University of California

Los Angeles

FINDING OPTIMAL EXPERIMENTAL DESIGNS FOR MODELS IN BIOMEDICAL STUDIES VIA PARTICLE SWARM OPTIMIZATION

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Biostatistics

by

Jiaheng Qiu

2014

Abstract of the Dissertation

FINDING OPTIMAL EXPERIMENTAL DESIGNS FOR MODELS IN BIOMEDICAL STUDIES VIA PARTICLE SWARM OPTIMIZATION

by

Jiaheng Qiu

Doctor of Philosophy in Biostatistics

University of California, Los Angeles, 2014

Professor Weng Kee Wong, Chair

The theory of optimal experimental design provides insightful guidance on resource allocation for many dose-response studies and clinical trials. However, as more and more complicated models are developed, finding optimal designs has become an increasingly difficult task; therefore, the availability of an efficient and easy-to-use algorithm to find optimal designs is important for both researchers and practitioners. In recent years, nature-inspired algorithms, like Particle Swarm Optimization(PSO), have been successfully applied to many non-statistical disciplines, such as computer science and engineering, even though there is no unified theory to explain why PSO works so well. To date, there is only limited work in the mainstream statistical literature that applies PSO to solve statistical problems.

In my dissertation, I review PSO methodology and show it is an easy and effective algorithm to generate locally D- and c-optimal designs for a variety of nonlinear statistical models commonly used in biomedical studies (Qiu et al. [2014]). I develop a new version of PSO called Ultra-dimensional PSO (UPSO) to find D-optimal designs for multi-variable exponential and Poisson regression models with up to five variables and all pairwise interactions. I use the proposed novel search strategy to find minimally supported D-optimal designs and ascertain conditions under which such optimal designs exist for such models. A remarkable discovery in my work is that locally D-optimal designs for such models can have many more support points than the number of parameters in the model. This result is both new and interesting because almost all D-optimal designs have equal or just one or two more number of points than the the number of parameters in the mean response function, see the examples in monographs by Fedorov [1972], Atkinson and Donev [1992], and recent papers by Yang and Stufken [2009], Yang [2010]. This discovery also disproves the conjecture by Wang et al. [2006] that for M-variable interaction models (M $>$ 2), D-optimal designs are also minimally and equally supported and have a similar structure as D-optimal designs for 2-variable model.

In addition to single objective optimal designs, I apply PSO to find optimal designs for estimating parameters and interesting characteristics in continuation-ratio (CR) model with non-constant slopes. Such a model has a great potential in dose finding studies because it takes both efficacy and toxicity into consideration. The optimal design I am interested in constructing is a three-objective optimal design, which provides efficient estimates for efficacy, adverse effect and all parameters in the CR model. This work is quite new because there are virtually no three-objective designs for a trinomial model reported in the literature. Through multiple objective efficiency plots, practitioners can construct the desired compound optimal design by selecting appropriate weighted average of three optimal criteria in a more flexible and informative way.

I also conduct simulation studies for parameters selection in PSO, and compare the performance of PSO with other popular deterministic and metaheuristic algorithms in terms of the CPU time and the closeness of the generated designs to the optimal designs. I show that PSO outperforms its competitors in finding D- and c-optimal designs for different models I consider in my dissertation.

Glossary of important symbols

A	a matrix
$\|A\|$	the determinant of a matrix A
A^T	the transpose of a matrix A
c_{eff}	c-efficiency
D_{eff}	D-efficiency
E	the identity matrix
$f(x)$	the vector of regression functions
$I(\xi)$	Fisher information matrix
K	the number of distinct support points in design ξ
M	the number of variables in the multivariable Poisson and exponential model
N	the number of the total sample size
n	the number of common support points in all candidate designs searching for the optimum in a deterministic algorithm
n_+	the number of support points with positive weight in all candidate designs searching for the optimum in a deterministic algorithm
n_{flock}	flock size or population size used in a metaheuristic algorithm
P	number of unknown parameters in the mean function of a model
t	iteration number
u	a random variable with uniform distribution on [0,1]
χ	the design space; dose range
x_i	the i^{th} distinct support point $\in \chi$. In a dose response study, x_i is the i^{th} dosage applied

w_i	the weight for the i^{th} distinctive support point
Ω	the set of all possible w on a discretized space
ξ	a design defined on χ
Ξ	the set of all designs defined on χ
δ_x	the one-point design putting unit mass at x
θ	the parameter vector in a regression model
Ψ	the concave optimality criterion function
ψ	the directional derivative of Ψ
$\eta(x, \theta)$	the mean function of a nonlinear regression model

The dissertation of Jiaheng Qiu is approved.

Hongquan Xu

Gang Li

Elliot M. Landaw

Thomas R. Belin

Weng Kee Wong, Committee Chair

University of California, Los Angeles
2014

To my parents

who

always support me pursuing my dream.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Dr. Weng Kee Wong for his inspiring and constructive advice to my projects. I sincerely appreciate his countless hours of reflecting, reading, encouraging, and most of all his patience throughout the entire process.

I'm very grateful to Dr. Beate Ritz and Dr. Michelle Wilhelm for hiring me as a graduate student researcher in the Department of Epidemiology at UCLA since 2009. I appreciate their generous financial support and giving me the opportunity to apply what I learned in class to numerous projects.

Finally, I would thank my beloved family and friends for supporting me and encouraging me as always. Their willingness and excitement made the completion of this long journey an enjoyable experience.

VITA

2005	B.S. (Statistics), Nankai University, Tianjin, China.
2008	M.S. (Bioinformatics), Nankai University, Tianjin, China.
2010	M.S. (Biostatistics), UCLA.
2012–2013	Graduate student instructor, Department of Biostatistics, UCLA.
2009–2014	Graduate student researcher, Department of Epidemiology, UCLA.

CHAPTER 1

Introduction and Background

This chapter provides the introductory part for my dissertation, and background material for constructing an optimal design using different design criteria. I also review several search algorithms commonly used for finding optimal designs, including deterministic algorithms and metaheuristic algorithms.

1.1 Introduction

Optimal experimental designs have been gaining attention in the past few decades (Atkinson [1996]). A main reason is the rising cost in conducting experiments and the increasing realization in more applied fields that optimal design ideas can save costs substantially without sacrifice in statistical efficiency. Some real examples are given in Dette and Beidermann [2003], Dette et al. [2011], Woods et al. [2006], Lopez-Fidalgo et al. [2009] and Gilmour and Trinca [2011], where the applications range from designing reaction kinetics studies to estimating maximum tolerated dose (MTD) or most effective dose (MED) in dose finding studies in phase I and II clinical trials. Berger and Wong [2009] also provided a collection of concrete applications of optimal designs to real problems that ranges from biomedicine to social science researches.

Nonlinear models are widely used to study outcomes or responses in biomedical experiments. This means that we assume a known nonlinear functional relationship between the mean response and the independent variables, apart from unknown parameters that determine the shape and properties of the mean response. One common goal in the study is to estimate some or all parameters in the mean function. Typical dose-response models usually involves the drug dosage as the only explanatory variable in the model. I call them univariable models in my dissertation. When there are multiple factors in the mean function of the model, I call them multivariable models. Such models are often used to study the joint effects of multiple drugs or agents on the response variable. In the case of modeling multiple response variables, I call them multivariate models. Examples of such models are in drug studies when both efficacy and adverse outcomes are simultaneously modeled. This model classification terminology will be used throughout my dissertation.

Given a study objective or an optimality criterion and a model, the design problem is to select the right number of combination levels of independent variables to observe the outcome and what these levels are. The design optimality criterion for nonlinear models depends on the values of a subset or all the model parameters, and therefore nominal values (or best guesses for these parameters) are required before the optimal design can be implemented. Because they depend on these nominal values, the optimal designs are called locally optimal, a term coined by Chernoff [1953]. Such optimal designs usually represent the fist step in an optimal design finding strategy and is the simplest to construct and study.

Analytical descriptions of the locally optimal design for a nonlinear model are rarely available unless the model is very simple. When they do exist, they are usually complicated; see for example, the analytical description for the locally D-optimal design for estimating the two parameters in the logistic model (Silvey [1980]). Further, the formula or analytical description of the optimal design in a nonlinear model is invariably derived under a set of mathematical assumptions that may or may not apply in practice. As more and more complicated models are developed, finding optimal designs has become an increasingly challenging task for researchers. When the analytical description of an optimal design is difficult or even impossible to derive, it is desirable to have a flexible and effective algorithm that can find a variety of optimal designs quickly and reliably.

There are algorithms for finding optimal designs. A few of them can be proven to converge to the optimal designs, and prominent ones include Fedorov's and Wynn's algorithms for generating D- and c-optimal designs (Fedorov [1972], Wynn [1972]). D-optimal designs are useful for estimating all parameters in the mean function, and c-optimality targets minimizing the asymptotic variance of the function of parameters estimates of interest. For the few algorithms that can be shown to converge mathematically, problems may still exist including (i) they take too long to converge, (ii) they may fail to converge for more complicated setups that they are not designed for, such as nonlinear mixed effects models, and (iii) numerical issues due to rounding problems or the intrinsic nature of the sequential process. For example, many algorithms produce clusters of support points as the algorithm proceeds and these clusters require periodic and judicious collapsing into the correct number of support points, which is usually unknown.

Recently, nature-inspired metaheuristic algorithms have been successfully applied to solve many tough engineering and computer science problems (see examples in Whitacre [2011a,b]). These algorithms do not guarantee that the global optimum can always be found, but frequently get to (or close to) the optimum after several iterations. Among them, Particle Swarm Optimization (PSO) seems to be the most promising one in recent years due to its repeated successes in solving a large class of applied problems. PSO gains popularity from its flexibility, ease of implementation, and general applicability to solve (or nearly solve) complex optimization problems without having to make specific assumptions on the objective function. The main goal of my dissertation is to investigate the capability and performance of PSO in finding D- and c-optimal designs for a variety of models commonly used in biomedical studies.

1.2 Design theory for approximate designs

In this section I briefly introduce fundamental theory for constructing an optimal design and provide several simple examples.

1.2.1 Approximate design

Here and throughout, my focus is on a approximate design ξ, which is a probability measure defined on a given compact (i.e. closed and bounded) design space χ (Kiefer [1974]). Clearly, $\int_{\chi} \xi dx = 1$. If ξ has K points, we denote it by

$$\left\{ \begin{array}{cccc} x_1 & x_2 & ... & x_K \\ w_1 & w_2 & ... & w_K \end{array} \right\},$$

where w_i is the proportion of the total observations to be taken at the distinct point x_i, $i = 1, 2, ..., K$. A special and important case is the one-point design putting unit mass at the point x. I denote this design by δ_x throughout my dissertation.

The total number N of observations for the study is predetermined by cost or practical considerations. When N is known, the design ξ allocates Nw_i observations to points x_i, $i = 1, ..., K$. Note that these observation numbers may not be integers. In practice, each Nw_i in an approximate design is rounded to the nearest integer subject to that they sum to N. The general problem in the search for an optimal design is that once an optimality criterion and a statistical model are given, we need to determine *a*) K the number of support points required, *b*) where the points x_i's are, and *c*) their corresponding weights w_i's.

1.2.2 Information matrix

Following convention, the criterion function is formulated in terms of the design ξ through the information matrix. To fix ideas, consider a linear regression model with N independent observations $y_1, ..., y_N$,

$$y_i = f^T(x_i)\beta + \epsilon_i, \tag{1.1}$$

where $f(x_i)$ is a $P \times 1$ vector of regression functions, β is a $P \times 1$ vector of unknown parameters, and ϵ_i is an unobserved measurement error. Define $F = \begin{bmatrix} f(x_1) & f(x_2) & .... & f(x_N) \end{bmatrix}^T$ as the $N \times P$ design matrix. Suppose ϵ_i's are *i.i.d* normally distributed random variables, each with mean 0 and variance σ^2. Clearly, the maximum likelihood estimator (MLE) of β is $\hat{\beta} = (F^T F)^{-1} F^T y$, and the covariance matrix of $\hat{\beta}$ is given by $Var(\hat{\beta}) = \sigma^2 (F^T F)^{-1}$.

The Fisher information matrix is defined by the expectation of the square of the first derivative of the total log likelihood function with respect to β. For independent observations taken at $x_1, \ldots, x_N$, the total information matrix under model (1.1) is proportional to

$$\begin{aligned} I_N &= \sum_{i=1}^{N} f(x_i) f^T(x_i) \\ &= F^T F. \end{aligned}$$

For a $K-$point approximate design ξ, its normalized information matrix is proportional to

$$I(\xi) = \mathrm{E}_\xi f(x) f^T(x) = \sum_{i=1}^{K} w_i f(x) f^T(x).$$

We note that $K \geq P$ for $I(\xi)$ to be non-singular.

Example: Quadratic regression model As a specific example, consider the regression model given by

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i,$$

where ϵ_i's are *i.i.d* normally distributed random variables, each with mean 0 and variance σ^2. The regression function at point x_i is $f(x_i) = (\ 1,\ \ x_i,\ \ x_i^2\ )^T$. For a $K-$point design ξ, the normalized information matrix is

$$I(\xi) = \sum_{i=1}^{K} w_i f(x_i) f^T(x_i) = \sum_{i=1}^{K} w_i \begin{bmatrix} 1 & x_i & x_i^2 \\ x_i & x_i^2 & x_i^3 \\ x_i^2 & x_i^3 & x_i^4 \end{bmatrix},$$

apart from a multiplicative constant.

1.2.3 Design criteria

Let ξ be a design defined on the space χ, and Ξ be the set of all possible designs on χ. Here are some optimality design criteria frequently used for linear models.

D-optimality:

This criterion seeks to minimize the volume of the confidence ellipsoid for all parameters in the mean function: $\min\limits_{\xi \in \Xi} \log|I(\xi)^{-1}|$.

A-optimality:

This criterion seeks to minimize the average of the variances of the parameter estimates: $\min\limits_{\xi \in \Xi} \text{trace}[I(\xi)^{-1}]$.

c-optimality:

Suppose there is interest in estimating $c(\theta)$ a linear combination of the parameters. This criterion seeks to minimize the asymptotic variance of $c(\hat{\theta})$, which is $\nabla^T Var(\hat{\theta})\nabla \propto \nabla^T I^{-1}(\xi)\nabla$, and ∇ is the partial derivative of $c(\hat{\theta})$. The design criterion is: $\min\limits_{\xi \in \Xi} \nabla^T I^{-1}(\xi)\nabla$.

1.2.4 General equivalence theorem

The above criteria are formulated in terms of the information matrix $I(\xi)$. An important piece of work by Kiefer and Wolfowitz is the equivalence theorem, which allows us to verify if an approximate design is optimal among all designs on χ when the given criterion function $\Psi(I(\xi))$ is convex or concave (Kiefer and Wolfowitz

[1960], Kiefer [1974]). First I review the concept of convex/concave functions.

Convex/concave function

A function $g : Q \to R^1$ is convex if Q is a convex set and the following inequality holds for any $x, y \in Q$ and $0 \leq \alpha \leq 1$:

$$g((1-\alpha)x + \alpha y) \leq (1-\alpha)g(x) + \alpha g(y).$$

If g is convex, then $-g$ is concave. For example, $-x^2$ is a concave function on $[-\infty, \infty]$. An important property of a convex function is that any local minimum is necessarily global. This can be seen as follows.

Suppose x is a local optimum; that is, there is an open neighborhood U where $g(x) \leq g(u)$ for $\forall u \in U$. For arbitrary $y \in R^1$, select $(1-\alpha)x + \alpha y$ close to x such that

$$\begin{aligned} g(x) &\leq g((1-\alpha)x + \alpha y) \quad &&\text{for small } \alpha > 0 \\ &\leq (1-\alpha)g(x) + \alpha g(y) \quad &&\text{since } g \text{ is convex} \end{aligned}$$

Rearranging terms, we get $g(x) \leq g(y)$ which indicates that x is also the global optimum. Similarly, a local maximum in a concave function is necessarily the global maximum as well.

All optimality criteria given in section 1.2.3 are convex functions of the information matrix. In my dissertation, I choose to work with concave criterion functionals, and so the criteria are

D-optimality: $\Psi(I(\xi)) = \log|I(\xi)|$

c-optimality: $\Psi(I(\xi)) = -c^T I^{-1}(\xi)c$

A-optimality: $\Psi(I(\xi)) = -\text{trace}[I(\xi)^{-1}]$

As an illustration, to prove D-optimality $\Psi(\xi) = \log|I(\xi)|$ is a concave function of $I(\xi)$, it is enough to show that for any information matrices $I(\xi_1)$ and $I(\xi_2)$

and all $0 \leq \alpha \leq 1$,

$$\log|(1-\alpha)I(\xi_1)+\alpha I(\xi_2)| \geq (1-\alpha)\log|I(\xi_1)|+\alpha\log|I(\xi_2)|. \tag{1.2}$$

For this purpose, I use the Simultaneous Diagonalization Theorem: Let A be a symmetric and positive definite matrix, and B be a symmetric matrix. There exists a non-singular matrix U such that $UAU^T = E$ (the identity matrix), and $UBU^T = D$ (a diagonal matrix with diagonal entries d_i's). Assuming $I(\xi)$ is a $P \times P$ matrix, apply the Simultaneous Diagonalization Theorem to the left-hand side of inequality (1.2)

$$\begin{aligned}
LHS &= \log|(1-\alpha)I(\xi_1)+\alpha I(\xi_2)| + \log|UU^T| - \log|UU^T| \\
&= \log|(1-\alpha)UI(\xi_1)U^T + \alpha UI(\xi_2)U^T| - \log|UU^T| \\
&= \log|(1-\alpha)E + \alpha D| - \log|UU^T| \\
&= \log\prod_{i=1}^{P}(1-\alpha+\alpha d_i) - \log|UU^T| \\
&= \sum_{i=1}^{P}\log(1-\alpha+\alpha d_i) - \log|UU^T| \\
&\geq \sum_{i=1}^{P}((1-\alpha)\log(1)+\alpha\log d_i) - \log|UU^T| \\
&= (1-\alpha)\log|E| + \alpha\sum_{i=1}^{P}\log d_i - \log|UU^T| \\
&= (1-\alpha)\log|E| + \alpha\log|D| - \log|UU^T| \\
&= (1-\alpha)\log|UI(\xi_1)U^T| + \alpha\log|UI(\xi_2)U^T| - \log|UU^T| \\
&= (1-\alpha)\log|I(\xi_1)| + \alpha\log|I(\xi_2)|
\end{aligned}$$

General equivalence theorem

Let ξ be a design defined on the space χ, and let Ψ denote a concave functional of $I(\xi)$. The general equivalence theorem states the following three conditions are equivalent (Atkinson and Donev [1992]).

1. the design ξ^* maximizes $\Psi(I(\xi))$.

2. the design ξ^* minimizes $\max\limits_{x\in\chi} \psi(x,\xi^*)$, where

$$\psi(x,\xi) = \lim_{\alpha\to 0^+} \frac{\partial}{\partial\alpha} \Psi\{I((1-\alpha)\xi + \alpha\delta_x)\} - \Psi\{I(\xi)\} \tag{1.3}$$

is the directional derivative of Ψ at ξ in the direction of δ_x, and δ_x is a measure putting unit mass at x.

3. $\psi(x,\xi^*) \leq 0$ for all $x \in \chi$ with equality at the support points of the design ξ. The inequality is frequently referred as the checking condition for whether ξ^* is optimal or not.

We also note from 3) that for any non-optimal design ξ, $\max\limits_{x\in\chi}\psi(x,\xi) > 0$.

Checking conditions for optimality criteria

We use the equivalence theorem above to verify whether a design ξ is optimal by checking the directional derivative $\psi(x,\xi) \leq 0$ for any $x \in \chi$. Because the criterion is a continuous concave function, the limit in (1.3) exists, and each concave criterion results in a different checking condition as follows.

D-optimality:

$$\psi(x,\xi) = f^T(x)I^{-1}(\xi)f(x) - P; \tag{1.4}$$

A-optimality:

$$\psi(x,\xi) = f^T(x)I^{-2}(\xi)f(x) - \text{trace}[I^{-1}(\xi)]; \tag{1.5}$$

c-optimality:

$$\psi(x,\xi) = (f^T(x)I^{-1}(\xi)\nabla)^2 - \nabla^T I^{-1}(\xi)\nabla. \tag{1.6}$$

As an illustrative example, I show how the directional derivative $\psi(x,\xi)$ is derived for D-optimality. Since the objective function of D-optimality is $\Psi(\text{I}(\xi))$=log$|I(\xi)|$,

and

$$\begin{aligned}
\Psi((1-\alpha)I(\xi)+\alpha I(\delta_x))-\Psi(I(\xi)) &= \log|(1-\alpha)I(\xi)+\alpha I(\delta_x)|-\log|I(\xi)| \\
&= \log|(1-\alpha)I(\xi)I(\xi)^{-1}+\alpha I(\delta_x)I(\xi)^{-1}| \\
&= \log|(1-\alpha)E+\alpha I(\delta_x)I(\xi)^{-1}| \\
&= \log|E+\alpha(I(\delta_x)I(\xi)^{-1}-E)| \\
&= g(\alpha) \\
&= g(0)+g'(\alpha)|_{\alpha=0}+... \\
&= g'(\alpha)|_{\alpha=0}+...
\end{aligned}$$

By the fact that for an arbitrary matrix A,

$$\frac{\partial|E+\alpha A|}{\partial\alpha}=\text{trace}[A],$$

The inequality (1.3) in the equivalence theorem leads to the checking condition for D-optimality by

$$\begin{aligned}
\psi(x,\xi) &= \lim_{\alpha\to 0^+}\frac{1}{\alpha}(\Psi((1-\alpha)I(\xi)+\alpha I(\delta_x))-\Psi(I(\xi))) \\
&= \frac{\partial|E+\alpha(I(\delta_x)I(\xi)^{-1}-E)|}{\partial\alpha} \\
&= \text{trace}[I(\delta_x)I(\xi)^{-1}-E] \\
&= \text{trace}[I(\delta_x)I(\xi)^{-1}]-P,
\end{aligned}$$

and (1.4) follows because $I(\delta_x)=f(x)f^T(x)$.

