a combined weight of $(4q-3)^{1/2}/(q(4q-3)^{1/2}+2q(q-1))$ equally distributed among support points of the form $(1,0,\ldots,0),\ldots,(0,\ldots,0,1)$, and a combined weight of $4r_1/(4q-3)^{1/2}$ equally distributed among points of the form $(1/2,1/2,0,\ldots,0),\ldots,(0,\ldots,0,1/2,1/2)$. When $q=3$, they numerically identified the A-optimal as the weighted $\{q,3\}$ simplex-centroid design with $(r_1,r_2,r_3)=(0.1418,0.1873,0.0128)$, where r_1,r_2 are as before and r_3 is now the weight at each of the support point of the form $(1/3,1/3,1/3,0\ldots,0),\ldots,(0,\ldots,0,1/3,1/3,1/3)$. Table 1 shows these optimal designs and the $\{q,2\}$ simplex-centroid designs which are A- or D-optimal for the quadratic models.

We next consider two third-degree polynomial models for mixture studies. The first one is the cubic model without 3-way effect polynomial mixture model, i.e.

$$E(y)=\sum_{i=1}^{q}\beta_i x_i+\sum_{1\leqslant i<j\leqslant q}\beta_{ij}x_ix_j+\sum_{1\leqslant i<j\leqslant q+1}\gamma_{ij}x_ix_j(x_i-x_j). \tag{9}$$

The D-optimal design was found by Mikaeili (1989) to be equally supported at the the

following design points: C_1^{q+1} points given by $x_i = 1, x_j = 0, \quad i \neq j, i = 1, \ldots, q$, $2C_2^{q+1}$ points given by $x_i = 1 - x_j = \frac{1}{2}(1 - \frac{1}{\sqrt{5}}), \quad i \neq j, i, j = 1, \ldots, q$, and $x_k = 0, k \neq i, j$. The second third-degree polynomial model of interest is the full cubic model with all 2-factor and 3-factor interactions:

$$E(y) = \sum_{i=1}^{q} \beta_i x_i + \sum_{1 \leqslant i < j \leqslant q} \beta_{ij} x_i x_j + \sum_{1 \leqslant i < j \leqslant q} \gamma_{ij} x_i x_j (x_i - x_j) + \sum_{1 \leqslant i < j < k \leqslant q+1} \beta_{ijk} x_i x_j x_k. \quad (10)$$

The D-optimal design was also found by Mikaeili (1993) and is equally supported at the following design points: C_1^{q+1} points given by $x_i = 1, x_j = 0, \quad i \neq j, i = 1, \ldots, q$, $2C_2^{q+1}$ points given by $x_i = 1 - x_j = \frac{1}{2}(1 - \frac{1}{\sqrt{5}}), \quad i \neq j, i, j = 1, \ldots, q$, $x_k = 0, k \neq i, j$, and C_3^{q+1} points given by $x_i = x_j = x_k = 1/3$, $x_l = 0$, $l \neq i, j, k; i, j, k = 1, 2, \ldots, q + 1$.

An analytical description of the optimal design is clearly desirable but there are also obvious limitations of such a theoretical approach. This is because (i) formulae exist only for relatively simple problems and even when they exist, they are complicated and are obtained only after long and tedious mathematical calculation that at times require exploiting the mathematical properties of the specific regression functions, (ii) frequently artificial assumptions are imposed to arrive at a closed form solution and (iii) the mathematical derivation typically works only for one specific model and quickly breaks down for a sub-model or a slight generalization of the model. An example of the latter situation is finding D-optimal designs for the above two cubic models with one or two missing interaction terms.

For these reasons, search algorithms are more practical approaches to find optimal designs than one that relies on mathematics alone. Statistical software packages like SAS, JMP and Design-Expert have some routines for finding selected types of optimal designs for some mixture models, however, the scope still seems limiting. Our experience is that most statistical software packages do not have the flexibility of generating an optimal design for a user-specified model directly, numerically or otherwise.

The next section describes a PSO-based algorithm with great potential for finding many

types of optimal designs quickly for a wide variety of mixture models, including optimal designs for sub-models in (10) for which analytical results remain elusive.

3. Particle Swarm Optimization with Projection Capabilities

Given the mixture model and the optimality criterion, PSO begins its search for the optimal mixture design with a population of randomly generated candidate designs or particles that cover the design space. Each design sequentially adapts its movement toward where it believes is the optimum and does so with an velocity that depends on its current location and locations that other particles believe is the optimum. Usually these movements are governed by two key equations with a few parameters that includes random vectors that are responsible for the stochastic movement of the designs.

For our problem, we first use a projection function to optimize over the hypercube instead of the simplex. This simplifies the optimization problem and speeds up the computation process. To fix ideas, suppose the given mixture model has q factors and we wish to find a k-point optimal design. Let $m = k \times (q+1)$ and let $\Xi = [0,1]^m$ denote the m-dimensional hypercube. Define the $m \times 1$ vector, $\tilde{\xi} = (\mathbf{x}_1^{'}, \ldots, \mathbf{x}_k^{'}, \mathbf{p}^{'})^{'} \in \Xi$, where $\mathbf{x}_i$ is a $q \times 1$ vector in $[0,1]^q$, $i = 1, \ldots, k$, and $\mathbf{p} \in [0,1]^k$. Let $\Xi^* = \Xi \setminus \{\tilde{\xi} = (\mathbf{x}_1^{'}, \ldots, \mathbf{x}_k^{'}, \mathbf{p}^{'})' \in \Xi \mid 1_k' \cdot \mathbf{p} = 0 \text{ or } 1_q' \cdot \mathbf{x}_i = 0 \text{ for some } i\}$. To transform $\tilde{\xi}$ into a proper design ξ, we define the projection function $P : \Xi^* \longrightarrow (S^{q-1})^k \times S^{k-1}$ by

$$P(\tilde{\xi}) = (\frac{\mathbf{x}_1^{'}}{(1_q' \centerdot \mathbf{x}_1)}, \ldots, \frac{\mathbf{x}_k^{'}}{(1_q' \centerdot \mathbf{x}_k)}, \frac{\mathbf{p}'}{(1_k' \centerdot \mathbf{p})})'. \tag{11}$$

The projection function P is invariant in the sense that $P \circ P(\tilde{\xi}) = P(\tilde{\xi})$ and the design ξ has support on $\mathbf{x}_i^{'}/(1_q' \centerdot \mathbf{x}_i), i = 1, \ldots, k$ and the components in $\mathbf{p}'/(1_k' \centerdot \mathbf{p})$ are the corresponding weights. The notation $\xi = P(\tilde{\xi})$ signifies that the design ξ is transformed from $\tilde{\xi}$ via the projection P.

