5-2013

Statistical Algorithms for Optimal Experimental Design with Correlated Observations

Chang Li
Utah State University

STATISTICAL ALGORITHMS FOR OPTIMAL EXPERIMENTAL DESIGN WITH CORRELATED OBSERVATIONS

by

Chang Li

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mathematical Sciences

Approved:

Daniel C. Coster Major Professor	James Powell Committee Member
Christopher Corcoran Committee Member	Drew Dahl Committee Member
Adele Cutler Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013

Abstract

Statistical Algorithms for Optimal Experimental Design with Correlated Observations

by

Chang Li, Doctor of Philosophy

Utah State University, 2013

Major Professor: Daniel C. Coster
Department: Mathematics and Statistics

This research is in three parts with different although related objectives. The first part developed an efficient, modified simulated annealing algorithm to solve the D-optimal (determinant maximization) design problem for 2-way polynomial regression with correlated observations. Much of the previous work in D-optimal design for regression models with correlated errors focused on polynomial models with a single predictor variable, in large part because of the intractability of an analytic solution. In this research, we present an improved simulated annealing algorithm, providing practical approaches to specifications of the annealing cooling parameters, thresholds, and search neighborhoods for the perturbation scheme, which finds approximate D-optimal designs for 2-way polynomial regression for a variety of specific correlation structures with a given correlation coefficient. Results in each correlated-errors case are compared with the best design selected from the class of designs that are known to be D-optimal in the uncorrelated case: annealing results had generally higher D-efficiency than the best comparison design, especially when the correlation parameter was well away from 0.

The second research objective, using Balanced Incomplete Block Designs (BIBDs), was to construct weakly universal optimal block designs for the nearest neighbor correlation structure and multiple block sizes, for the hub correlation structure with any block size, and

for circulant correlation with odd block size. We also constructed approximately weakly universal optimal block designs for the block-structured correlation.

Lastly, we developed an improved Particle Swarm Optimization(PSO) algorithm with time varying parameters, and solved D-optimal design for linear regression with it. Then based on that improved algorithm, we combined the nonlinear regression problem and decision making, and developed a nested PSO algorithm that finds (nearly) optimal experimental designs with each of the pessimistic criterion, index of optimism criterion, and regret criterion for the Michaelis-Menten model and logistic regression model.

(79 pages)

Public Abstract

Statistical Algorithms for Optimal Experimental Design with Correlated Observations

by

Chang Li, Doctor of Philosophy

Utah State University, 2013

Major Professor: Daniel C. Coster
Department: Mathematics and Statistics

The first part of my dissertation demonstrates that a modified simulated annealing algorithm can successfully determine highly efficient D-optimal designs for second order polynomial regression for a variety of correlated error structures.

In the second part, I solved weak universal optimal block designs for the nearest neighbor correlation structure and multiple block sizes, for the hub correlation structure with any block size, and for circulant correlation with odd block size.

In the third part, we propose an improved Particle Swarm Optimization (PSO) algorithm with time varying parameters. Then combining the theorem of decision making and PSO, we innovated nested PSO algorithms with all of these three criteria and make comparison among the quality of solutions found from the three criteria.

Acknowledgments

I really appreciate my advisor, Professor Daniel C. Coster, for his guidance and help in my research. I am especially grateful for his help in the revision of my dissertation.

I would like to thank my committee members, Professors James Powell, Adele Cutler, Christopher Corcoran, and Drew Dahl, for their teaching, advice, and help in my study.

In my studying period in the department of Mathematics and Statistics, I received a lot of support, advice, and help from the faculty, staff, and schoolmates in this department. I am also indebted to the department of Mathematics and Statistics for the financial support.

Finally, I also appreciate my parents for their love, help, and encouragement.

Chang Li

Contents

List of Tables

Chapter 1

Introduction

This research is in optimal experimental design: specifically, finding solutions to the mathematically intractable and computationally intensive problem of finite sample size optimal design when observations are correlated rather than independent. Such optimal experiments are of increasing practical relevance in applied science when responses are known to be correlated and there is a demand for statistical accuracy (i.e., optimality) from experiments that are expensive and time consuming to perform. Examples would include genome mapping experiments and microarray analysis, where genetic association automatically dictates dependence among observations, and biological engineering requiring small but precise experiments.

Three separate, although related, research objectives were developed. The first required the development and implementation of an efficient simulated annealing (SA) algorithm with an original modification to solve the D-optimal (determinant maximization) design problem for multi-way polynomial regression with correlated observations, an important extension of standard (uncorrelated) response surface methodology to the correlated errors case.

The creative part of this modified simulated annealing algorithm required the division of the underlying perturbation scheme into more tractable sub-parts resulting in a more dynamic scheme and a better defined threshold for searching in the neighborhood of the target (optimal) solution. This improved algorithm overcomes the limitation of standard optimization hill-climbing algorithms by allowing the search process to extend beyond local optima. The algorithm has been implemented successfully for multiple specifications of correlation structures, including cyclic, hub, and nearest neighbor (defined elsewhere) structures.

The second objective of this research continued with the common theme of optimal design with correlated observations but focused on the design objective of weak universal optimality in block-treatment designs. In particular, and in contrast to traditional uncorrelated optimal design, the order of observations within blocks is critical for optimality. An efficient way to construct weakly universal optimal block designs with various correlation structures and block sizes is presented along with proofs that the conditionals for optimality are satisfied.

The third research objective combined decision making theory and Particle Swarm Optimization (PSO) and featured nested PSO algorithms and three criterion functions with application to the Michaelis-Menten model and the two parameter logistic regression model. Comparisons were made among the quality of solutions found from the three criteria. The three criteria reflect different levels of "optimism" and "pessimism" associated with the decision making process in the PSO algorithm and may be adjusted to achieve different solutions to the design problem. For example, when using the "index of optimism" criterion, the settings of 0.3 (the decision maker is relatively pessimistic), 0.5 (the decision maker compromises between the pessimistic and optimistic case) and 0.7 (the decision maker is relatively optimistic) were used, respectively, and solution quality compared on the design objective function.

A more complete specification of the three research objectives and accompanying literature review follows in Chapter 2. Chapter 3 presents results for the first research objective involving the SA algorithm and D-optimality, Chapter 4 contains theory and applications for the second research objective, and Chapter 5 deals with the PSO algorithm and results for two types of models. Discussion and suggestions for future research are in Chapter 6. Appendices contain annotated examples of Matlab code used to produce numerical results.

Chapter 2

Research Problems and Literature Review

2.1 D-optimality for Polynomial Regression with Correlated Observations

D-optimality is a popular criterion for optimal experimental design. Consider the model for polynomial regression as in [1]

$$y_i = f_i(x)'\beta + \epsilon_i \tag{2.1}$$

where i=1...n, β is a k-vector of parameters, and $f_i(x) = (f_{1i}(x), f_{2i}(x), \ldots f_{ki}(x))$ is a k vector of polynomial functions of x, and n is the number of observations.Our purpose is to estimate the coefficient vector β, or part of the vector β of primary interest.

In some experimental settings, the observations may be correlated according to various structures or patterns.Motivation for this research in optimal designs with correlated observations can be found in [2]. [3] introduced optimal design with correlated observations in detail.

The simulated annealing (SA) algorithm is a probabilistic "hill climbing" algorithm for optimization in the absence of an analytical solution. This algorithm derives from the principle of annealing metal: heat the metal to a high temperature first, then decrease the temperature slowly. As the temperature is decreased, the molecules in the metal tend from unordered to ordered.

The probabilistic feature of the SA algorithm mimics this behavior in metal by allowing transitions to less ideal "solutions" during the cooling stage which, in turn, provides for the opportunity to leave local optimal, something deterministic algorithms may fail to do.

[4] proposed a simulated annealing algorithm for D-optimal design with uncorrelated observations. The simulated annealing algorithm with a reheating process is introduced in

[5] and [6]. In [7], Zhu solved the 1-way D-optimal design for polynomial regression with correlated observations using a simulated annealing algorithm. [8] produced D-optimal designs with block effects, which can be considered as a special case of the D-optimal design problem with correlated observations, since the block effects can be incorporated into the correlation structure.

Most previous work only considered the simplest case, that is, optimal design for 1-way polynomial regression. However, in real world problems, the response variable is usually influenced by multiple effects and their interactions. This kind of problem is more complicated, and has not been solved by existing algorithms or their generalizations.

2.1.1 Model:

The full model for second order 2-way polynomial regression is presented in [9] and [10]. The model for the second order 2-way polynomial regression is:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i}^2 + \beta_4 x_{2i}^2 + \beta_5 x_{1i} x_{2i} + \epsilon_i \tag{2.2}$$

where i=1,2 ... n, and each of the x_{1i} and x_{2i} are in [-1,1], where ϵ_i has mean 0, variance σ^2 but are not necessarily independent.

The design matrix is: $X = (x_{ij})_{n\times 6}$, The first column is all 1's, the other 5 columns correspond to the values of $X_1, X_2, X_1^2, X_2^2, X_1X_2$, respectively . That is, each column of X corresponds to one design variable (or their square or interaction effect) in the model.

D-optimality aims to maximize of the determinant of the information matrix, where the information matrix for these models is:

$$M = X'V^{-1}X \tag{2.3}$$

where

$$V = cov(Y) = \sigma^2(\rho_{ij})_{n\times n} \tag{2.4}$$

is the variance covariance matrix of the errors. Some common correlation structures for V are introduced below.

2.1.2 Correlation structures

We define commonly used correlation structures below for a single correlation parameter ρ: 2.2(i) Circulant correlation: see [1]:

$$cov(y_i, y_j) = \begin{cases} \sigma^2 & \text{i=j} \\ \rho\sigma^2 & |i-j|\text{=1 or } |i-j|\text{=n-1} \\ 0 & \text{otherwise} \end{cases}$$

The correlation matrix is of the form:

$$R = \begin{pmatrix} 1 & \rho & 0 & 0 & ... & \rho \\ \rho & 1 & \rho & 0 & ... & 0 \\ 0 & \rho & 1 & \rho & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ \rho & 0 & 0 & 0 & ... & 1 \end{pmatrix}$$

2.2(ii) Nearest Neighbor correlation: see [8]:

$$cov(y_i, y_j) = \begin{cases} \sigma^2 & \text{i=j} \\ \rho\sigma^2 & |i-j|\text{=1} \\ 0 & \text{otherwise} \end{cases}$$

The correlation matrix is of the form:

$$R = \begin{pmatrix} 1 & \rho & 0 & 0 & ... & 0 \\ \rho & 1 & \rho & 0 & ... & 0 \\ 0 & \rho & 1 & \rho & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ 0 & 0 & 0 & 0 & ... & 1 \end{pmatrix}$$

2.2(iii) Autoregressive correlation: see [2]:

$$cov(y_i, y_j) = \sigma^2 \rho^{|i-j|}, \text{ where } i, j = 1, 2 \ldots n.$$

2.2(iv) Completely symmetric block structure: see [11]:

$$\left(\begin{array}{c|c|c|c} R & R12 & ... & R1b \\ \hline R21 & R & ... & R2b \\ \hline . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \hline 0 & 0 & ... & R \end{array}\right) \tag{2.5}$$

Here R is a $k \times k$ matrix with the elements on the main diagonal =1, and all other elements=ρ, (k is the common block size). ρ is the correlation coefficient for the observations in the same block. Rij is a $k \times k$ block with all elements =ρ_{ij}. In this paper we take all of the ρ_{ij} equal to the same coefficient ρ'.

Note that one commonly used block correlation structure is proposed by [12]:

$$cov(Y) = \sigma^2 (I_b \otimes V) \tag{2.6}$$

with $V = (1 - \rho)\mathbf{I}_k + \rho \mathbf{J}_k$. Here $\mathbf{J}_k$ is the $k \times k$ matrix with all of the elements =1. This is a special case of 2.2(iv) with Rij=0.

Hub correlation is presented in [1]. The correlation structure is a $k \times k$ matrix :

$$R = \begin{pmatrix} 1 & \rho & \rho & \rho & ... & \rho \\ \rho & 1 & 0 & 0 & ... & 0 \\ \rho & 0 & 1 & 0 & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ \rho & 0 & 0 & 0 & ... & 1 \end{pmatrix}$$

2.2 weak universal optimal block design

[13] present the definition of and a sufficient condition for weak universal optimality in balanced block design with correlated observations. The definition is: X^{**} is weakly universal optimal relative to $\mathcal{X}^*$ for covariance matrix V if it minimize $\Psi(D(X^*, V))$ over $\mathcal{X}^*$ for every convex Ψ invariant under permutation of coordinates and such that

$$\Psi(bD) > \Psi(D), \forall b > 1$$

where $\mathcal{X}^*$ is a set of eligible designs, usually it is all of the BIBDs with certain parameters.

Their sufficient condition is: X^{**} is weakly universally optimal relative to $\mathcal{X}^*$ for covariance matrix V if

(i) $D(X^{**}, V)$ is completely symmetric (CS)

$$(ii) trace(D(X^{**}, V)) = min_{X^* \in \mathcal{X}^*} trace(D(X^*, V)) \tag{2.7}$$

Note that (ii) is also known as A-optimality criterion. A matrix is completely symmetric (CS) if it is in the form $aI_k + bJ_k$, here a and b are scalars, and J_k is a $k \times k$ matrix with all elements=1. [13] also define for X^* in $\mathcal{X}^*$, $D(X^*, V) = cov(\hat{t_0}|V)$, where t_0 is the unique minimum variance linear unbiased estimator (BLUE) of t under V = V_0, which is

usually known as the least squares (LS) estimator. $cov(\hat{t_0}|V)$ is the covariance matrix of the least squares (LS) estimator t_0 under design X^* and covariance matrix V.