For a low-dimensional design space, such as when we have only a single agent in the study, the optimality of a design ξ can be easily verified by plotting the graph of $\psi(x,\xi)$ versus $x\in\chi$ and visually examining if $\psi(x,\xi)$ is bounded above by 0 with equality at the support points. Such a graph is called the equivalence plot.

Examples of optimal designs

In this subsection, I show how to use the general equivalence theorem to verify optimality of a few types of design criteria. Assume observations are independent in the following homoscedastic quadratic regression model

$$y_i = f^T(x_i)\beta + \epsilon_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i,$$

where $\epsilon_i \sim Normal(0,1)$, $i = 1, 2, ..., N$.

a) D-optimal design for quadratic regression on design space [-1, 1]

Consider the design

$$\xi = \left\{ \begin{array}{ccc} -1 & 0 & 1 \\ 1/3 & 1/3 & 1/3 \end{array} \right\},$$

which has equal weight at -1,0 and 1. Its normalized information matrix is

$$\begin{aligned} I(\xi) &= \frac{1}{3}\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ &= \frac{1}{3}\begin{bmatrix} 3 & 0 & 2 \\ 0 & 2 & 0 \\ 2 & 0 & 2 \end{bmatrix}. \end{aligned}$$

To verify whether ξ is D-optimal, we calculate

$$\begin{aligned} \psi(x,\xi) &= \begin{bmatrix} 1 & x & x^2 \end{bmatrix} I^{-1}(\xi) \begin{bmatrix} 1 & x & x^2 \end{bmatrix}^T - 3 \\ &= 3\begin{bmatrix} 1 & x & x^2 \end{bmatrix}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0.5 & 0 \\ -1 & 0 & 1.5 \end{bmatrix}\begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} - 3 \\ &= 4.5x^4 - 4.5x^2. \end{aligned}$$

Figure 1.1 shows that ξ is D-optimal because $\psi(x,\xi)$ is bounded above by 0 with equality at its support points -1,0 and 1.

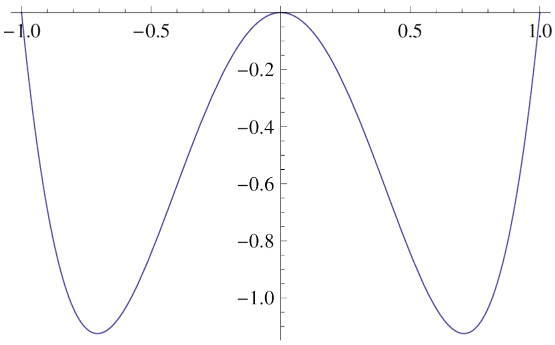


Figure 1.1: The equivalence plot of $\psi(x, \xi)$ for the D-optimal design for the homoscedastic quadratic regression on [-1,1].

b) A-optimal design for quadratic regression on design space [-1, 1]

Consider the design

$$\xi = \left\{ \begin{array}{ccc} -1 & 0 & 1 \\ 1/4 & 1/2 & 1/4 \end{array} \right\}$$

which has $1/4$ weight at points -1 and 1, and puts the rest of the weight at 0. The normalized information matrix for the design is

$$\begin{aligned} I(\xi) &= \frac{1}{4}\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \end{aligned}$$

It is easy to verify that the checking condition (1.5) of A-optimality for the quadratic model is given by

$$\begin{aligned} \psi(x, \xi) &= \begin{bmatrix} 1 & x & x^2 \end{bmatrix} I^{-2}(\xi) \begin{bmatrix} 1 & x & x^2 \end{bmatrix}^T - \text{trace}[I^{-1}(\xi)] \\ &= \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \begin{bmatrix} 8 & 0 & -12 \\ 0 & 4 & 0 \\ -12 & 0 & 20 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} - 8 \\ &= 20x^4 - 20x^2. \end{aligned}$$

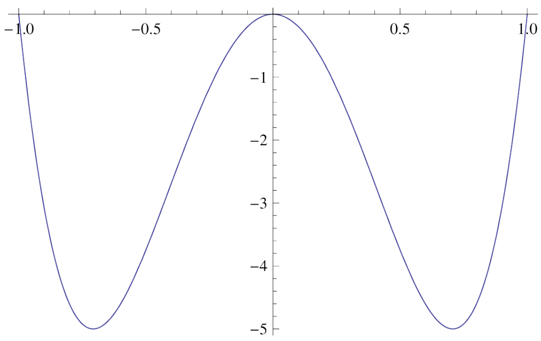


Figure 1.2: The equivalence plot of $\psi(x,\xi)$ for the A-optimal design for the homoscedastic quadratic regression on [-1,1].

The plot of $\psi(x,\xi)$ for the A-optimal design for quadratic regression is shown in Figure 1.2, which confirms ξ is A-optimal since $\psi(x,\xi)$ is bounded above by 0 with equality at the set of support points {-1,0,1}.

Efficiency and efficiency lower bound

When the optimal design is known, we can measure the quality of an arbitrary design ξ by comparing it to the optimal design. For example, assume the D-optimal design is ξ_D for a model with P unknown parameters in the mean function. For an arbitrary design ξ, its D-efficiency is defined by

$$D_{eff} = (\frac{|I(\xi)|}{|I(\xi_D)|})^{1/P}.$$

When comparing to the c-optimal design ξ_c for estimating $c(\theta)$, its c-efficiency is defined by

$$c_{eff} = \frac{\nabla^T I^{-1}(\xi_c)\nabla}{\nabla^T I^{-1}(\xi)\nabla},$$

where ∇ is the partial derivative of $c(\theta)$. Note that if a design ξ has an efficiency of $\frac{1}{k}$, it means we need k replicates of the design ξ to achieve the same criterion value as one replicate of the optimal design.

Working with approximate designs has many advantages. One of them is that if a design is not optimal, one can also use convex analysis results to ascertain how close the design is to the optimum without knowing the latter by means of the efficiency lower bound. Details can be found in standard design monographs such as Fedorov [1972], Silvey [1980] and Berger and Wong [2009]. Define Ξ the set of all designs for a regression model on χ. For an arbitrary design $\xi \in \Xi$, let $\Psi(I(\xi))$ be the user-selected concave functional, δ_x be the one-point design at x, and $\psi(x,\xi)$ be the directional derivative of $\Psi(I(\xi))$ at ξ in the direction of δ_x. Proposition IV. 28 in Pazman [1986] (page 118) shows that if $\Psi(I(\xi)) > -\infty$ and $\max_{x\in\chi}\psi(x,\xi) = \zeta$ for some $\zeta > 0$, then

$$\Psi(I(\xi)) \geq \sup\{\Psi(I(\xi)) : \xi \in \Xi\} - \zeta.$$

When a design ξ is not optimal, its closeness to the optimal design can be readily assessed. For example, recall that for D-optimality, the directional derivative at ξ in the direction of δ_x is

$$\psi(x,\xi) = f^T(x)I^{-1}(\xi)f(x) - P,$$

where P is the number of unknown parameters in the mean function. By Proposition IV. 28, since D-optimal design ξ_D satisfies $\Psi(I(\xi_D)) = \sup\{\Psi(I(\xi)) : \xi \in \Xi\}$, for any design ξ, we have $\log|I(\xi)| \geq \log|I(\xi_D)| - \log e^{\zeta}$ and it follows that

$$D_{eff}(\xi) \geq e^{-\frac{\zeta}{P}} = D_{lb}(\xi),$$

where $\zeta = \max_{x\in\chi} \text{trace}[I(\delta_x)I^{-1}(\xi)] - P$. This D-efficiency lower bound $D_{lb}(\xi)$ attains unity if and only if $\zeta = 0$, whereupon ξ is D-optimal.

Local optimality

In all previous sections, I review the concepts and fundamental theory of optimal designs for linear models. All the results can be readily extended to nonlinear models. The main difference is that in a nonlinear model, the information matrix depends on the values of P unknown parameters $\theta = (\theta_1, ..., \theta_P)^T$ in the mean function $\eta(x, \theta)$. The usual and simplest approach is to assume nominal values θ^0 for θ are available. Nominal values are typically obtained from pilot studies or data from similar experiments. Then $\eta(x, \theta)$ is linearized approximately by a first order Taylor's expansion at θ^0 as follows,

$$\mathrm{E}(y) = \eta(x, \theta) \doteq \eta(x, \theta^0) + \frac{\partial \eta}{\partial \theta}|_{\theta^0}(\theta - \theta^0).$$

Note that throughout my dissertation, I adopt the numerator layout notation in the derivative of η with respect to θ, which means $\frac{\partial \eta}{\partial \theta} = (\frac{\partial \eta}{\partial \theta_1}, ..., \frac{\partial \eta}{\partial \theta_P})$ is a row vector. With $\beta = \theta - \theta^0$, an optimal design ξ for estimating β, or equivalently, θ can be constructed using the following linear model

$$\mathrm{E}(y) - \eta(x, \theta^0) \doteq f^T(x, \theta^0)\beta,$$

where $f^T(x, \theta_0) = \frac{\partial \eta}{\partial \theta}|_{\theta^0}$. The normalized information matrix from the design ξ for $\eta(x, \theta)$ becomes

$$I(\xi, \theta^0) = \mathrm{E}_\xi\, f(x, \theta^0) f^T(x, \theta^0),$$

and so depends on the nominal values θ_0. Therefore the desired optimal design for nonlinear models can be found using the same method as if we have a linear model after we replace θ in the gradient of $\eta(x, \theta)$ by θ^0. Because such optimal designs depend on the nominal values θ^0, they are termed locally optimal by Chernoff [1953]. All design criteria and checking conditions apply to nonlinear models after replacing the mean function in the nonlinear model by its gradient.

Example: Compartmental model Compartmental models are commonly used to study the drug movement through the body. For a single observation x, a one-compartment open model is represented by

$$\eta(x, \theta) = \theta_3(\exp(-\theta_2 x) - \exp(-\theta_1 x)), \quad x > 0.$$

with $\theta_1 > \theta_2 > 0$ and $\theta_3 > 0$.

Let $\theta = (\theta_1, \theta_2, \theta_3)^T$ be the vector of model parameters, and let $\theta^0 = (\theta_1^0, \theta_2^0, \theta_3^0)^T$ be the nominal values for θ. A direct calculation shows the gradient of the approximated mean function at the point x is

$$f^T(x, \theta^0) = (\frac{\partial\eta(x,\theta)}{\partial\theta_1}, \frac{\partial\eta(x,\theta)}{\partial\theta_2}, \frac{\partial\eta(x,\theta)}{\partial\theta_3})|_{\theta^0},$$

with

$$\begin{aligned}\frac{\partial\eta(x,\theta)}{\partial\theta_1} =& x\theta_3\exp(-\theta_1 x)\\ \frac{\partial\eta(x,\theta)}{\partial\theta_2} =& -x\theta_3\exp(-\theta_2 x)\\ \frac{\partial\eta(x,\theta)}{\partial\theta_3} =& \exp(-\theta_2 x) - \exp(-\theta_1 x).\end{aligned}$$

For a K-point design ξ with weights $w_1, \ldots, w_K$ as the proportions of observations taken at support points $x_1, ..., x_K$, its normalized information matrix is $I(\xi, \theta) = \sum_{i=1}^K w_i f(x_i, \theta^0) f^T(x_i, \theta^0)$, apart from an unimportant multiplicative constant. Note that in the rest of my dissertation, I will write θ instead of θ^0 as nominal values when there is no possible confusion.

1.3 Search algorithms

Optimization algorithms have been developed for several decades. A detailed early history of optimization algorithms was discussed in Yang [2010c]. In the dissertation, I briefly discuss two main categories of optimization algorithms: deterministic and stochastic algorithms. For all algorithms I discussed here, the common stopping criteria include the maximal number of iterations is reached, or the generated solution is close enough to the optimum. Specifically, when finding a D-optimal design, I set the stopping criterion as the D-efficiency lower bound $D_{lb}(\xi) > 0.999$ for a generated design ξ. For c-optimality, I set the stopping criterion as $\zeta \leq 10^{-4}$ for the generated design ξ, where $\zeta = \max_{x\in\chi}(f^T(x)I^{-1}(\xi)\nabla)^2 - \nabla^T I^{-1}(\xi)\nabla$, and ∇ is the gradient of the combination of parameters of interest.

1.3.1 Deterministic algorithms

An algorithm is said to be deterministic if "at any point of execution there is, at most, one possible way to proceed" (Ueberhuber [1997]). In other words, a deterministic algorithm is rigorous and specific in the sense that the algorithm always goes through the same path given a particular input; thus all the states and values of the functions are repeatable. Deterministic algorithms include many conventional algorithms such as linear programming, nonlinear programming, gradient-based and gradient-free algorithms (Parsopoulos and Vrahatis [2010]). Typically, the deterministic algorithm for finding optimal designs works with a finite design space $\chi = \{x_1, x_1, ..., x_n\}$ with the n points discretized in some way from the original continuous design space. The goal of the algorithm is to optimize a vector of weights $w \in \Omega = \{w = (w_1, w_2, ..., w_n) : \sum_{i=1}^{n} w_i = 1, w_i \geq 0\}$. The value w_i is the proportion of observations taken at x_i. It is worth noting that the number of design points n is user-specified depending on the complexity of the design criterion and the model. Specifically in describing deterministic algorithms, I use $w^{(t)}$ to denote the weights vector of an approximate design on the discretized design space χ at the t^{th} iteration.

1.3.1.1 Vertex direction method (VDM)

This deterministic algorithm was proposed by Fedorov [1972] and Wynn [1972], and later named vertex direction method by Wu [1978]. For a linear regression model (1.1) with P unknown parameters, the steps of the algorithm are as follows (Tsay [1976]):

1) Choose a starting design $w^{(0)} = (w_1^{(0)}, w_2^{(0)}, ..., w_n^{(0)})$ with $n > P$ points such that $I(w^{(0)})$ is nonsingular. A common choice of w_0 is to have equal weight at all the n design points, and all weights add up to 1.

2) For the current design $w^{(t)} = (w_1^{(t)}, w_2^{(t)}, ..., w_n^{t)})$, $t = 0, 1, 2...$, find

$$i_{max} = \underset{i}{\text{argmax}}\ d(i, i, w^{(t)}), \ \ 1 \leq i \leq n$$

where $d(i, j, w^{(t)}) = f^T(x_i)I^{-1}(w^{(t)})f(x_j)$. Then the new design at the $t + 1^{th}$ iteration is given by

$$w_i^{(t+1)} = \begin{cases} (1 - \kappa^{(t)})w_i^{(t)} & i \neq i_{max}, \\ \kappa^{(t)} + (1 - \kappa^{(t)})w_i^{(t)} & i = i_{max}, \end{cases}$$

where $\kappa^{(t)} = \frac{d(i_{max}, i_{max}, w^{(t)})/P - 1}{d(i_{max}, i_{max}, w^{(t)}) - 1}$.

3) Repeating step 2) when the stopping criterion is not met, we obtain a sequence of designs $\{w^{(t)}\}$. Fedorov [1972] proved $|I(w^{(t)})| \to \sup_{w \in \Omega}|I(w)|$. By the equivalence theorem, $\lim_{t \to \infty} d(i, i, w^{(t)}) = P$ for arbitrary $1 \leq i \leq n$ and this means we can stop iterations when $d(i, i, w^{(t)})$ gets close enough to P.

This algorithm shares the same idea with steepest-ascent method, because $\kappa^{(t)}$ moves $w^{(t)}$ towards “the direction corresponding to the largest derivative in any direction”. A drawback of this algorithm is $w^{(t)}$ may end up with too many support points as iterations progress, and they need to be carefully collapsed to fewer points and restart the algorithm.

1.3.1.2 Cocktail algorithm (CA)

The cocktail algorithm was proposed by Yu [2011] for finding D-optimal designs. The name cocktail originates from the hybrid of three algorithms that have been mostly used for finding D-optimal designs: Multiplicative algorithm(MA), Vertex direction method (VDM) and Vertex exchange method (VEM). All three algorithms search for D-optimal designs on a discretized design space $\chi = \{x_1, ..., x_n\}$. Yu [2011] showed that CA performs much faster than each of its component algorithms for several univariable model. CA also inherits the monotonic convergence property from three algorithms, i.e. as $t \to \infty$, $|I(w^{(t)})| \to \sup_{w\in\Omega}|I(w)|$. Below is the description of three algorithms.

- Vertex direction method (VDM)

As described in section 1.3.1.2, for the current design $w = (w_1, w_2, ..., w_n)$, I define $w^{new} = VDM(w)$ by

$$w_i^{new} = \begin{cases} (1-\kappa)w_i & i \neq i_{max}, \\ \kappa + (1-\kappa)w_i & i = i_{max}, \end{cases}$$

where $\kappa = \frac{d(i_{max}, i_{max}, w)/P - 1}{d(i_{max}, i_{max}, w) - 1}$, and $i_{max} = \underset{i}{\text{argmax}}\ d(i, i, w^{(t)}),\ \ 1 \leq i \leq n.$

- Nearest neighbor exchange (NNE)

The NNE algorithm is similar to the Vertex exchange method (VEM) developed by Bohning (1986). It aims to reallocate weights of nearby points in the neighborhood. Note that it operates on support points with positive weight only. For univariable models, there is a natural order in the finite design space

$$\chi = \{x_i, i = 1, ..., n\}.$$

For the j^{th} point, we find its adjacent point $j^* \in \{1, ..., n\}$, such that their L^1-norm $||x_j - x_{j^*}||$ is minimized. For the current design $w = (w_1, w_2, ..., w_n)$, NNE is to perform weight exchange between j^{th} and j^{*th} support points by

$$w_i^{new} = \begin{cases} w_i & i \neq \{j, j^*\}, \\ w_i - \tau & i = j, \\ w_i + \tau & i = j^*, \end{cases}$$

where $\tau = \min\{w_j, \max\{-w_{j^*}, \tau^*(j, j^*)\}\}$, and

$$\tau^*(j, j^*) = \frac{d(j^*, j^*, w) - d(j, j, w)}{2(d(j, j, w)d(j^*, j^*, w) - d^2(j, j^*, w))}$$

where $d(j, j^*, w) = f^T(x_j)I^{-1}(w)f(x_{j^*})$. After iterating j from 1 to n design points in w, the new design w^{new} is generated. I denote the entire procedure as $w^{new} = NNE(w)$.

- Multiplicative algorithm (MA)

The MA was proposed by Silvey et al. [1978]. Its updating rule for the current design $w = (w_1, w_2, ..., w_n)$, denoted by $w^{new} = MA(w)$, is

$$w_i^{new} = \frac{d(i, i, w)}{P} w_i, i = 1, ..., n$$

The CA algorithm consists of the following steps.

1) Choose a starting $n > P$ points design $w^{(0)}$ such that $I(w^{(0)})$ is non-singular. A common choice of w_0 is to assign equal weight to the n distinct design points, and all the weights add up to 1.

2) In the t^{th} iteration, there are three sub-steps, denoted by $t + \frac{1}{3}$, $t + \frac{2}{3}$, $t + 1$ $(t = 0, 1, 2...)$ as follows

2.1) Generate $w^{(t+\frac{1}{3})} = VDM(w^{(t)})$;

2.2) Exclude non-support points in $w^{(t+\frac{1}{3})}$, and re-index the n_+ design points with positive weight $w^{(t+\frac{1}{3})}$ from (1) to (n_+). Generate $w^{(t+\frac{2}{3})} = NNE(w^{(t+\frac{1}{3})})$;

2.3) Exclude non-support points in $w^{(t+\frac{2}{3})}$, and re-index the n_+ design points with positive weight $w^{(t+\frac{2}{3})}$ from (1) to (n_+). Generate $w^{(t+1)} = MA(w^{(t+\frac{2}{3})})$.

3) Repeat 2) until $\max_{i\in\{1,...,n_+\}} d((i), (i), w^{(t)})$ gets close enough to P, or the maximal number of iterations is reached.

1.3.2 Stochastic algorithms

Stochastic algorithms have random components in their search for the optimum using a heuristic (or metaheuristic). They are different from deterministic algorithms because randomization, a key feature of stochastic algorithms, makes each step of a stochastic algorithm unrepeatable. There are two types of stochastic algorithms: heuristic and metaheuristic, though their difference is minor and some literature has used these two terms interchangeably. "Heuristic is a solution strategy by trial-and-error to produce acceptable solutions to a problem in a reasonably practical time" (Yang [2010c]). Recently, metaheuristic algorithms refer to all stochastic algorithms with randomization and local search.

As early as 1940, heuristic algorithm was first used by Alan Turing in the Second World War to break the German Enigma ciphers (Yang [2010c]). Since 1960s, there have been an explosive development in stochastic algorithms. In my dissertation, I use these metaheuristic algorithms to maximize an objective function $h(\cdot)$. Unlike deterministic algorithms I have described early on, metaheuristic algorithms usually do not require $h(\cdot)$ to be convex/concave nor differentiable (Storn and Price [1997]). Such an advantage enables them to cope with much broader types of optimization problems. In the following sections, I introduce a couple of important metaheuristic algorithms in their chronological order.

1.3.2.1 Genetic algorithm (GA)

The Genetic algorithm was proposed by John Holland and his students in 1960s and first summarized in his book: Adaptation in Natural and Artificial systems (Holland [1975]). Based on Charles Darwin's theory of natural selection, it mimics the biological evolution by employing the idea of "crossover and recombination, mutation, and selection" to artificial systems. Over the years, researchers have proposed many variations based on the canonical GA, and successfully applied them to solve problems such as graph coloring, pattern recognition, traveling salesman problem and multi-objective engineering optimization, etc. An advantage of GA is its multiple offspring enables parallelism, which enhances the search speed enormously. However, GA is sensitive to the selection of important parameters and an improper choice of parameters may lead to non-convergence or unreasonable results. GA can be summarized in the following steps (Yang [2010c]):

1) Define a fitness function or selection criterion $h(\cdot)$.

2) Initialize a population of individuals $z_1^{(0)}, ..., z_{n_{flock}}^{(0)}$. Each individual represents a chromosome, which is a potential solution to the optimization problem.