Our particle swarm optimization is based on the projection function P in Eq. (11) as

follows. We first initialize a random population of n candidates of k-point designs from Ξ^*. We define two notion at each stage of the iteration: let (i) $\tilde{\xi}_i^{pbest}$ denote the personal best position for the i^{th} particle, i.e. the design $\tilde{\xi}_i^{pbest}$ provides the optimal value for the criterion among all the positions that the i^{th} particle has ever visited, and (ii) let $\tilde{\xi}^{gbest}$ denote the global best position, i.e. ξ^{gbest} provides the optimal value for the criterion among all the positions that all of the particles have ever visited. The strategy for the i^{th} particle, $\tilde{\xi}_i$ at the t^{th} iteration is as follows:

- Generate a new velocity v_i^t to reach to next position given by $v_i^t = w_t v_i^{t-1} + c_1\epsilon_1(\tilde{\xi}_i^{gbest} - \tilde{\xi}_i^{t-1}) + c_2\epsilon_2(\tilde{\xi}_i^{pbest} - \tilde{\xi}_i^{t-1})$, where v_i^{t-1} is the velocity generated at the $t-1$ iteration, w_t is the inertia weight, c_1, c_2 are two pre-specified positive constants, and ϵ_1, ϵ_2 are $m \times 1$ uniform random vectors.
- The next location for the i^{th} particle is

$$\tilde{\xi}_i^t = \tilde{\xi}_i^{t-1} + \chi v_i^t, \tag{12}$$

  where χ is a pre-specified positive constant. If $\tilde{\xi}_i^t$ is not in Ξ^*, we project $\tilde{\xi}_i^t$ to a location closest to the boundary of Ξ^*.
- Obtain the current design ξ^t by projecting $\tilde{\xi}_i^t$ to the simplex using i.e. $\xi_i^t = P(\tilde{\xi}_i^{\,t})$.
- Evaluate the objective function at the new design ξ_i^t.
- Update the current best for each particle ξ_i^{pbest} and $\tilde{\xi}_i^{pbest}$.
- Update the inertia weight $w_{t+1} = g(w_t)$, where g is a user-selected monotonic decreasing function.

After updating all particles, $\tilde{\xi}_i^t$, we identify $\tilde{\xi}^{gbest}$ and ξ^{gbest}, and repeat the procedure. The procedure terminates and reports ξ^{gbest} as our "best" design after a pre-specified maximal number of iterations is reached or when the criterion value does not change much according to some user-specified tolerance level.

In the above PSO steps, the constant c_1 is the cognitive learning factor and the constant c_2

is the social learning factor. Following Kennedy (1997), the convention is to set $c_1 = c_2 = 2$. The constant χ is the constriction factor and usually is set to be 1. The function g is a pre-selected non-increasing function to control the time-varying inertia weight w_t. In our work, we set the function g to be the linear decreasing function such that w_t varies from 0.9 to 0.4.

The key advantage of this modified PSO is that it operates on the simple hypercube first before it projects any non-feasible point into $(S^{q-1})^k \times S^{k-1}$. This simplifies and makes the computation more efficient. If we had directly implemented PSO to search for the optimal mixture design and worked with the constraints in the simplex, our experience is that some of the sequentially generated particles "flew" outside the simplex and the subsequent work required to ignore them or bring them back to the simplex can complicate and prolong the search for the best design considerably. This proposed PSO based on the projection is called ProjPSO.

4. Numerical Results

We now demonstrate our ProjPSO algorithm is an effective way to generate different types of optimal designs for various mean functions in a mixture experiment. For comparison sake, the examples we work with include results already reported in the literature and also new results where analytical formulae for the optimal designs are not available. In the latter case, we used an equivalence theorem to confirm its optimality and if the PSO-generated design is not optimal, we assessed its proximity to the optimal (without knowing the optimum) via an efficiency lower bound obtained from the equivalence theorem. Details in Pazman (1986).

4.1 *ProjPSO codes*

We implemented the ProjPSO algorithm written in MATLAB codes and they are available from the authors upon request. Our PC has a 2.67GHz Intel(R) Core(TM) i7 CPU. We always start with a modest size of the particles and a modest number of the iterations and

increase them when the dimensionality of the experimental region gets larger or the model has more parameters. We follow convention and use default values for the other parameters in PSO algorithm, for example, setting c_1 and c_2 both equal to 2 in ProjPSO. In almost all our examples, PSO was able to generate designs which were optimal or very close the theoretical optimal designs after a few minutes of CPU time. For example, to find the D-optimal design for the full cubic model with 3 factors, we implemented the codes using 1024 particles and 200 iterations and it took ProjPSO around 2.5 minutes to produce the analytical D-optimal design.

For each mixture problem, the main generic part of the code remains intact and only the information matrix and the criterion need to be changed in the code. For all our examples, we first searched among all designs with design points equal to the number of parameters in the model using the ProjPSO algorithm. Our guiding principle was larger number of particles or larger number of iterations for more complex models. This is because the time required to generate the optimal design is usually fast and the differences in additional computational time required when the number of particles or iterations are increased is usually not long. For instance, in the examples below, the number of particles we chose to generate the optimal designs for the linear Scheffé polynomial models were $64, 128$ and 256 for $q = 3, 4$ and 5 and the corresponding number of iterations used were $200, 400$ and 800.

[Figure 1 about here.]

We applied ProjPSO to find the optimal designs when there are q factors and $q = 3, 4$ and 5. The generated A- and D-optimal designs for Scheffé's linear mixture models are all numerically the same as the theoretical A- and D-optimal designs reported in Table 1. Figure 1 is an illustrative PSO-generated plot that shows the particles representing the support points of different designs at various stages in the search when $q = 3$. The sub-figures display the support points of the 64 particles at the $1^{st}, 5^{th}, 10^{th}, 20^{th}, 30^{th}$ and the 40^{th} iterations

and show the very rapid convergence of the ProjPSO procedure to the optimum. In this case, ProjPSO only takes less than 3 seconds for 40 iterations. When we applied the ProjPSO algorithm to find D-optimal designs for the Scheffé's quadratic mixture model (3), we also obtained the $\{q, 2\}$ simplex-centroid design in Table 1 and shown by Kiefer to be D-optimal (Kiefer, 1961).

As specific examples, we applied the ProjPSO algorithm to search for these A-optimal designs using 1024 particles 600 and 400 numbers of iterations for $q = 3$ and 4 respectively. The PSO-generated designs for these two cases were found to be identical to the theoretical optimal designs for all practical purposes.

We also applied ProjPSO to models with more complex structures. Consider the full cubic model with $q = 3$ and the cubic model does not contain 3-way effects. The corresponding best designs generated by ProjPSO are all the same these reported in Mikaeili (1989, 1993). Here the number of iterations in ProjPSO is set to be 400, but the number of particles are 128 and 512 for the cubic model without 3-way effect and full cubic model respectively. In addition to A- and D-optimal criteria, we also study I-optimal design for the Scheffé's quadratic and cubic mixture models. To generate the numerical I-optimal designs, we set 1024 particles and iterate ProjPSO 400 times. The results are shown in Table 2. The equivalence plot of the I-optimal design for the cubic Scheffé model is shown in Figure 2.