For a block design with correlated observations, we usually have $V = cov(Y) = \sigma^2 I_b \otimes R$. The matrix R depends on the correlations among observations in each block.

Balanced Incomplete Block Designs (BIBD) provide a foundation for this research about weak universal optimal block designs. A BIBD is a block design with v treatments, b blocks, each block having size k. Incomplete means $k < v$, and balanced means each treatment appears once in each of r blocks, and each pair of treatments appear together in the same number of blocks, this number is denoted by λ. For a BIBD, the parameters v, k, b, r and λ satisfy:

$$vr = kb \tag{2.8}$$

and

$$\lambda = \frac{r(k-1)}{v-1} \tag{2.9}$$

The construction of all kinds of BIBDs is discussed in detail in [14], and these construction is our foundation of weak universal optimal block designs. Some other foundational results about the construction of BIBD's with block size k=3 or 4 are presented by [15], [16], [8] and [17]. In several former papers and books, like [15] and [14], a BIBD is represented by triple parameters (v,k,λ), and denoted by (v,k,λ)-BIBD. In this paper we keep on this notation.

An example of BIBD: If we take k=4, v=5, then by formula (8) and(9),

we can take r=4, b=5 and $\lambda = 3$. The BIBD can be constructed in this way:

$$(1,2,3,4),(2,3,4,5),(3,4,5,1),(4,5,1,2),(5,1,2,3). \tag{2.10}$$

The application of BIBD in real world is introduced in chapter 14 of [18]. BIBD is especially useful when the block size is fixed or limited. For example, if we want to do an experiment of eye-drops to several persons, and take the eyes of each person as a block, then the block size can only be two. Or if we want to do an experiment of several detergents,

but we only have 3 operators, and the speed of washing is the same in any one session but differ from session to session, then the block size can only be 3.

Optimal block designs have been studied in many papers. [19] introduced optimal block designs with correlated observations under various circumstances. [1] introduced the weak universal optimal block design with a circulant correlation matrix in each block. The circulant correlation matrix is in the form:

$$R = \begin{pmatrix} 1 & \rho & 0 & 0 & ... & \rho \\ \rho & 1 & \rho & 0 & ... & 0 \\ 0 & \rho & 1 & \rho & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ \rho & 0 & 0 & 0 & ... & 1 \end{pmatrix} \tag{2.11}$$

The core of Zhu's research is the Theorem 3 in section 2 of that paper. In that section, he proposed the formula for the computation of the covariance matrix:

$$cov(Q) = k^2 \sum_{j=1}^{b} P_j'(I_k - \frac{1}{k}J_k)R(I_k - \frac{1}{k}J_k)P_j \tag{2.12}$$

Here Q is the matrix of the adjusted treatment total of each treatment, $P_j, j = 1, 2 \ldots b$ is a $k \times v$ matrix : $(P_j)_{li} = 1$ iff i is on the lth position of the jth block. For example: if the

elements in order in block j is (1,2, ... k). Then

$$P_j = \left(\begin{array}{cccc|ccc} 1 & 0 & ... & 0 & 0 & ... & 0 \\ 0 & 1 & ... & 0 & 0 & ... & 0 \\ . & . & . & . & 0 & ... & 0 \\ . & . & . & . & 0 & ... & 0 \\ . & . & . & . & 0 & ... & 0 \\ 0 & 0 & ... & 0 & 0 & ... & 0 \\ 0 & 0 & ... & 1 & 0 & ... & 0 \end{array}\right) \tag{2.13}$$

Another example: if the elements in order in block j is (v,v-1, ... v-k+1). Then

$$P_j = \left(\begin{array}{cccc|ccc} 0 & 0 & ... & 0 & 0 & ... & 1 \\ 0 & ... & 0 & 0 & ... & 1 & 0 \\ . & . & . & . & . & ... & . \\ . & . & . & . & . & ... & . \\ . & . & . & . & . & ... & . \\ 0 & 0 & ... & 0 & 0 & ... & 0 \\ 0 & 0 & ... & 0 & 1 & ... & 0 \end{array}\right) \tag{2.14}$$

That is, each row of P_j has one 1 and (v-1) 0's. k columns of P_j has one 1 and (k-1) 0's, and other (v-k) columns are all zeros.

Zhu uses cov(Q) to represent the covariance matrix of the LS estimator instead of $cov(\hat{t_0}|V)$, and in this paper we retain his notation. From the construction of P_j, we can see it has k elements = 1, and other elements = 0. The core of the right side of the formula is (denote it as W):

$$W = (I_k - \frac{1}{k}J_k)R(I_k - \frac{1}{k}J_k) \tag{2.15}$$

As above, J_k is a $k \times k$ matrix of all ones, and R is a $k \times k$ matrix defined by (4). Zhu also showed that

$$cov(Q_i, Q_{i'}) = k^2 \sum_{i,i' \in B_j} w_{h(i,j)h(i',j)} \tag{2.16}$$

if i, i' are in the same block j. Here w is the element in matrix W, B_j is block j, and h(i,j)=ℓ (ℓ =1,2 ... k) if i is on the ℓ^{th} position of the j^{th} block. Consequently, cov(Q) is a $v \times v$ matrix since there are v treatments.

In section 3, [7] introduced the construction of weak universal optimal block designs with a circulant correlation matrix with block size=3 based on the Steiner triple system introduced in [20].

2.3 Particle Swarm Optimization algorithm in experimental design and decision making

Particle Swarm Optimization is a heuristic search method proposed by [21]. This algorithm is a bionic algorithm which simulates the preying behavior of a bird flock. In the Particle Swarm Optimization algorithm, each solution of the optimization problem is considered to be a "bird" in the search space, and we call it a"particle". The whole population of the solution is termed as a "swarm," and all of the particles are searched by following the current best particle in the swarm. Each particle has a an associated optimization function, which determines the particle's fitness value, and a velocity, which determines the direction and distance of the search. As the PSO algorithm proceeds, for each particle, we track two "best" values: the first value is the best for the individual particle by itself so far, which is denoted by "pbest"; the second value is the best solution from the whole population so far, denoted by "gbest". When the algorithm terminates, gbest is the declared to be the solution of our problem.

Associated with each particle is a velocity, v, and position, x. The velocity and position of each particle are updated from iteration i to i+1 by:

$$v_{i+1} = \omega v_i + c_1 rand(pbest_i - x_i) + c_2 rand(gbest - x_i) \tag{2.17}$$

$$x_{i+1} = x_i + v_i \tag{2.18}$$

Here v_i is the velocity of the particle in the ith iteration, x_i is the position of the particle in the ith iteration. ω is called the inertia weight. $pbest_i$ and gbest are the local best position for particle i and global best position for all of the particles, respectively. Term "rand" is a random number in [0,1], while c_1, c_2 are "learning factors", with c_1 termed the " cognitive learning factor" , and c_2 the " social learning factor"([21]).

From the formulas, we can see the update of v is composed of three parts: the first part is the inertia velocity before the change; the second part is the cognitive learning part, which represents the learning process of the particle from its own experience; the third part is the social learning part, which represents the learning process of the particle from the experience of other particles.

2.3.1 Experimental design and the Fisher information matrix

An experimental design ξ which has n support points can be written in the form:

$$\xi = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ \xi_1 & \xi_2 & \cdots & \xi_n \end{pmatrix}$$

Here $x_i, i = 1 \ldots n$ are the values of the support points within the allowed design region, and ξ_i are the weights, which sum to 1, and represent the relative frequency of observations at the corresponding design point.

The general form of the regression model can be written as y=f(θ , ξ)+ϵ. Here f(θ , ξ) can be either linear or nonlinear function, θ is the vector of unknown parameters, and ξ is the vector of design(includes the information for both weight and the value of the support point). The range of θ is Θ, and the range of ξ is Ξ. The value of a design is computed from the Fisher information matrix, which is usually obtained as the negative of the matrix of the second derivatives (with respect to θ) of the log likelihood function.

In many case, the Fisher information matrix involves the unknown parameter θ, and

is denoted by $I(\theta, \xi)$. One popular criterion is to minimize the function $log|I^{-1}(\theta, \xi)|$. In this context, $log|I^{-1}(\theta, \xi)|$ is considered to be a loss function.

2.3.2 Models with unknown parameters

One typical example of regression with a Fisher information matrix involving unknown parameters is the Michaelis-Menten model, which is presented by [22]:

$$y = \frac{ax}{b+x} + \epsilon, x > 0 \tag{2.19}$$

For which the information matrix at a point x is defined by

$$M(x, \theta) = (\frac{ax}{b+x})^2 \begin{pmatrix} \frac{1}{a^2} & -\frac{1}{a(b+x)} \\ -\frac{1}{a(b+x)} & \frac{1}{(b+x)^2} \end{pmatrix} \tag{2.20}$$

And the information matrix for a design ξ is

$$I(\theta, \xi) = \sum_{i=1}^{k} \xi_i M(x_i, \theta) \tag{2.21}$$

Here ξ_i is the mass function at x_i .

For the Michaelis-Menten model on design space $X = [0, \tilde{x}]$, [22] showed that an optimal design is supported at 2 support points, and one of which is $\tilde{x}$. So ξ is a vector $(x_1, \xi_1)'$.

Another typical example is two parameter logistic regression model ([23]), in which the probability of response is assumed to be $p(x; \theta) = 1/(1 + exp(-b(x - a)))$. Here $\theta = (a, b)^T$ is the unknown parameter vector.

The information matrix of this model is:

$$\int \begin{pmatrix} b^2 p(x,\theta)(1-p(x,\theta)) & -b(x-a)p(x,\theta)(1-p(x,\theta)) \\ -b(x-a)p(x,\theta)(1-p(x,\theta)) & (x-a)^2 p(x,\theta)(1-p(x,\theta)) \end{pmatrix} d\xi(x) \tag{2.22}$$

2.3.3 Essential elements of decision making

A decision making problem is composed of four elements ([24]):

(i) a number of actions to be taken;

(ii)a number of states which can not be controlled by the decision maker;

(iii)objective function: payoff function or loss function which depends on both an action and a state (our objective is to maximize the payoff function or minimize the loss function);

(iv) criterion: by certain criterion, the decision maker decide which action to take.

In the function $log|I^{-1}(\theta,\xi)|$, θ is in the set of states which are out of our control and design ξ is an action to be taken.

2.3.4 Optimization criterion for decision making

Decision-making with loss functions is proposed in several papers, like [25]. Clearly, our objective is to minimize the loss functions. Based on the loss function, there are several popular criterion for decision making:

(i) Pessimistic criterion: The pessimistic decision maker always considers the worst case, that is, suppose θ will maximize the loss function. The decision maker will take the action that minimizes the loss function on the worst case. This criterion is also known as minimax criterion. The formula for this criterion is:

$$min_{\xi}(max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|) \tag{2.23}$$

(ii) Index of optimism criterion: usually the decision maker will trade off from optimism and pessimistic in decision making. This derives index of optimism criterion which take the weighted average of maximum and minimum of the loss function. The weight is called index of optimism, which is between 0 and 1. It reflects the content of optimism of the decision maker. The formula for this criterion is:

$$min_{\xi}[(1-\alpha)max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|+\alpha min_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|] \tag{2.24}$$

Here α is the index of optimism.

(iii) Minimax regret criterion: in this criterion, our objective is to minimize the maximum possible regret value. The regret value is defined by the difference between the loss under certain action and the minimum loss possible under the same state. The formula for this criterion is:

$$min_{\xi} max_{\theta \in \Theta} RV(\theta, \xi) \tag{2.25}$$

Here $RV(\theta, \xi) = log|I^{-1}(\theta, \xi)| - min_{\xi} log|I^{-1}(\theta, \xi)|$

The significance for criterion (ii) and (iii) are: usually the decision maker will trade off from optimism and pessimism in decision making. This derives index of optimism criterion which take the weighted average of maximum and minimum of the possible loss. The weight is called index of optimism, which reflects the content of optimism of the decision maker.

Some times after the decision maker made a decision, he or she may regret when certain states appear. In this case we want to minimize the maximum regret value, which is the distance between the loss value of the action he take and the minimum loss value possible in the relevant state. Regret value is also called opportunity costs, represent regret in the sense of lost opportunities.

Chapter 3

Simulated Annealing Algorithm for D-optimal Design

In my dissertation, a modified, improved simulated annealing algorithm to approximately solve for D-optimal design for 2-way polynomial regression with correlated observations is proposed. This algorithm is applicable to any number of observations, not necessarily a multiple of the dimension of the parameter vector. It conquers the shortcoming of previous work, which mainly concentrated on the case that n (the number of observations) is a multiple of k (the number of the coefficients to be estimated, or equivalently, the dimension of the parameter β). We also provide a reinforced version of our simulated annealing algorithm with a reheating process.

3.1 Improved simulated annealing algorithm for 2-way second-order polynomial regression with correlated observations

3.1.1 Research Objective:

D-optimal design for polynomial regression with uncorrelated observations is presented in [26] and [27].