3) For the current t^{th} population, $t = 0, 1, ...$, evaluate the fitness of all the individuals in the population $h(z_1^{(t)}), ..., h(z_{n_{flock}}^{(t)})$,

4) Create a new population $z_1^{(t+1)}, ..., z_{n_{flock}}^{(t+1)}$ by performing crossover, and mutation, fitness proportionate reproduction, etc.;

5) Repeat step 3) and 4) and evolve the population until the stopping criterion is met.

1.3.2.2 Simulated Annealing (SA)

Simulated Annealing (SA) algorithm was developed by Kirkpatrick et al. [1983]. As the name implies, SA was inspired by the annealing process when cooling a metal into a crystalline state with minimum energy. The law of thermodynamics states that at temperature Υ the probability of a decrease in energy level Δh is given by

$$\Pr(\Delta h){=}\exp(-\frac{\Delta h}{k_B \Upsilon}),$$

where k_B is a constant known as Boltzmann's constant, Υ is the temperature for controlling the annealing process, and Δh represents the change of energy levels. By introducing $\Pr(\Delta h)$ into the algorithm, SA reflects the idea of Metroplis algorithm (details of the algorithm can be found in Metropolis et al. [1953]), which not only allows improvement of the objective function but also accepts non-ideal changes with some probability. An attractive feature of SA is that the convergence to its global optimal solution is guaranteed if enough randomness is used in combination with very slow cooling (Yang [2010c]). SA has been widely used for solving different areas of optimization problems. For example, Haines

[1987] applied SA to find exact optimal designs for linear models. Selima and Alsultanb [1991] showed SA outperforms $K-$means algorithms for solving the clustering problem.

The procedure of SA to maximize an objective function $h(\cdot)$ is defined as follows.

1) Initialize temperature Υ_0 and starting point $z^{(0)}$ at time 0, which represents a candidate solution. Evaluate the fitness $h(z^{(0)})$.

2) Define annealing schedule i.e how temperature $\Upsilon(t)$ decreases as a function of time $t = 0, 1, 2...$

3) Move $z^{(t)}$ to a new location $z^{(t+1)}$ randomly, calculate $h(z^{(t+1)})$ and the energy level change $\Delta h^{(t+1)} = h(z^{(t)}) - h(z^{(t+1)})$. In my dissertation, the goal is to maximize the objective function $h(\cdot)$, therefore if $\Delta h^{(t+1)} < 0$ then $z^{(t+1)}$ is better and we accept it. If $\Delta h^{(t+1)} > 0$ then accept $z^{(t+1)}$ with probability $\exp(-\frac{\Delta h^{(t+1)}}{\Upsilon})$. Note that in SA, Boltzmann's constant k_B is usually dropped since it was introduced to cope with different materials.

4) Repeat step 3) to update $z^{(t)}$ and $h(z^{(t)})$ until $\Upsilon \to 0$.

In SA, it is crucial to choose an appropriate decreasing mode for Υ. The higher the temperature, the larger the probability of accepting a worse position. Two commonly used annealing schedules (or cooling schedules) are:

linear: $\Upsilon(t) = \Upsilon_0 - \beta t$

geometric: $\Upsilon(t) = \Upsilon_0 \alpha^t$, $0 < \alpha < 1$

1.3.2.3 Differential Evolution (DE)

The differential evolution algorithm was developed by Storn and Price [1997]. It searches global optimum in a continuous space by utilizing a population of J-dimensional vectors $z_1, ..., z_{n_{flock}}$. The theoretical framework of DE is relatively simple and requires only a few tuning parameters, but it performs well in convergence in many applications (Kachitvichyanukul [2012]). Storn and Price [1997] claimed that DE, based on simulation results using multiple testing functions, outperforms other popular metaheuristic algorithms including Adaptive Simulated Annealing (ASA), the Annealed Nelder and Mead approach (ANM), the Breeder Genetic Algorithm (BGA), the EASY Evolution Strategy and the method of Stochastic Differential Equations. Other researchers have also shown DE outperforms GA in solving many engineering problems such as the design of scannable circular antenna arrays (Panduro et al. [2009]), enhancement of total transfer capability (Chandrasekar and Ramana [2012]), etc. The DE algorithm codes and some applications are available in various programming platforms at `http://www1.icsi.berkeley.edu/~storn/code.html`.

The algorithm consists of the following steps.

1) Initialize a population of J-dimensional candidate vectors $z_i^{(0)} = (z_{i1}^{(0)}, z_{i2}^{(0)} ..., z_{iJ}^{(0)})$, and evaluate the fitness $h(z_i^{(0)})$, $i = 1, 2, ..., n_{flock}$.

2) At t^{th} iteration ($t = 0, 1, 2, ...$), update the current generation $z_i^{(t)}$,$i = 1, 2, ..., n_{flock}$ by the following three sub-steps:

- *Mutation*

For each vector $z_i^{(t)}, i = 1, 2, ..., n_{flock}$, generate a mutant vector by

$$m_i^{(t+1)} = z_{r_a}^{(t)} + CF \times (z_{r_b}^{(t)} - z_{r_c}^{(t)})$$

where r_a, r_b, r_c are integers randomly drawn without replacement from $\{1, 2, ..., n_{flock}\}$, and $CF \in [0, 2]$ is a constant factor controlling the amplification of the differential variation $z_{r_b}^{(t)} - z_{r_c}^{(t)}$. A large CR often speeds up convergence, but a smaller CR enables DE to search the space in a more detailed manner.

- *Crossover*

To increase the diversity of vectors, the crossover trial vector is introduced as: $c_i^{(t+1)} = (c_{i1}^{(t+1)}, c_{i2}^{(t+1)}, ..., c_{iJ}^{(t+1)})$, where

$$c_{ij}^{(t+1)} = \begin{cases} m_{ij}^{(t+1)} & \text{if } u_j \leq CR \text{ or } j = r_i \\ z_{ij}^{(t)} & \text{if } u_j > CR \text{ and } j \neq r_i \end{cases},$$

$j = 1, 2, ..., D$. Here u_j is a random number which follows uniform distribution in [0,1], and CR is the user determined crossover constant$\in [0, 1]$. For the i^{th} vector, a number r_i is randomly chosen from $1, 2, ..., J$ which ensures that $c_i^{(t+1)}$ gets at least one element from $m_{ij}^{(t+1)}$.

- *Selection*

Assuming $h(\cdot)$ is the objective function to be maximized, the i^{th} vector at $t + 1^{th}$ generation is defined by

$$z_i^{(t+1)} = \begin{cases} c_i^{(t+1)} & \text{if } h(z_i^{(t)}) < h(c_i^{(t+1)}) \\ z_i^{(t)} & \text{if } h(z_i^{(t)}) \geq h(c_i^{(t+1)}) \end{cases}$$

3) Repeat step 2) until stopping criteria are satisfied.

The only two tuning parameters in DE is CF and CR. Through simulation studies of a number of testing functions, Storn and Price [1997] suggested that CF=0.5 is usually a good initial choice, and a relatively large CR, say 0.9, is appropriate as a first try in order to see if a quick solution is possible. In terms of the total population size, Storn and Price [1997] recommended a reasonable choice for n_{flock} is between $5D$ and $10D$ according to their experience.

1.3.2.4 Particle Swarm Optimization (PSO)

Nature-inspired metaheuristic algorithms have been gaining popularity in the last two decades and recently have gained dominant status both in academia and industrial applications (Whitacre [2011a,b]). One of the most prominent examples of a nature-inspired algorithms is Particle Swarm Optimization (PSO) proposed by Kennedy and Eberhart [1995]. A main appeal is its simplicity, ease of implementation and its ability to provide quality solution to an optimization problem very fast. Another appeal is that it makes no assumption on the function to be optimized and researchers from a widening range of disciplines report high success rates of finding an optimal or nearly optimal solution to a complex optimization problem. An attempt to explain its success from a biological viewpoint is given in Garnier et al. [2007], and an overview of PSO is available in Poli et al. [2007].

The concept of particle swarm optimization stems from studying interactions in a simplified social system (Eberhart and Shi [2001]). Particles in PSO are usually described as mimicking the social behavior of a flock of birds in search of food. The PSO algorithm begins after the user inputs two key starting values including (i) the flock size and (ii) maximum number of iterations. An initial flock of particles is randomly generated and they represent candidate designs for the optimum, each with the same user-specified number of points.

There are two basic equations that drive movement for the each particle in the PSO search to optimize an objective function $h(\cdot)$. The i^{th} particle at time t is determined by its position through the vector $z_i^{(t)}$ and its movement by the velocity vector $v_i^{(t)}$. At time $t+1$, its position $z_i^{(t+1)}$ and velocity $v_i^{(t+1)}$ are evolved by the following two equations:

$$v_i^{(t+1)} = \omega^{(t)} v_i^{(t)} + \gamma_1 u_{1i}^{(t)} \odot (p_i^{(t)} - z_i^{(t)}) + \gamma_2 u_{2i}^{(t)} \odot (p_g^{(t)} - z_i^{(t)}), \tag{1.7}$$

and

$$z_i^{(t+1)} = z_i^{(t)} + v_i^{(t)}. \tag{1.8}$$

The initial velocity for each particle is randomly assigned from $U(0,1)$. The inertia weight $\omega^{(t)}$ modulates the influence of the former velocity. Throughout the dissertation, I follow the suggestion in Eberhart and Shi [2000] and set $\omega^{(t)}$ to be a linear function that decreases from 0.9 to 0.4 during an optimization run.

Such setting helps particles to explore the entire space during the early part of the search and exchange information before converging to the global optimum. Particles therefore have less chance to converge prematurely or get trapped in the local optimum. In (1.7) the personal best position $p_i^{(t)}$ is the best position found by the i^{th} particle up to time t, which conceptualizes self-knowledge. The global best position $p_g^{(t)}$ is the overall best position determined by the whole flock after sharing information up to time t, which is referred to as social knowledge. This means that up to time t, the personal best for particle i is $pbest_i = h(p_i^{(t)})$ and the population best for the entire flock is $gbest = h(p_g^{(t)})$. The two random vectors in the PSO algorithm are $u_{1i}^{(t)}$ and $u_{2i}^{(t)}$, and usually their components are taken to be independent random variables from $U(0,1)$. The constant γ_1 and γ_2 are constants that determine how each particle moves toward its own personal best position $p_i^{(t)}$ and in the direction of the global best position $p_g^{(t)}$. The default values for these two constants in the PSO codes are $\gamma_1 = \gamma_2 = 2$ and they really seem to work well in practice for all univariable problems that I have investigated so far. Note that in (1.7) the product in the last two terms is Hadamard product.

The PSO generic code is available on many websites such as `http://www.swarmintelligence.org` and in books on metaheuristic methods like Yang [2010b]. MATLAB also has a toolbox for running the PSO code (`http://www.mathworks.com/matlabcentral/fileexchange/7506`). The pseudo code for the PSO procedure for a flock of size n_{flock} is as follows.

```
(1) Initialize particles
    (1.1) Initiate positions $z_i$ and velocities $v_i$ for $i = 1, \ldots, n_{flock}$.
    (1.2) Calculate the fitness values $h(z_i)$ for $i = 1, \ldots, n_{flock}$.
    (1.3) Determine the personal best positions $p_i = z_i$ and the global position $p_g$.
(2) Repeat until the stopping criterion is satisfied.
    (2.1) Calculate particle velocity according to Equation (1.7).
    (2.2) Update particle position according to Equation (1.8).
    (2.3) Calculate the fitness values $h(z_i)$.
    (2.4) Update personal and global best positions $p_i$ and $p_g$.
(3) Output $p_g = \arg\max h(z)$ with $gbest = h(p_g)$.
```

CHAPTER 2

Optimal Designs for Univariable Biomedical Models

In this chapter, I discuss the statistical background and apply PSO to find D- and c-optimal designs for common univariable models in the biomedical studies. By univariable I mean there is only one dependent variable (response variable) and one independent variable (explanatory variable) in the model. These models may appear small in terms of the number of parameters that they have. However, as noted in Konstantinou et al. [2011], finding optimal designs for such models can still be problematic using traditional numerical methods or analytically. The statistical models have a known mean response function $\eta(x, \theta)$ apart from the values of the parameter vector θ.

In what follows, I present D- and c-optimal designs found by PSO for the following model: (i) an one-compartment model used in pharmacokinetics, (ii) 2 and 3-parameter logistic models for studying binary responses, (iii) a double exponential model for studying tumor regrowth rate, (iv) a 2-parameter survival model, (v) an inverse polynomial model and (vi) a 4-parameter heteroscedastic Hill model. These models are selected to facilitate comparisons with known results from the literature. Other than the 4-parameter heteroscedastic Hill model, all have independent normal errors, each with mean zero and constant variance.

2.1 Motivation

Sheiner et al. [1989] pointed out that "a dose-ranging study must begin with a parametric model for patient-specific dose-response curves". In biomedical research such as dose-response studies, nonlinear univariable models, including the compartmental model and the Hill model, are widely used. By postulating these models, practitioners try to answer fundamental questions such as what is the nature of the dose-response relationship? What is the optimal dose (Ruberg [1995])? The answers to these questions usually rely on the precise estimation of the unknown parameters in the model. As the cost of biomedical researches rises nowadays, it is crucial to efficiently estimate these quantities by choosing multiple dose levels via D- or c-optimal designs.

Optimal experimental design theory has been developed extensively for almost a century, and they can provide insightful guidance on how to allocate multiple dose levels. As more and more complicated models are developed, finding optimal designs has become an increasingly difficult task for researchers, therefore the availability of an efficient and easy-to-use algorithm to find different types of optimal design is both useful and timely. Recently, metaheuristic algorithms including Particle swarm optimization (PSO) have been successfully applied to solve many tough engineering and computer science problems. These algorithms do not guarantee that the global optimum can be always found, but frequently converge to the optimum or near optimum after a few iterations. In the context of my work, all the objectives are concave functionals and so I can easily verify if the PSO-generated design is global optimum using the equivalence theorem. In what is to follow, much of the work is on applications of PSO to find a variety of optimal designs for models commonly used in biomedical science.

2.2 Implementation of Particle Swarm Optimization to find locally optimal designs

I have described of PSO in Chapter 1. When applying PSO to find a K-point optimal design

$$\xi = \left\{ \begin{array}{cccc} x_1 & x_2 & ... & x_K \\ w_1 & w_2 & ... & w_K \end{array} \right\},$$

the user first selects the flock size, the maximal number of iterations and parameters for the design problem. The latter includes the lower bound lb and upper bound ub of the design space and nominal values of model parameters for the particular problem at hand. PSO begins searching by randomly generating a flock with each particle representing a candidate design ξ. These candidate designs all have the same fixed number of points, K is equal to or greater than the number of parameters in the mean response function.

For the i^{th} particle, its position at time t , i.e. a candidate design, is $z_i^{(t)} = (x_1, x_2, ..., x_K, w_1, w_2, ..., w_{K-1})^T$. There are two constraints imposed on each particle: $x_i \in [lb, ub]$ and $\sum_{i=1}^{K-1} w_i \leq 1$. If a candidate design doesn't satisfy the two constraints, we punish this position $z_i^{(t)}$ by assigning an extremely small value -10^{10} to its fitness value $h(z_i^{(t)})$, eliminating such an infeasible solution. For particles that fly outside of the design space, some repair mechanisms will be applied. I elaborate on them in Chapter 5, where I compare how different repair mechanisms impact the performance of PSO.

The rest of the PSO parameters like the inertia weight, cognitive and social learning parameters are all set to their default values. Consequently, we only have to fuss with the flock size and the number of iterations. This simplifies the process and is an appealing feature of PSO. Since in our examples, the design optimality criterion is concave, the generated design can be verified using the general equivalence theorem derived from the directional derivative of the criterion functional. We therefore check the quality of the PSO generated design either by the equivalence plot or calculating its efficiency lower bound.

The number of support points required by an optimal design is unknown. It is possible that repeated searches by PSO with different number of iterations and different flock sizes generate a K-point design that does not meet the equivalence theorem conditions. That means the optimal design is supported at more than K points and so we continue the search to all designs with $K + 1$ points. My experience is that such a strategy always produce a locally optimal design and I only need to search among designs with $K+j$ points where j is usually 1 or 2. On the other hand, if the number of support points required by the optimal design is over-specified, then PSO will report an optimal design with some identical points or some points with extremely small weights. The weights at these identical points are then summed to obtain the optimal design. Such situations arise when the optimal design has a singular information matrix and an example of such a case is provided below when we want to find the locally optimal design for estimating the area under curve (AUC) in the 3-parameter compartmental model.

In my dissertation, All PSO codes are programmed in MATLAB 2013a, and run on a workstation with Intel i7-4770 and 16GB ram.

2.3 Locally D-optimal designs for compartmental models

Compartmental models are widely used to model transport of material in biological systems. Teorell [1937] was one of the first researchers to use compartmental models to investigate "*in vivo* absorption, distribution, metabolism, and excretion" of a drug. Dominguez and Pomerene [1945] used a one-compartment open model to describe the time course of absorption rate of creatinine from both plasma and urinary excretion data. Such model was also applied by Borzelleca and Lowenthal. [1966] in a pharmacokinetic study for the rectal administration of a drug. Compartmental models are not limited to pharmaceutical applications. In veterinary science, for example, Fresen [1984] used a 3-paramter compartmental model to study the effect of theophylline on horses.

To illustrate how to find a locally D-optimal design via PSO, I focus on the one-compartment open model with first-order absorption input, as described by Atkinson et al. [1993]

$$\eta(x, \theta) = \theta_3(\exp(-\theta_2 x) - \exp(-\theta_1 x)),\ x > 0.$$

with $\theta_1 > \theta_2 > 0$ and $\theta_3 > 0$. As shown in section 1.2.4.7, the gradient of $\eta(x, \theta)$ is

$$f^T(x, \theta) = (x\theta_3\exp(-\theta_1 x), -x\theta_3\exp(-\theta_2 x),\exp(-\theta_2 x) - \exp(-\theta_1 x)).$$

Hereby the normalized information matrix is $I(\xi, \theta) = \sum_{i=1}^{K} w_i f(x_i, \theta) f^T(x_i, \theta)$, which depends on nominal values of $\theta = (\theta_1, \theta_2, \theta_3)^T$. To implement PSO to find locally D- and c-optimal design for the 3-parameter compartmental model, I assume for comparison purposes the same dose interval and the same set of nominal parameters used in Atkinson and Donev [1992] with $\theta_1 = 4.298$, $\theta_2 = 0.05884$ and $\theta_3 = 21.8$. These values were obtained from the least square estimates of the parameters from Fresen's data set (Fresen [1984]).

As a first attempt, I use 100 iterations and a flock size of 100 particles each with 3 support points in the PSO algorithm to find the locally D-optimal design for this compartmental model. This is a five-dimension optimization problem where we want to find the best choices of x_1, x_2, x_3, w_1, w_2 to maximize $|I(\xi, \theta)|$. The constraints I put on each particle are $0 < x_1 < x_2 < x_3 < 30$, $0 < w_1 < 1, 0 < w_2 < 1, 0 < w_3 = 1 - w_1 - w_2$. The PSO generated design is equally supported at 0.2288, 1.3886 and 18.4168, which coincides with the locally D-optimal design to four decimal places in Atkinson and Donev [1992] on page 264. The equivalence plot confirms the optimality of the design over all designs on the designated dose interval and I do not need to search further. The CPU time required was 0.408 seconds. If I have used just 50 particles and 50 iterations, the same design would also have been obtained in 0.381 seconds.

2.4 Locally c-optimal designs for estimating the time to maximum concentration in compartmental models

Three common study features of a drug are its time to maximum concentration in the targeted compartment, time to maximum concentration in the compartment and the average time it spends inside the compartment. I discuss only the latter two objectives for space consideration because finding optimal design for the first objective can be carried out in a similar way. The locally c-optimal design for estimating the time to maximum concentration of a drug can be found by differentiation directly from the model. This time as a function of the model parameters is

$$t_{max}(\theta) = \frac{\log\theta_1 - \log\theta_2}{\theta_1 - \theta_2}.$$

The goal then is to choose a design to minimize the asymptotic variance of the estimated time given by

$$\nabla^T I(\xi, \theta)^- \nabla.$$

Here $\nabla^T = \frac{\partial t_{max}(\theta)}{\partial \theta^T} = (a/\theta_1 - b)/a^2, (b - a/\theta_2)/a^2, 0)$, $a = \theta_1 - \theta_2$, $b = \log\theta_1 - \log\theta_2$, and $I(\xi, \theta)^-$ is a generalized inverse of the information matrix. Using the same set of nominal values of the parameters, I use PSO to minimize the variance of $g(\hat{\theta})$ by choice of (x_1, x_2, w_1), $0 < x_1 < x_2 < 30$, $0 < w_1 < 1$, $w_2 = 1 - w_1$.

The locally optimal design for estimating the time to maximum concentration is a c-optimal design with only 2 support points, which means that its information matrix is singular because the matrix is now a sum of two rank-one matrices. To get around having to find the inverse of singular information matrix $I(\xi, \theta)$, I follow the convention and add a small multiple ι to the 3×3 identity matrix E_3 and work with the invertible matrix

$$I_\iota(\xi, \theta) = I(\xi, \theta) + \iota E_3.$$

I implement PSO to find the optimal values of x_1, x_2, w_1 subject to $0 < x_1 < x_2 < 10$ and $0 < w_1 < 1$ with $w_2 = 1 - w_1$. The PSO parameters I use are $\iota = 10^{-6}$ and 200 particles all with $K = 2$ points. Expecting a singular information matrix for the optimal design, I allow for a larger number of iterations and after 1000 iterations, PSO generates a two-point design supported at 0.1793 and 3.5658 with weight 0.3938 at the latter point.

2.5 Locally c-optimal designs for estimating the area under the curve (AUC) in compartmental models

A similar procedure is used to find the locally c-optimal design for estimating the total exposure of the drug inside the compartment. By its definition, the function is

$$\begin{aligned} AUC &= \int_0^\infty \eta(x, \theta) dx \\ &= \theta_3 (\frac{1}{\theta_1} - \frac{1}{\theta_2}). \end{aligned}$$

Proceeding as before, I apply PSO to minimize the asymptotic variance of the estimated AUC. Setting $K = 2$ and using 100 particles and 1000 iterations, PSO takes 6.185 seconds to generate a two-point design supported at 0.2326 and 17.6339 with weight 0.0135 at the smaller point, which coincides with the locally c-optimal design to four decimal places in Atkinson and Donev [1992] on page 264.