[Figure 2 about here.]

[Figure 3 about here.]

We also used ProjPSO to determine optimal designs for several submodels (or incomplete models) obtained by deleting a few interaction terms from the full cubic model. As far as we know, theoretical optimal designs for these submodels or incomplete (IC) models remain unknown. We used 1024 particles and iterated 400 times. To ensure the generated designs are D-optimal for the submodels, we used an equivalence theorem for each submodel and

checked for its optimality. Figure 3 is an example of graphical version of equivalence theorem for the submodel IC Model 1 in Table 3 and since the 3-dimensional plot is bounded above by 0, optimality of the reported corresponding design in Table 3 is confirmed.

[Figure 4 about here.]

[Table 1 about here.]

[Table 2 about here.]

[Table 3 about here.]

[Table 4 about here.]

In addition to the Scheffé type models, we also apply ProjPSO to generate D-optimal designs for the other mixture models to demonstrate the flexibility of our ProjPSO. Here three Becker's models, (5) to (7), with $q = 3$ and Kasatkin's polynomials, (8), with 3, 4, and 5 orders are considered. In our ProjPSO code, we only need to change the regressor set-up according to the target model. The numerical best designs for all 6 models are satisfied the corresponding equivalence theorem. The equivalence plot for the D-optimal design of Kasatkin's polynomial with order 5 is shown in Figure 3.

4.2 *Constrained Mixture Experiments*

4.2.1 *The linear log contrast models.* This subsection shows ProjPSO is flexible and can be directly modified to find optimal designs for related mixture models. Consider the linear log contrast model proposed by Aitchison and Bacon-Shone (1984) for studying a certain type of mixture experiments. Such models continue to be of interest and were recently studied by Huang and Huang (2009a) and, Huang and Huang (2009b) where they found exact D- and A-optimal designs for linear log contrast and quadratic log contrast models. Chan (1988)

found the continuous D-optimal design for the log contrast model,

$$E(y) = \beta_0 + \sum_{i=1}^{q-1} \beta_i \log(x_i/x_q).$$

To ensure a D-optimal design exists, additional constraints on all the factors are required. One common way to do this is to select a constant $\delta \in (0, 1)$ with the conditions $\delta \leqslant x_i/x_j \leqslant 1/\delta$, for all $1 \leqslant i, j \leqslant q$ as added constraints on the design region S^{q-1}.

As an illustration, consider the log contrast model with $q = 3$. Chan (1988) showed that for a given δ, the D-optimal design has 3 points and is supported equally at $(1/(1+2\delta), \delta/(1+2\delta), \delta/(1+2\delta))$, $(\delta/(1+2\delta), 1/(1+2\delta), \delta/(1+2\delta))$, $(\delta/(1+2\delta), \delta/(1+2\delta), 1/(1+2\delta))$, or $(1/(2+\delta), 1/(2+\delta), \delta/(2+\delta))$, $(\delta/(2+\delta), 1/(2+\delta), 1/(2+\delta))$, $(1/(2+\delta), \delta/(2+\delta), 1/(2+\delta))$.

To find the optimal design using ProjPSO, we redefined the regressors as $\log(x_i/x_q)$ and also amended the projection operator in ProjPSO so that it projects into the right space that includes the additional constraints, $\delta \leqslant x_i/x_j \leqslant 1/\delta$ for all i. We used ProjPSO to find the D-optimal designs when $\delta = 0.145$ and 0.2. Using a flock size of 1024 and 100 number of iterations, ProjPSO took approximately 11 seconds of CPU time to generate the D-optimal designs below, which also agree with the result in Chan (1988). For each of these two δ's, there are two optimal designs equally supported at 3 points. For $\delta = 0.145$, one set of support points is $\mathbf{x_1}' = (0.1124, 0.1124, 0.7752)$, $\mathbf{x_2}' = (0.7752, 0.1124, 0.1124)$ and $\mathbf{x_3}' = (0.1124, 0.1124, 0.7752)$ and the other set is $\mathbf{x_1}' = (0.4662, 0.4662, 0.0676)$, $\mathbf{x_2}' = (0.0676, 0.4662, 0.4662)$ and $\mathbf{x_3}' = (0.4662, 0.0676, 0.4662)$. For $\delta = 0.2$, one set of support points is $\mathbf{x_1}' = (0.7143, 0.1429, 0.1429)$, $\mathbf{x_2}' = (0.1429, 0.7143, 0.1429)$ and $\mathbf{x_3}' = (0.1429, 0.1429, 0.7143)$ and the other set is $\mathbf{x_1}' = (0.0909, 0.4545, 0.4545)$, $\mathbf{x_2}' = (0.4545, 0.0909, 0.4545)$ and $\mathbf{x_3}' = (0.4545, 0.4545, 0.0909)$.

It is interesting to note here that this is one situation where we did not obtain good results using the default values $c_1 = c_2 = 2$ in the PSO algorithm. This may be due to the smaller

design space resulting from the several constraints. Our experience suggests that setting $c_1 = c_2 = 0.5$ seems to work well for log contrast models.

4.2.2 *Another constrained mixture problem.* Our last example concerns mixture experiments with constraints on the components imposed physically by the problem. Because of practical or cost considerations, upper or lower bound constraints are imposed on some of the x_i's with user-specified constants L_i and U_i, such that $L_i \leqslant x_i \leqslant U_i$, $i = 1, 2, ..., q$. A commonly cited example is in the making of a 3-component fruit juice comprising watermelon, pineapple and orange. The limits on the first component are .4 and .8, the limits on the second component are .1 and .5 and the limits on the third component are .1 and .3. The design question is how to blend these components so that the drink tastes best subject to some cost constraints. In this case, the mixture has a larger requirement on the watermelon component because it is the cheapest among the three. Examples where mixture experiments have constraints on the components abound in pharmaceutical problems as well. For instance in tablet formulations, typically a D-optimal design is sought in the constrained mixture design with limits imposed on the various ingredients, see for example El-Malah et al. (2006), Jin et al. (2008), Nahata and Saini (2008) and Nahata and Saini (2009).

ProjPSO algorithm can also be modified directly to generate optimal designs for mixture experiments with constraints on its components. We discuss one such application due to space consideration. Choisnard et al. (2005) employed a mixture design to study amphiphilic cyclodextrin nano particles. There were three variables z_1, z_2 and z_3 and their constraints were $0.4 \leqslant z_1 \leqslant 0.7$; $0 \leqslant z_2 \leqslant 0.6$ and $0 \leqslant z_3 \leqslant 0.6$. Working with pseudo-variables, Crosier (1986) searched for an optimal design inside the pseudo-variable simplex region defined by x_1, x_2 and x_3 with the following constraints: $0 \leqslant x_1 \leqslant 0.5$; $0 \leqslant x_2, x_3 \leqslant 1$ and $\sum_{i=1}^{3} x_i = 1$. Because this region for the pseudo-variables is not the regular simplex, we modified our

ProjPSO algorithm to ensure that all design points are projected correctly into the proper region.