In [27], the authors found the D-optimal design for 2-way (i.e., 2 predictors, X1 and X2) 2nd-degree (i.e., a model including quadratic and cross-product terms in each of X1 and X2) polynomial regression with uncorrelated observations based on 9 factorial points (the combination of -1, 0 1) in detail. The method of Box and Draper provides a convenient way to approximate the maximum value of the determinant of the information matrix $|X'X|$. One way to approximately solve the D-optimal design problem to polynomial regression with correlated observations is to use the best D-optimal design for polynomial regression with uncorrelated observations but for the specified correlation structures. We call this the

"uncorrelated method."

However, the uncorrelated method will seldom find the globally D-optimal design for any specific correlation structure because the support points are unlikely to be in -1, 0, 1 and the order of the observations themselves will impact the D-criterion.

"uncorrelated method" usually can not get ideal result. For example, for the circulant correlation structure with n=9, when $\rho = 0.4$, the "best" determinant of "uncorrelated method" is 10688, while the "best" determinant of our improved simulated annealing algorithm is 68277. What this means is that much greater D-efficiency (ratio of the maximized determinants) is available using our methods versus the "uncorrelated method" approach. We use the "uncorrelated method" as a benchmark for the potential or realized improvements obtained from our SA algorithm. In practice, if an experimenter has some idea of the magnitude of the correlation, ρ, and the structure of the dependency (circulant, hub, nearest neighbor, and so on), our best designs will be more efficient and lead to more precise estimation of model parameters and model predictions.

3.2 The Principle of Simulated Annealing

The simulated annealing (SA) algorithm belongs to a class of heuristic probabilistic hill-climbing algorithms, see [7] and [4]. The SA algorithm attempts to globally maximize an energy function E(X) for X in a specified state space (a design region for our D-optimality problem), by moving about the state space according to a transition mechanism defined by random perturbations of the current solution, X_c, to a new candidate solution, X_n. Let $dE = E(X_n) - E(X_c)$, if $dE > 0$, accept X_n as the current solution. Otherwise, accept X_n as the current solution with probability $exp(dE/T_c)$, where T_c is the current value of a temperature control parameter, T. Thus, there is positive probability that the algorithm will move to a poorer design, which is the key feature of the SA search algorithm, as it provides for the possibility that the algorithm will escape a local maximum. As the algorithm proceeds, the temperature decreases, making it less likely that designs with lower energy will be accepted. Convergence of the SA algorithm to a highly efficient design (a globally optimal solution is never guaranteed to be found), depends on the convergence to

a stationary distribution of the underlying Markov chain, which typically requires a large number of iterations as well as a suitably chosen transition scheme over the state space.

3.2.1 Simulated Annealing Algorithm for D-optimal Design for 2-Way Polynomial Regression

For 2-way polynomial regression, the $n \times 6$ design matrix is fully determined by the values of X_1 and X_2, each in [-1, 1]. Therefore, at each iteration of our simulated annealing algorithm, a new design matrix is obtained by perturbing the current values of X_1 and X_2. We denote the current values of X_1 and X_2 by X_{1c} and X_{2c} and new values by X_{1n} and X_{2n}, respectively.

In many applications of simulated annealing, the values of only one current design point are perturbed (by some random mechanism) at each iteration, and typically a systematic pass is made through all design points in this manner, and the process repeated until "convergence" is achieved according to a specified stopping condition. Alternatively, all design points are perturbed simultaneously. However, both of these traditional methods were found to be inefficient for our D-optimal design with correlated errors. Thus, we used a modification that improved convergence and solution quality. Our modification was to divide the design points into three parts, of equal or nearly equal size, and perturb all points in each part in an "inner" loop, while systematically doing this for each of the three parts. This represented a middle ground for the perturbation scheme between the two traditional perturbation methods, one at each extreme, as described above.

Our modified simulated annealing algorithm was as follows:

Step 1: Initialize starting temperature, T_0, finishing temperature T_f, temperature reduction coefficient r, perturbation neighborhood control parameter g_0, and initial design matrix X_0. Control parameter g_c is chosen from [0, 1] and is used to adjust the size of the perturbations as the algorithm proceeds. Calculate the energy function of the current design, $E(X_c) = Det(X'V^{-1}X)$.

Divide the n design points (rows of X) into three parts. If n = 3k, for some positive integer k, then each part has = n/3 design points. If n = 3k+1, the first two parts have

k design points and the third part has k+1. Similarly, if n = 3k+2, the first part has k points, and the other two have k+1 design points.

Step 2: Outer Loop:

Cycle through each of the 3 parts of X systematically, repeating the following inner loop:

Inner Loop:

(i) Let Z_1 and Z_2 be n x 1 vectors with each element of Z_i (i = 1,2) sampled at random from [-1, 1] for those design points belonging to the current part of X under consideration. All remaining elements of Z_i are set equal to 0.

(ii) Generate new candidate design points $X_{1n} = X_{1c} + gZ_1$ and $X_{2n} = X_{2c} + gZ_2$. If any element of X_{1n} or X_{2n} falls outside [-1, 1], set the value to the closest boundary value of the design region.

(iii) Determine $E(X_n)$.

(iv) If $dE = E(X_n) - E(X_c) > 0$, accept the new design by setting $X_c = X_n$. Otherwise, compare exp(dE/T) with a random number chosen uniformly from [0,1] multiply by a coefficient 1.01^c. If exp(dE/T) is greater than this number, we set $X_c = X_n$. If not, keep the X_c unchanged . Step 3: If $T_c < T_f$, stop. Otherwise, increment the counter c to c+1, set $T_c = rT_{c-1}, g_c = rg_{c-1}$, c=c+1 and return to Step 2.

Reheating: the annealing algorithm is often reinforced by using "reheating." Specifically, after the usual stopping condition based on the temperature is reached in Step 3, the process is repeated, often several times, by reheating to the original starting temperature, and continuing at Step 2. In Table 6.6, we present results of the algorithm for n = 12 and three correlation structures without and with reheating.

Reduction Control Parameter r: this tuning parameter is chosen by the user, but is often set about 0.98 - 0.99 for geometric rate of reduction in the temperature.

Perturbation Control Parameter g: Typically, g_0 is set close to 1, allowing large perturbations in design points at early iterations. As solution quality improves and the temperature decreases, g_c also decreases, localizing perturbations to a smaller neighborhood

of the current design which is more likely to be close to a global optimum when iteration counter c is large.

3.3 Improvements from this algorithm compared with a standard simulated annealing algorithm

1. There are 2 vectors, X_{1c} and X_{2c}, to be changed. In this case, the standard simulated annealing algorithm , which treats the perturbation vector Z as a whole, does not produce satisfactory results. In our modified algorithm, we divide the Z (and consequently the perturbation process) into 3 parts, and make perturbations part by part. This method ensures that we do not miss any corner of the design region, and is much more precise than the usual annealing method. Additionally, this part-by-part perturbation scheme allows the number of observations to be any number, not necessarily to be multiple of the number of coefficients. This makes our algorithm more flexible since it can be applied to experiments with any number of observations.

2. We shrink the search neighborhood and increase the threshold for accepting a perturbation each time we lower the temperature. That is,when the temperature is high, we search in a wide neighborhood and are more likely to jump out of the local optimum. At each time we lower the temperature,we make the perturbation neighborhood smaller and make the acceptance threshold higher so it becomes harder to leave a local optimum. We implement this approach by multiplying the scale number g by the reduction coefficient r and multiplying the random number to be compared with dE by a coefficient, 1.01^c at each time we decrease the temperature. Here c initially is 0, and will increase by 1 each time we decrease the temperature.

This approach is in accordance with the idea of simulated annealing, that is: when the temperature becomes lower, the "molecules" are less active and tend to an equilibrium stabilization. This modification resulted in improved relative efficiency of the final design.

3. In each part of step 3, we repeat the iterations until the improvement is less than a small threshold value multiple times. This guarantees we go to the next step only when the improvement is negligible and none in the current step. In other words, we do not

miss any valuable improvement. We take the threshold as $0.02\times$ determinant of the current information matrix as the threshold value.

3.4 Results and comparison with D-optimal design for 2-way second-order polynomial regression with uncorrelated observations

In this paper, we use the D-optimal designs in [10] to compute $|X'V^{-1}X|$, and compare them with the results from our simulated annealing algorithm.

Since the most often used correlation parameters are 0.1 and 0.4, in the tables below, we mainly use these 2 parameters in the computation and comparison. In table 8, we list the result for circulant correlation structure with various ρ and n.

In Table 3.1 through Table 3.4, we present the comparisons of the simulated annealing results and the "uncorrelated method" when observations number n is a multiple of 6 using each of the autoregressive, circulant, nearest neighbor and block correlation structures. Tables 3.1-3.3 present results of the SA algorithm for the autoregressive, circulant and nearest neighbor structure for designs of size 6, 12 and 18 and correlation parameter of 0.1 and 0.4, and Table 3.4 presents results of the SA algorithm for the block structure for designs of size 12 and correlation parameter of 0.1 and 0.4, along with comparisons with the best "uncorrelated method" design.

Table 3.1: 2-way polynomial regression with autoregressive correlation

n	ρ	Uncorrelated Determinant	Annealing Determinant
6	0.1	281.5	281.2
6	0.4	732.4	751.8
12	0.1	17368.0	17769.0
12	0.4	43921.0	45108.0
18	0.1	258700.0	272620.0
18	0.4	399710.0	889690.0

Table 3.2: 2-way polynomial regression with circulant correlation

n	ρ	Uncorrelated Determinant	Annealing Determinant
6	0.1	269.3	279.0
6	0.4	1007.8	1047.0
12	0.1	17413.0	17815.0
12	0.4	64500.0	65894.0
18	0.1	198120.0	206010
18	0.4	705880.0	1091400

Table 3.3: 2-way polynomial regression with nearest neighbor correlation

n	ρ	Uncorrelated Determinant	Annealing Determinant
6	0.1	281.7	279.1
6	0.4	732.5	742.5
12	0.1	26325.0	32901.0
12	0.4	50901.0	74276.0
18	0.1	198350.0	206010.0
18	0.4	734690.0	1175800.0

Table 3.4: 2-way polynomial regression with block correlation

n	ρ	Uncorrelated Determinant	Annealing Determinant
12	0.1	25088.0	21138.0
12	0.4	39018.0	39870.0

From these tables,we see that when ρ=0.1, the determinants obtained by simulated annealing and the uncorrelated method are similar. However, when ρ=0.4, the determinants from the simulated annealing algorithm are much higher than the results of the uncorrelated method. When n gets larger (especially when n=18), the ratio increases to well above

1, so the D-efficiency of the annealing design is relatively much better than that of the "uncorrelated method."

For the case that the observations number n is not a multiple of the dimension of the parameter vector, we take n=7 in Table 3.5. We find in all of these cases, the results of the simulated annealing algorithm are much better than the results of the uncorrelated method.

Table 3.5: 2-way polynomial regression with n=7

Correlation Structure	ρ	Uncorrelated Determinant	Annealing Determinant
Nearest Neighbor	0.1	962.3	1036.8
Nearest Neighbor	0.4	2655.1	3213.2
Circular	0.1	975.7	1027.2
Circular	0.4	3545.5	3631.6
Auto Regress	0.1	951.3	1028.1
Auto Regress	0.4	1757.5	2382.4

Table 3.6 provides the comparison of reheated simulated annealing with non-reheated simulated annealing. From this table, we can see that with the addition of the reheating process, the results are much better than the non-reheating process.

Table 3.6: 2-way polynomial regression with n=12, compare reheated simulated annealing with non-reheated simulated annealing

Correlation Structure	ρ	Non-reheated Determinant	Reheated Determinant
Nearest Neighbor	0.4	264350	317470
Circular	0.4	294140	529960
Auto Regress	0.4	55234	67548

We also present a comparison of the support points for the circulant correlation structure with n=9 in table 3.7:

Table 3.7: Comparison of support points between simulated annealing and uncorrelated method for circulant correlation

Support Point	Uncorrelated	ρ =0.1	ρ =0.2	ρ =0.4
1	-1, -1	-1, -1	-1, -1	-1, -1
2	-1, 0	-1, -0.05	-1, 0	-1, 0.42
3	-1, 1	-1, 1	-1,1	-1,1
4	0, 0	-0.14, 0.06	-0.24, 1	-0,05, -0.01
5	0, 1	0 , 1	0.07 , -0.15	0,1
6	0, -1	0.03, -1	0.23, 1	0.38, -1
7	1, 1	1 , 1	1, 0,25	1,1
8	1, -1	1, -1	1,-1	1,-1
9	1, 0	1, -0.0088	1 , 1	1, -0.08
Uncorrelated Determinant		4542.7	4600.5	10688.0

From the above table, we can see for ρ=0.1, the support points of the simulated annealing results are very close to uncorrelated method. When ρ becomes larger, the support points of simulated annealing have larger separation from the uncorrelated method support points. Table 3.8 is circulant correlation structure with various ρ and n.