It is interesting to observe what happens if I have used starting designs all with $K = 3$ points. The PSO generated design obtained using 200 particles and 500 iterations is supported at 0.2337, 17.6269 and 17.7176 with weight distribution at these points equal to 0.0135, 0.8983 and 0.0882. Increasing the number of iterations to 1000 results in a design supported at 0.2332, 17.6336 and 17.6626 with weight distribution at these points equal to 0.0135, 0.9535 and 0.03296, respectively. These results are consistent with the expectation that a longer iteration and/or more particles usually produces a higher quality solution. It also shows a very nice feature of PSO in that when I over-specify the number of support points the optimal design has, PSO can also automatically find the optimal design directly; in the above case, the 3 points found above get increasingly closer to 2 points as more iterations are used, leaving the weight at the smaller point unchanged.

2.6 Locally D-optimal Designs for quadratic logistic models

For modeling binary responses, logistic models are among the most popular ones because of their simplicity and ease of interpretation. Frequently, we have simple logistic models with two parameters and sometimes quadratic logistic models with three parameters. In this section I consider locally D-optimal designs for estimating all model parameters. Probably the first description of the locally D-optimal design for the simple logistic model was given in a doctoral thesis of Ford [1976] for the prototype interval $[-1, 1]$ and reported in Silvey [1980]. The formula is complicated for a relatively simple model. When I ran PSO to verify Ford's design, I was unable to produce the same result. A corrected formula for the locally D-optimal design on an arbitrary interval was given in Sebastiani and Settimi [1997] and I was able to verify their results using PSO.

Quadratic logistic models are sometimes employed to explore possible curvature in the model or to estimate an interesting characteristic of an agent in a dose-response study. For example, in radiology and radiotherapy, there is often interest in estimating the ratio of the coefficients associated with the linear and quadratic terms in the quadratic logistic model (Taylor [1990]). Selected locally c-optimal designs for the quadratic logistic model were found theoretically in Fornius and Nyquist [2010] after re-parameterizing the model in the following way using their notation:

$$\log(\frac{\mathrm{E}y}{1-\mathrm{E}y}) = \alpha + \beta(x-\mu)^2.$$

Here y is the binary response taking on values 0 or 1 with certain probabilities at dose x. By a first order Taylor's expansion, I get the gradient vector of the mean function as

$$f^T(x,\theta) = (1, (x-\mu)^2, -2\beta(x-\mu))$$

and the normalized information matrix of a $K-$point design ξ is

$$I(\xi,\theta) = \sum_{i=1}^{K} w_i \lambda(x_i,\theta) f(x_i,\theta) f^T(x_i,\theta)$$

where $\lambda(x,\theta) = \frac{\exp(-\alpha-\beta(x-\mu)^2)}{(1+\exp(-\alpha-\beta(x-\mu)^2))^2}$.

As usual, I begin the search for the locally D-optimal design among all 3-point designs first. As an example, suppose the design interval is $[-3, 1]$ and the nominal values for the 3 parameters are $\alpha = 2$, $\beta = 3$ and $\mu = 0$. With 128 particles and 150 iterations, PSO takes 6.623 seconds to find a design equally supported at -0.7270, 0 and 0.7270. The equivalence plot confirms the D-optimality of this design. If fewer number of iterations were used, say 50 iterations, the pattern of the optimal design will also emerge quickly and clearly, except that the weights are only approximately equal and the extreme support points are less symmetric about 0. PSO took 3.104 seconds to produce the design and also reports the design has an efficiency of 99.98%.

Interestingly, when the maximum probability of response is high, there are 4-point locally D-optimal designs. For instance suppose the design interval is $[-1, 1]$ and the nominal values for the 3 parameters are $\alpha = 3$, $\beta = -5$ and $\mu = 0$. With a

Table 2.1: Locally D-optimal designs for estimating the three parameters in the quadratic logistic model for different nominal values and different design intervals.

(α, β, μ)	Design space	Locally D-optimal designs
$(0, -1, 0)$	$[-1, 1]$	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$
$(0, -1, 0)$	$[-2, 2]$	$\begin{pmatrix} -1.4073 & 0.0000 & 1.4073 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$
$(3, -1, 0)$	$[-1, 1]$	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$
$(3, -1, 0)$	$[-2, 2]$	$\begin{pmatrix} -2.0000 & -1.2506 & 1.2506 & 2.0000 \\ 0.3061 & 0.1939 & 0.1939 & 0.3061 \end{pmatrix}$
$(3, -1, 0)$	$[-4, 4]$	$\begin{pmatrix} -2.0609 & -1.3239 & 1.3239 & 2.0609 \\ 0.2966 & 0.2034 & 0.2034 & 0.2966 \end{pmatrix}$

flock size of 256 and the number of iterations set at 200, PSO takes 18.317 seconds to find a design symmetrically supported at $-0.9217, -0.5921, 0.5921$ and 0.92170. The weights at -0.9217 and at -0.5921 are 0.2966 and 0.2034, respectively. Figure 2.1 is generated from the P-code from our websites and shows the equivalence plot of this PSO generated 4-point design for the quadratic logistic model. The plot is bounded above by 0 throughout the scaled design interval $[-1, 1]$ with equality at the support points of the PSO generated design and so the figure confirms its D-optimality. Table 2.1 displays locally D-optimal designs for estimating the three parameters in the quadratic logistic regression model for various nominal values of the parameters and on different design intervals.

As always here and elsewhere, in order to ensure a higher chance that PSO will generate the optimal design, I set the flock size and the maximal number of iterations larger than are usually necessary. Frequently, smaller flock size and smaller number of iterations will suffice, which means shorter CPU time can usually also

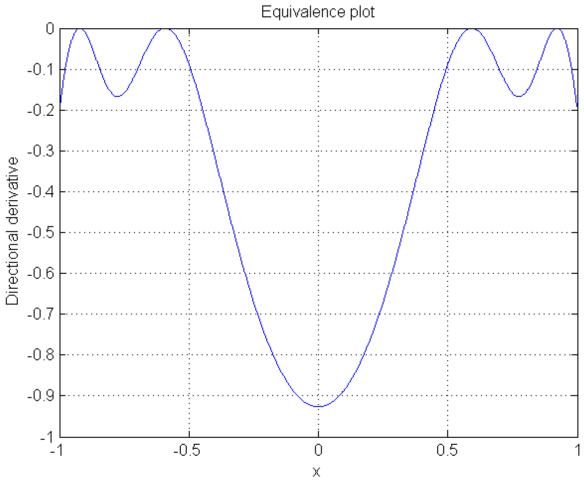


Figure 2.1: Equivalence plot of the D-optimal criterion for the PSO-generated 4-point design for the quadratic logistic model when $(\alpha, \beta, \mu) = (3, -5, 0)$.

produce the optimal design.

2.7 Locally D-optimal designs for a double exponential model

Double exponential regrowth model was developed by Demidenko [2004] to describe the dynamics of post-irradiated tumors based on the two-compartment model. Tumor cells may be categorized to two states as proliferating or quiescent. Assuming proliferating cells divide with constant rate ν, and quiescent cells die with a constant rate γ, the natural logarithm of the tumor volume of the two kinds of cells is expressed by

$$y = \alpha + \ln[\beta e^{\nu x} + (1-\beta)e^{-\gamma x}] + \varepsilon,$$

where $\varepsilon \sim N(0, \sigma^2)$.

After linearizing the mean function by a Taylor's first order expansion of the mean function, the gradient vector is

$$f^T(x,\theta) = (1, \frac{e^{\nu x} - e^{\gamma x}}{\beta e^{\nu x} + (1-\beta)e^{-\gamma x}}, \frac{\beta x e^{\nu x}}{\beta e^{\nu x} + (1-\beta)e^{-\gamma x}}, \frac{-(1-\beta)xe^{-\gamma x}}{\beta e^{\nu x} + (1-\beta)e^{-\gamma x}})$$

and $\theta = (\alpha, \beta, \nu, \gamma)^T$ is the vector of model parameters.

Li and Balakrishnan [2011] have shown that $|I(\xi, \theta)|$ depends only on β and $\nu + \gamma$, which implies that only two nominal values are required to generate the locally D-optimal design, i.e. a nominal value for β and a nominal value for the sum of the two parameters ν and γ. Proceeding as before and using codes from our websites, one can verify that PSO readily generates locally D-optimal designs that match those in Li and Balakrishnan [2011]. For instance, suppose I use the default input values for this example in the code, namely set $\beta = \nu + \gamma = 0.2$ and the interval is $[0, 10]$. Then PSO with 100 particles and 100 iterations produces in 1.041 seconds the locally D-optimal design equally supported at 0, 2.660, 6.707 and 10 reported in Table 2.2 in Li and Balakrishnan [2011]. Likewise, PSO with the same flock size and same number of iterations, taking 1.085 seconds, produces the optimal design as shown in the last row of their Table where $\beta = 0.8$ and $\nu + \gamma = 1$. Other cases including the case for verifying the c-optimal designs reported in their paper can be similarly verified.

2.8 Locally D-optimal designs for an inverse polynomial model

The inverse polynomial model has been frequently applied to describe the response to various stimuli in biological and agricultural studies (Cobby et al. [1986]). For example, Sparrow [1979] claimed that the inverse quadratic was the curve that best described the relationship between crop yield and fertilizer input. Here I focus on the inverse quartic polynomial below taken from Cobby et al. [1986]:

$$\mathrm{E}(y_i) = \eta(x_i; \theta) = \frac{x_i + \alpha}{\beta_0 + \beta_1(x_i + \alpha) + \beta_2(x_i + \alpha)^2},$$

where $\theta^T = (\beta_0, \beta_1, \beta_2, \alpha)$, with $0 \leq \alpha < (\beta_0\beta_2)^{1/2}$,$\beta_2 \geq 0$ and $|\beta_1| < 2(\beta_0\beta_2)^{1/2}$. By a Taylor's first order expansion of the mean function, the gradient vector is

$$f^T(x, \theta) = (\frac{\partial \eta}{\partial \beta_0}, \frac{\partial \eta}{\partial \beta_1}, \frac{\partial \eta}{\partial \beta_2}, \frac{\partial \eta}{\partial \alpha}),$$

with

$$\begin{aligned} \frac{\partial \eta}{\partial \beta_j} &= -\ z_i^{j+1}/p_i^2 \ \ (j = 0, 1, 2), \\ \frac{\partial \eta}{\partial \alpha} &= (\beta_0 - \beta_2 z_i^2)/p_i^2, \end{aligned}$$

where $z_i = x_i + \alpha$ and $p_i = \beta_0 + \beta_1 z_i + \beta_2 z_i^2$.

PSO is able to find the same locally D-optimal designs reported in the paper. However, different set of parameters requires different number of iterations. For example, if $\alpha = 0.1$,$\beta_1 = -0.8$, PSO takes only 100 iterations to find the optimum, but with $\alpha = 0.1$, $\beta_1 = 0.8$ PSO requires 200 iterations.

2.9 Locally c-optimal designs for a survival model

Konstantinou et al. [2011] investigated a two-parameter exponential model with type I right censored data, where all individuals entered the study at the same time and stayed until a user-specified time c or until failure, whichever was earlier. Right-censoring occurs when survival times are greater than c. Let $t_1, ..., t_n$ be the observed values for n subjects, and each is assigned experiment condition x_i. We consider an approximate design ξ of the form as follows

$$\xi = \left\{ \begin{array}{cccc} 1 & 2 & ... & K \\ w_1 & w_2 & ... & w_K \end{array} \right\}$$

Here the design space $\chi = \{1, ..., k, ..., K\}$, and w_k is the proportion of observations corresponding to condition k. The exponential regression model with probability density function $g(t_i)$ and a survival function $S(t_i)$ are given respectively by

$$g(t_i) = e^{\alpha+\beta x_i} \exp(-t_i e^{\alpha+\beta x_i})$$

and

$$S(t_i) = \exp(-t_i e^{\alpha+\beta x_i}).$$

Without loss of generality, we assume that the first m observations are failure times and rest $n-m$ observations are right censored. A direct calculation shows that if all observation pairs $(t_i, x_i), i = 1, \ldots, n$ are independent, the log-likelihood is

$$l(\theta; x_1, ..., x_n) = \log(\prod_{i=1}^{m} g(t_i) \prod_{i=m+1}^{n} S(t_i)) = \sum_{i=1}^{m}(\alpha + \beta x_i) - \sum_{i=1}^{n} t_i \exp(\alpha + \beta x_i).$$

It follows that the information matrix of design ξ is

$$\begin{aligned} I(\xi, \theta) &= \sum_{i=1}^{n}(1 - \exp(-ce^{\alpha+\beta x_i})) \begin{bmatrix} 1 & x_i \\ x_i & x_i^2 \end{bmatrix} \\ &= n\sum_{k=1}^{K} w_k(1 - \exp(-ce^{\alpha+\beta k})) \begin{bmatrix} 1 & k \\ k & k^2 \end{bmatrix}. \end{aligned}$$

The information matrix $I(\xi, \theta)$ depends on unknown parameters $\theta = (\alpha, \beta)^T$ because the model is nonlinear.

In the study by Konstantinou et al. [2011], the design space χ is $\{1, 0\}$ representing two treatment conditions, i.e. treated or not. The maximum time for observing outcomes in the study is $c = 30$ so that all observations are right censored if the outcome is not observed by that time. Using only 20 particles and 50 iterations, PSO is able to find the locally D-optimal designs as claimed in Konstantinou et al. [2011], all of which are equally supported at two points.

In practice, the parameter β in the model always has a clear biological interpretations and so it is often of interest. If we have a randomized control trial, β represents the effect on the hazard of death when the new treatment is compared with the placebo condition. An appropriate design to use here for estimating β is the locally c-optimal design that gives the smallest asymptotic variance of the estimate. This design is the same as the optimal design for minimizing the

Table 2.2: Weights of selected locally c-optimal designs for the Survival Model.

Nominal values	w_1	w_2
$\alpha = -2.163$, $\beta = -0.1$	0.498	0.502
$\alpha = -2.163$, $\beta = -0.405$	0.491	0.509
$\alpha = -2.163$, $\beta = -1.526$	0.425	0.575
$\alpha = -2.163$, $\beta = -2.623$	0.324	0.676

criterion

$$\begin{bmatrix} 0 & 1 \end{bmatrix} I^{-}(\xi, \theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

For this two-parameter model, it can be shown that the c-optimal design is always supported at 1 and 0 but with unequal weights that depend on the nominal values. Using 20 particles and 50 iterations, PSO is able to find and verify all the c-optimal designs for the four sets of nominal values in Konstantinou et al. [2011] and these weights are shown in Table 2.2.

2.10 Locally D-optimal design for a 4-parameter heteroscedastic Hill model

The Hill model is a commonly used nonlinear sigmoid model in pharmacodynamics studies (Khinkis et al. [2003]). In its original form (Hill [1910]), the model describes the drug concentration-effect relationship between an independently observed response y_{ij} and ith concentration x_i as

$$y_{ij} = \frac{(E_{con} - B)(\frac{x_i}{IC_{50}})^m}{1 + (\frac{x_i}{IC_{50}})^m} + B + \varepsilon_{ij} = \eta(x_i, \theta) + \varepsilon_{ij}.$$

The control effect at zero drug concentration is E_{con} , and B denotes the background effect at infinite drug concentration. The IC_{50} is the dose inducing a 50% decrease in the maximal effect $(E_{con} - B)$, and it is a measurement of the drug potency. The slope of the curve is controlled by m (Levasseur et al. [1998]).

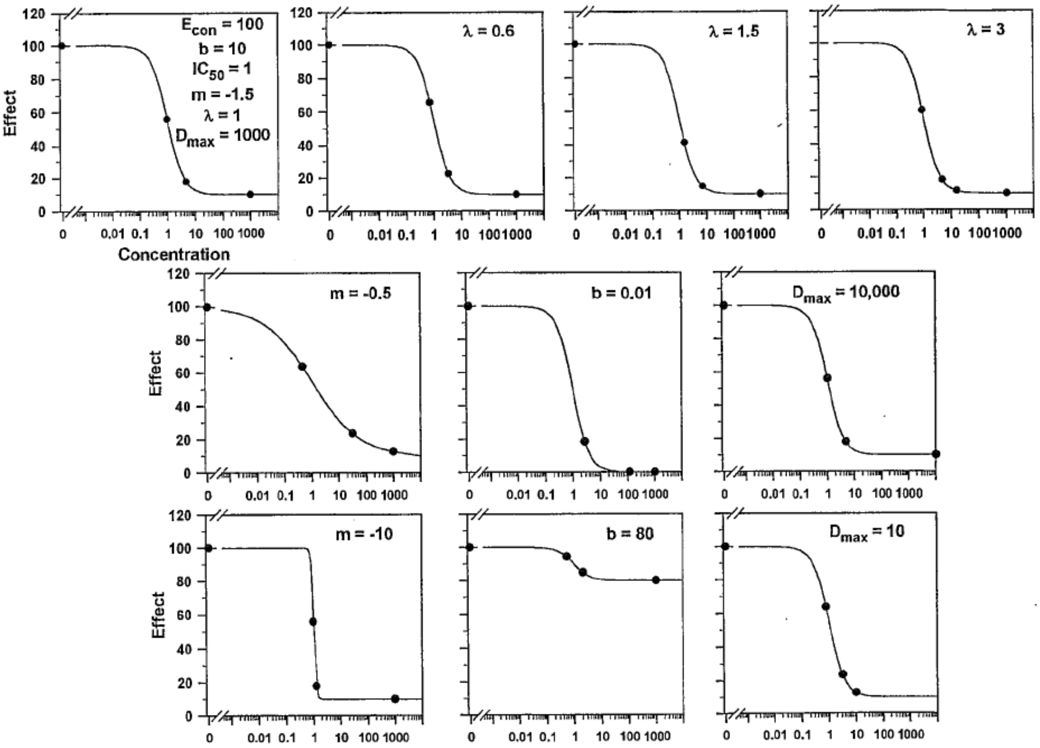


Figure 2.2: Influence of changing parameters on the shape of the Hill model. The figure is taken from Khinkis et al. [2003].

Figure 2.2 and its title is taken from Khinkis et al. [2003], which shows different shapes of Hill model under different parameters. Note that if m is negative, the curve decreases with rising concentrations, indicating an inhibitory effect of the drug (Khinkis et al. [2003]). Hill model assumes there are no systematic errors among the independent observations y_{ij}'s such that $\mathrm{E}(y_{ij}) = \eta(x_i, \theta)$. It is also assumed $\varepsilon_{ij} \sim Normal(0, \sigma_0 \eta(x_i, \theta)^{2\lambda})$, which means that the variance of the response is proportional to the mean response raised to power 2λ. Note that unlike other parameters in the Hill model, λ is a user-specified parameter. Khinkis et al. [2003] claimed that their experiments confirmed the appropriateness of this heteroscedastic model.

To find the locally D-optimal design for the Hill model, the information matrix for an observation at x_i is calculated as follows, assuming the nominal value of the vector of parameters is, say $\theta = (E_{con}, B, IC_{50}, m)^T$. Let

$$f^T(x_i, \theta) = (\frac{\partial\eta(x_i,\theta)}{\partial E_{con}}, \frac{\partial\eta(x_i,\theta)}{\partial B}, \frac{\partial\eta(x_i,\theta)}{\partial IC_{50}}, \frac{\partial\eta(x_i,\theta)}{\partial m}),$$

where

$$\begin{aligned}
\frac{\partial\eta(x_i,\theta)}{\partial E_{con}} =& \frac{(x_i/IC_{50})^m}{(1+x_i/IC_{50})^m} \\
\frac{\partial\eta(x_i,\theta)}{\partial B} =& \frac{1}{1+(x_i/IC_{50})^m} \\
\frac{\partial\eta(x_i,\theta)}{\partial IC_{50}} =& -\frac{(B-E_{con})(x_i/IC_{50})^m \log(x_i/IC_{50})}{(1+(x_i/IC_{50})^m)^2} \\
\frac{\partial\eta(x_i,\theta)}{\partial m} =& \frac{(B-E_{con})m(x_i/IC_{50})^m}{IC_{50}(1+(x_i/IC_{50})^m)^2}.
\end{aligned}$$

Hence for independent observations $x_1, ..., x_n$, we have $F = \begin{bmatrix} f(x_1) & f(x_2) & ... & f(x_n) \end{bmatrix}^T$ and the total information matrix is proportional to

$$I(\xi, \theta) = F^T W F$$

where $W = diag(\eta(x_1, \theta)^{-2\lambda}, ..., \eta(x_n, \theta)^{-2\lambda})$.

Khinkis et al. [2003] showed that locally D-optimal design for the Hill model has 4 support points. Accordingly a seven-dimension PSO is built with particles $(x_1, x_2, x_3, x_4, p_1, p_2, p_3)$, where $0 < x_1 < x_2 < x_3 < x_4$,$0 < p_1 < 1, 0 < p_2 < 1$,$0 < p_3 < 1$, $0 < p_4 = 1 - p_1 - p_2 - p_3$. Keeping the same nominal parameters $E_{con} = 1.7, B = 0.137, \lambda = 0.794$, and changing values of IC_{50} and m for each drug as did in Khinkis et al. [2003], PSO is able to find the locally D-optimal designs in Table 2.3 which agree with the results in Khinkis et al. [2003] to three decimal places. The numbers of iterations and particles of PSO I use here are small. For example, PSO finds the four support points for drug “AG2034” using only 300 iterations and 100 particles.

Table 2.3: Locally D-optimal designs found by PSO for Hill model with common nominal values: $E_{con} = 1.7, B = 0.137, \lambda = 0.794$. By varying IC_{50} and m. Seven D-optimal designs are obtained with equally weighted support points.