Choisnard et al. (2005) used the full quadratic model in their work without explanation but because this model contains the intercept, along with the constraint that the components sum to unity, the information matrix of any design for this model is singular. Accordingly we worked with an illustrative cubic mixture model without the 3-way effect, i.e.

$$E(y) = \sum_{i=1}^{q} \beta_i x_i + \sum_{1 \leqslant i < j \leqslant q} \beta_{ij} x_i x_j + \sum_{1 \leqslant i < j \leqslant q+1} \gamma_{ij} x_i x_j (x_i - x_j). \tag{13}$$

There are two constraints for the variables, x_i's. One is $\sum_{i=1}^{3} x_i = 1$ and another is $x_1 \leqslant 0.5$. To find the optimal design, we modified ProjPSO by including the additional constraint and found a 9-point optimal design using 1024 particles and 1000 iterations. This design $\xi^3_{pso,D}$ is equally supported at the 9 support points $\mathbf{x_i}' = (x_1, x_2, x_3)$ shown as columns in this matrix:

0.5000	0.3645	0.0000	0.2135	0.5000	0.0000	0.0000	0.2135	0.0000
0.5000	0.3178	0.0000	0.7865	0.0000	1.0000	0.2764	0.0000	0.7236
0.0000	0.3178	1.0000	0.0000	0.5000	0.0000	0.7236	0.7865	0.2764

Figure 5 is a multi-dimensional plot of the directional derivative of this generated design $\xi^3_{pso,D}$, which is $f(x)'M^{-1}(\xi^3_{pso,D})f(x) - 9$. The plot shows the derivative is always bounded above by 0 with equality at the support points and so confirms the D-optimality of this design.

[Figure 5 about here.]

5. Conclusions

There are several common methods for finding optimal designs for mixture experiments. In this section, we mention some of them and discuss comparative advantages of PSO techniques. Early search algorithms that have been proposed for finding optimal designs are quite comprehensively reviewed in Cook and Nachtsheim (1980). Roughly they include:

Dykstra's method (Dykstra, 1971), Wynn-Mitchell's method (Mitchell and Miller, 1970; Wynn, 1972), DETMAX Mitchell (1974) and modified Fedorov's method. Monographs on optimal design usually devote a chapter on algorithmic construction of optimal designs.

Statistical software packages like SAS and JMP typically have a few menus for finding optimal designs and some have specific codes for generating optimal designs for several mixture models. The types of optimal designs available in conventional packages are usually for Scheffé polynomial models and frequently limited to D- or A-optimal designs and sometimes also for I-optimal designs. Different packages employ different methods for finding optimal design; for instance, SAS uses the exchange coordinate type algorithms which requires a candidate point set be pre-specified, and JMP uses the candidate-free exchange algorithm for finding the optimal design. When a model is not available, it is not always clear how to generate the desired optimal design or if such an option is even possible. For instance, we were unable to find a statistical package capable of generating the D-optimal designs in Table 4 directly.

There is a R package called AlgDesign that generates optimal designs for several mixture models. This package uses the Federov exchange algorithm under the name optFederov, and claims to calculate the approximate designs for D-, A- and I-criteria. Optimal designs for mixture experiments are obtained using the function gen.mixture and details are freely available at http://cran.r-project.org/web/packages/AlgDesign/AlgDesign.pdf. The function "optFederov" in AlgDesign quits when no more profitable exchanges are possible.

We now compare PSO performance with AlgDesign which seems like the most appropriate program to use since it finds optimal continuous designs for different models and criteria in a mixture experimental problem. Results found from AlgDeisgn and our ProjPSO algorithm were basically the same but we observed optimal designs found from the latter are sometimes slightly better in terms of the criterion value. For example, for the full cubic model with

three factors, the optimal design ξ_{AD-D} found by AlgDesign has 33 design points whereas the one found by ProjPSO ξ_{PSO-D} has 10 points. The relative D-efficiency of the two designs is $\{det(M(\xi_{AD-D}))/det(M(\xi_{PSO-D}))\}^{1/10} = 0.9985$. As another example, for the quadratic model with four factors, AlgDesign produced ξ_{AD-A}, a 25-point A-optimal design and ProjPSO produced a design ξ_{PSO-A} with only 10 points. The relative A-efficiency of the two designs is $trace\ M(\xi_{PSO-A})^{-1}/trace\ M(\xi_{AD-A})^{-1} = 0.9668$. In either case, the PSO-generated design wins. A possible explanation for the discrepancy in the results is that AlgDesign does not use an equivalence theorem to verify optimality or incorporate it as part of the stopping criterion.

To implement optFederov, AlgDesign requires a candidate set of points to be pre-specified first. The grid set we chose was 100 levels uniformly spread out for each factor. This common requirement in AlgDesign and several other packages means that the optimal design found depends on this initial grid set of points. PSO works on the a continuous domain and differentiates itself from this and other algorithms by not requiring the user to specify a candidate set of points at the onset. We view this feature of PSO a distinct advantage over its competitors.

Other advantages of our ProjPSO algorithm over current methods are (1) our experience is that the time required to generate the optimal design is generally a lot faster than many of the current methods; a detailed comparison will be reported in another paper; (2) we have applied ProjPSO successfully to find optimal designs for models not covered in the standard software packages, suggesting that it is versatile and can generate optimal designs for a broader range of models; for instance, ProjPSO finds the I-optimal design for the cubic Scheffé model with three factors quickly and Figure 2 confirms its I-optimality. We were not able to find current packages that will produce such an optimal design; (3) in the few cases we had examined, ProjPSO always produced an optimal design with fewer points and higher

efficiencies compared with other methods; this can be an advantage if taking observations at a new "site" is expensive or laborious; (4) the ProjPSO algorithm is easy to build after downloading the basic PSO codes from various websites and then making changes to them; it is also freely available to interested reader by writing to Dr. Chen, and (5) many current methods proceed by adding a point to the current design sequentially, necessitating the user to collapse and redistribute the accumulated points and the cumulative weights to selected sites as the iteration progresses. PSO does not require such an ad-hoc procedure, which can be cumbersome.

PSO also compares favorably with other heuristic algorithms in one major aspect. For example, in genetic algorithms (GA), all tuning parameters have to be carefully selected before the algorithm works well. Our experience to date is that tuning parameters in PSO seems easy to use and are not sensitive for solving optimal design problems. Frequently, the default values for the tuning parameters work well for a broad range of design problems. For instance, we always set $c_1 = c_2 = 2$ in our work here and when we applied PSO to find minimax optimal designs for nonlinear models in Chen et al. (2012). The only exception we encountered so far is that for log contrast model where setting $c_1 = c_2 = 0.5$ seems to work better than setting them equal to 2. Our experience this far suggests that for finding an optimal continuous design, only two parameters in the PSO algorithm may require changes; the flock size and the number of iterations. A larger size of randomly generated flock of birds covers a broader range of the search space and so is suitable for more complex and high dimensional problems. A larger number of iterations minimizes the chance of early termination and allows PSO additional time to find the optimum, which it usually does not need for solving our design problems. Our typical value for a flock size is 256 and 512 if the model is more complex. A typical iteration number that we used is 300. In addition, a

further advantage of PSO over GA is that there are fewer parameters in PSO to adjust than in GA (http://www.swarmintelligence.org/tutorials.php).