Table 3.8: Circulant correlation structure with various ρ and n

ρ \ n	6	7	8	9	10	11	12
0.1	268.9	1077.5	2523.2	4417.6	6738.3	16975.0	17413.0
0.2	340.6	1261.1	3666.9	7672.5	16211.0	21788.0	47836.0
0.3	479.6	1958.1	5540.0	16406.0	31880.0	6.89220	109250.0
0.4	1038.6	4046.7	13514.0	68277.0	133670.0	268800.0	516290.0

Chapter 4

Construction of Weak Universal Optimal Block Design

Balanced block designs and in particular balanced incomplete block designs (BIBDs) have been in widespread use in agricultural, ecological, pharmaceutical, and industrial research for many years. This is - in part - a consequence of the need for efficient estimation of treatment effects in settings where blocking of experimental units is expected to be useful for improved precision but physical constraints on the available experimental units dictates block sizes less than the number of treatments (incomplete blocks). The "balance" achieved in these designs is reflected in the fact that treatment effects are still estimated with equal precision (equal variance) after adjustment for block effects. Typically, treatments are assigned "at random" to units within each block, as the order of observations does not impact the variance of treatment effects. However, this "balance" characteristic is only true for BIBDs with uncorrelated observations. With correlated observations within each block, the order of the observations matters and this order impacts variance and hence any notion of "balance". Thus, there is need of research into construction of these useful designs when any one of a number of different correlation structures might exist within each block of units.

[1] is one providing a foundation in the research about the construction of weak universal optimal block designs in the presence of correlations. However, some shortcomings of Zhu's results are:

1. Other correlation structures and block sizes might be more applicable. Zhu's research is limited to the construction of weak universal optimal block designs for a circulant correlation matrix with block size k=3. This is the simplest case because for a circulant correlation matrix with k=3, the requirement that the correlation between treatments in each block be the same is automatically satised (i.e., the correlation structure is also known

as "complete symmetric").

Additionally, since Zhu's method is based on a specific property of Steiner triple systems, this construction approach does not generalize to other correlation structures and block sizes.

2. For circulant correlation matrix, from (2.15) (see chaper 2) Zhu obtained that $W = R - \frac{1}{k}(1 + 2\rho)J_k$. However, this holds only for the circulant structure. For other kinds of correlation matrix, the matrix structure of formula (2.12) is more complex, and covariance matrix for the adjusted treatment means (cov(Q)) depends on the order (or arrangement) of the treatments in each block.

To solve these problems, in this section, we introduce an efficient way to construct weak universal optimal block designs with various correlation structures and block sizes. First, I get lemma 1:

Lemma1: For any BIBD, the condition (ii) of weak universal optimal design (A-optimality criterion) is satisfied.

Proof: Based on (16), we have $cov(Q_i, Q_i) = k^2 \sum_{i \in B_j} w_{\ell\ell}$ if i is on the ℓ^{th} position of the j^{th} block. So

$$trace(cov(Q)) = \sum_{i=1\ldots v} k^2 \sum_{i \in B_j} w_{\ell\ell} \tag{4.1}$$

Notice that the element $w_{\ell\ell}$ on the diagonal of W correspond to the position ℓ of each block. That is, once the position ℓ (no matter in which block) is occupied by an element (no matter which one), $w_{\ell\ell}$ is added in the formula (4.1) once. Since there are b blocks, each position ℓ is occupied b times. That is, each element $w_{\ell\ell}$ is added in (4.1) exactly b times, no matter how do we arrange the treatments. So finally we have $trace(cov(Q)) = bk^2 \sum_{\ell=1\ldots k} w_{\ell\ell}$.

It means under the A optimality criterion, all of the BIBDs with the same parameters are equally good. So the condition (ii) for weak universal optimal is satisfied since the trace of any design based on BIBD attains the minimum.

Since all of our constructions are based on BIBD, based on lemma 1, we only have to prove our designs satisfy condition (i) in the proofs below. The main idea is to construct

a block group based on each block of the original BIBDs. We split each W to three parts: one part is a constant times J, another part is R, yet another part is an irregular matrix. Our construction will make $N_{ii'}$ equals a constant for any i and i', and find a tricky way to make T satisfy the condition (i).

Here $N_{ii'}$ is the number of times that treatment i and i' are in the same block and are correlated. In the proof, we show $cov(Q_i, Q_{i'})$ is a constant for either $i = i'$ or $i \neq i'$.

Since the parameter λ and r of the BIBDs will be changed after our design, in this paper, we always denote the parameter before our design as λ' and r', and the parameter after our design as λ and r.

4.1 Weak universal optimal block design for nearest neighbor correlation with block size 3 to 6

Design 2.1: Weak universal optimal block design for nearest neighbor correlation with block size 3 to 6

For block size k=3, construct a (v, 3, λ')-BIBD by the method in [8]. In each block B, denote the 3 treatments in the block in order as (1,2, 3). Then we generate another 2 blocks based on the original one: B2 = (1; 3; 2);B3 = (2; 1; 3). The result is a (v, 3, $3\lambda'$)-BIBD design.

For k=4, construct a (v, 4, λ')-BIBD by the process in [15]. Then in each block B, denote the 4 treatments in the block in order as (1,2, 3,4). Then we generate another block based on the original one in this order: block $B' = (2, 4, 1, 3)$. The result is a (v, 4, $2\lambda'$)-BIBD design.

For k=5, based on a (v, 5, λ')-BIBD constructed in [14], for each block B, denote the 5 treatments in the block in order as (1,2,3,4,5). Then we generate another 4 blocks based on the original one in this order: block $B_2 = (1, 4, 2, 5, 3), B_3 = (3, 1, 5, 2, 4), B_4 = (2, 1, 4, 3, 5), B_5 = (4, 5, 1, 3, 2)$. This result is a (v, 5,$5\lambda'$)-BIBD design.

For k=6, based on a (v, 6, λ')-BIBD constructed in [14], for each block B, note the 6 treatments in the block in order as (1,2,3,4,5,6). Then we generate another 2 blocks based on the original one in this order: block $B_2 = (2, 4, 6, 1, 3, 5), B_3 = (3, 6, 2, 5, 1, 4)$. This result

is a (v, 6, $3\lambda'$)-BIBD design.

For each block size, we call the original block and blocks constructed based on it as a "block group."

Theorem 2.1: Design 2.1 is a weak universal optimal block design for all of the BIBDs with the same k and r value. Proof: Expanding formula (2.15), we obtain:

$$W = (R - \frac{1}{k}(RJ + JR) + \frac{1}{k^2}JRJ) = R - \frac{2+2\rho}{k}J + \frac{k+2(k-1)\rho}{k^2}J - \frac{1}{k}T \\ = R - \frac{k+2\rho}{k^2}J - \frac{1}{k}T \tag{4.2}$$

Here T is a $k \times k$ matrix of the form: a $(k-2) \times (k-2)$ matrix=$2\rho J$ in the middle, 0 on the four corners, and (k-2) repetitions of ρ on each of the the four sides (except the four corners). That is,

$$T = \left(\begin{array}{c|ccc|c} 0 & \rho & ... & \rho & 0 \\ \hline \rho & 2\rho & ... & 2\rho & \rho \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \rho & 2\rho & ... & 2\rho & \rho \\ \hline 0 & \rho & ... & 2\rho & 0 \end{array}\right) \tag{4.3}$$

Let the middle term in (4.2) be W', so $W = R + W' - \frac{1}{k}T$. Next we analyze each part, W' and T, of W.

From (16), we have

$$cov(Q_i, Q_{i'}) = k^2 \sum_{i,i' \in B_j} w'_{h(i,j)h(i',j)} + k^2 \sum_{i,i' \in B_j} r_{h(i,j)h(i',j)} - k \sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} \tag{4.4}$$

Here w' is an element in matrix W', t is the element in matrix T, B_j is block j, and h(i,j)=$\ell(\ell = 1, 2 \ldots k)$ if i is on the ℓ^{th} position of the j^{th} block. The first part of the value of $cov(Q_i, Q_{i'})$, which is based on W', is only a function of λ, and does not depend on the

arrangement of the treatments in blocks. Here $N_{ii'}$ is the number of times that treatment i and i' are in the same block and are correlated.

In contrast, the second part of the value of $cov(Q_i, Q_{i'})$, which is based on R and T, is related to the arrangement of the treatments in blocks. Thus the second part which is based on R and T needs specific attention.

The key step of the proof is to show that under our construction, $\sum_{i,i' \in B_j} r_{h(i,j)h(i',j)}$ and $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)}$ are constants and independent of the arrangement of treatments in each block. We will show in both cases $\sum_{i,i' \in B_j} r_{h(i,j)h(i',j)}$ and $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)}$ are constants and independent of the arrangement of treatments in each block.

Computing $cov(Q_i, Q_{i'})$:

Case 1: k is even: Suppose k=2n.

Suppose the replication of each treatment for the original BIBD is r', then the replication of each treatment for our construction is $r = kr'/2$. From the construction of the block, we can see $N_{ii'} = \lambda'$ (so $\sum_{i,i' \in B_j} r_{h(i,j)h(i',j)}$ is a constant), $\lambda = \lambda' k/2$.

(i) $i = i'$. From our construction, each treatment appears in the head or tail of T once (under this condition $t_{h(i,j)h(i',j)} = 0$), and appears in the middle of T (k/2-1) times (under this condition $t_{h(i,j)h(i',j)} = 2\rho$). So from the structure of the matrix T, for each block group, $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = 2(n-1)\rho = (k-2)\rho$. Since each treatment is included in r' block groups, totally $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = (k-2)r'\rho = \frac{k-2}{k} r\rho$.

Thus, from (2.12) and (2.16), and since each treatment appears in exactly r blocks, and cov(Q) is a $v \times v$ matrix, we have

$$cov(Q_i, Q_i) = k^2[r(1 - \frac{k+2\rho}{k^2})] - k \times \frac{k-2}{k} r\rho] = r(k^2 - k - k\rho) \tag{4.5}$$

That is, all of the elements on the main diagonal of cov(Q) are of the same value.

(ii) $i \neq i'$. If the sum of the position order number of a pair of treatments is 2k+1(like they are on position k and k+1, k-1 and k+2), we say they are symmetric to the middle. If a pair of treatments are symmetric to the middle, then they will appear in the middle of T (n-1) times (under this condition $t_{h(i,j)h(i',j)} = 2\rho$), and appear in the corner

of T once (under this condition $t_{h(i,j)h(i',j)} = 0$); if a pair of treatments are not symmetric to the middle, then they will appear on the side but not corner of T twice (under this condition $t_{h(i,j)h(i',j)} = \rho$),, and appear in the middle of T (n-2) times (under this condition $t_{h(i,j)h(i',j)} = 2\rho$),. So from the structure of the matrix T, we can see for each block group, $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = 2(n-1)\rho = (k-2)\rho$. Then

$$cov(Q_i, Q_i') = \lambda'[k^2 \times \rho - k^2 \times \frac{k}{2} \times \frac{k+2\rho}{k^2} - k \times (k-2)\rho] = \lambda'(k\rho - k^2/2) \tag{4.6}$$

is a constant (since k and ρ are constants) for $i \neq i'$. That is, all of the elements that are not on the main diagonal of cov(Q) are of the same value.

Combining (4.5) and (4.6), condition (i) for weak universal optimality is satisfied.

Case2: k is odd. Suppose the replicates of each treatment for the original BIBD is r', then the replication of each treatment for our construction is kr'. We denote it as r= kr' . From the construction of the block, we can see $N_{ii'} = 2\lambda'$ (so $\sum_{i,i' \in B_j} r_{h(i,j)h(i',j)}$ is a constant), $\lambda = k\lambda'$.

(i) $i = i'$. From our construction, each treatment appears in the head or tail of T twice (under this condition $t_{h(i,j)h(i',j)} = 0$), and appears in the middle of T (k-2) times (under this condition $t_{h(i,j)h(i',j)} = 2\rho$). So from the structure of the matrix T, we can see for each block group, $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = 2(k-2)\rho$. Since each treatment is included in r' block groups, totally $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = 2(k-2)r'\rho = \frac{2k-4}{k} r\rho$.

So from (2.12) and (2.16), and notice each treatment appears in exactly r blocks, and cov(Q) is a $v \times v$ matrix, we have

$$cov(Q_i, Q_i) = k^2[r(1 - \frac{k+2\rho}{k^2}) - \frac{2k-4}{k^2} r\rho] = r[k^2 - k - 2(k-1)\rho)] \tag{4.7}$$

That is, all of the elements on the main diagonal of cov(Q) are of the same value.

By the same argumentation as k is even, under the A optimality criterion, all of the BIBDs with the same parameter r and k are equally good. So condition (ii) for weak universal optimal is satisfied since the trace of our design attains this same minimum.

(ii) $i \neq i'$. From our construction each pair of treatments will appear in the middle of T (k-3) times (under this condition $t_{h(i,j)h(i',j)} = 2\rho$), in the corner of T once (under this condition $t_{h(i,j)h(i',j)} = 0$), and on the side but not corner of T twice (under this condition $t_{h(i,j)h(i',j)} = \rho$). So from the structure of the matrix T, we can see for each block, $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = 2(k-2)\rho$.

$$cov(Q_i, Q_i') = \lambda'[k^2 \times 2\rho - k^2 \times k \times \frac{k+2\rho}{k^2} - k \times 2(k-2)\rho] = \lambda'(-k^2 + 2k\rho) \tag{4.8}$$

is a constant(since k and ρ are constants) for $i \neq i'$. That is, all of the elements that are not on the main diagonal of cov(Q) are of the same value.

Combining (4.7) and (4.8), condition (i) for weak universal optimality is satisfied. The proof is completed.

For example, for k=4, v=5, based on the BIBD in formula (10) our design will generate one block based on each original block in this way: (2,4,1,3), (3,5,2,4), (4,1,3,5), (5,2,4,1), (1,3,5,2).