Drug	IC_{50}	m	Support points			
TMTX	0.00875	-1.79	0	0.00773	0.02965	8.95
MTX	0.0223	-2.74	0	0.02056	0.04950	22.3
AG2034	0.453	-0.825	0	0.32042	5.56703	453
AG2032	0.0774	-3.49	0	0.07263	0.144756	77.4
AG2009	111	-1.03	0	53.9007	377.2057	1500
AG337	0.468	-1.54	0	0.40495	1.93184	468
ZD1694	0.0429	-1.69	0	0.03761	0.15624	42.9

2.11 Summary

Particle Swarm Optimization algorithm seems like a very powerful, interesting but under-utilized tool for solving optimization problems in the pharmaceutical industry and more so in general statistical research work. I have shown in this chapter that PSO is an efficient and flexible method for finding optimal experimental designs for several biomedical studies, but clearly the applications are not restricted to biomedicine. A further strong point for PSO is that, as a metaheuristic algorithm, it does not respect the technical requirements imposed on the problem to obtain the locally optimal designs. For example, Konstantinou et al. [2011] and Li and Balakrishnan [2011] assumed technical conditions to arrive at the theoretical descriptions of the optimal designs. PSO does not have to incorporate the technical conditions in its search, suggesting that PSO can generate optimal designs for a wider class of problems, including optimal design problems in Marschner [2007], Ogungbenro et al. [2009], Biswas and Lopez-Fidalgo [2013].

CHAPTER 3

Locally Optimal Designs for Multivariable Biomedical Models

In this chapter, I focus on finding locally D-optimal designs for multivariable Exponential and Poisson regression models of up to five variables with all two way interactions in a user-selected restricted design space. Such models are useful to investigate the joint and interaction effects of multiple agents in biomedical studies. The context is in toxicology but techniques developed here can be applied to other models and find optimal designs for user-specified study goals or objectives. A main aim of my work is to show that a modified version of the Particle Swarm Optimization (PSO) algorithm can find locally D-optimal designs for estimating all model parameters in a generalized linear model with all pairwise interaction terms accurately and at minimal cost. Additionally, I use the proposed novel search strategy to find minimally supported D-optimal designs and ascertain conditions under which such optimal designs exist for such models.

A remarkable discovery in my work is that locally D-optimal designs for such models can have many more support points than the number of parameters in the model. This result is both new and interesting because almost all locally D-optimal designs have equal or just one or two more number of points than the the number of parameters in the mean response function; see the examples in Yang and Stufken [2009], Yang [2010a], and many examples in design monographs in Silvey [1980], Pazman [1986] and Berger and Wong [2009].

The remainder of the chapter is organized as follows. Section 3.1 provides the motivation of the project. In section 3.2, I briefly review locally D-optimal design for Exponential and Poisson regression models in the literature. In section 3.3, I propose a new algorithm called Ultra-dimensional Particle Swarm Optimization (UPSO) to find locally D-optimal designs for these models to jointly study effects of up to five toxicants and all pairwise interaction effects and provide guidelines for choice of tuning parameters in the UPSO algorithm, including flock size and number of iterations. Section 3.4 presents locally D-optimal designs found from the algorithm and conditions under which minimally-supported D-optimal designs exist for the Exponential and Poisson regression models. In Section 3.5, I consider conditional D-optimal designs found by setting all nominal values for all coefficients in the interaction terms equal to zero and investigate their robustness properties to mis-specification in the nominal values. Section 3.6 contains a real application in toxicology where I show the implemented design can be substantially improved using a locally D-optimal design generated by UPSO. Section 3.7 concludes with a summary and discussion of recent applications of particle swarm optimization techniques to search for other types of optimal designs.

3.1 Motivation

Over the past few decades, multiple drug therapies have been used extensively to treat various diseases such as cancers, AIDS and rheumatoid arthritis. Combinatorial drugs approach enjoys several therapeutic benefits, including increased efficacy, reduction in drug toxicity, delay the development of resistant organisms or cells by enhancing efficacy synergism or toxicity antagonism beyond the capability of a single drug dose (Chou [2006]). However, studying the joint efficacy or toxicity effects from several agents is complicated and while advances in estimating their joint effects have been made, research in developing informed design strategies has lagged.

There are only a handful of papers that address optimal design issues for generalized linear models with multiple regressors. Locally D-optimal design for the additive logistic regression model with two variables was given in Haines et al. [2007] and explicit formulas for D, A, and E-optimal designs for additive multi-factor logistic and probit models were provided in Yang et al. [2011]. Others focused on constructing minimally supported designs, which are designs with the same number of points as the number of parameters in the mean function of the response variable. Russell et al. [2009] established a sufficient condition for the existence of a minimally supported locally D-optimal design for the additive Poisson regression model with several agents. Li and Majumdar [2009] provided a sufficient condition for the existence of a minimally supported locally D-optimal design for a generalized linear model with several independent variables. However their results were complicated and their examples were confined to Poisson models with a polynomial predictor of a single variable up to degree 2. Minimally supported optimal designs are useful when it is expensive to take observations at a new site or a new combination of dosages from the agents. A disadvantage of such designs is that they cannot provide a lack of fit test to assess adequacy of the model.

The above literature review suggests that work in this area is limited and an analytic approach to find the optimal design for a generalized linear model is difficult and becomes increasingly more so as the model gets more complex with more variables. Numerical searches seem to be the only practical way to find optimal designs for such models with several variables. D-optimality is the easiest to study analytically, and some algorithms to search for locally D-optimal designs with several variables have been developed recently. For example, Sitter and Wu [1993a] studied locally D- and c-optimal designs for a two-variable additive model using a geometric approach and later extended the results to find locally D-optimal designs when there are more than two variables in an additive generalized linear model (Sitter and Wu [1993b]). Another notable example is Wang et al. [2006] who applied the Nelder-Mead algorithm to find minimally supported D-optimal designs for the Poisson model with one or two toxicants with an interaction term. Yu [2011] proposed a Cocktail algorithm that quickly finds a D-optimal design for a bivariate polynomial model with second order term and interactions. The latest advance in algorithm development came from Yang et al. [2013] who claimed their algorithm can find locally D-optimal designs for nonlinear models with any number of variables faster than current algorithms, such as Fedorov's V-algorithm described in Chapter 4 of Silvey [1980]. They supported their claim with four examples using models with a single response and one or two variables. Three models have a single independent variable and the fourth has two variables including their interaction term. Their algorithm is promising but it has not been tested how well it works for finding locally D-optimal designs in high dimensional problems or other types of optimal designs.

In general, there is still a dire lack of effective and general purpose algorithms for finding optimal designs in high dimensions. A main reason is that the dimension of the constrained optimization problem becomes increasingly large very fast as more variables are included in the model. In the following sections, I show PSO offers exciting promise to tackle high dimensional design problems.

3.2 Statistical background

Our interest is in constructing locally optimal designs for studying effects of M toxicants and their interactions and we have resources to observe independent responses from K distinct dose combinations of the toxicants. One common type of outcome is the number of organisms or cells that survive when we apply the k^{th} dose combination $x_k = (x_{k1}, ..., x_{kM})^T$, and x_{km} is the dose of the m^{th} toxicant, $k = 1, ..., K$. In the Exponential model, the number of surviving cells is assumed to follow a normal distribution with mean λ_k. Alternatively, the outcome can be modeled as a count variable y_k of organism or cells that survive when the k^{th} combination dose is applied using a Poisson regression with mean λ_k. Both Exponential and Poisson regression models have the same mean λ_k but different variances. The Exponential model assumes $Var(y_k|x_k) = \sigma^2$ and the Poisson model assumes $Var(y_k|x_k) = \lambda_k\sigma^2$. In practice, the latter is more commonly used. I include corresponding results for the Exponential model because I wish to compare relative merits of the implemented design by toxicologists in their study using an Exponential model in section 6, and (ii) I want to demonstrate our method can be directly applied to find optimal designs for other types of models.

A model with an additive mean structure is appropriate if it is believed that there are no interactions among the different agents. When there are multiple agents, low order interactions often exist among different toxicants and high order interactions are usually negligible relative to low order interactions. Accordingly, we consider a M-variable model with a mean structure given by

$$\begin{aligned}\lambda_k &= \exp(\beta_0 + \sum_{m=1}^{M} \beta_m x_{km} + \sum_{m=1}^{M-1} \sum_{m'>m}^{M} \beta_{mm'} x_{km} x_{km'}) \\ &= \exp(\beta^T f(x_k)) \qquad\qquad k = 1, ..., K,\end{aligned}$$

where $f^T(x_k) = (1, x_{k1}, .., x_{kM}, x_{k1}x_{k2}, ..., x_{kM-1}x_{kM})$. The zero dose corresponds to $\exp(\beta^T f(0)) = \exp(\beta_0) = \lambda_0$. We assume that $\beta_m < 0$ $(m = 1, ..., M)$ because we expect the effect of each agent is such that fewer cells will survive when the dose of the agent is increased. Klaassen [1980] proposed the concepts of synergism and antagonism, which have been well accepted in toxicology. When the joint effects of two toxicants are larger than their additive effects, i.e. $\beta_{mm'} < 0$, we have synergism. On the other hand, we have antagonism when the combined effects are smaller than the additive effects in which case $\beta_{mm'} > 0$. In what is to follow, the ratios $r_{mm'} = \beta_{mm'}/(\beta_m\beta_{m'}), 1 \le m < m' \le M$ will play an important role in the study of locally optimal designs. In the mean function, the number of parameters P increases as M increases. More specifically, $P = {}^{M}C_0 + {}^{M}C_1 + {}^{M}C_2 = 0.5M^2 + 0.5M + 1$.

Throughout, my focus is on finding locally D-optimal designs and so nominal values are required to construct them. An assumption is that any combination of toxicants should do some harm to the cells, and so this translates to

$$\frac{\lambda_k}{\lambda_0} = \exp(\sum_{m=1}^{M} \beta_m x_{km} + \sum_{m=1}^{M-1} \sum_{m'>m}^{M} \beta_{mm'} x_{km} x_{km'}) \leq 1. \tag{3.1}$$

The assumption (3.1) places a restriction on the range of valid values for the parameters as will be shown in the next section.

3.2.1 Locally D-optimal approximate design and D-efficiency lower bound

In this chapter, I consider finding approximate designs proposed by Kiefer [1974]. Designs are viewed as probability measures defined on a user-selected compact space χ. We represent such a design with K points by $\xi = \{x_1, ..., x_K; w_1, ..., w_K\}$, where w_k is the proportion of observations to be taken at the support point x_k, $0 < w_k \leq 1$ and $\sum_{k=1}^{K} w_k = 1$. For a pre-determined total sample size N, the implemented design takes roughly $n_k = Nw_k$ at the k^{th} support point, $k = 1, 2, ..., K$ subject to $Nw_1 + Nw_2 + \cdots + Nw_K = N$.

D-optimality is the most commonly used design criterion. It seeks to maximize the determinant of Fisher information matrix $I(\xi, \beta)$ by choice of the design so that the volume of joint confidence ellipsoid of all parameters in the mean function is minimized. The Fisher information matrix from the design ξ for the above model with mean λ_k has the form

$$I(\xi, \beta) = F^T W F,$$

where $F = (f(x_1), ..., f(x_K))^T$ and W is the matrix of second derivatives of the log-likelihood. For the Exponential model, $W = \text{diag}\{w_1\lambda_1^2, ..., w_K\lambda_K^2\}$ and for the Poisson model, $W = \text{diag}\{w_1\lambda_1, ..., w_K\lambda_K\}$. Because $|I(\xi, \beta)|$ depends on the unknown parameters β's, we assume nominal values for the β's are available to construct locally optimal designs (Chernoff [1953]). In practice, nominal values for β's are first elicited from experts or found from similar or pilot studies and then used to find the locally D-optimal design by maximizing $|I(\xi, \beta)|$. Data from the implemented locally optimal design is then used to estimate the β's again and they then serve as nominal values for the construction of the next locally optimal design. This procedure is usually repeated a few times before the estimated β's are stable.

As shown in Chapter 1, I can use D-efficiency $D_{eff}(\xi)$ or its lower bound $D_{lb}(\xi)$ to evaluate the quality of the PSO generated design ξ. My experience is that for high-dimensional problems such as the ones I am about to report, the lower bound $D_{lb}(\xi)$ can be very conservative. This means that $D_{lb}(\xi)$ can grossly underestimate the true efficiency of the design ξ. For example, a design ξ for the 5-variable Poisson model may show a lower bound of 31% even though its D-efficiency is 93%. One can use the equivalence plot to verify optimality of the design graphically but its utility may become increasingly limited as the number of toxicants increases and we have a high-dimensional plot to interpret. An alternative way, which I adopt here, is to use $D_{lb}(\xi)$ as the criterion to conservatively judge the quality of a design and regard it as numerically D-optimal design for all practical purposes if $D_{lb}(\xi) > 0.999$. All PSO generated designs reported here satisfy this lower bound and so are D-optimal.

3.3 Ultra-dimensional Particle Swarm Optimization (UPSO)

To find a K-point optimal design ξ, we choose vectors $(x_{k1}, ..., x_{kM}, w_k)$, $k = 1, ..., K$ to optimize the determinant of the Fisher information matrix. Because the weights w_k's sum to unity, the dimension of the search space for the locally

D-optimal design ξ_D is $(M+1)K-1$. Even for minimally supported design when $K=P$, the dimension of the search space is 95 for a 5-variable model with main effects and all pairwise interactions. Such high dimensional optimization problems pose a challenge for current optimal design algorithms, which include the Nelder-Mead simplex algorithm or Fedorov's algorithm and its several modified versions. For example, Wang [2002] searched for the locally D-optimal design using the Nelder-Mead simplex algorithm and reported that he could only produce results for the Poisson model with two main effects and interaction. Encouraged by the successful experience of applying PSO to univariable models in Chapter 2, I further extend PSO to find locally D-optimal designs for Exponential and Poisson model that allows the interaction terms. To enable PSO find the optima in a high dimensional constrained search space, I incorporate the following modifications in the algorithm and name the modified version as "Ultra-dimensional PSO".

Time-varying acceleration coefficients γ_1 *and* γ_2. In the standard PSO, γ_1 and γ_2 are two constants that determine how each particle moves toward its own personal best position p_i or the global best position p_g. Their user-selected values can affect PSO performance in high dimensional optimization problems (Ratnaweera et al. [2004]). Since finding locally D-optimal designs for Poisson or Exponential models with multiple toxicants is a high dimensional optimization problem, I apply their recommended modification in PSO and allow these two parameters to vary as the iteration progresses. Specifically, following their recommendations, I let γ_1 be a linearly decreasing function from 2.5 to 0.5 and let γ_2 be a linearly increasing function from 0.5 to 2.5 over the full range of the search iterations. This variation also allows high diversity in the early stage of the search and prevents premature convergence of particles.

Repair mechanism. Our problem is to find a locally optimal design in a constrained design space, but particles can wander outside the design space and generate infeasible solutions. The repair strategy in UPSO is to pull such particles back to the nearest boundary of the design space:

$$z_{i,d}^{(t)} = \begin{cases} ub_d & \text{if } z_{i,d}^{(t)} > ub_d \\ lb_d & \text{if } z_{i,d}^{(t)} < lb_d \\ z_{i,d}^{(t)} & \text{otherwise} \end{cases}$$

where $z_{i,d}^{(t)}$ is the d^{th} component of the particle i's position at time t, and ub_d and lb_d are the upper and lower bounds of the d^{th} component. The rationale for our strategy is that locally D-optimal designs frequently have support points at the boundary of the design space, see Li and Majumdar [2009], for example.

Ultra-dimensional search space. PSO is a metaheuristic algorithm, which means that it is not guaranteed to find the optimum. For high dimensional design problems, such as the ones discussed here, I find that the probability of finding the locally optimal design increases when each particle of the flock has many more points than the number of parameters in the mean function. This is a new, interesting and highly unusual search technique for finding optimal designs in the literature but it works well for our problems. Typically, the extra design points used in the search would end up as either points with a zero weight or coincide with other support points, in which case we combine their weights.

As an illustration, consider the problem of finding the locally D-optimal design for the 3-variable Poisson and Exponential models when all coefficients of the interaction terms are zero. This is the same as having nominal values for all $r_{ij} = 0$. Figure 3.3 shows simulated probabilities of finding the conditional D-optimal design for the 3-variable Exponential and Poisson models, which I have earlier on verified to have 8 support points. In the simulation study, I changed the search using different flocks where each flock may all have 8 to 16 points and the number of particles in each flock varies from 20 to 200. For each set of parameter combinations, I repeated the simulation 40 times and calculated the probability of generating the locally D-optimal design for both the Exponential and Poisson

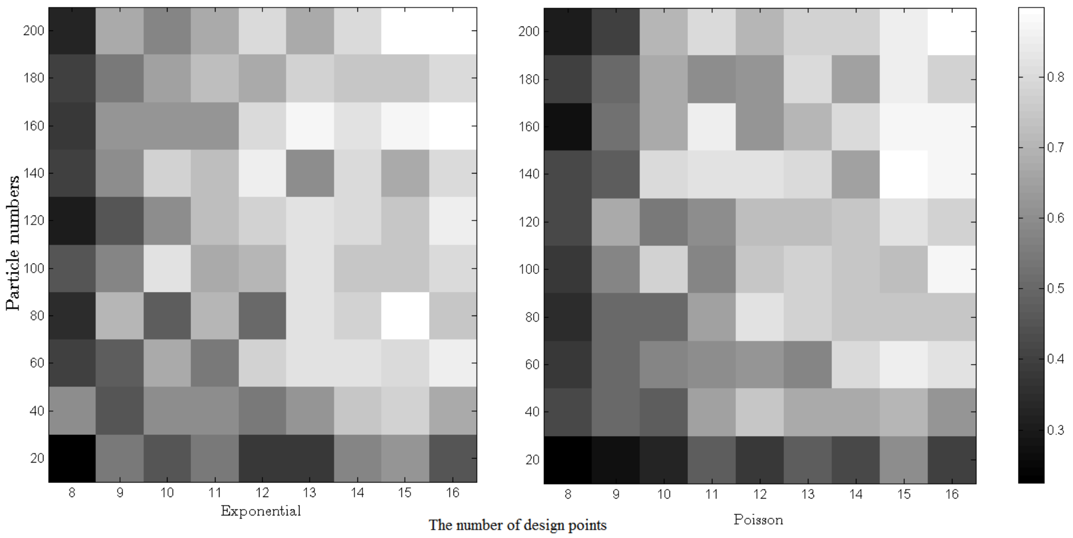


Figure 3.1: The probabilities of UPSO generating the conditional D-optimal design in 40 replicates for a 3-variable Exponential or Poisson model. The number of particles varies from 20 to 200 and the number of design points in each particle of the flock varies from 8 to 16.

models via UPSO. These probabilities are shown in Figure 3.3 and the results show that as each particle in the flock has increasingly more design points (from 8 to 16) and number of particles increases from 20 to 200, the probability of finding the conditional D-optimal design increases from less than 10% to over 90%.

Similar patterns have also been observed for finding the locally D-optimal designs for the 4 and 5-variable Exponential and Poisson models when not all $r_{ij} = 0$. For space consideration, I omit details and the results. My overall recommendation for UPSO to work well is that if the number of support points in the locally D-optimal design is K, I propose the rule of thumb of choosing the number of design points for each particle in the flock is $2K$ and the number of particles is roughly $20K$. In practice, K is unknown and I suggest having a flock with number of design points about twice the number of parameters in the mean function. Clearly, when each particle has more design points, more iterations are required for UPSO to converge to optimum. I recommend using 2000 iterations to search for the optimum for 3-variable, 5000 for 4-variable model and 10,000 for 5-variable models. This rough rule of thumb is likely to ensure that UPSO generate the locally D-optimal design with high probability.

3.4 Results

In toxicity study with multiple toxicants, I follow the definition in Wang et al. [2006] and define the individual effective dose (IED) as expected survival rate at the dose x_{km} from the m^{th} toxicant alone by

$$q_{km} = \frac{\exp(\beta_0 + \beta_m x_{km})}{\exp(\beta_0)} = \exp(\beta_m x_{km}) \in (0, 1].$$

Clearly, if c_m is the user-specified lower bound of the IED for the m^{th} toxicant, $m = 1, ..., M$, I have $c_m \le q_{km} \le 1$. Further, I derive information matrix for the models in terms of IED levels $q's$ defined on the induced design space $\prod_{m=1}^{M}[c_m, 1]$ for q_{km}'s rather than the design space for the doses x_{km}'s. Doing so can simplify the calculation of the optimal designs and provide insight into their properties as the following result shows.

THEOREM 1. *For a M-variable Exponential or Poisson model with 2-way interactions,*

a) the locally D-optimal design on the induced space depends on the model parameters only through the interaction terms $r_{mm'} = \beta_{mm'}/(\beta_m\beta_{m'}), 1 \le m < m' \le M$ and $c_1, \ldots, c_M$;

b) the D_{eff} of an arbitrary design on the induced space depends on the model parameters only through the interaction terms $r_{mm'} = \beta_{mm'}/(\beta_m\beta_{m'}), 1 \le m < m' \le M$ and $c_1, \ldots, c_M$.