The numerous optimal designs found here on a regular or irregular simplex for mixture experiments and our earlier success using PSO to find minimax optimal designs for several nonlinear models further reinforce the great potential of PSO as a novel, easy and powerful way to generate optimal experimental designs. An immediate application is to follow up work in Zhang et al. (2012) to further modify the ProjPSO to search for multiple-objective optimal designs for mixture models. On a longer term goal, our aim is to create an interactive web site for practitioners to generate tailor-made optimal designs where users can select the optimality criterion and input a fully specify the mixture model for their problem and let ProjPSO does its job.

Acknowledgements

This manuscript originated at The Sir Isaac Newton Institute at Cambridge, England when Wong was a visiting fellow and a scientific adviser for a six month workshop in the design and analysis of experiment at the Institute. He would like to thank the Institute for the support during his repeated visits there in the second half of 2011. The research of Chen and Wang were partially supported by the National Science Council of Taiwan. Wang was partially supported by the Taida Institute of Mathematical Sciences. Chen was also partially supported by the Mathematics Division of the National Center for Theoretical Sciences (South) in Taiwan.

References

Aitchison, J. and Bacon-Shone, J. (1984). Log contrast models for experiments with mixtures. *Biometrika*, 71:323–330.

Becker, N. G. (1968). Models for response of a mixture. *J. Roy. Statist. Soc. Ser. B*, 31(2):107–112.

Becker, N. G. (1978). Models and designs for experiments with mixtures. *Austral. J. Statist.*, 20(3):195–208.

Chan, L. Y. (1988). Optimal design for a linear log contrast model for experiments with mixtures. *J. Statist. Plann. Inf.*, 20:105–113.

Chan, L. Y. (2000). Optimal design for experiments with mixtures: a survey. *Commun. Statist.-Theor. Meth.*, 29:342–373.

Chen, R. B., Chang, S. P., Wang, W., and Wong, W. K. (2012). Optimal experimental designs via particle swarm optimization methods. Preprint.

Choisnard, L., A., G., M., B., Putaux, J. L., and D., W. (2005). Efficient size control of amphiphilic cyclodextrin nanoparticles through a statistical mixture design methodology. *J. Pharm Parmaceut Sci*, 8:593–600.

Cobby, J. M., Chapman, P. F., and Pike, D. J. (1986). Design of experiments for estmating inverse quadratic polynomial responses. *Biometricsi*, 42:659–664.

Cook, R. D. and Nachtsheim, C. J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22:315–323.

Cornell, J. A. (1990). *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data (2nd ed.)*. Wiley, New York.

Crosier, R. B. (1986). The geometry of constrained mixture experiments. *Technometrics*, 28(2):95–102.

Dykstra, O. (1971). The augmentation of experimental data to maximize $|X'X|^{-1}$. *Technometrics*, 13:682–688.

El-Malah, Y., Nazzal, S., and Khanfar, N. M. (2006). D-optimal mixture design: optimization of ternary matrix blends for controlled zero-order drug release from oral dosage forms.

Drug Dev Ind Pharm., 32:1027–1218.

Galil, Z. and Kiefer, J. (1977). Comparison of simplex designs for quadratic mixture models. *Technometrics*, 19(4):445–453.

He, Q. and Guan, Y. (1990). Note on simplex-centroid design of degree 3: D-optimality. *J. Northeast Uni. Technology*, 15(5):504–507.

Huang, M. K. and Huang, M. N. L. (2009a). D_s-optimal designs for quadratic log contrast model for experiments with mixtures. *Commun. Statist.-Theor. Meth.*, 38:1607–1621.

Huang, M. K. and Huang, M. N. L. (2009b). ϕ_p-optimal designs for a linear log contrast model for experiments with mixtures. *Metrika*, 70:239–256.

Huang, M. N. L., Hsu, H. L., Chou, C. J., and Klein, T. (2009). Model-robust D- and A-optimal designs for mixture experiments. *Statistica Sinica*, 19:1055–1075.

Jin, X., Zhang, Y., Xiao, L., and Zhao, Z. (2008). Optimization of extended zero-order release gliclazide tablets using D-optimal mixture design. *Yakugaku Zasshi*, 182:1475–1483.

Kasatkin, O. G. (1974). On the construction of D-optimal design on a simplex (in russian). In *Application of Mathematical Methods for Muti-component Systems Investigation*, pages 43–51. Metallurgia, Moscow.

Kennedy, J. (1997). The Particle Swarm: Social Adaptation of Knowledge. In *Evolutionary Computation, 1997., IEEE International Conference on*, pages 303–308. IEEE.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE.

Khuri, A. I. (2009). *Linear model methodology*. Chapman and Hall, London.

Kiefer, J. (1961). Optimal designs in regression problems II. *Annals of Mathematical Statistics*, 32:298–325.

Kiefer, J. (1974). General equivalence theory for optimum designs (approximate theory).

Annals of Statistics, 2(5):849–879.

Kiefer, J. (1975). Optimal design: Variation in structure and performance under change of criterion. *Biometrika*, 62(2):277–288.

Kiefer, J. and Wolfowitz, J. (1960). The equivalence of two extremum problems. *Canad. J. Math.*, 12:363–366.

Liu, S. and Neudecker, H. (1995). A V-optimal design for scheffé polynomial model. *Stat. & Prob. Letters*, 23:253–258.

Mandal, N. K. and Pal, M. (2008). Optimum mixture design using deficiency criterion. *Communications in Statistics-Theory and Methods*, 37:1565–1575.

Mikaeili, F. (1989). D-optimum design for cubic without 3-way effect on the simplex. *Journal of Statistical Planning and Inference*, 21:107–115.

Mikaeili, F. (1993). D-optimum design for full cubic on q-simplex. *Journal of Statistical Planning and Inference*, 35:121–130.

Mitchell, T. J. (1974). An algorithm for the construction of D-optimal experimental designs. *Technometrics*, 16:203–211.

Mitchell, T. J. and Miller, F. L., J. (1970). Use of design repair to construct designs for special linear models. In *Math. Div. Ann. Progr. Rept. (ORNL-4661)*, pages 130–131. Oak Ridge National Laboratory, Oak Ridge, TN.

Nahata, T. and Saini, T. R. (2008). D-optimal designing and optimization of long acting microsphere-based injectable formulation of aripiprazole. *Drug Dev Ind Pharm.*, 34:668–675.

Nahata, T. and Saini, T. R. (2009). Formulation optimization of long-acting depot injection of aripiprazole by using d-optimal mixture design. *PDA J Pharm Sci Technol.*, 63:113–122.