If we take $\rho = 0.4$, by formula

$$\begin{aligned} W = (R - \frac{1}{k}(RJ + JR) + \frac{1}{k^2}JRJ) = R - \frac{2+2\rho}{k}J + \frac{k+2(k-1)\rho}{k^2}J - \frac{1}{k}T \\ = R - \frac{k+2\rho}{k^2}J - \frac{1}{k}T \end{aligned} \tag{4.9}$$

We get

$$W = (R - \frac{1}{4}(RJ + JR) + \frac{1}{16}JRJ) = R - \frac{1+\rho}{2}J + \frac{4+6\rho}{16}J - \frac{1}{4}T \tag{4.10}$$

Here

$$T = \begin{pmatrix} 0 & 0.4 & 0.4 & 0 \\ 0.4 & 0.8 & 0.8 & 0.4 \\ 0.4 & 0.8 & 0.8 & 0.4 \\ 0 & 0.4 & 0.4 & 0 \end{pmatrix} \tag{4.11}$$

By the computation in MATLAB, we get

$$W = \begin{pmatrix} 0.7 & 0 & -0.4 & -0.3 \\ 0 & 0.5 & -0.1 & -0.4 \\ -0.4 & -0.1 & 0.5 & 0 \\ -0.3 & -0.4 & 0 & 0.7 \end{pmatrix} \tag{4.12}$$

$$cov(Q) = \begin{pmatrix} 83.2 & -19.2 & -19.2 & -19.2 & -19.2 \\ -19.2 & 83.2 & -19.2 & -19.2 & -19.2 \\ -19.2 & -19.2 & 83.2 & -19.2 & -19.2 \\ -19.2 & -19.2 & -19.2 & 83.2 & -19.2 \\ -19.2 & -19.2 & -19.2 & -19.2 & 83.2 \end{pmatrix} \tag{4.13}$$

4.2 Weak universal optimal block design for hub correlation for any block size

Hub correlation is presented in [1]. The correlation structure is a $k \times k$ matrix:

$$R = \begin{pmatrix} 1 & \rho & \rho & \rho & ... & \rho \\ \rho & 1 & 0 & 0 & ... & 0 \\ \rho & 0 & 1 & 0 & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ \rho & 0 & 0 & 0 & ... & 1 \end{pmatrix}$$

For a (v,k, λ')-BIBD, we can always construct a weak universal optimal block design with $\lambda = k\lambda'$. The basic idea is expanding each block to a block group with k blocks.

Design 3.1: Based on a (v,k, λ')-BIBD constructed by [14], in each block, denote the k treatments in the block in the order (1,2 ... k), then we construct k-1 blocks based on the original one, in the i^{th} (i=2 ... k) block, the element i is on the top, and other elements can be in any order. We refer to these k blocks as a "block group."

Theorem 3.1: Design 3.1 is a weak universal optimal block design for BIBD's with the same parameters.

Proof: Suppose the replication of each treatment for the original BIBD is r', then the replication of each treatment for our construction is $r = kr'$. We denote it as r. Clearly r is multiple of k. From the construction of the block group, we can see that $N_{ii'} = 2\lambda', \lambda = k\lambda'$.

Expanding formula (15), we obtain

$$\begin{aligned} W = (R - \frac{1}{k}(RJ + JR) + \frac{1}{k^2}JRJ) &= R - \frac{2+2\rho}{k}J + \frac{k+2(k-1)\rho}{k^2}J - \frac{1}{k}T \\ &= R - \frac{k+2\rho}{k^2}J - \frac{1}{k}T \end{aligned} \tag{4.14}$$

$$T = \begin{pmatrix} (2k-4)\rho & (k-2)\rho & (k-2)\rho & (k-2)\rho & ... & (k-2)\rho \\ (k-2)\rho & 0 & 0 & 0 & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ (k-2)\rho & 0 & 0 & 0 & ... & 0 \end{pmatrix}$$

Let the middle term be W', so $W = R + W' - \frac{1}{k}T$. Next we analyze each part, R, W' and T, of W. From (2.7), we have

$$cov(Q_i, Q_{i'}) = k^2 \sum_{i,i' \in B_j} w'_{h(i,j)h(i',j)} + k^2 \sum_{i,i' \in B_j} r_{h(i,j)h(i',j)} - k \sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} \tag{4.15}$$

Here w' is an element in matrix W', r is the element in matrix R, t is the element

in matrix T, B_j is block j, and h(i,j)=*ell* (l=1,2 ... k) if i is on the $\ell^t h$ position of the j^{th} block. The first part of the value of $cov(Q_i, Q_{i'})$, which is based on W', is only a function of λ, and does not depend on the arrangement of the treatments in blocks. In contrast, the second part of the value of $cov(Q_i, Q_{i'})$, which is based on R and T, is related to the arrangement of the treatments in blocks. Thus the second part which is based on R and T needs specific attention. So the key step of the proof is to show that under our construction, $\sum_{i,i' \in B_j} r_{h(i,j)h(i',j)}$ and $\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)}$ are constants and independent of the arrangement of treatments in each block.

From (2.16), let's compute $cov(Q_i, Q_{i'})$.

If $i = i'$, notice that in our design 3.1, for each treatment, if it appears on the top in one block, then it appears in other places in the other k-1 blocks of the same block group. Since each treatment is included in r' block groups, in total

$$\sum_{i,i' \in B_j} t_{h(i,j)h(i',j)} = \lambda'[(2k-4)r'\rho] = \lambda' \frac{2k-4}{k} r\rho \tag{4.16}$$

for i=1... v.

So from (2.12) and (2.16), and notice each treatment appears in exactly r blocks, and cov(Q) is a $v \times v$ matrix, we have

$$cov(Q_i, Q_i) = k^2[r(1 - \frac{k+2\rho}{k^2}) - \frac{2k-4}{k^2} r\rho] = r[k^2 - k - 2(k-1)\rho)] \tag{4.17}$$

That is, all of the elements on the main diagonal of cov(Q) are of the same value.

Next step is computing $cov(Q_i, Q_{i'})$ for the case $i \neq i'$.

Notice that in our design 3.1, for each group of k blocks, each treatment appears on the top once, so $N_{ii'} = 2$. Thus in each block group, $\sum t_{h(i,j)h(i,j)}$ is composed of k elements, $(k-2)\rho$ appearing twice and 0 appearing (k-2) times. That is, $\sum t_{h(i,j)h(i',j)} = 2(k-2)\rho$.

So we have

$$cov(Q_i, Q_{i'}) = \lambda'[k^2 \times 2\rho - k^2 \times k \times \frac{k+2\rho}{k^2} - k \times 2(k-2)\rho] = \lambda'(-k^2 + 2k\rho) \tag{4.18}$$

is a constant(since k and ρ are constants) for $i \neq i'$. That is, all of the treatments that are not on the main diagonal of cov(Q) are of the same value.

Combining (4.17) and (4.18), condition (i) for weak universal optimality is satisfied. The proof is completed.

We confirm our formulas in the proof of theorem 3.1 with block size=4. From those formula,we get $trace(cov(Q)) = (48 - 24\rho)b$ and $cov(Q_i, Q_i') = 32\rho - 32(1+\rho) + 4(4+6\rho) - 4 \times 4\rho = -16 + 8\rho$.

These results coincide with those from Matlab.

Based on the BIBD in formula (10) with k=4 and v=5, our design will generate 3 blocks based on each original block in this way: (2,4,1,3), (3,1,2,4), (4,1,2,3); (3,5,2,4), (5,3,2,4), (4,5,3,2); (4,1,3,5), (1,5,3,4), (5,4,3,1); (5,2,4,1), (1,2,4,5), (2,1,4,5); (1,3,5,2). (3,2,1,5), (2,3,1,5).

By the computation in MATLAB, we get

$$W = \begin{pmatrix} 0.3 & -.0.1 & -0.1 & -0.1 \\ -0.1 & 0.7 & -0.3 & -0.3 \\ -0.1 & -0.3 & 0.7 & -0.3 \\ -0.1 & -0.3 & -0.3 & 0.7 \end{pmatrix} \tag{4.19}$$

$$cov(Q) = \begin{pmatrix} 38.4 & -38.4 & -38.4 & -38.4 & -38.4 \\ -38.4 & 38.4 & -38.4 & -38.4 & -38.4 \\ -38.4 & -38.4 & 38.4 & -38.4 & -38.4 \\ -38.4 & -38.4 & -38.4 & 38.4 & -38.4 \\ -38.4 & -38.4 & -38.4 & -38.4 & 38.4 \end{pmatrix} \tag{4.20}$$

4.3 Weak universal optimal block design for circulant correlation with odd block size

Circulant correlation is introduced in [1]. The correlation matrix is of the form:

$$R = \begin{pmatrix} 1 & \rho & 0 & 0 & ... & \rho \\ \rho & 1 & \rho & 0 & ... & 0 \\ 0 & \rho & 1 & \rho & ... & 0 \\ . & . & . & . & & \\ . & . & . & . & & \\ . & . & . & . & & \\ \rho & 0 & 0 & 0 & ... & 1 \end{pmatrix}$$

Design 4.1 Weak universal optimal block design for circulant correlation with odd block size:

Based on a (v,k, λ')-BIBD constructed in [14] with odd block size k (suppose k=2n-1) in each block, denote the k treatments in the block in the order (1,2 ...2n-1), then we construct $\frac{k-1}{2} - 1$ additional blocks, each block constructed based on the previous block. In the ith block, take the (n-1) treatments in the even positions (positions 2,4,6 ...2n-2) of the (i-1)th block and put them in order to the first n-1 positions, and put the remaining n treatments in the same order to the remaining positions.

For example, for k=5, based on a (v,5,λ')-BIBD, for each block B, denote the 5 treatments in the block in order as (1,2,3,4,5). Then we generate another one block based on the original one in this order: block $B_2 = (2, 4, 1, 3, 5)$, This result is a (v, 5,2λ')-BIBD design. For k=7, based on a (v,5,λ')-BIBD, for each block B, denote the 7 treatments in the block in order as (1,2,3,4,5,6,7). Then we generate another 2 blocks based on the original one in this order: block $B_2 = (2, 4, 6, 1, 3, 5, 7)$, , $B_3 = (4, 1, 5, 2, 6, 3, 7)$,This result is a (v, 7,3λ')-BIBD design.

Theorem 4.1: Design 4.1 is a weak universal optimal block designfor all of the BIBDs with the same r value.

Proof: Suppose the replication of each treatments for the original BIBD is r', then the replication of each treatments for our construction is $\frac{(k-1)r'}{2}$. We denote it as r. From the construction of the block, we can see $\lambda = \frac{k-1}{2}\lambda'$, .

Expanding formula (2.15), we obtain:

$$W = (R - \frac{1}{k}(RJ + JR) + \frac{1}{k^2}JRJ) = R - \frac{2+4\rho}{k}J + \frac{k+2k\rho}{k^2}J = R - \frac{1+2\rho}{k}J \quad (4.21)$$

Let the last terms be W', so $W = R - W'$ From (16), we have

$$cov(Q_i, Q_{i'}) = k^2 \sum_{i,i' \in B_j} r_{h(i,j)h(i',j)} - k^2 \sum_{i,i' \in B_j} w'_{h(i,j)h(i',j)} \quad (4.22)$$

Here w' is an treatment in matrix W', r is the treatment in matrix R, B_j is block j, and h(i,j)=l(l=1,2 ... k) if i is on the lth position of the jth block.

The second part of the value of $cov(Q_i, Q_{i'})$, which is based on W', is only a function of λ, and does not depend on the arrangement of the treatments in blocks. In contrast, the first part of the value of $cov(Q_i, Q_{i'})$, which is based on R is related to the arrangement of the treatments in blocks.

Our purpose is to show that under our construction, $\sum_{i,i' \in B_j} r_{h(i,j)h(i',j)}$ are constants, that is, $N_{ii'}$ is a constant for any i and i'. Here $N_{ii'}$ is the number of times that treatment i and i' are in the same block and are correlated.

We will begin our proof with treatment n. Since all of the treatments are cyclic symmetric, the analysis of n can be applied to any other treatment.

Let's consider the circulant distance of n and other treatments. In the original block, the distance between treatment n and n-1 and the distance between treatment n and n+1 is 1, the distance between treatment n and n-2 and the distance between treatment n and n+2 is 2, ... the distance between treatment n and 1 and the distance between treatment n and 2n-1 is n-1.

After the construction, the distance between treatment n and n-1 and the distance between treatment n and n+1 is (n-1) (that is, the farthest distance), the distance between treatment n and n-2 and the distance between treatment n and n+2 is 1, ... the distance between treatment n and 1 and the distance between treatment n and 2n-1 is n-2. That is, except n+1 and n-1, the distance between n and any other treatments gets closer by 1 unit. Since we repeat this process $\frac{k-1}{2} - 1 = n - 2$ times, this construction can guarantee n and other treatments will be neighbor circularly exactly once. So $N_{ii'} = 1$ under our construction.

Computing $cov(Q_i, Q_i)$: (i) $i = i'$.

From (2.12) and (2.16), and since each treatment appears in exactly r blocks, and cov(Q) is a $v \times v$ matrix, we have

$$cov(Q_i, Q_i) = k^2 r - k^2 \times r \times \frac{1 + 2\rho}{k} = r(k^2 - k - 2k\rho) \tag{4.23}$$

That is, all of the values on the main diagonal of cov(Q) are of the same value.