Proof of THEOREM 1a:

I provide here justifications for the technical results for the Poisson model; corresponding results for the Exponential model can be proved similarly and are omitted. For the M-variable Poisson model with all two-way interactions, the number of parameters in the mean function is $P =^M C_0 +^M C_1 +^M C_2 = 1 + M + M(M-1)/2$ and the regression vector at the k^{th} dose combination is $f(x_k) = \{1, x_{k1}, \ldots, x_{kM}, x_{k1}x_{k2}, \ldots, x_{kM-1}x_{kM}\}^T$. If the design ξ assigns w_i proportion of the total number of doses at $x_i, i = 1, \ldots, K$, the Fisher information matrix is

$$\begin{aligned}
I(\xi, \beta) &= \sum_{k=1}^{K} w_k \lambda_k f(x_k) f(x_k)^T \\
&= \sum_{k=1}^{K} w_k \text{exp}(\beta_0) (\prod_{m=1}^{M} \text{exp}(\beta_m x_{km}))(\prod_{m=1}^{M-1} \prod_{m'>m}^{M} \text{exp}(\beta_{mm'} x_{km} x_{km'})) f(x_k) f(x_k)^T \\
&= \text{exp}(\beta_0) \{ \sum_{k=1}^{K} w_k (\prod_{m=1}^{M} q_{km})(\prod_{m=1}^{M-1} \prod_{m'>m}^{M} \text{exp}(r_{mm'} \text{ln} q_{km}) \text{ln} q_{km'}))) f(x_k) f(x_k)^T \},
\end{aligned}$$

where the last equation follows from the definition of $r_{mm'}$ and $\beta_m x_{km} = \text{ln} q_{km}, m = 1, 2, \ldots, M$. Let $Q(\xi)$ be the quantity inside the curly brackets; and let

$$f(\text{ln} q_k) = \{1, \text{ln} q_{k1}, \ldots, \text{ln} q_{kM}, \text{ln} q_{k1} \text{ln} q_{k2}, \ldots, \text{ln} q_{kM-1} \text{ln} q_{kM}\}^T,$$

and let $P \times P$ diagonal matrix

$$\tau = \text{diag}\{1, \beta_1^{-1}, \ldots, \beta_M^{-1}, (\beta_1 \beta_2)^{-1}, \ldots, (\beta_{M-1} \beta_M)^{-1}\}.$$

Clearly, $f(x_k) = \tau f(\text{ln} q_k)$, $I(\xi, \beta) = \text{exp}(\beta_0) \tau Q(\xi) \tau^T$ and if S_P is the set of all permutations on the set $\{1, .., P\}$, by Leibniz's formula for the determinant of a matrix, we have

$$\begin{aligned}|I(\xi,\beta)| &= \sum_{\sigma\in S_P}\text{sgn}(\sigma)\prod_{i=1}^{P} I(\xi,\beta)_{i,\sigma(i)}\\ &= (\exp(\beta_0))^P\sum_{\sigma\in S_P}\text{sgn}(\sigma)\prod_{i=1}^{P}(\tau_i\tau_{\sigma(i)}Q(\xi)_{i,\sigma(i)})\\ &= (\exp(\beta_0))^P\sum_{\sigma\in S_P}\text{sgn}(\sigma)(\prod_{m=1}^{M}\beta_m)^{-2M}\prod_{i=1}^{P}Q(\xi)_{i,\sigma(i)}\\ &= \frac{(\exp(\beta_0))^P}{(\prod_{m=1}^{M}\beta_m)^{2M}}\sum_{\sigma\in S_P}\text{sgn}(\sigma)\prod_{i=1}^{P}Q(\xi)_{i,\sigma(i)}\\ &= \frac{(\exp(\beta_0))^P}{(\prod_{m=1}^{M}\beta_m)^{2M}}|Q(\xi)|.\end{aligned}$$

Consequently, maximizing $|I(\xi,\beta)|$ is equivalent to maximizing $|Q(\xi)|$, which only depends on $r_{mm'}$ and the induced design space lower bounds $c_1,...,c_M$.

Proof of THEOREM 1b:

From the proof of Theorem 1a, the determinant of any design is proportional to the determinant of $Q(\xi)$. If we let $Q(\xi_D)$ denote the corresponding expression for the locally D-optimal design ξ_D, we have

$$D_{eff}(\xi) = (\frac{|I(\xi,\beta)|}{|I(\xi_D,\beta)|})^{\frac{1}{P}} = (\frac{|Q(\xi)|}{|Q(\xi_D)|})^{\frac{1}{P}},$$

which only depends on $r_{mm'}$ and the induced design space lower bounds $c_1,...,c_M$.

The induced design space is predicated on having good prior knowledge for each toxicant from pilot experiments or experiences with similar toxicants. Good estimates of the range of IEDs and the lower bounds $c_m, m = 1, 2, \cdots, M$ are required; otherwise, mis-specification in the nominal values can result in inefficiency of the implemented design. Theorem 1 shows that when the locally optimal designs are described in terms of q_{km}'s, they depend on the nominal values of β_m's only through the ratios $r_{mm'}$s. This implies that we may without loss of generality set $\beta_m = -0.1$, $m = 0, ..., M$ for simplicity in the search for the optimal design.

In the rest of this section, I discuss conditions under which a minimally supported D-optimal designs exist for our models and show a conjecture by Wang et al. [2006] is wrong. I also describe general structure of the locally D-optimal designs when it is feasible.

3.4.1 Minimally supported D-optimal designs

For many linear models, such as the polynomial models, the locally D-optimal design is usually minimally supported i.e. the number of support points in the optimal design is equal to the number of unknown parameters in the model. Such optimal designs are ubiquitous and easier to study since they require equal replicate at each design point. Locally D-optimal designs for the Poisson models with up to two variables were also found to be minimally supported by Minkin [1987] and Wang et al. [2006], regardless of the design space or the nominal values of the interaction term. I implemented UPSO and was able to verify their results, including their reported locally D-optimal designs for both the Exponential and Poisson models up to two variables with an interaction term.

Theorem 1 shows locally D-optimal designs for the Exponential and Poisson models depend on the model parameters only through $r_{mm'}$. It is not clear how these values affect whether a minimally supported locally D-optimal design exists for our models. Similarly, it is interesting to ask the same question for the selected values of $c_1, \ldots, c_M$. For simplicity, I assume that $r_{mm'} = r$ and the lower bound of the induced design space for each i^{th} IED is equal, i.e. $c_m = c, m = 1, ..., M$, for some user-selected value c. I start finding locally D-optimal design for $c = 0.01$ which is a practically unrestricted design space, and gradually increase c by step of 0.01 to investigate the effect of c on the structure of locally D-optimal designs for the Exponential and Poisson models. By assumption (3.1) an upper bound can be derived for the interaction term r. Clearly, when $r < 0$, the inequality (3.1) always holds. When $r > 0$ the left-hand side of (3.1) is a convex function of

q_{km}'s and so it attains its maximum value when $q_{km} = c_m$ or 1. As an example, if we have a 3-variable model, (3.1) holds if and only if $(c_m)^3\text{exp}(r(\log(c_m))^2)^3 < 1$, which implies $r < 0.22$ if $c_m = c = 0.01$. Similarly, the upper bound for r is 0.14 for a 4-variable model and is 0.11 for a 5-variable model.

I find minimally supported D-optimal designs for several possible values of $r_{mm'} = r = 0.05, 0, -0.1, -0.5, -1, -5, -10$. By studying the UPSO generated minimally supported D-optimal designs in the design space $[c, 1]^M$, I discover that locally D-optimal designs have the same structure for the Exponential and Poisson model with M variables and $M = 3, 4$ and 5. There is 1 control point $(1, ..., 1)$, ${}^MC_1$ pure component design points at dose combinations of the form $(s, 1, ..., 1)$ and $s = \max\{c, 0.368\}$ for the Exponential model and $s = \max\{c, 0.135\}$ for the Poisson model. Additionally, there are ${}^MC_2$ binary blend points at dose combinations of the form $(u, u, 1, ...1)$, where $u = \max\{c, d\}$ and the value of d depends on r and the model. Table 3.1 lists the values of d for various setups.

Wang et al. [2006] conjectured that for M-variable interaction model ($M > 2$), locally D-optimal designs are also minimally and equally supported and have a similar structure as D-optimal designs for 2-variable model. However, using UPSO I find that locally D-optimal designs are not necessarily minimally supported when there are 3 or more toxicants in the Poisson or Exponential models. UPSO suggests D-optimal design becomes non-minimally supported when c surpasses a threshold C. For different r's, I determine the threshold C on the induced design space that produces a minimally supported D-optimal design for Exponential and Poisson regression models (Table 3.1). One notable property is that these C's depend on r only but not on the number of variables in the Poisson and Exponential model.

In practice, it is unlikely that the toxicants under investigation all have the same degree of potency or lethality (same c's) and interactive ability (same $r_{mm'}$'s). When the toxicants have unequal c_m's and unequal $r_{mm'}$'s, UPSO can be applied

Table 3.1: Values of d and the threshold C that produces the minimally supported D-optimal design for 3, 4 and 5-variable Exponential and Poisson regression models with all 2-way interactions.

r	0.05	0	-0.1	-0.5	-1	-5	-10
$d_{expoential}$[1]	0.348	0.368	0.400	0.481	0.539	0.699	0.763
$d_{Poisson}$[2]	0.105	0.135	0.181	0.291	0.368	0.583	0.670
$C_{expoential}$[3]	0.48	0.51	0.54	0.62	0.67	0.79	0.84
$C_{Poisson}$[4]	0.20	0.26	0.32	0.45	0.53	0.71	0.77

[1] d value for Exponential models;

[2] d value for Poisson models ;

[3] Threshold C for Exponential models;

[4] Threshold C for Poisson models.

to ascertain the threshold level for each set of user-specified parameters in the design problem and determine the locally optimal design. For example, if we have a 3-variable Poisson model with nominal values $\beta_{mm'} = 0$, the lower bounds are $[c_1 = 0.1; c_2 = 0.3; c_3 = 0.3]$, UPSO generated a D-optimal design equally and minimally supported at 7 points. However, when the lower bounds are $[c_1 = 0.1; c_2 = 0.3; c_3 = 0.4]$, the locally D-optimal design has 8 points and so is no longer minimally supported.

3.4.2 Non-minimally supported conditional D-optimal designs for Exponential and Poisson regression models

Yang and Stufken [2009], Yang [2010a], Dette et al. [2011] showed that for univariate nonlinear models, D-optimal designs usually have the same or just one or two more support points than the number of parameters in the mean function. However this is not the case for multivariate Exponential and Poisson models when the assumed common lower bound c for each toxicant exceeds a threshold value and all $r_{mm'}$s have the same value r. Table 3.1 lists the threshold values for selected values of r and show that they do not depend on the number of toxicants in the model but that the threshold values depend on whether the model is Exponential or Poisson.

We are particularly interested in the locally non-minimally supported D-optimal designs because the extra support points are appealing to practitioners and allow tests for model adequacy to be conducted. I focus on the case when all $r_{mm'} = 0$ for the reason that in practice, although we believe that there are some interactions among toxicants, we may not have prior knowledge about the direction or the magnitude of the interactions before carrying out the experiment. A practical way is to find the locally D-optimal design for the model assuming the nominal values of the coefficients of the interaction terms are all 0. Wang et al. [2006] called such optimal designs "conditional D-optimal designs" - a term I use in the rest of the paper as well. I use UPSO to find them on induced design spaces $[c, 1]$, and find that the structure of these non-minimally supported D-optimal designs depend on the magnitude of the common assumed value of c and whether we have a Poisson or an Exponential model. In particular, there are model dependent constants b_i's that place limits on c resulting in different structures for the locally D-optimal designs. Table 3.2 below shows some of these b_is' values.

I now describe the various structures of the locally D-optimal designs for the Poisson model with 3, 4 and 5 variables. Remarkably, the same structures hold for the corresponding Exponential models as well.

3-variable model. When $b_1 \leq c$, there are 8 support points including 1 control point located at (1,1,1) with weight w_0; ${}^3C_1$ points at dose combinations of the form $(c, 1, 1)$ each with weight w_1; ${}^3C_2$ points at dose combinations of the form $(c, c, 1)$ each with weight w_2; and an additional point at dose (c, c, c) with weight w_3, and $w_0 > w_1 > w_2 > w_3$.

4-variable model. When $b_1 \leq c < b_2$, there are 15 support points including 1 control point with weight w_0; ${}^4C_1$ points at dose combinations of the form

$(c, 1, 1, 1)$ each with weight w_1; ${}^4C_2$ points at dose combinations of the form $(c, c, 1, 1)$ each with weight w_2, ${}^4C_3$ points at dose combinations of the form $(c, c, c, 1)$ each with weight w_3, and $w_0 > w_1 > w_2 > w_3$. When $c \geq b_2$ there is an additional point at (c, c, c, c) with weight $w_4 < w_3$.

5-variable model. When $b_1 \leq c < b_2$, there are 26 support points including 1 control with weight w_0; ${}^5C_1$ points at dose combinations of the form $(c, 1, 1, 1, 1)$ each with weight w_1; ${}^5C_2$ points at dose combinations of the form $(c, c, 1, 1, 1)$ each with weight w_2, ${}^5C_3$ points at dose combinations of the form $(c, c, c, 1, 1)$ each with weight w_3 and $w_0 > w_1 > w_2 > w_3$. When $c \geq b_2$, there are ${}^5C_4$ more points at dose combinations of the form $(c, c, c, c, 1)$ each with weight $w_4 < w_3$. When $b_2 > c \geq b_3$ an additional point at (c, c, c, c, c) with weight $w_5 < w_4$.

Table 3.2: Constants b_i's that determine whether additional points are required by the locally D-optimal designs for the Poisson and Exponential models.

	b_1	b_2	b_3
Exponential	0.52	0.69	0.88
Poisson	0.27	0.47	0.69

3.5 Discussion

In this section I discuss the robustness properties of the conditional D-optimal design. The latter is a practical concern because mis-specification of the nominal values can affect the quality of inference from the implemented design. I also comment on the performance of PSO and some of the recent algorithms proposed for finding optimal designs.

3.5.1 Robustness of the conditional D-optimal designs

Practitioners may be concerned with the robustness of the conditional D-optimal designs when the interaction terms r'_{ij}s deviate from zero. To investigate the effect of mis-specification of nominal values in the interaction terms, I calculate the efficiency of conditional D-optimal designs for 3-variable Exponential and Poisson models for different nominal values of r_{ij}'s. There are 3 interaction terms r_{12}, r_{13}, r_{23} in the models and I consider two situations. The first scenario assumes that only one interaction is not 0 but the conditional D-optimal design is mistakenly used. The second scenario has all non-zero three interactions terms with a common value equal to r. Figure 3.5.1 shows how D-efficiencies of conditional D-optimal design change when one or three interactions are mis-specified from -5 to 0.4, and the common lower bound of design space c varies from 0.1 to 0.9. The two plots on the first row show the D-efficiencies for the Exponential models, and the second row are the corresponding plots for the Poisson model. The two plots on the left hand side are for both models with one interaction mis-specified, and the right hand side are for models with three interaction terms mis-specified.

For both models, I observe that when c has a small value, the conditional D-optimal design is very inefficient if r_{ij} is far from 0. However, as the lower bound increases, such design becomes more robust to deviation from the condition that $r_{ij} = 0$. For the Poisson model, my finding is consistent with results from Wang et al. [2006]'s results for 2 variables. Comparing the two models, I find that the conditional D-optimal designs for the Poisson model is more robust than that for the Exponential model. For example, when $c = 0.5$, the conditional D-optimal design for the 3 variable Poisson model with one interaction mis-specified has an efficiency of 0.93 even when the true value of r is as low as -5. The corresponding result for the Exponential model is only 0.73. In addition, the two plots show the conditional D-optimal design is more robust when fewer number of interactions are mis-specified, which is expected.

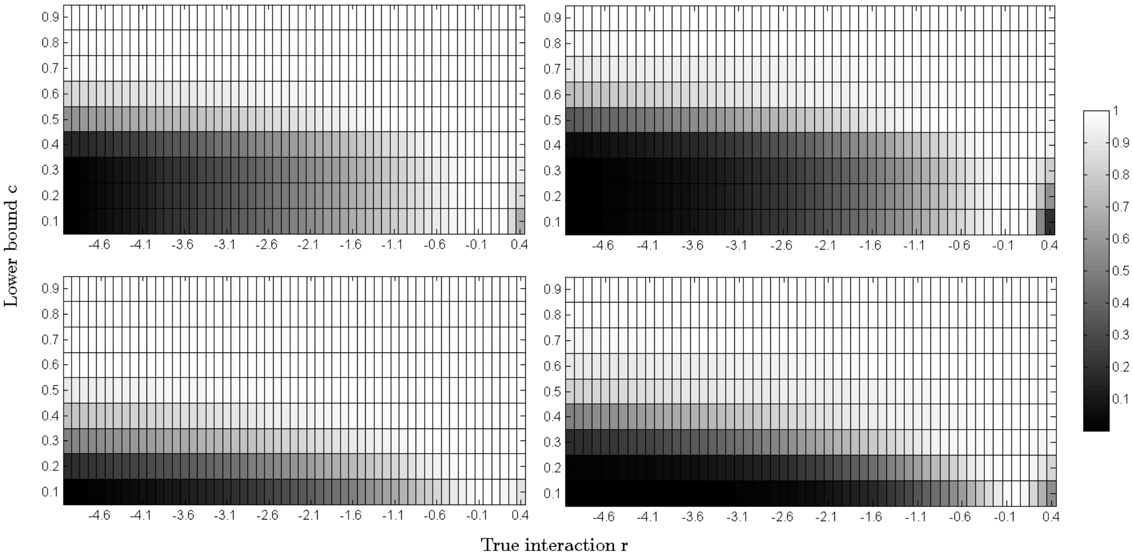


Figure 3.2: D-efficiencies of the conditional D-optimal design for the 3-variable Exponential or the Poisson model using different sets of nominal values. The two plots in the first row are for the Exponential model and the two in the second row are for the Poisson model. The two plots in the first column are for models with one interaction term misspecified, and the plots in the second column are for models with three interaction terms mis-specified.

3.5.2 Verify optimality via PSO

In UPSO, all design points and their weights are rounded to four decimal places. This is equivalent to having UPSO search on a discretized space with a grid size of 10,000. Such fine grid size incurs a huge computational burden to verify the optimality of the generated design ξ for any algorithm. In the numerical work, optimality of a design can be confirmed by applying the standard PSO to find the maximum violation in the equivalence theorem by determining the maximum positive value in the checking condition $\zeta = \max_{x \in \chi} \text{trace}[I(\delta_x, \beta) I^{-1}(\xi_D, \beta)] - P$ as shown in Chapter 1. This value is then used to compute the D_{lb} for the current design. If the lower bound has a value of unity, this indicates the current design is D-optimal. Verifying a design optimality is a relatively easy task using PSO since the optimization problem now has dimension only M, the number of variables. In practice, we repeat the search for the maximum violation several times since PSO, while effective, does not guarantee that the global optimum is always found.

3.6 Application

Lutz et al. [2005] explored 2-way interaction effects for the induction of Micronuclei in Mouse Lymphoma Cells among three genotoxic agents. Two of them are methylating agents: methyl methanesulfonate (MMS) and N-methyl-N-nitrosourea (MNU) and the third is a topoisomerase-II inhibitor genistein (GEN). They conducted three experiments using two agents at a time and fitted the number y_i of cells containing one or more micornuclei per 1000 binucleated cells using a 2-variable Exponential model with interaction to study the combination effects of any two agents mixture. Table 3.3 shows the 36 dose combinations used in their study, each with 2 replicates resulting total 72 runs (denoted as Lutz design).

I model the number cells containing zero micornuclei per 1000 binucleated cells instead since the negative effects of these toxicants are of interest here. I refit the data from their experiments by 3-variable Poisson and Exponential models with the same mean given by

$$\begin{aligned}E(1000 - y_i) = \exp(&\beta_0 + \beta_1\text{MMS} + \beta_2\text{MNU} + \beta_3\text{GEN} \\ &+ \beta_{12}\text{MMS*MNU} + \beta_{13}\text{MMS*GEN} + \beta_{23}\text{MNU*GEN}).\end{aligned}$$

Judging by the value of the fitted full log likelihood function and the AIC criteria, both Exponential and Poisson models provide a better fit to all the experimental data than the two-agent models used by Lutz et al. [2005]. Therefore in the application I show how experiment can be efficiently designed based on 3-variable Exponential and Poisson regression model with pairwise interactions, especially when there is only limited prior information available.

I look for the conditional D-optimal design assuming there is zero prior knowledge about the magnitude or direction of drug-drug interactions. Since the conditional D-optimal design is independent of nominal values of β_m's (-0.1 was arbitrarily chosen here), the only quantities needed here are lower bounds c_i's of the IED design space of three agents, according to THEOREM 1. Unfortunately the authors didn't provide the exact values for c_i's in their paper though they have conducted pilot experiments for each single agent. From observing the number of uninduced cells out of 1000 by different dose combinations of three agents, and according to the paper description that "the concentration range used for the three chemicals was defined by similar effect magnitude based on their pilot experiments", I have to roughly guess IED lower bounds $c_1 = c_2 = c_3 = 0.7$ for doses of MMS, MNU and GEN range in intervals [0, 0.3], [0, 2.1], [0,0.045]. It is worth noting that the generated conditional D-optimal design may lose efficiency, since these IED lower bound guesses are very possible to deviate from the truth.

Using UPSO, I obtain the conditional D-optimal designs for Exponential and Poisson model, which are non-minimally supported (Table 3.5). This indicates the lower bound of dose range for each agents should be used in the drug combinations.

The conditional D-optimal design for Exponential model puts more weight on control point and less weight on blend point (c_1, c_2, c_3) than does design for Poisson model.

To fairly evaluate the efficiencies of Lutz design and the proposed conditional D-optimal design, I compare them to the locally D-optimal design using parameter estimates from our fitted model as nominal values (denoted as post-hoc D-optimal design), which is surely the most efficient design for the current available data. For Exponential and Poisson models, Table 3.4 shows the fitted $\hat{\beta}_m$'s, pairwise interactions $\hat{\beta}_{mm'}$'s and lower bounds $\hat{c}_i$'s of IED design space.

By examining the structure of the generated designs, I find the conditional and post-hoc D-optimal designs have the same support points but with different weights (Table 3.5). Assuming total 72 runs, we would implement a design with the number of runs at each support point shown in parentheses in the table. Comparing to post-hoc D-optimal design, the proposed design has a relative D-efficiency of 99.38% for the Exponential model and a D-efficiency of 99.90% for the Poisson model. In contrast, Lutz design has a D-efficiency of 25.57% for the Exponential model and a D-efficiency of 24.13% for the Poisson model. This shows substantial efficiencies and improved inference can be had using an informed optimal design for estimating the model parameters, even when there is only limited prior information available. This application also confirms my finding in section 5.1 that the conditional D-optimal design is more robust to deviation from the condition of $r_{mm'} = 0$ when IED lower bounds c_i's are large.

3.7 Summary

Encouraged by the successes of applying PSO to find locally D- and c-optimal designs for univariable models in Chapter 2, I tackle high-dimensional design problems described herein. My research shows that the proposed UPSO is a

Table 3.3: Number(y_i) of binucleated cells that show one or more micronuclei per 1000 binucleated cells using different dose (μM) combinations of MMS, MNU and GEN with 2 replications at each of the dose combination levels.