Pal, M. and Mandal, N. K. (2008). Minimax designs for optimum mixtures. *Statistics &*

Probability Letters, 78:608–615.

Pazman, A. (1986). *Foundations of Optimum Experimental Design*. D. Reidel Publishing Company.

Scheffé, H. (1958). Experiments with mixtures. *J. Royal Statist. Soc., Ser. B*, 20:344–360.

Sobolev, N. N. and Chemleva, T. A. (1976). Construction of simplicial-linear mdoels of composition-property diagrams. *Industrial Laboratory USSR*, 42:103–108.

Uranisi, H. (1964). Optimal design for the special cubic regression on the q-simplex. In *Mathematical Reports*, volume 1, pages 7–12. General Education Department, Kyushu University.

Wynn, H. P. (1972). Results in the theory and construction of D-optimum experimental designs. *J. Roy. Statis. Soc. Ser. B.*, 34:133–147.

Yu, G. and Guan, Y. (1993). A-optimal design of parameter estimation for mixture models of two degree. *Journal of Northeast University of Technology*, 14(3):307–311.

Zhang, C., Wong, W. K., and Peng, H. (2012). Dual-objective optimal designs for mixture experiments. *Australian and New Zealand Journal of Statistics*. In press.

July 5, 2012

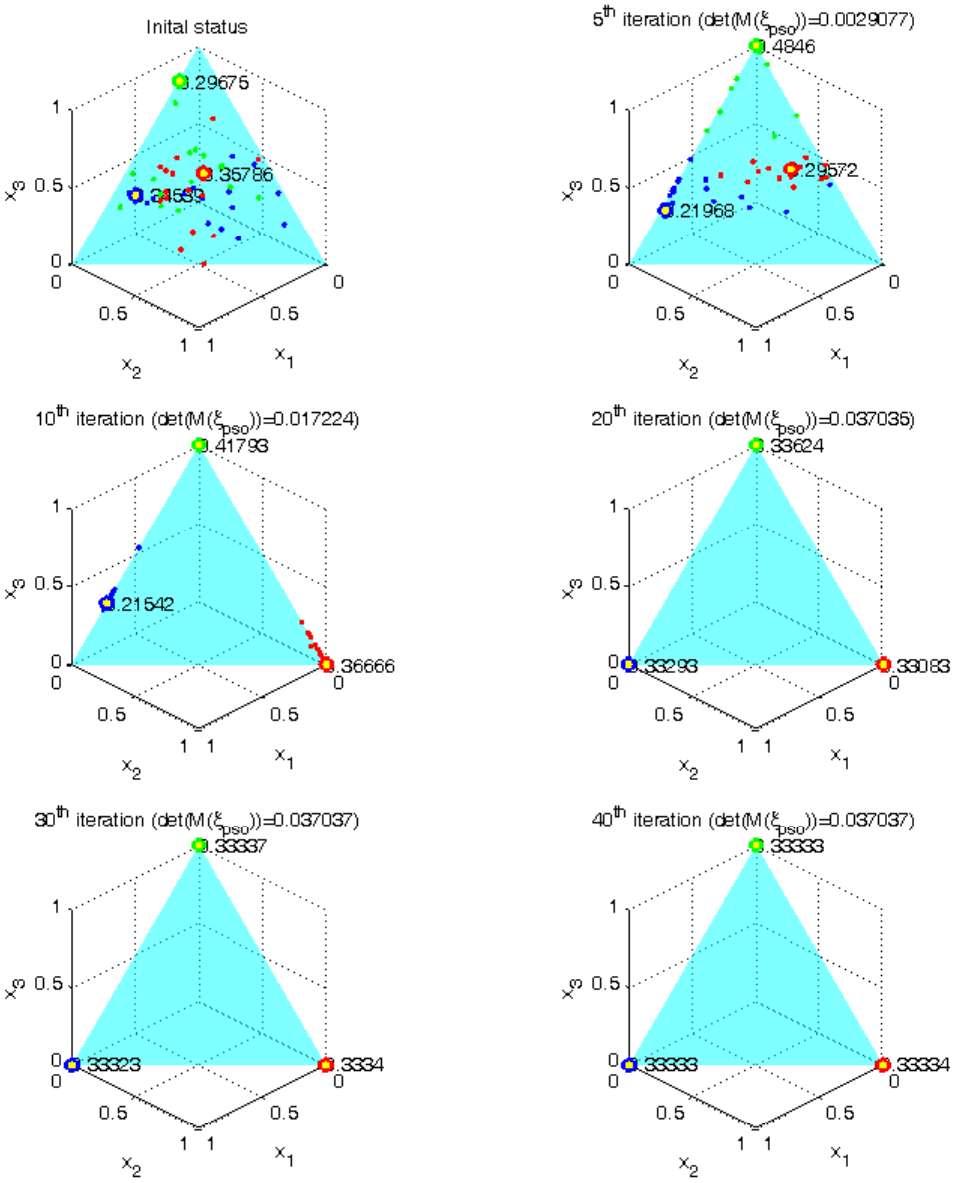

Figure 1. The movement of particles in the PSO search for the D-optimal design for the linear mixture model with $q = 3$ factors. Each subfigure displays the PSO-generated mixture design at a particular iteration. At the 40^{th} iteration, ProjPSO seems to have converged in 4 seconds of CPU time to the D-optimal design equally supported at the vertices.

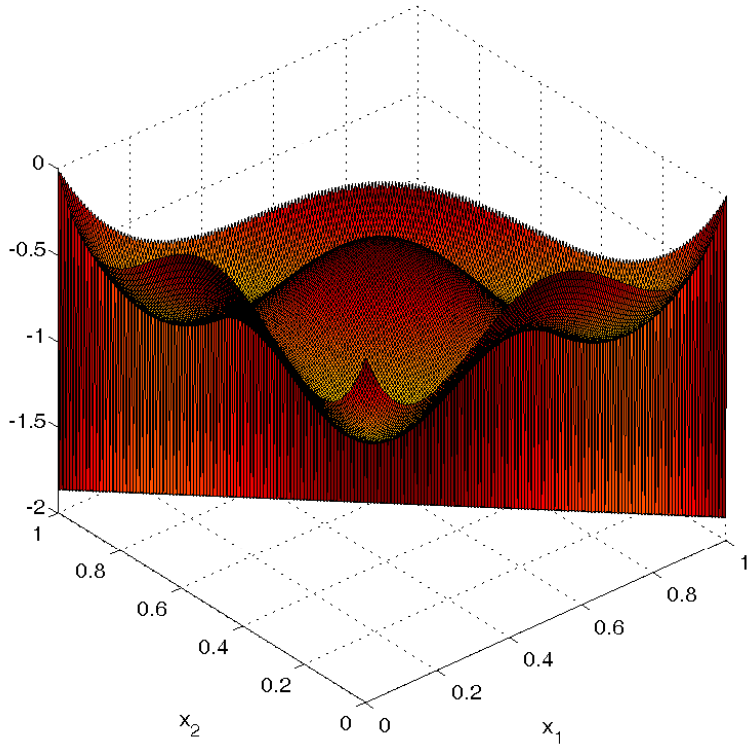

Figure 2. The plot of the directional derivative of the PSO-generated design confirms that ξ_{PSO} is I-optimal design for cubic Scheffé model.