(ii) $i \neq i'$.

$$cov(Q_i, Q_i') = \lambda'(k^2 \times \rho - k^2 \times \frac{k-1}{2} \times \frac{1+2\rho}{k}) = \lambda'(k\rho + \frac{k-k^2}{2}) \tag{4.24}$$

is a constant(since k and ρ are constants) for $i \neq i'$. That is, all of the treatments that are not on the main diagonal of cov(Q) are of the same value.

Combining (4.23) and (4.24), condition (i) for weak universal optimal is satisfied.

For example, if we take k=5, v=6, then we can take r=5, b=6 and $\lambda = 4$. The BIBD can be constructed in this way: (1,2,3,4,5), (2,3,4,5,6), (3,4,5,6,1), (4,5,6,1,2), (5,6,1,2,3), (6,1,2,3,4).

Then for circulant correlation, our design will generate one block based on each original block in this way: (2,4,1,3,5), (3,5,2,4,6), (4,6,3,5,1), (5,1,4,6,2), (6,2,5,1,3), (1,3,6,2,4).

By the computation in MATLAB, we get

$$W = \begin{pmatrix} 0.6625 & 0.0625 & -0.3375 & -0.3375 & 0.0625 \\ 0.0625 & 0.6625 & 0.0625 & -0.3375 & -0.3375 \\ -0.3375 & 0.0625 & 0.6625 & 0.0625 & -0.3375 \\ -0.3375 & -0.3375 & 0.0625 & 0.6625 & 0.0625 \\ 0.0625 & -0.3375 & -0.3375 & 0.0625 & 0.6625 \end{pmatrix} \tag{4.25}$$

$$cov(Q) = \begin{pmatrix} 80 & -32 & -32 & -32 & -32 & -32 \\ -32 & 80 & -32 & -32 & -32 & -32 \\ -32 & -32 & 80 & -32 & -32 & -32 \\ -32 & -32 & -32 & 80 & -32 & -32 \\ -32 & -32 & -32 & -32 & 80 & -32 \\ -32 & -32 & -32 & -32 & -32 & 80 \end{pmatrix} \tag{4.26}$$

4.4 Weak universal optimal block design for block-structured correlation

Theorem 5.1 For any kind of block-structured correlation, any BIBD is a weak universal optimal block design.

Proof: For the blocks not on the diagonal, we have

$$W = (I_k - \frac{1}{k}J_k)\rho_{ij}J_k(I_k - \frac{1}{k}J_k) = 0 \tag{4.27}$$

For the blocks on the diagonal, we have

$$W = (I_k - \frac{1}{k}J_k)R(I_k - \frac{1}{k}J_k) = R - \frac{1}{k}(1 + (k-1)\rho)J_k \tag{4.28}$$

So from (2.12), we have $cov(Q_i, Q_i) = r(1 - \frac{1+(k-1)\rho}{k})$.

For $cov(Q_i, Q_{i'})$ for the case $i \neq i'$, we have

$$cov(Q_i, Q_i') = \lambda k^2 \rho - \lambda k[1 + (k-1)\rho] \tag{4.29}$$

is a constant for $i \neq i'$.

So the condition (i) for weak universal optimal is satisfied.

In fact, if we change the matrix R to the other correlation structures considered, we can see our designs in section 2 also work for this block-structured correlation, since the matrix that is not on the main diagonal contributes nothing to cov(Q).

Chapter 5

Combinatorial Particle Swarm Optimization for Experimental Design

5.1 Motivation

An immediate limitation of the methods of Chapters 3 and 4 is that the models are linear - that is, linear in the parameters. When models are nonlinear in the parameters, and the design objective function remains maximization of the (Fisher) information matrix, the problem becomes theoretically intractable because the information matrix is - itself - a function of the unknown parameters. Traditionally, non-linear optimal design methods have been sequential ([10]) in nature, using current results from partial experiments to improve the selection of the next design point. This has practical limitations. A second limitation of the methods of Chapters 3 and 4 is that the designs are exact -they have integer weights on selected design points/treatments. If the design size changes, both the support points and the weights of those points would change.

Given these two limitations, our third research objective was to develop an algorithm that could overcome both the linear model and the integer weight limitations and produce reasonable if not optimal designs, still with the objective of maximizing the (Fisher) information of some function of this information matrix. To this end, we implemented modified and nested Particle Swarm Optimization (PSO) algorithms with multiple decision making criteria to determine the gain in efficiency that might be achieved. Examples of potential improvements are presented using two types of non-linear models: the Michaelis-Menton model andte 2-parameter logistic regression model.

Application of the basic Particle Swarm Optimization algorithm to maximization and minimization problems and a nested PSO algorithm for the pessimistic criterion are pre-

sented by [28]. Chen's paper is a milestone in the research of applying particle swarm optimization to experimental design. The combination of decision making and particle swarm optimization in engineering has been studied in several previous papers, see for example [29] and [30].

However, the combination of PSO algorithm and other decision making criteria, including the index of optimism criterion and minimax regret criterion, are seldom considered in previous research.

In this research section, we propose an improved algorithm by combining the PSO algorithm and various decision making criteria, and use time varying parameters proposed by [31], [32], and [33] to find efficient designs for non-linear models and to compare the results of the PSO methodology to that of the simulated annealing algorithm to determine which method might provide the better results even when the models are linear.

5.2 Main improvement of our algorithm

(i) [32] introduced the time varying formula for c_1 and c_2:

$$c_1 = (c_{upper} - c_{low}) \times \frac{maxiter - iter}{maxiter} + c_{low} \tag{5.1}$$

$$c_2 = (c_{upper} - c_{low}) \times \frac{iter}{maxiter} + c_{low} \tag{5.2}$$

Here iter is the current number of iteration and maxiter is the maximum number of iterations. c_{upper} and c_{low} are the upper and lower bounds of the learning factors, respectively. In this algorithm, following the result of [32], we take $c_{upper} = 2$, $c_{low} = 0.75$.

Consequently, as the PSO algorithm proceeds, the cognitive learning factor is linearly decreased and the social learning factor is linearly increased.

(ii) [33] proposed the time varying formula for ω:

$$\omega = (\omega_1 - \omega_2) \times \frac{maxiter - iter}{maxiter} + \omega_2 \tag{5.3}$$

ω_1 and ω_1 are the upper and lower bounds of ω, respectively. In our algorithm, following the result of [33], we usually set $\omega_1 = 0.9$, $\omega_2 = 0.4$.

We use improvements (i) and (ii) because this approach is in accordance with the idea of particle swarm optimization: at the beginning, each bird has a large cognitive learning factor and small social learning factor, and each bird searches mainly by its own experience. After a period of time, as each bird gets more and more knowledge from the bird population, it relies increasingly on the social knowledge for its search. In addition, the effect of inertia velocity will decrease over time since the particles get more and more information from cognitive learning and social learning in the process of searching, so they rely increasingly on their learning instead of the inertia.

After applying these improvement, we get the improved pso algorithms:

5.3 Basic algorithm for minimization/ maximization problem

Initialization process

1.1. For each of the n particles, initialize particle position xi and velocity vi with random values.

1.2. Evaluate the fitness value of each particle according to the objective function.

update process:

2.1. Update the velocity of particles by formula (2.17). Here v_i are limited to an interval $[v_{min}, v_{max}]$. If any value of v_i is out of the bounds, then we will take the corresponding upper bound or lower bound.

2.2. Based on the velocity, update the position of particles by formula (2.18)

2.3. update the fitness value, then update pbest and gbest based on that.

If the stopping criteria is satisfied, output the pbest and gbest.

If not, update c_1, c_2 and ω by formula (5.1), (5.2) and (5.3), and repeat the update process 2.1.

Clearly, this basic algorithm can be used to solve either minimization or maximization problem. For minimization problem, the update process of pbest and gbest in 2.3 is: for each particle, if the updated fitness value $<$ the fitness value of current pbest, then pbest

is updated to the new solution; otherwise, pbest remains unchanged. gbest is the particle pbest for the particle that achieves the minimum of the pbest fitness values over the whole population of particles.

For maximization problem, the update process of pbest and gbest in 2.3 is: for each particle,if the updated fitness value $>$ the fitness value of current pbest, then pbest is updated to the new solution; otherwise, pbest keeps unchanged. gbest is the particle which take maximum of the fitness value of pbest.

This algorithm is an efficient way to obtain D-optimal design for linear regression, especially when the observations are correlated. The model for linear regression can be written as in [7]

$$y_i = f_i(x)'\beta + \epsilon_i \tag{5.4}$$

where i=1...n, β is a k vector of parameters, and $f_i(x) = (f_{1i}(x), f_{2i}(x), \dots f_{ki}(x))$ is a k vector of polynomial functions of x, and n is the number of observations. The design matrix is: $X = (x_{ij})_{n\times d}$, and D-optimality aims to maximize of the determinant of the information matrix, where the information matrix for these models is:

$$I = X'V^{-1}X \tag{5.5}$$

$$V = cov(Y) = \sigma^2(\rho_{ij})_{n\times n} \tag{5.6}$$

is the variance covariance matrix of the errors. In the linear regression, the swarm is the design matrix.

5.4 Nested PSO algorithms and their application

For regression with Fisher information matrix involving unknown parameters, we need two "*swarms*" of particles (one is ξ, another is θ), and solve it by using a nested PSO algorithm. These two swarms of particles are used in different layers of iterations. In each layer, the fitness value is determined by one of the two swarms of particles. For convenience,

we note the two swarms corresponding to ξ and θ as swarm 1 and swarm 2, the position as x_i and y_i, and the velocity as xv_i yv_i, respectively.

5.4.1 pso algorithm for pessimistic(minimax) criterion

define $f_{actions}(\theta,\xi) = max_{\theta\in\Theta}(log|I^{-1}(\theta,\xi)|))$. Then this optimization problem is to find $min_{\xi} f_{actions}(\theta,\xi)$. Clearly $f_{actions}(\theta,\xi)$ is based on the particle swarm θ, and $min_{\xi} f_{actions}(\theta,\xi)$ is based on the particle swarm ξ.

Initialization process

1.1. For each of the n particles in each of the two swarms ξ, and θ, initialize particle position x_i, y_i and velocity xv_i, yv_i with random vectors.

1.2. evaluate the fitness value $f_{actions}(x)$ of each particle according to the objective function by basic algorithm . Then compute the local and global best position based on that.

update process:

2.1. update velocity xv_i of particles in swarm 1 by formula (2.17). Here xv_i are limited into an interval $[v_{min}, v_{max}]$. If any value of v_i is out of the bound, then we will take corresponding upper bound or lower bound.

2.2. based on the velocity, update the position of particles in swarm 1 by formula (2.18)

2.3. based on the new position, update the fitness value $f_{actions}(x)$ by basic algorithm. Then update pbest and gbest based on that.

If the stopping criteria is satisfied, output the gbest and related fitness value. If not, update c_1, c_2 and ω by formula (5.1), (5.2) and (5.3), and repeat the update process.

From the algorithm, we can see the process of evaluating $f_{actions}(x)$ is the inner circulation, the process of evaluating $min_{\xi} f_{actions}(\theta,\xi)$ is the outer circulation.

5.4.2 pso algorithm for index of optimism criterion

define $f_{actions}(\theta,\xi) = (1-\alpha)max_{\theta\in\Theta}log|I^{-1}(\theta,\xi)| + \alpha min_{\theta\in\Theta}log|I^{-1}(\theta,\xi)|$. Our object is to find $min_{\xi} f_{actions}(\theta,\xi)$.

Initialization process:

1.1. For each of the n particles in each of the 2 swarms ξ, and θ, initialize particle position

x_i, y_i and velocity xv_i, yv_i with random vectors.

1.2. evaluate the fitness value $max_{\theta\in\Theta} log|I^{-1}(\theta,\xi)|$ and $min_{\theta\in\Theta} log|I^{-1}(\theta,\xi)|$ by basic algorithm. Then initialize the $f_{actions}(x)$ and local and global best position.

The update process is similar to pso algorithm for pessimistic(minimax) criterion. The only difference is in the update process, we will compute both $max_{\theta\in\Theta} log|I^{-1}(\theta,\xi)|$ and $min_{\theta\in\Theta} log|I^{-1}(\theta,\xi)|$ by basic PSO algorithm and take the weighted average of them as the fitness value.

5.4.3 pso algorithm for minimax regret criterion

define $RV(\theta,\xi) = log|I^{-1}(\theta,\xi)| - min_{\xi} log|I^{-1}(\theta,\xi)|$. Then this optimization problem is to find $min_{\xi} max_{\theta\in\Theta} RV(\theta,\xi)$. So this is a 3-fold nested algorithm.

Initialization process:

1.1. For each of the n particles in each of the 2 swarms ξ, and θ, initialize particle position x_i, y_i and velocity xv_i, yv_i with random vectors.

1.2. compute the fitness value $min_{\xi} log|I^{-1}(\theta,\xi)|$ by basic algorithm. Based on that, compute $RV(\theta,\xi)$. Then initialize the local and global best position based on that.

update process:

2.1. update velocity yv_i of particles in swarm 2 by formula (2.17)

2.2. based on the velocity, update the position of particles in swarm 2 by formula (2.18)

2.3 update the fitness value $max_{\theta\in\Theta} RV(\theta,\xi))$ by basic algorithm.