MMS	MNU	GEN	Replicate 1	Replicate 2
0	0	0	16	11
0	0	0	16	13
0.1	0	0	29	26
0.2	0	0	52	88
0.3	0	0	160	210
0	0.7	0	51	50
0	1.4	0	93	125
0	2.1	0	172	279
0.1	0.7	0	79	141
0.1	1.4	0	131	256
0.2	0.7	0	200	230
0.2	1.4	0	235	253
0	0	0	16	21
0	0	0	20	23
0.1	0	0	36	31
0.2	0	0	65	57
0.3	0	0	140	135
0	0	0.015	37	32
0	0	0.03	69	79
0	0	0.045	168	175
0.1	0	0.015	61	47
0.1	0	0.03	91	89
0.2	0	0.015	106	85
0.2	0	0.03	145	152
0	0	0	24	25
0	0	0	24	27
0	0.7	0	82	73
0	1.4	0	165	175
0	2.1	0	279	302
0	0	0.015	77	76
0	0	0.03	120	131
0	0	0.045	209	261
0	0.7	0.015	74	114
0	0.7	0.03	138	112
0	1.4	0.015	206	187
0	1.4	0.03	224	258

Table 3.4: Parameter estimates and estimated lower bounds of the induced design space for the Exponential and Poisson models.

	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_{12}$	$\hat{\beta}_{13}$	$\hat{\beta}_{23}$	$\hat{c}_1$	$\hat{c}_2$	$\hat{c}_3$
Exponential	6.909	-0.492	-0.125	-4.200	-0.0649	15.782	1.398	0.863	0.769	0.828
Poisson	6.911	-0.506	-0.128	-4.288	-0.0472	16.308	1.458	0.859	0.765	0.825

Table 3.5: D-optimal designs for Exponential and Poisson model

MMS-q_{k1}	1	1	1	$\hat{c}_1$	$\hat{c}_1$	$\hat{c}_1$	1	$\hat{c}_1$
MNU-q_{k2}	1	1	$\hat{c}_2$	1	$\hat{c}_2$	1	$\hat{c}_2$	$\hat{c}_2$
GEN-q_{k3}	1	$\hat{c}_3$	1	1	1	$\hat{c}_3$	$\hat{c}_3$	$\hat{c}_3$
MMS-x_{k1}	0	0	0	0.3	0.3	0.3	0	0.3
MNU-x_{k2}	0	0	2.1	0	2.1	0	2.1	2.1
GEN-x_{k3}	0	0.045	0	0	0	0.045	0.045	0.045
w_i[a]	0.139 (10)	0.134 (10)	0.134 (10)	0.134(10)	0.124 (9)	0.124 (9)	0.124 (9)	0.088 (5)
w_i[b]	0.133	0.127	0.124	0.129	0.113	0.130	0.122	0.123
w_i[c]	0.131 (10)	0.128 (9)	0.128 (9)	0.128 (9)	0.123 (9)	0.123 (9)	0.123 (9)	0.117 (8)
w_i[d]	0.129	0.126	0.124	0.127	0.120	0.127	0.123	0.124

[a] Weight of the support point of conditional D-optimal design for Exponential model. Values in the parentheses are rounded number of runs assuming 72 runs.

[b] Weight of the support point of post-hoc D-optimal design for Exponential model at nominal values fitted by data.

[c] Weight of the support point of the conditional D-optimal design for Poisson model.

[d] Weight of the support point of post-hoc D-optimal design for Poisson model at nominal values fitted by data.

powerful and flexible tool for finding locally D-optimal designs to estimate main effects of multiple agents and their interactions in an Exponential and Poisson model. The algorithm is metaheuristic and so I expect that it be applicable to find other optimal designs for different types of models with a more complicated mean structure as well. Our search strategy is new and unique in that it requires searches among candidate designs with many points than the anticipated number of support points in the optimal design. Interestingly, unlike typical optimal designs in low dimensional problems, the resulting locally D-optimal designs for our models have many more points than the number of parameters in the mean function.

Algorithms are key tools for finding optimal designs for more complex models in an increasingly high dimensional world. The traditional expectation is that algorithms have to be shown to converge theoretically. We argue that this is a desirable feature to have in the algorithm but can be limiting in applications. Silvey et al. [1978] rightly pointed out that "What is important about an algorithm is not whether it converges, but whether it is effective in the sense that it guarantees arbitrarily closeness to the optimum and how fast this approach is". I show here that the UPSO algorithm seems to meet this requirement quite well for designing the studies considered in the paper. I hope my work motivates others to further explore alternative optimization strategies developed and widely used in computer science and engineering fields to bear on solving statistical problems.

CHAPTER 4

Multi-objective Optimal Design for a Multivariate Model

In this chapter, I extend my work to find optimal designs for the continuation-ratio (CR) model. Such a model has great potential in dose finding studies because it considers both the efficacy and toxicity. Unlike other multivariate models, the CR model simultaneously models both probabilities of observing efficacy and adverse effect without having to assume the correlation between them.

The optimal design I am interested in constructing is a three-objective compound optimal design that provides efficient estimates for the most effective dose (MED), the maximum tolerated dose (MTD) for a user specified toxicity rate, and for all parameters in the CR model. I show that PSO can successfully find multi-objective compound optimal designs for the CR model. Further, I investigate the proper choice of weights for three optimal criteria in multi-objective designs under different parameters settings. By using efficiency plots, researchers and practitioners can construct the desired compound optimal design through appropriate weights combination of three optimal criteria in a more flexible and informative way.

4.1 Motivation

In the process of a drug discovery, efficacy and toxicity of the drug are the two most important endpoints. In phase I clinical trials, one of the main goals is to describe the dose-limiting toxicities (DLT) and estimate the maximum tolerated dose (MTD). Phase II clinical trials focus on providing more information about the efficacy of the drug in addition to evaluating the safety of the drug. For example, the most effective dose (MED) that produces the maximum efficacy is a common quantity of interest. For both economical and ethical reason, it is ideal to design a study which takes both efficacy and toxicity into consideration.

As shown in Cook and Wong [1994], we may find a multi-objective optimal design that incorporates various practical goals in an experiment, so that efficient estimates of multiple quantities of interest can be obtained simultaneously. Of course, multi-objective design does not imply equal interests on different objectives. In practice researcher may give more emphasize on one or two objectives as the primary, and treat the rest as the secondary objective. Such interest can be quantified as the efficiency of the multi-objective design for estimating the corresponding quantity. For example in a model based dose response study, a researcher wishes to find a three-objective optimal design such that the efficiencies for its two primary objectives, estimating MTD and MED, are both equal to or greater than 0.9. Under the condition that the two primary objectives are satisfied, the researcher also wants the design to maximize the precision for estimating all parameters in the model, which serves as the secondary objective. The question is, does such a design exist? If yes, how do we find it? For simple cases such as finding two-objective optimal designs for linear and quadratic polynomial regression models, Cook and Wong [1994] gave the analytical solution of such designs. However, there is no explicit solution of multi-objective designs for complicate models such as multivariate models. In addition, it seems that there is no algorithm can find a multi-objective optimal design for any user-specified weights combination of objective criteria.

Encouraged by the successes in finding locally D- and c-optimal designs for univariable and multivariable models in Chapter 2 and 3, I now apply PSO to find the multi-objective optimal design for a multivariate model. More importantly, I show PSO generated efficiencies plot can help researchers and practitioners construct the desired multi-objective optimal design in a more flexible and informative way.

4.2 Continuation-ratio (CR) models

There is a sizable literature that tried to simultaneously study efficacy and toxicity by postulating multivariate parametric models. For example, to model dose-response curves, Heise and Myers [1996] proposed a binary logistic model which simultaneously measures toxicity (yes/no), and efficacy (yes/no), assuming these two outcomes are correlated. A similar idea was presented by Fedorov and Wu [2007] who proposed a dichotomized outcome pair $(y_1, y_2)^T$ from a bivariate normal model, where y_1 is the efficacy response and y_2 is the toxicity response. It follows that, there are four possible outcomes and the utility function of interest is $\pi(x) = Pr(y_1 = 1, y_2 = 0|x)$. This model is complicated and difficult to use in practice because it requires user to specify the correlation structure between toxicity and efficacy. Alternatively, we may combine the two outcomes corresponding to the presence of toxicity together and reduce the four types of responses into three. Doing so greatly simplifies the model by eliminating the correlation between toxicity and efficacy. In the resulting trinomial models, the responses of patients are now exclusively and exhaustively classified into three categories: "no effect" (neither toxicity or efficacy found); "efficacy" with no toxicity; and "adverse reaction" for toxicity regardless of whether efficacy is presented or not.

To model the trinomial responses above, Thall and Russell [1998] proposed the following constant slope proportional odds (PO) model for a patient's response given dose x.

$$\begin{aligned}\log(\pi_3(x)/(1-\pi_3(x))) =& a_1 + bx\\ \log((\pi_2(x)+\pi_3(x))/\pi_1(x)) =& a_2 + bx,\end{aligned}$$

where $\pi_1(x)$, $\pi_2(x)$ and $\pi_3(x)$ correspond to, respectively, probabilities of observing "no reaction", "efficacy without toxicity" and "toxicity" at dose x. For any dose x, $\sum_{i=1}^{3} \pi_i(x) = 1$. The model assumes a constant effect of dose exists across the cumulative logits. Such assumption may be violated for a trinomial model, and is unlikely to be valid when there are more than three types of responses.

The continuation-ratio (CR) model is a more flexible alternative because it does not have the same assumption as the PO model does (Agresti, 1990, Chapter 9). In the CR model, slopes of the two equations b_1 and b_2 can be constant or not. Assuming $b_1, b_2 > 0$, the CR model is

$$\begin{aligned}\log(\pi_3(x)/(1-\pi_3(x))) =& a_1 + b_1x\\ \log(\pi_2(x)/\pi_1(x)) =& a_2 + b_2x.\end{aligned} \tag{4.1}$$

It is straightforward to show for this model, probability of "no reaction" is

$$\pi_1(x) = \frac{1}{(1+e^{a_1+b_1x})(1+e^{a_2+b_2x})};$$

probability of "efficacy without toxicity" is

$$\pi_2(x) = \frac{e^{a_2+b_2x}}{(1+e^{a_1+b_1x})(1+e^{a_2+b_2x})};$$

and probability of "toxicity" is

$$\pi_3(x) = \frac{e^{a_1+b_1x}}{1+e^{a_1+b_1x}}.$$

Fan and Chaloner [2004] were only interested in finding the dose x that maximizes probability of "efficacy without toxicity" $\pi_2(x)$, which is selected as the most effective dose (MED). If $b_1 = b_2$ then MED has a closed form given by

$$MED = -(a_1 + a_2)/2b_1,$$

but if $b_1 \neq b_2$, there is no closed form solution of MED. In their paper, several locally c-optimal designs were provided for estimating the MED under several nominal parameter settings.

In dose-response studies, side effects and drug toxicity may be very serious and need to be well controlled. One of the main targets in clinical trials is to find out the "dose that is closest to an acceptable level of toxicity" (Richards et al. [1997]). The target dose is defined as Maximum Tolerated Dose (MTD), which is the highest dose of a treatment agent that produces the dose limiting toxicity (DLT) in a proportion ρ of patients. In the CR model, it follows that

$$\pi_3(MTD, a_1, b_1) = \rho.$$

Here the probability ρ is user-specified and varies depending on what types of adverse effect is of interest. For life threatening side effects, ρ should be a small value. To find the MTD, practitioners usually start from a safe dose that produces a small ρ and gradually increase the dose to its highest acceptable level. The dose level at ρ is called lethal dose $LD100\rho$ (Zhu and Wong [2000]). For example, a commonly used one is the "median lethal dose" denoted by $LD50$. For both economical and ethical reason, it is necessary to take both MED and MTD (or $LD100\rho$) into consideration at the design stage.

4.3 Information matrix $I(\xi, \theta)$ of the CR model with $b_1 \neq b_2$

Consider a unit design δ_x which puts all weights on a dose x. Without loss of generality assume there is one subject assigned with the dose x, and the observed trinomial outcome for the subject is $y = (y_1, y_2, y_3)^T$ with $\sum_{i=1}^{3} y_i = 1$. Its expected value is $\mathrm{E}(y) = \pi(x)^T = (\pi_1(x), \pi_2(x), \pi_3(x))^T$ as previously defined. Denote the parameters in the CR model by $\theta = (a_1, b_1, a_2, b_2)^T$. To find the information matrix $I(\delta_x, \theta)$ of δ_x for the CR model, I use the method developed by Zocchi and Atkinson [1999] for multinomial logistic models. Denote the left hand side of (4.1) by

$$\eta(x) \;=\; \begin{bmatrix} \log(\pi_3(x)/(\pi_1(x)+\pi_2(x))) \\ \log(\pi_2(x)/\pi_1(x)) \\ \log(\pi_1(x)+\pi_2(x)+\pi_3(x)) \end{bmatrix}$$

and its right hand side by

$$X\theta = \begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \end{bmatrix}.$$

By the numerator-layout notation, we have

$$\begin{aligned} \frac{\partial \eta(x)}{\partial \pi(x)} &= \begin{bmatrix} \frac{\partial \eta(x)}{\partial \pi_1(x)} & \frac{\partial \eta(x)}{\partial \pi_2(x)} & \frac{\partial \eta(x)}{\partial \pi_3(x)} \end{bmatrix} \\ &= \begin{bmatrix} -(\pi_1(x)+\pi_2(x))^{-1} & -(\pi_1(x)+\pi_2(x))^{-1} & \pi_3(x)^{-1} \\ -\pi_1(x)^{-1} & \pi_2(x)^{-1} & 0 \\ 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

Therefore the derivative of $\pi(x)$ with respect to θ is derived by the chain rule

$$\begin{aligned} G(x) &= \frac{\partial \pi(x)}{\partial \theta} = \frac{\partial \pi(x)}{\partial \eta(x)} \frac{\partial \eta(x)}{\partial \theta} = (\frac{\partial \eta(x)}{\partial \pi(x)})^{-1} X \\ &= \begin{bmatrix} -\pi_1(x)\pi_3(x) & -\frac{\pi_1(x)\pi_2(x)}{\pi_1(x)+\pi_2(x)} & \pi_1(x) \\ -\pi_2(x)\pi_3(x) & \frac{\pi_1(x)\pi_2(x)}{\pi_1(x)+\pi_2(x)} & \pi_2(x) \\ (\pi_1(x)+\pi_2(x))\pi_3(x) & 0 & \pi_3(x) \end{bmatrix} \begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

For the dose x, since observation $y^T \sim Multinomial(1, \pi(x)^T)$, we get the likelihood function as

$$L(\theta; x) = \prod_{j=1}^{3} \pi_j^{y_j}(x),$$

and so the log-likelihood is $l(\theta;x) = y^T \log\pi(x)$, which leads to the score vector for x by the chain rule

$$\begin{aligned}
\frac{\partial l(\theta;x)}{\partial\theta} &= \frac{\partial l(\theta;x)}{\partial\log\pi(x)}\frac{\partial\log\pi(x)}{\partial\pi(x)}\frac{\partial\pi(x)}{\partial\theta} \\
&= y^T \begin{bmatrix} \pi_1^{-1}(x) & 0 & 0 \\ 0 & \pi_2^{-1}(x) & 0 \\ 0 & 0 & \pi_3^{-1}(x) \end{bmatrix} G(x),
\end{aligned}$$

and the unit information matrix $I(\delta_x, \theta)$ for the observation y at dose x is given by

$$\begin{aligned}
I(\delta_x,\theta) &= \mathrm{E}(\frac{\partial l(\theta;x)}{\partial\theta})^T(\frac{\partial l(\theta;x)}{\partial\theta}) \\
&= G^T(x) \begin{bmatrix} \pi_1^{-1}(x) & 0 & 0 \\ 0 & \pi_2^{-1}(x) & 0 \\ 0 & 0 & \pi_3^{-1}(x) \end{bmatrix} (\mathrm{E}yy^T) \\
&\quad \begin{bmatrix} \pi_1^{-1}(x) & 0 & 0 \\ 0 & \pi_2^{-1}(x) & 0 \\ 0 & 0 & \pi_3^{-1}(x) \end{bmatrix} G(x).
\end{aligned}$$

Since $y^T \sim Multinomial(1, \pi(x)^T)$, we have

$$\mathrm{E}yy^T = \mathrm{E}y\mathrm{E}y^T + Var(y) = \begin{bmatrix} \pi_1(x) & 0 & 0 \\ 0 & \pi_2(x) & 0 \\ 0 & 0 & \pi_3(x) \end{bmatrix},$$

and so,

$$\begin{aligned}
I(\delta_x,\theta) &= G^T(x)\begin{bmatrix} \pi_1^{-1}(x) & 0 & 0 \\ 0 & \pi_2^{-1}(x) & 0 \\ 0 & 0 & \pi_3^{-1}(x) \end{bmatrix} G(x) \\
&= \begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} (\pi_1(x)+\pi_2(x))\pi_3(x) & 0 & 0 \\ 0 & \frac{\pi_1(x)\pi_2(x)}{\pi_1(x)+\pi_2(x)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & x & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
&= (\pi_1(x)+\pi_2(x))\pi_3(x) \begin{bmatrix} 1 & x & 0 & 0 \\ x & x^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
&\quad + \frac{\pi_1(x)\pi_2(x)}{\pi_1(x)+\pi_2(x)} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & x \\ 0 & 0 & x & x^2 \end{bmatrix}).
\end{aligned}$$

For a $K-$point approximate design ξ as defined in Chapter 1, assume there are w_i proportion of subjects assigned with the dose x_i, $i = 1, 2, ..., K$. The normalized information matrix for the design ξ is

$$I(\xi,\theta) = \sum_{i=1}^{K} w_i I(\delta_{x_i},\theta)$$

4.3.1 Equivalence theorem for locally D-optimal design for CR models

Similarly as the univariate case, for multivariate models the directional derivative of $\Psi(I(\xi,\theta)) = \log|I(\xi,\theta)|$ at ξ in the direction of δ_x is

$$\psi(x,\xi,\theta) = \text{trace}[I(\delta_x,\theta)I^{-1}(\xi,\theta)] - P$$

where P is the number of parameters. In the CR model with $b_1 \neq b_2$, $P = 4$.

The multivariate version of equivalence theorem states that the following statements are equivalent (Fedorov et al. [2002]):

1. The design ξ^* maximizes $\Psi(I(\xi, \theta))$;

2. The design ξ^* minimizes $\max_{x \in \chi} \psi(x, \xi, \theta)$;

3. $\max_{x \in \chi} \psi(x, \xi^*, \theta) = 0$, and it achieves its maximum at the support points of the design.

4.4 Locally c-optimal designs for estimating the MED and MTD of the CR model with $b_1 \neq b_2$

Recall MED is the dose which maximizes π_2 the probability of having efficacy without toxicity. We derive MED by solving $\frac{d\pi_2(x)}{dx} = 0$. After a few steps of simplification, we get MED as the implicit solution of the following equation

$$g(x, \theta) = b_2(1 + \exp(-a_1 - b_1 x)) - b_1(1 + \exp(a_2 + b_2 x)) = 0.$$

To find the locally c-optimal design for estimating the MED, we need to get its gradient ∇_{MED} first. I follow Atkinson and Haines [1996]'s way to handle such equation $g(x, \theta)$ without an explicit solution for MED as follows.

By the implicit function theorem, if the function $g(x, \theta)$ has continuous first derivatives, and $MED(\theta)$ is continuous, then

$$\frac{\partial g(x,\theta)}{\partial \theta}|_{MED} = \frac{\partial g(x,\theta)}{\partial x}|_{MED} \frac{\partial x}{\partial \theta}|_{MED}$$

and so,

$$\begin{aligned} \nabla_{MED}^T &= \frac{\partial x}{\partial \theta}|_{MED} \\ &= \left[\frac{\partial g(x,\theta)}{\partial x}|_{MED}\right]^{-1} \frac{\partial g(x,\theta)}{\partial \theta}|_{MED} \end{aligned}$$

where

$$\frac{\partial g(x,\theta)}{\partial x}|_{MED} = -b_1 b_2(\exp(-a_1 - b_1 MED) + \exp(a_2 + b_2 MED)),$$

and

$$\frac{\partial g(x,\theta)}{\partial \theta}|_{MED} = \begin{bmatrix} -b_2\text{exp}(-a_1 - b_1MED) \\ -b_2MED\ \ \text{exp}(-a_1 - b_1MED) - (1+\text{exp}(a_2+b_2MED)) \\ -b_1\text{exp}(a_2+b_2MED) \\ 1+\text{exp}(-a_1-b_1MED) - b_1MED\,\text{exp}(a_2+b_2MED) \end{bmatrix}^T .$$

It follows that, the locally c-optimal design for estimating the MED is to maximize the following quantity

$$-\nabla^T_{MED}I^{-1}(\xi,\theta)\nabla_{MED}.$$

In the CR model, since

$$\log(\pi_3(x)/(1-\pi_3(x))) = a_1 + b_1x,$$

a simple calculation shows $MTD = \frac{1}{b_1}(\text{logit}(\rho) - a_1)$. Similarly, the locally c-optimal design for finding MTD is to maximize

$$-\nabla^T_{MTD}I^{-1}(\xi,\theta)\nabla_{MTD},$$

where $\nabla^T_{MTD} = \frac{\partial MTD}{\partial\theta} = (-\frac{1}{b_1}, \frac{a_1-\text{logit}(\rho)}{b_1^2}, 0, 0)$. In the traditional dose finding designs, MTD is usually set as the dose level at which two of six patients experienced toxicity, see Fumoleau et al. [2013] for example. Therefore in the rest of the chapter, I set $\rho = 0.3$, but other values can be directly used as well.