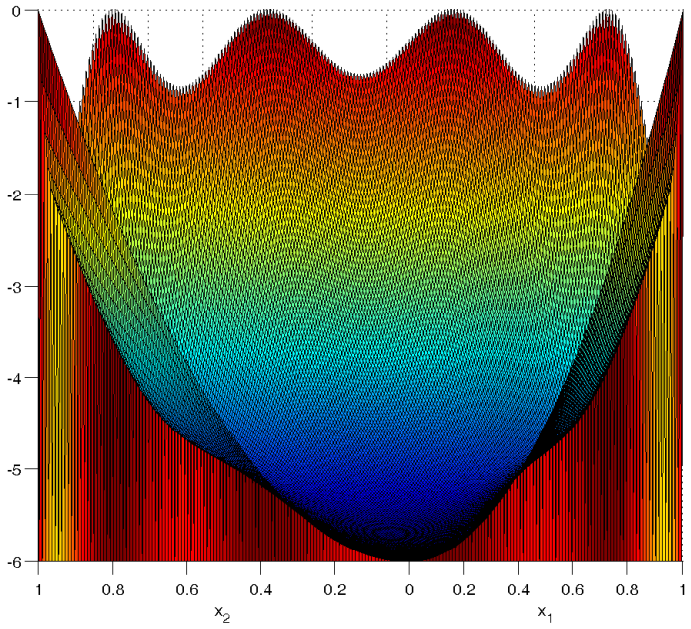

Figure 3. The plot of the directional derivative of the PSO-generated design confirms that ξ_{PSO} is D-optimal design for Katkasin polynomial model with order 5.

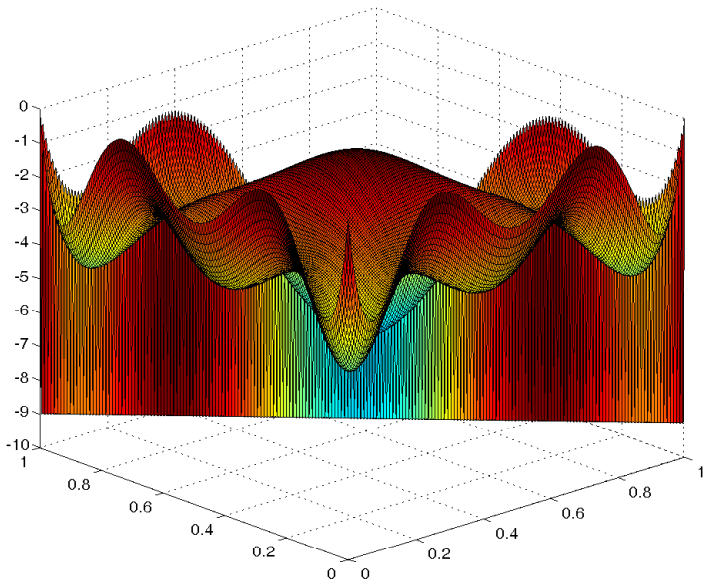

Figure 4. The plot of the directional derivative of the PSO-generated design confirms that ξ_{PSO} is D-optimal design for IC Model 1.

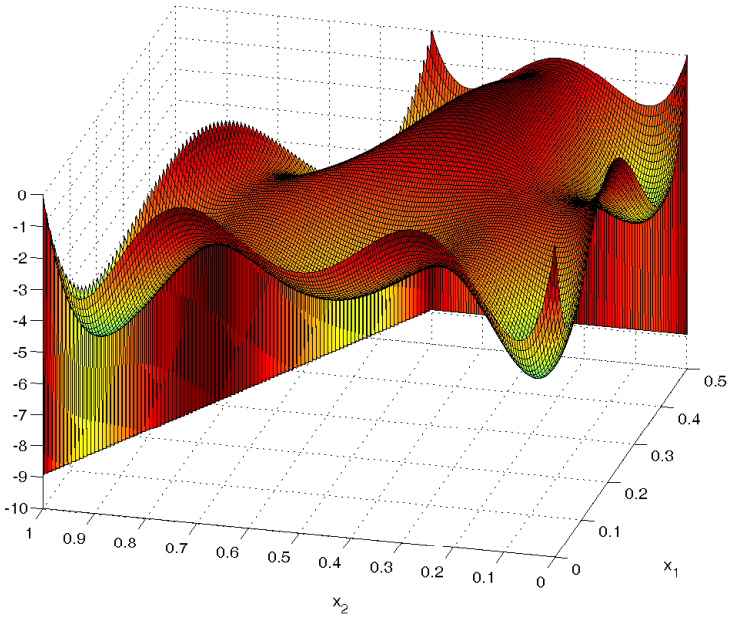

Figure 5. The plot of the directional derivative of the PSO-generated design confirms that ξ_{PSO} is D-optimal design on the restricted design space $\sum_{i=1}^{3} x_i = 1$, $0 \leqslant x_1 \leqslant 0.5$ and $0 \leqslant x_2, x_3 \leqslant 1$.

Table 1
Different types of PSO-generated designs matched theoretical results for different Scheffé polynomial models.

model	type	criterion	type of support points	weights	comment
1	Linear ($q = 3$)	D	(1, 0, 0)	1/3 ×3	Scheffé (1958)
2	Linear ($q = 4$)	D	(1, 0, 0, 0)	1/4 ×4	Scheffé (1958)
3	Linear ($q = 5$)	D	(1, 0, 0, 0, 0)	1/5 ×5	Scheffé (1958)
4	Linear ($q = 3$)	A	(1, 0, 0)	1/3 ×3	Scheffé (1958)
5	Linear ($q = 4$)	A	(1, 0, 0, 0)	1/4 ×4	Scheffé (1958)
6	Linear ($q = 5$)	A	(1, 0, 0, 0, 0)	1/5 ×5	Scheffé (1958)
7	Quadratic ($q = 3$)	D	(1, 0, 0) (1/2, 1/2, 0)	1/6 ×3 1/6 ×3	Kiefer (1961)
8	Quadratic ($q = 4$)	D	(1, 0, 0, 0) (1/2, 1/2, 0, 0)	1/10 ×4 1/10 ×6	Kiefer (1961)
9	Quadratic ($q = 3$)	A	(1, 0, 0) (1/2, 1/2, 0) (1/3, 1/3, 1/3)	0.142 ×3 0.187 ×3 0.0128 ×1	Yu and Guan (1993)
10	Quadratic ($q = 4$)	A	(1, 0, 0, 0) (1/2, 1/2, 0)	0.094 ×4 0.104 ×6	Yu and Guan (1993)