2.4. update velocity xv_i of particles in swarm 1 by formula (2.17)

2.5. based on the velocity, update the position of particles by in swarm 1 formula (2.18)

2.6. update the fitness value(the loss function) $min_{\xi} max_{\theta\in\Theta} RV(\theta,\xi)$ by basic algorithm. Then update pbest and gbest based on that.

If the stopping criteria is satisfied, output the gbest and related fitness value. If not, update c_1, c_2 and ω by formula (5.1), (5.2), and (5.3), and repeat the update process.

5.5 Result and comparison

From table 5.1 and 5.2, we see that when ρ=0.1, the determinants obtained by

simulated annealing is a little higher than the result of PSO method. However, when ρ=0.4, the determinants from the PSO algorithm are much higher than the results of the simulated annealing.

Table 5.1: Basic PSO for linear regression with circulant correlation structure

n	ρ	Annealing Determinant	PSO Determinant
6	0.1	37.5	33.9
6	0.4	100	114.3

Table 5.2: Basic PSO for linear regression with nearest neighbor correlation structure

n	ρ	Annealing Determinant	PSO Determinant
6	0.1	37.2	33.5
6	0.4	74.9	84.3

From tables 5.3 and 5.4, we see the gbest of index of optimism criterion is better than that of the pessimistic criterion and minimax regret criterion, and gbest is inversely proportional to α. That is because the pessimistic criterion always consider the worst case, but index of optimism criterion takes a trade off between optimistic case and pessimistic case. When α increase, the extent of optimistic get larger, so the loss function get smaller(and therefore better).

Table 5.3: Different criterion with Michaelis-Menten model

criterion	gbest	support point 1 and weight	support point 2 and weight
Pessimistic	8.9996	50.1889 0.5007	200 0.4993
Index of optimism with α=0.7	5.6371	28.9594 0.5140	200 0.4860
Index of optimism with α=0.5	6.1237	91.5995 0.2158	200 0.7842
Index of optimism with α=0.3	7.5770	118.1915 0.1648	200 0.8352
Minimax regret	7.7660	39.5151 0.5648	200 0.4352

Table 5.4: Different criterion with two parameter logistic regression model

criterion	gbest	support points
Pessimistic	4.1104	-0.3384 1.0064 1.6533 2.6503
Index of optimism with α=0.7	3.3675	0.0227 0.5731 0.3514 -0.4051
Index of optimism with α=0.5	3.4405	-0.4583 0.6799 0.0676 2.2800
Index of optimism with α=0.3	3.6436	2.5594 1.6075 2.2055 -0.2563
Minimax regret	3.3282	0.6467 1.4097 -0.2367 0.5244

Chapter 6

Discussion

The first part of my dissertation demonstrates that a modified simulated annealing algorithm can successfully determine highly efficient D-optimal designs for second order polynomial regression on $[-1, 1]^2$ for a variety of correlated error structures and with the design size, n, not limited to a multiple of the number of regression parameters. The combination of (i) a "middle ground" perturbation scheme, (ii) the use of a parameter that controls the size of the neighborhood for the perturbations, and (iii) re-heating, leads to designs that - while not likely globally optimal - are better than those obtained by searching among the set of designs known to be D-optimal for the uncorrelated errors case. In particular, when the true correlation parameter is well away from 0, the final SA design has much greater relative efficiency than the "best uncorrelated" comparison design.

The SA algorithm needs only a well-defined energy function to maximize, here the determinant of the information matrix. Thus, the same algorithm may be used for other design optimality criteria, for example, A- and E-optimality. In the absence of exact analytic optimal designs when errors are correlated, the SA algorithm is an attractive, easily implemented method to find highly efficient designs. Extensions to higher degree polynomial regression models are immediate, except for the likely need for longer run times and slower reduction of the temperature to allow for more effective searching over a larger design region.

Limitations of this approach are apparent. First, the value of the correlation parameter is specified in our examples as is the correlation structure itself. While the trend in improved D-efficiency as the correlation moves further from 0 and the n-size increases is generally apparent, and the final design points depart more from the usual vertices of the design region used in the optimal uncorrelated case, whether the final SA design will be of practical value

for the experimenter depends on the correlation parameter, which is usually unknown. If the true correlation is close to 0, the uncorrelated errors optimal designs are likely satisfactory, but there is potential for gain when this is not so.

In the second part, I solved weak universal optimal block designs for the nearest neighbor correlation structure and multiple block sizes, for the hub correlation structure with any block size, and for circulant correlation with odd block size. For circulant correlation with even block size and nearest neighbor correlation with block size more than 6, the problem becomes more complicated. How to make a general construction to a weak universal optimal block design for circulant correlation with even block size and nearest neighbor correlation with block size more than 6 is still a open question.

In the third part, combining the theorem of decision making and pso, we propose nested pso algorithms with all of these three criteria applied to the Michaelis-Menten model and the two parameter logistic regression model and make comparison among the quality of solutions found from the three criteria. For index of optimism criterion, we set the index of optimism=0.3 (the decision maker is relatively pessimistic), 0.5 (the decision maker compromises between the pessimistic and optimistic case) and 0.7 (the decision maker is relatively optimistic) respectively.

Comparison of PSO and simulated annealing:

Simulated annealing algorithm is an efficient way to solve unweighted optimal design with 2 way linear regression. Specifically, for matrix with variable in high dimension (in our section 1 the variable is usually 2 12 $\times 1$ vectors or even more complicated), simulated annealing is more efficient than other algorithms, like PSO. That is because our improved simulated annealing algorithm allows us to improve the solution part by part, so we do not miss any corner of the design region. On the other side, PSO is not very good at solving problems with complicated matrix.

However, PSO algorithm is a efficient way to solve nonlinear and weighted optimal design, which can not be solved by simulated annealing. For matrix with variable in relatively low dimension (like for 1-way polynomial regression, the variable is a 6×1 vector), the PSO

algorithm usually can get result that is a little better than simulated annealing.

References

[1] Zhu, Z., Coster, D. C., and Beasley, L., "Properties of a covariance matrix with an application to D-optimal design," *Electronic Journal of Linear Algebra*, Vol. 10, 2003, pp. 65–76.

[2] Dette, H., Kunert, J., and Pepelyshev, A., "Exact optimal designs for weighted least squares analysis with correlated errors," *Statistica Sinica*, Vol. 18, 2008, pp. 135–154.

[3] Goos, P., *The optimal design of blocked and split-plot experiments*, Springer, 2002.

[4] Lejeune, M., "Heuristic optimization of experimental designs," *European Journal of Operational Research*, Vol. 147, 2003, pp. 484–498.

[5] Dimitris, B. and Omid, N., "Robust optimization with simulated annealing," *Springer Science+Business Media, LLC.*, 2009.

[6] Abdullah, S., Golafshan, L., and Nazri, M., "Re-heat simulated annealing algorithm for rough set attribute reduction," *International Journal of the Physical Sciences*, Vol. 6(8), 2011, pp. 2083–2089.

[7] Zhu, Z., "Optimal experimental designs with correlated observations," *PhD dissertation, Department of Mathematics and Statistics, Utah State University*, 2004.

[8] Cheng, C., "Optimal regression designs under random block-effects models," *Statistica Sinica*, 2008, pp. 485–497.

[9] Boon, J. E., "Generating exact d optimal design for polynomial models," *SpringSim '07*, Vol. 2, 2007.

[10] Pukelsheim, F., *Optimal design of experiments*, SIAMr, 2006.

[11] Cadima, J., Calheiros, F., and Preto, I., "The eigenstructure of block-structured correlation matrices and its implications for principal component analysis," *Journal of Applied Statistics*, Vol. 37, 1971, pp. 577–589.

[12] Atkinsa, J. E. and Cheng, C., "Optimal regression designs in the presence of random block effects," *Journal of Statistical Planning and Inference*, Vol. 77, 1999, pp. 321–335.

[13] Kiefer, J. and Wynn, H. P., "Optimum Balanced Block and Latin Square Designs for Correlated Observations," *The Annals of Statistics*, Vol. 9, 1981, pp. 737–757.

[14] Kenneth, H. R. and Michaels, J. G., *Handbook of Discrete and Combinatorial Mathematics*, CRC Press, 2000.

[15] Cheny, K. and Wei, R., "A few more cyclic Steiner 2-designs," *The Electronic Journal of Combinatorics 13*, 2006.

[16] Lam, C. and Miao, Y., "On Cyclically Resolvable Cyclic Steiner 2-Designs," *Journal of Combinatorial Theory*, Vol. A 85, 1999, pp. 194–207.

[17] Chang, Y., "Some Cyclic BIBDs with Block Size Four," *Wiley Periodicals, Journal of Combinatorial Designs*, Vol. 12, 2004, pp. 177–183.

[18] Oehlert, G. W., *A first course in design and analysis of experiments*, CRC Press, 2000.

[19] Jin, B., "Optimal Block Designs with Limited Resources," *PhD dissertation, Virginia Polytechnic Institute and State University*, 2004.

[20] C. Lindner, C. R., *Design theory*, CRC Press, 1997.

[21] Kennedy, J. and Eberhart, R., "Particle swarm optimization," *Proceedings of International Conference on Neural Networks*, 1995, pp. 1942–1948.

[22] Dette, H. and Wong, W., "E optimal designs for the Michaelis Menten model," *Statistics and Probability Letters*, Vol. 44, 1999, pp. 405–408.

[23] King, J. andWong, W. K., "Minimax D-optimal Designs for the Logistic Model," *Biometrics*, Vol. 56(4), 2000, pp. 1263–1267.

[24] Diao, Z., Zhen, H. D., Liu, J., and Liu, G., *Operations research*, Higher Education Press, 2001.

[25] Fozunbal, M. and Kalker, T., "Decision-Making with Unbounded Loss Functions," *Information Theory, 2006 IEEE International Symposium*, 2006, pp. 2171 – 2175.

[26] Hoel, P., "Minimax Designs in Two Dimensional Regression," *The Annals of Mathematical Statistics*, Vol. 36, 1965, pp. 1097–1106.

[27] Box, M. J. and Draper, N. R., "Factorial Designs, the $|X'X|$ Criterion, and Some Related Matters," *Technometrics*, Vol. 13, 1971, pp. 731–742.

[28] Chen, R. B., Chang, S. P., Wang, W., and Wong, W. K., "Optimal Experimental Designs via Particle Swarm Optimization Methods," *Contributed*, 2011.

[29] Yang, R., Wang, L., and Wang, Z., "Multi-objective particle swarm optimization for decision-making in building automation," *IEEE/PES General Meetingl*, 2011, pp. 1–5.

[30] Yang, L. and Shu, L., "Application of Particle Swarm Optimization in the Decision-Making of Manufacturers' Production and Delivery," *Electrical, Information Engineering and Mechatronics 2011*, 2012, pp. 83–89.

[31] Kiranyaz, S., Pulkkinen, J., and Gabbouj, M., "Multi-dimensional particle swarm optimization for dynamic environments," *Innovations in Information Technology*, 2008, pp. 34–38.

[32] Cai, X., Cui, Z., Zeng, J., and Tan, Y., "Particle Swarm Optimization with Self-adjusting Cognitive Selection Strategy," *International Journal of Innovative Computing, Information and Control*, Vol. 4, 2008, pp. 943–952.

[33] Ratnaweera, A., Halgamuge, S., and Watson, H., "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *Evolutionary Computation, IEEE Transactions on*, Vol. 8, 2004, pp. 240– 255.