Table 2
PSO-generated D and I-optimal designs matched theoretical designs for various Scheffé polynomial models (cont'd).

model	type	criterion	type of support points	weights	comment
11	Cubic without	D	(1, 0, 0)	1/9 ×3	Mikaeili (1989)
	3-way effects		(0.2764, 0.7236, 0)	1/9 ×3	
	($q = 3$)		(0.7236, 0.2764, 0)	1/9 ×3	
12	Full Cubic ($q = 3$)	D	(1, 0, 0)	1/10 ×3	Mikaeili (1993)
			(0.2764, 0.7236, 0)	1/10 ×3	
			(0.7236, 0.2764, 0)	1/10 ×3	
			(1/3, 1/3, 1/3)	1/10 ×1	
13	Quadratic ($q = 3$)	I	(1, 0, 0)	0.1002 ×3	Liu and Neudecker (1995)
			(1/2, 1/2, 0)	0.2016 ×3	
			(1/3, 1/3, 1/3)	0.0949 ×1	
14	Cubic ($q = 3$)	I	(1, 0, 0)	0.0925 ×3	Liu and Neudecker (1995)
			(1/2, 1/2, 0)	0.1483 ×3	
			(1/3, 1/3, 1/3)	0.2776 ×1	

Table 3
PSO-generated D-optimal designs for new 3-factor submodels from Scheffé polynomial models.

model	type	support points	weights	comment
15	$\sum_i \beta_i x_i + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3$	(1, 0 , 0), (0, 1, 0)	0.0833 ×2	new
	$+ \sum_{i<j} \gamma_{ij} x_i x_j (x_i - x_j) + \beta_{123} x_1 x_2 x_3$	(0, 0, 1)	0.1111 ×1	
		(0.2764, 0.7236, 0), (0.2764, 0, 0.7236)	0.1111 ×4	
		(0.2113, 0.7887, 0), (0.7887, 0.2113, 0)	0.0833 ×2	
		(0.3333, 0.3333, 0.3333)	0.1111 ×1	
16	$\sum_i \beta_i x_i + \beta_{23} x_2 x_3$	(1, 0 , 0)	0.0938 ×3	new
	$+ \sum_{i<j} \gamma_{ij} x_i x_j (x_i - x_j) + \beta_{123} x_1 x_2 x_3$	(0, 0.2764, 0.7236), (0, 0.7236, 0.2764)	0.1250 ×2	
		(0.7887, 0.2113, 0), (0.7887, 0, 0.2113)	0.0937 ×2	
		(0.2113, 0.7887, 0), (0.2113, 0, 0.7887)	0.0937 ×2	
		(0.3333, 0.3333, 0.3333)	0.1250 ×1	
17	$\sum_i \beta_i x_i + \sum_{i<j} \beta_{ij} x_i x_j$	(1, 0 , 0)	0.1250 ×3	new
	$+\gamma_{13} x_1 x_3 (x_1 - x_3) + \gamma_{23} x_2 x_3 (x_2 - x_3)$	(0, 0.2764, 0.7236), (0.2764, 0, 0.7236)	0.1250 ×2	
		(0.7236, 0, 0.2764), (0, 0.7236, 0.2764)	0.1250 ×2	
		(0.5, 0.5, 0))	0.1250 ×1	
18	$\sum_i \beta_i x_i + \sum_{i<j} \beta_{ij} x_i x_j$	(1, 0 , 0)	0.1111 ×3	new
	$+\gamma_{13} x_1 x_3 (x_1 - x_3) + \gamma_{23} x_2 x_3 (x_2 - x_3) + \beta_{123} x_1 x_2 x_3$	(0, 0.2764, 0.7236), (0.2764, 0, 0.7236)	0.1111 ×2	
		(0.7236, 0, 0.2764), (0, 0.7236, 0.2764)	0.1111 ×2	
		(0.5, 0.5, 0))	0.1111 ×1	
		(0.3333, 0.3333, 0.3333)	0.1111 ×1	

Table 4
PSO-generated D-optimal designs for Becker and Kasatkin's mixture models with 3 factors.

model	type	support points	weights	comment
19	Becker Model 1	(1, 0 , 0)	0.1429 ×3	Becker (1968, 1978)
	$\sum_i \beta_i x_i + \sum_{i<j} \beta_{ij}(x_i x_j)^{(1/2)}$	(0.5, 0.5, 0)	0.1429 ×3	
	$+\beta_{123}(x_1 x_2 x_3)^{(1/3)}$	(0.3333, 0.3333,0.3333)	0.1429 ×1	
20	Becker Model 2	(1, 0 , 0)	0.1429 ×3	Becker (1968, 1978)
	$\sum_i \beta_i x_i + \beta_{12} x_1 x_2/(x_1 + x_2) + \beta_{13} x_1 x_3/(x_1 + x_3)$	(0.5, 0.5, 0)	0.1429 ×3	
	$+\beta_{23} x_2 x_3/(x_2 + x_3) + \beta_{123} x_1 x_2 x_3$	(0.3333, 0.3333, 0.3333)	0.1429 ×1	
21	Becker Model 3	(1, 0 , 0)	0.1429 ×3	Becker (1968, 1978)
	$\sum_i \beta_i x_i + \beta_{12} \min\{x_1, x_2\} + \beta_{13} \min\{x_1, x_3\}$	(0.5, 0.5, 0)	0.1429 ×3	
	$+\beta_{23} \min\{x_2, x_3\} + \beta_{123} \min\{x_1, x_2, x_3\}$	(0.3333, 0.3333, 0.3333)	0.1429 ×1	
22	Kasatkin 3rd Order Model	(1, 0, 0), (0, 1, 0)	1/4 ×2	Kasatkin (1974)
	$\sum_{i=1}^{2} \theta_i x_i + \sum_{i=0}^{1} \phi_i x_1 x_2 (x_1 - x_2)^i$	(0.2764, 0.7236, 0), (0.7236, 0.2764, 0)	1/4 ×2	
23	Kasatkin 4th Order Model	(1, 0, 0), (0, 1, 0)	1/5 ×2	Kasatkin (1974)
	$\sum_{i=1}^{2} \theta_i x_i + \sum_{i=0}^{2} \phi_i x_1 x_2 (x_1 - x_2)^i$	(0.1727, 0.8273, 0), (0.8273, 0.1727, 0)	1/5 ×2	
		(0.5, 0.5, 0)	1/5	
24	Kasatkin 5th Order Model	(1, 0, 0), (0, 1, 0)	1/6 ×2	Kasatkin (1974)
	$\sum_{i=1}^{2} \theta_i x_i + \sum_{i=0}^{3} \phi_i x_1 x_2 (x_1 - x_2)^i$	(0.1175, 0.8825, 0), (0.8825, 0.1175, 0)	1/6 ×2	
		(0.3574, 0.6426, 0), (0.6426, 0.3574, 0)	1/6 ×2	