Appendices

Appendix A

TYPICAL CODES AND COMMENTS

A.1 A Simulated Annealing Algorithm for D-optimal Design for 2-Way Polynomial Regression with Correlated Observations

```
% Simulated annealing in matlab
n=12                    %determine the number of observations
Z1=unifrnd(-1,1,n,1);
  Z2=Z1.^2;
Z3= unifrnd(-1,1,n,1);
 Z4=Z3.^2;
Z5=Z1.*Z3
Z=[Z1 Z2 Z3 Z4 Z5];              %initialize the matrix
V=eye(n);
m(1:n-1)=0.1;
V1=diag(m,-1);
V2=diag(m,1);
V=V+V1+V2;
 V(n,1)=0.1;
V(1,n)=0.1;                %V is the corrlation matrix



X=zeros(n,5,100);
O=ones(n,1);
for j=1:100
```

```
X(:,:,j)=Z;
Xc= X(:,:,1);
end
Xc=[0 Xc];              %Xc is the design matrix

Tstart= 50; % Start temperature
Tend= 1; % Stop temperature
r= 0.99;
g=0.3;
k=1.01;
c=0;
T= Tstart;
while T >= Tend  % The outer loop

 result= ones(1,100);
for i=3: 100                          % the  inner loop  1
 Z=zeros(n,1); %Z is the vector of perturbation


 for j=1:4
 Z(j)=g*random('unif',-1, 1,1,1);    % We do perturbation part by part
end


X1= Xc(:,2)+Z;
X2=X1.^2;
Z=zeros(n,1);
for j=1:4
 Z(j)=g*random('unif',-1, 1,1,1);
end

X3= Xc(:,4)+Z;
```

```
X4=X3.^2;
X5=X1.*X3;
          X(:,:,i+1) =  [X1 X2 X3 X4 X5];
Xn=[O X(:,:,i+1)];          % Xn is new candidate design matrix



         dE=det(Xn'*inv(V)* Xn)-det(Xc'*inv(V)*Xc);
       result(i)= (abs(det(Xn'*inv(V)* Xn))<1.02*abs(det(Xc'*inv(V)*Xc)));

if result(i-2)+ result(i-1)+ result(i)==0   %we repeat the iterations until the improvement
 is less than the   threshold value 3 times.

break
end



      Xn(find(Xn>1))=1;
Xn(find(Xn<-1))=-1;        % the boundary of Xn is [-1,1].
        if dE>0
         Xc= Xn;
         elseif  exp(dE/T ) >1.01^c*  random('unif',0,1)
 %1.01^c is the threshold value.

           Xc= Xn;
          end
           end

 result= ones(1,100);
for i=3: 100  % inner loop  2
Z=zeros(n,1);
for j=5:8
```

```
 Z(j)=g*random('unif',-1, 1,1,1);
end



X1= Xc(:,2)+Z;
X2=X1.^2;
Z=zeros(n,1);

for j=5:8
 Z(j)=g*random('unif',-1, 1,1,1);
end

X3= Xc(:,4)+Z;
X4=X3.^2;
X5=X1.*X3;
          X(:,:,i+1) =  [X1 X2 X3 X4 X5];
Xn=[O X(:,:,i+1)];



         dE=det(Xn'*inv(V)* Xn)-det(Xc'*inv(V)*Xc);
    result(i)= (abs(det(Xn'*inv(V)* Xn))<1.02*abs(det(Xc'*inv(V)*Xc)));
if result(i-2)+ result(i-1)+ result(i)==0
break
end



            Xn(find(Xn>1))=1;
           Xn(find(Xn<-1))=-1;
        if dE>0
         Xc= Xn;
         elseif  exp(dE/T ) >1.01^c*  random('unif',0,1)
```

```
            Xc= Xn;
          end
            end

 result= ones(1,100);
for i=3: 100   % inner loop  3
Z=zeros(n,1);

for j=9:12
 Z(j)=g*random('unif',-1, 1,1,1);
end




X1= Xc(:,2)+Z;
X2=X1.^2;
Z=zeros(n,1);

for j=9:11
 Z(j)=g*random('unif',-1, 1,1,1);
end

X3= Xc(:,4)+Z;
X4=X3.^2;
X5=X1.*X3;
          X(:,:,i+1) =  [X1 X2 X3 X4 X5];
Xn=[0 X(:,:,i+1)];


         dE=det(Xn'*inv(V)* Xn)-det(Xc'*inv(V)*Xc);
 result(i)= (abs(det(Xn'*inv(V)* Xn))<1.02*abs(det(Xc'*inv(V)*Xc)));
```

```
if result(i-2)+ result(i-1)+ result(i)==0
break
end


      Xn(find(Xn>1))=1;
Xn(find(Xn<-1))=-1;
          if dE>0
         Xc= Xn;
         elseif  exp(dE/T ) >1.01^c*  random('unif',0,1)
            Xc= Xn;
          end
            end

    T = r * T ; %lower the temperatures
   g=0.99*g;
   c=c+1;
end % We make make the perturbation neighborhood smaller and
the acceptance threshold higher at each time we lower the temperature so it
becomes harder to leave a local optimum.
det(Xc'*inv(V)*Xc)
Xc
```

A.2 Uncorrelated method

```
V=eye(12);

W=zeros(12);
for i=1:11
m=zeros(1,i);
```

```
m(1:i)=0.4^(12-i)
W=diag(m,12-i);
V=V+W;
end
V=V+V'-eye(12);  %V is the correlation matrix

X=zeros(5,12);

O=ones(1,12);



X(1,:)= [-1 -1 -1 0 0 0 1 1 1 -1 -1 1 ];
X(2,:)= X(1,:).^2;
X(3,:)= [-1 0 1 -1 0 1 -1 0 1 -1 1 -1 ];   % take the value given by  Box et al.
X(4,:)= X(3,:).^2;
X(5,:)= X(1,:).* X(3,:);

D=det([O; X(1,:); X(3,:); X(2,:); X(4,:) ; X(5,:)]* inv(V)*
[ O; X(1,:); X(3,:); X(2,:); X(4,:) ; X(5,:)]');
% apply these values to correlated case
D
```

A.3 PSO algorithm for pessimistic(minimax) criterion for logistic model

```
 % X is particle position, Y is theta.
 max_iterations=50;
                               no_of_particles=50;

      X= zeros(50,7);
      Xv=zeros(no_of_particles,7); );
   %X is a group of vectors including
 the information of our design, the first
```

```
4 elements are support points and  the last 3 are weight.


      Y= zeros(no_of_particles,2);
      Yv=zeros(no_of_particles,2);  % Y is the unknown parameters.

p_currentY=zeros(1,2);
p_currentX=zeros(1,7
c_upper=2;
c_low=0.75;

%initialise the particles and  velocity components
 fval=zeros(no_of_particles,1);
for x = 1: no_of_particles

            X(x,:) = [unifrnd(-0.5, 3,4,1);    unifrnd(0.23,0.26,3,1)  ];
        Xv(x,:) =[unifrnd(-0.5, 3,4,1);    unifrnd(0.23,0.26,3,1) ];
  % initialize the position and velocity of the swarm of X.

        p_bestX= X(x,:);
       current_fitness(x) = 1;
    p_best_fitness(x) = 1;

end
%decide on the global best among all the particles
    [g_best_val,g_best_index] = min(current_fitness);


        g_bestX= X(g_best_index,:);

   %main outer  particle swarm loop
for count = 1:50
```

```
    % c1, c2 and k are time varying parameters,
 we update them in every iteration of our outer loop.


c1= (c_upper - c_low)*(max_iterations-count)/ max_iterations+c_low;
c2=(c_upper - c_low)* count/ max_iterations+c_low;
k=0.5*(max_iterations-count)/ max_iterations+0.4;
for x= 1:no_of_particles

  %inner particle swarm loop

      for i= 1:no_of_particles

            Y(i,:) = [unifrnd(0, 2.5,1,1);  unifrnd(1, 3,1,1) ];
        Yv(i,:) =  [unifrnd(-.5, .1,1,1);  unifrnd(-.5, 1,1,1) ];
% initialize the   position and velocity of the swarm of Y

        p_bestY= Y(i,:);

           current_fitness(i) = 1;
    p_best_fitness(i) = 1;
 end
%decide on the global best among all the particles
    [g_best_val,g_best_index] = max(current_fitness);

        g_bestY= Y(g_best_index,:);
   g=0.9;

  for count = 1:30
c1= (c_upper - c_low)*(max_iterations-count)/ max_iterations+c_low;
c2=(c_upper - c_low)* count/ max_iterations+c_low;
```

```
g=0.9;
 for i= 1:no_of_particles

P1 =1/(1+exp(-Y(i,2)*(X(x,1)-Y(i,1))));
  M1=[Y(i,2)^2* P1*(1- P1), -Y(i,2)*(X(x,1)-Y(i,1)) * P1*(1- P1);
 -Y(i,2)*(X(x,1)-Y(i,1)) * P1*(1- P1), (X(x,1)-Y(i,1))^2* P1*(1- P1)];

P2 =1/(1+exp(-Y(i,2)*(X(x,2)-Y(i,1))));
  M2=[Y(i,2)^2* P2*(1- P2), -Y(i,2)*(X(x,2)-Y(i,1)) * P2*(1- P2);
 -Y(i,2)*(X(x,2)-Y(i,1)) * P2*(1- P2),  (X(x,2)-Y(i,1))^2* P2*(1- P2)];


 P3 =1/(1+exp(-Y(i,2)*(X(x,3)-Y(i,1))));

    M3=[Y(i,2)^2* P3*(1- P3), -Y(i,2)*(X(x,3)-Y(i,1)) * P3*(1- P3);
 -Y(i,2)*(X(x,3)-Y(i,1)) * P3*(1- P3),  (X(x,3)-Y(i,1))^2* P3*(1- P3)];

P4=1/(1+exp(-Y(i,2)*(X(x,4)-Y(i,1))));

    M4=[Y(i,2)^2* P4*(1- P4), -Y(i,2)*(X(x,4)-Y(i,1)) * P4*(1- P4);
 -Y(i,2)*(X(x,4)-Y(i,1)) * P4*(1- P4), (X(x,4)-Y(i,1))^2* P4*(1- P4)];


 M5= X(x,5)* M1+ X(x,6)* M2+X(x,7)* M3+ (1- X(x,5)- X(x,6)-  X(x,7)) * M4;
%M5 is the information matrix


               current_fitness(i) =log(det(inv(M5)));

         if current_fitness(i) > p_best_fitness(i)  % in pessimism criterion, we suppose
 the unknown parameters will maximize our loss function, so the inner loop
is used to solve   or maximization problem.
```

```
            p_best_fitness(i) = current_fitness(i);
                p_bestY= Y(i,:);
  end
    [g_best_val,g_best_index] = max(current_fitness);

        g_bestY= Y(g_best_index,:);

end        %this end correspond to     for i= 1:no_of_particles


for i= 1:no_of_particles   % update process of the unknown parameters Y.


            p_currentY= Y(i,:);
 %Update of the velocity of Y. If the velocity get out of  the bound,
then we either take the boundary number or keep it unchanged.
bv= g*Yv(i, :) + c1*rand*(p_bestY-p_currentY) + c2*rand*(g_bestY-p_currentY);

if length(find(bv<-.2))+ length(find(bv>1))>0

Yv(i,:) = Yv(i, :);
else

            Yv(i, :)= bv;
 end
%update process of the position of Y.
 b=(p_currentY+Yv(i,:));

      if    b(1)<0 |  b(1) >2.5

Y(i,:) = p_currentY;
```

```
elseif  b(2) <1 | b(2)>3
Y(i,:) = p_currentY;
else

Y(i,:) = p_currentY+Yv(i,:) ;
end



end
if g>0.4
   g=0.99*g;
else
g=g;

end

end%this end correspond to  for count = 1:30



fval(x)= current_fitness(g_best_index);
end
for x = 1:no_of_particles
%we take X( the design vectors) to minimize the  maximize our loss function,
 so the outer loop is used to  solve    minimization problem.


                    current_fitness(x) =  fval(x);
         if current_fitness(x) <p_best_fitness(x)
            p_best_fitness(x) = current_fitness(x);
                p_bestX= X(x,:);
```

```
  end
    [g_best_val,g_best_index] = min(current_fitness);

        g_bestX= X(g_best_index,:);

end  %this end correspond to    for x = 1:no_of_particles
  %Update of the velocity of X. If the velocity get out of  the bound,
then we will take the boundary number.

for x = 1:no_of_particles
            p_currentX= X(x,:);
av=k*Xv(x, 1:4) + c1*rand*(p_bestX(1:4)-p_currentX(1:4)) + c2*rand*(g_bestX(1:4)-p_currentX(1:4))
cv=k*Xv(x, 5:7) + c1*rand*(p_bestX(5:7)-p_currentX(5:7)) + c2*rand*(g_bestX(5:7)-p_currentX(5:7))

cv(find(cv<-.1))=-.1;

cv(find(cv> .1))=.1;
av(find(av<-.5))=-.5;
av(find(av> 1))=1;

              Xv(x, :)= [av cv];

 %update process of the position of X.


a1=p_currentX(1:4)+Xv(x,1:4);
a2=p_currentX(5:7)+Xv(x,5:7);
      if length(find(a1>3))+ length(find(a1<-.5))+ length(find(a1>.26))+ length(find(a2<.23)) >0
               X(x,:) = p_currentX;
else
```

```
X(x,:) = [a1 a2];
 end
end

end

 g_bestX       % this is the final result.
current_fitness(g_best_index)
```

Vita

Chang Li

Address:	Department of Mathematics and Statistics	Email:	chang.li@aggiemail.usu.edu
	Utah State University	Phone:	435-512-7641
	Logan, UT 84341		

EDUCATION:

Ph. D. candidate in Statistics, Utah State University expected May 2013.

Advisor: Professor Daniel Coster

Area of research: optimal design and statistical computing

M. S. in Statistics, Utah State University August 2009

Advisor: Professor Daniel Coster

Area of research: actuarial statistics

M. S. in Mathematics, Shandong University, China, July 2006.

Advisor: Professor Guojun Li

Specialty: Operations Research

Area of research: combinatorial optimization

B. S. in Mathematics, Shandong University, China, July 2003.

Graduate from Shandong Shiyan Senior Middle School, July, 1999

RESEARCH INTERESTS:

- Optimal experimental design
- Statistical computing
- Applications in actuarial mathematics

TEACHING EXPERIENCE:

- **Instructor in Utah State University**

  Elements of Algebra

  Intermediate Algebra

  College Algebra

  Calculus Techniques

- **Tutoring and Grading**

  Tutor for probability theory

  Grader for Partial Differential Equations, Engineering Mathematics and Statistics, Optimization, Introduction to Probability, Mathematical Statistics.

  Grading National University Entrance Examination, China, July 2004 and July 2005

CERTIFICATES:

- SAS advanced programmer Certificate
- SAS base programmer Certificate
- Actuary Certificate in P-1(level 10) and FM(level 10)
- Teaching Assistant Certificate, Utah State University

COMPUTER SKILLS:

- Proficient in SAS (statistical analysis system), Matlab, R, Microsoft Excel, Latex.
- Familiar with C, C++.