4.5 Equivalence theorem of the locally c-optimal design for estimating the MED

Fedorov et al. [2002] established the equivalence theorem to verify the optimality of a design for a multivariate model under the criterion $\Psi(I(\xi,\theta)) = \text{trace}[AI^{-1}(\xi,\theta)]$. I apply it here to find the locally c-optimal design for estimating the MED. The directional derivative of $\Psi(I(\xi,\theta)) = -\nabla^T_{MED} I^{-1}(\xi,\theta)\nabla_{MED}$ = -trace$[\nabla_{MED}\nabla^T_{MED} I^{-1}(\xi,\theta)]$ at ξ in the direction of δ_x is

$$\begin{aligned}\psi(x,\xi,\theta) &= \text{trace}[I(\delta_x,\theta)I^{-1}(\xi,\theta)(\nabla_{MED}\nabla^T_{MED})I^{-1}(\xi,\theta)] \\ &\quad -\text{trace}[(\nabla_{MED}\nabla^T_{MED})I^{-1}(\xi,\theta)].\end{aligned}$$

When ξ is optimal, $\psi(x,\xi,\theta)$ is bounded above by 0 with equality at the support points. Similarly we can derive the equivalence theorem for locally c-optimal design to estimate other quantities such as MTD.

4.6 Equivalence of compound and constrained optimal designs

In practice, there are a few objectives in a study and it is desirable to incorporate them at the design stage. There are two typical approaches to construct a multiple-objective design: the compound and constrained optimal design (Cook and Wong [1994]). Because different objectives $\Psi_i(I(\xi,\theta))$ $(i = 1, ..., m)$ may have different magnitude in their values, we define each $\Psi_i(I(\xi,\theta))$ in terms of the design efficiency $e_i(\xi)$ as did in Cook and Wong [1994]. For example, the D-efficiency for an arbitrary design ξ is defined by $e_1(\xi) = (\frac{|I(\xi,\theta)|}{|I(\xi_D,\theta)|})^{1/P}$, and its c-efficiency is $e_2(\xi) = \frac{\nabla_c^T I^{-1}(\xi_c,\theta)\nabla_c}{\nabla_c^T I^{-1}(\xi,\theta)\nabla_c}$, where ξ_D is the locally D-optimal design and ξ_c is the locally c-optimal design, ∇_c is the gradient of $c(\theta)$ with respect to θ. To find a multi-objective compound optimal designs, we consider a concave functional of these efficiencies and maximize each of them.

Suppose there are two competing objectives Ψ_1 and Ψ_2 of interest. Denote Δ_s the set of designs ξ that maximize $\Psi_2(I(\xi,\theta))$ subject to the constraint that $\Psi_1(I(\xi,\theta)) \geq s$, where s is a user-selected constant. The constrained optimal design is defined by

$$\xi_s = \arg\max_{\xi\in\Delta_s} \Psi_1(I(\xi,\theta)),$$

where the maximum is taken over the entire Δ_s.

The two-objective compound optimal design, for any user-selected constant $\lambda \in [0,1]$, is to find a design ξ_λ that maximizes the weighted average of two concave functionals

$$\Psi(\xi|\lambda) = \lambda\Psi_1(I(\xi,\theta)) + (1-\lambda)\Psi_2(I(\xi,\theta)).$$

Comparing to the constrained optimal design, the compound optimal design is easier to determine. This is because a concave combination of concave functionals is still concave for a fixed λ, and the equivalence theorem is just a weighted mean of the checking conditions for the optimal design for each objective. However, the weighting parameter λ may be hard to interpret in practice. Question is posed like if we set $\lambda = 0.5$, does it mean we put an equal emphasize on two objectives $\Psi_1(\xi)$ and $\Psi_2(\xi)$?

For linear models, Cook and Wong [1994] pointed out that for a two-objective compound optimal design ξ_λ that maximizes $\lambda\Psi_1(I(\xi,\theta)) + (1-\lambda)\Psi_2(I(\xi,\theta))$, it is equivalent that ξ_λ maximizes $\Psi_2(I(\xi,\theta))$ subject to $\Psi_1(I(\xi,\theta)) \geq \Psi_1(I(\xi_\lambda,\theta))$ which is the primary objective. Therefore the process of finding a multi-objective optimal design is to first formulate the optimal design problem as a constrained design ξ_s, as then find out the λ such that its compound optimal design ξ_λ is equivalent to the constrained optimal design ξ_s.

Clyde and Chaloner [1996] further extended the work to nonlinear models for three or more objective criteria. Consider a three objective compound optimal designs ξ_λ maximizing

$$\Psi(\xi|\lambda) = \lambda_1\Psi_1(I(\xi,\theta)) + \lambda_2\Psi_2(I(\xi,\theta)) + \lambda_3\Psi_3(I(\xi,\theta)),$$

where $\lambda = (\lambda_1, \lambda_2, \lambda_3)^T$, λ_i's $\in [0,1]$ and $\sum_{i=1}^3 \lambda_i = 1$. It is equivalent that ξ_λ maximizes $\Psi_3(I(\xi,\theta))$ subject to $\Psi_1(I(\xi,\theta)) \geq \Psi_1(I(\xi_\lambda,\theta))$ and $\Psi_2(I(\xi,\theta)) \geq \Psi_2(I(\xi_\lambda,\theta))$.

4.7 Three-objective locally optimal design for the CR model via PSO

In Fan and Chaloner [2004], only MED was of interest and the authors focused on finding locally c-optimal design for estimating the MED. This study extends their work to construct a design that efficiently estimates MED and MTD simultaneously. Other parameters such as b_1 and b_2 determine the slopes of three possible curves and so it is important to make sure the generated optimal design provides good estimates for all parameters in the CR model. Accordingly, in addition to estimating MED and MTD, I incorporate D-optimality as well. For a given λ, I construct a three-objective compound optimal design by maximizing the weighted sum of three concave functionals

$$\Psi(\xi|\lambda) = \lambda_1\Psi_1(I(\xi,\theta)) + \lambda_2\Psi_2(I(\xi,\theta)) + \lambda_2\Psi_3(I(\xi,\theta))$$

where $\lambda = (\lambda_1, \lambda_2, \lambda_3)^T$, λ_i's $\in [0,1]$ and $\sum_{i=1}^{3} \lambda_i = 1$. Here $\Psi_1(I(\xi,\theta))$ is the log c-efficiency of the design ξ for estimating the MTD

$$\begin{aligned} \Psi_1(I(\xi,\theta)) &= \log(\frac{\nabla_{MTD}^T I^{-1}(\xi_{MTD},\theta)\nabla_{MTD}}{\nabla_{MTD}^T I^{-1}(\xi,\theta)\nabla_{MTD}}) \\ &= \log(e_1), \end{aligned}$$

$\Psi_2(I(\xi,\theta))$ is log c-efficiency of the design ξ for estimating the MED

$$\begin{aligned} \Psi_2(I(\xi,\theta)) &= \log(\frac{\nabla_{MED}^T I^{-1}(\xi_{MED},\theta)\nabla_{MED}}{\nabla_{MED}^T I^{-1}(\xi,\theta)\nabla_{MED}}) \\ &= \log(e_2), \end{aligned}$$

and $\Psi_3(I(\xi,\theta))$ is log D-efficiency of the design ξ for estimating all parameters in the CR model

$$\begin{aligned} \Psi_3(I(\xi,\theta)) &= \log(\frac{|I(\xi,\theta)|}{|I(\xi_D,\theta)|})^{\frac{1}{P}} \\ &= \log(e_3). \end{aligned}$$

The checking condition for three-objective compound optimal design is just a weighted sum of the checking conditions for the optimal design for each objective, i.e. we have ξ is optimal for the three-objective problem if and only if for all $x \in \chi$

$$\begin{aligned} &\lambda_1 \frac{\text{trace}[I(\delta_x, \theta) I^{-1}(\xi, \theta)(\nabla_{MTD} \nabla_{MTD}^T) I^{-1}(\xi, \theta)]}{\text{trace}[(\nabla_{MTD} \nabla_{MTD}^T) I^{-1}(\xi, \theta)]} \\ &+\lambda_2 \frac{\text{trace}[I(\delta_x, \theta) I^{-1}(\xi, \theta)(\nabla_{MED} \nabla_{MED}^T) I^{-1}(\xi, \theta)]}{\text{trace}[(\nabla_{MED} \nabla_{MED}^T) I^{-1}(\xi, \theta)]} \\ &\qquad +\lambda_3 \frac{\text{trace}[I(\delta_x, \theta) I^{-1}(\xi, \theta)]}{P} \leq 1, \end{aligned}$$

with equality at support points.

Fox each fixed λ in [0,1], I use the default parameter settings in the standard PSO as described in Chapter 1 to find the three-objective compound optimal design ξ_λ for the CR model. Particles that wander outside of the design space are pulled back to its nearest boundary $[lb, ub]$:

$$z_i^{(t)} = \begin{cases} ub & \text{if } z_{i,}^{(t)} > ub \\ lb & \text{if } z_i^{(t)} < lb \\ z_i^{(t)} & \text{otherwise} \end{cases}$$

where $z_i^{(t)}$ is the particle i's position at time t. The rationale for our strategy is that optimal designs frequently have support points at the boundary of the design space, see Li and Majumdar [2009] for example. PSO successfully verifies all locally D-optimal and c-optimal designs for the CR model with different nominal values in Fan and Chaloner [2004].

Furthermore, as mentioned by Zhu and Wong [2000], the locally c-optimal design for finding $LD100\rho$ is a single point design supported at $LD100\rho$. Consider the c-optimal design for estimating the $LD30$ in the CR model with nominal values $(a_1 = -3.3, b_1 = 0.5)$. PSO uses 100 particles and 1000 iterations to find the one point design located at point 4.905, which can be verified by $LD30 = \frac{1}{0.5}(\text{logit}(0.3) + 3.3) = 4.9054$. Encouraged by these successes, I don't use the

Ultra-dimensional PSO that I developed for multivariable Poisson and exponential models. Using the standard PSO, I find that all compound optimal designs for the CR model are supported at four or fewer points.

Three-objective compound optimal designs are successfully generated by PSO for several sets of nominal values of parameters and objective weights λ_i's. The three sets of nominal values I choose as examples here represent typical types of dose response curves, i.e. the maximum efficacy varies from 0.3 to 0.9, see Figure 4.1, 4.2 and 4.3. In each figure, probability curves are shown in the left plot, and the middle one is the equivalence plot for compound optimal designs in the unrestricted design space. Table 4.1 shows some results of compound optimal designs with equal weights λ_i's=1/3 for all criteria on the unrestricted designs space. We can see that generated optimal designs are supported at three or four points, and the weights of these support points are not equal.

To further investigate the influence of design space to the compound optimal design, I shrink the design space of these three cases to a smaller asymmetric space [-2,7]. PSO is still able to find the compound optimal designs for all three cases, as shown in Table 4.2. Interestingly, the previous four point optimal designs become three points, with one or two points located on the boundaries of the restricted design space. Also the middle support point is not the same as the ones in optimal designs on unrestricted design space. All the equivalence plots of generated compound optimal designs on the restricted space are shown in the right plots of Figure 4.1, 4.2 and 4.3, which prove that all the PSO generated three-objective designs are optimal among all designs on [-2,7].

4.8 Different efficiencies of the compound optimal design ξ_λ

Let us revisit the question posed at the beginning of the chapter. Can we find a three-objective optimal design ξ for the CR model such that its D-efficiency $e_3(\xi)$

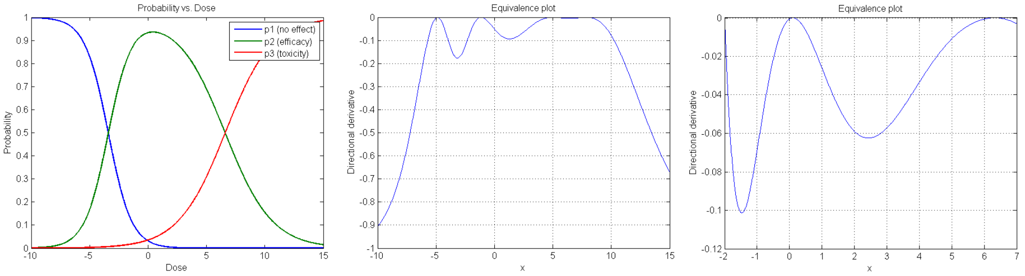


Figure 4.1: CR model with nominal values: $(a_1 = -3.3, b_1 = 0.5, a_2 = 3.4, b_2 = 1)$.

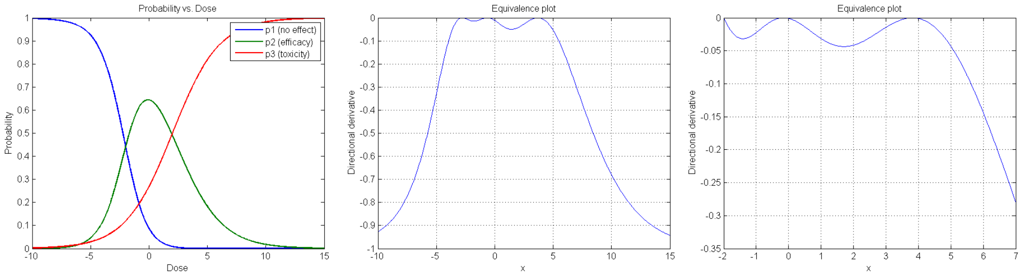


Figure 4.2: CR model with nominal values: $(a_1 = -1, b_1 = 0.5, a_2 = 2, b_2 = 1)$.

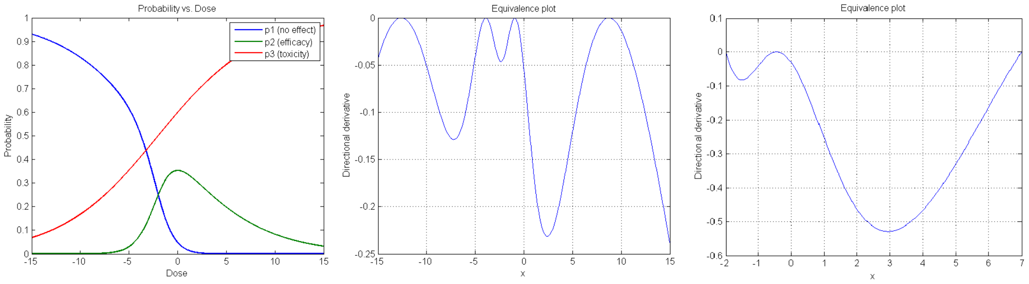


Figure 4.3: CR model with nominal values: $(a_1 = 0.4, b_1 = 0.2, a_2 = 2, b_2 = 1)$.

Mean Responses from the CR model with corresponding nominal values (left); directional derivatives of the compound criterion evaluated at the PSO-generated multiple-objective design on the unrestricted design space (middle); and directional derivatives of the compound criterion evaluated at the PSO-generated multiple-objective design on the design space [-2,7] (right).

Table 4.1: Three-objective compound optimal designs for estimating the MTD,MED and all parameters with $\rho = 0.3$, λ_1 =λ_2 = 1/3 on the unrestricted designs space.

(a_1, b_1, a_2, b_2)	$x_1(w_1)$	$x_2(w_2)$	$x_3(w_3)$	$x_4(w_4)$
(-3.3, 0.5, 3.4, 1)	-4.875 (0.102)	-1.139 (0.464)	5.016 (0.322)	7.874 (0.112)
(-1, 0.5, 2, 1)	-2.790 (0.202)	-0.637 (0.513)	3.683 (0.284)	-
(0.4, 0.2, 2, 1)	-12.610 (0.366)	-3.918 (0.158)	-0.942 (0.470)	8.727 (0.006)

Table 4.2: Three-objective compound optimal designs for estimating the MTD,MED and all parameters with $\rho = 0.3$, λ_1 =λ_2 = 1/3 on $\chi = [-2, 7]$.

(a_1, b_1, a_2, b_2)	$x_1(w_1)$	$x_2(w_2)$	$x_3(w_3)$
(-3.3, 0.5, 3.4, 1)	-2.000 (0.152)	0.1045 (0.502)	6.328 (0.345)
(-1, 0.5, 2, 1)	-2.000 (0.330)	-0.156 (0.403)	3.820 (0.267)
(0.4, 0.2, 2, 1)	-2.000 (0.356)	-0.438 (0.319)	7.000 (0.325)

is maximized subject to both its c-efficiencies $e_1(\xi)$, $e_2(\xi)$ for MTD and MED are equal to or great than 0.9? Such a design may or may not exist depending on how competitive the criteria are and also the efficiency requirements in the constraints. If it exists, what weighting parameter λ_i's should be chosen in the compound optimal design?

To answer the question we need to investigate the impact of different combinations of λ_1 and λ_2 on the efficiencies of the compound optimal design ξ_λ relative to the three criteria. I first discretize λ_1 and λ_2 using a grid size of 0.05, and find the corresponding compound optimal design ξ_λ for each pair of λ_1 and λ_2 combination subject to $\lambda_1 + \lambda_2 \leq 1$. The efficiencies $e_i(\xi_\lambda)$ under the ith criterion are displayed in Figure 4.4, 4.5 and 4.6 for different nominal values sets. In these figures, the upper left plot shows the c-efficiencies of $e_1(\xi_\lambda)$ as measured by the locally c-optimal design ξ_{MTD} for MTD; the upper right plot shows c-efficiencies of $e_2(\xi_\lambda)$ relative to the locally c-optimal design ξ_{MED} for MED; the bottom left plot shows D-efficiencies of $e_3(\xi_\lambda)$; bottom right shows minimal efficiencies of all three efficiencies.

These efficiency plots provide an answer to the question posed at the beginning of the section. The answer is no, at least for the three sets of nominal values I show here; there is no compound optimal design ξ_λ that maximizes D-optimality with subject to $e_1(\xi_\lambda) \geq 0.9$ and $e_2(\xi_\lambda) \geq 0.9$ no matter what combination of λ_i's we choose. By observing the efficiency plots of three criteria under different nominal values sets, I find $e_1(\xi_\lambda)$ and $e_2(\xi_\lambda)$ seem to be very competitive with each other (See Figure 4.4 for example). When one criterion gains, the other loses substantially, and vice versa.

For each set of the nominal values, I find out the combination of λ_i's that produces the maximum of the minimum efficiencies, as shown in Table 4.3. For example, $\lambda_1 = 0.55, \lambda_2 = 0.35, \lambda_3 = 0.1$ is the weights combination that produces the maximum of the minimum efficiency as 0.63 for all three criteria. That means with this choice for λ, we can find a compound optimal design ξ_λ maximizes the precision for estimating all parameters and at the same time has at least 63% efficiency for estimating the MED and MTD.

In addition, I find that for all the three nominal values sets, the compound optimal designs maintain high D-efficiency for most of the combinations of λ_i's. This indicates for those nominal values that consistently produce high D-efficiency for different criteria weight λ_i's, we may reduce the three-objective compound optimal design to two-objective for estimating the MTD and MED only so that the optimal design problem is simplified.

Table 4.3: Efficiencies of ξ_λ as measured by ξ_D,ξ_{MTD} and ξ_{MED} .

(a_1, b_1, a_2, b_2)	λ_1	λ_2	λ_3	$e_1(\xi_\lambda)$	$e_2(\xi_\lambda)$	$e_3(\xi_\lambda)$
(-3.3, 0.5, 3.4, 1)	0.55	0.35	0.1	0.63	0.63	0.87
(-1, 0.5, 2, 1)	0.7	0.3	0	0.82	0.85	0.88
(0.4, 0.2, 2, 1)	0.6	0.4	0	0.75	0.75	0.81

4.9 Summary

In this chapter, I generalize the work of Fan and Chaloner [2004] and Zhu and Wong [2000] to find the three-objective design for the CR model. I show PSO is a powerful and flexible tool to find a three-objective compound optimal design for simultaneously estimating MED that produces maximal efficacy, the maximum tolerated dose (MTD) and all parameters in the CR model with non-constant slope $b_1 \neq b_2$. I am able to find the compound optimal designs under different combinations of weights λ, and investigate the impact of λ on efficiencies relative to different criteria. Such technique enables researchers and practitioners to construct the desired compound optimal design through appropriate weights combination of three optimal criteria in a more flexible and informative way.

Of course, the techniques developed here are broadly applicable to other criteria and other nonlinear models. The PSO codes that I provided can be directly amended to find other multiple-objective optimal design problems.

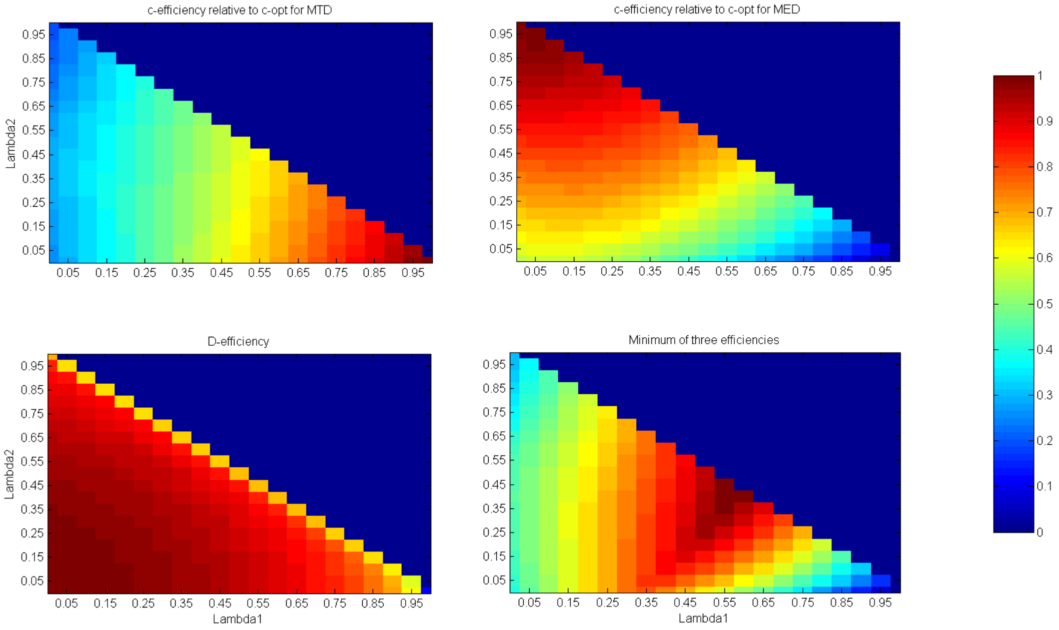


Figure 4.4: Different efficiency plots of compound optimal designs for the CR model with nominal values: $(a_1 = -3.3, b_1 = 0.5, a_2 = 3.4, b_2 = 1)$.

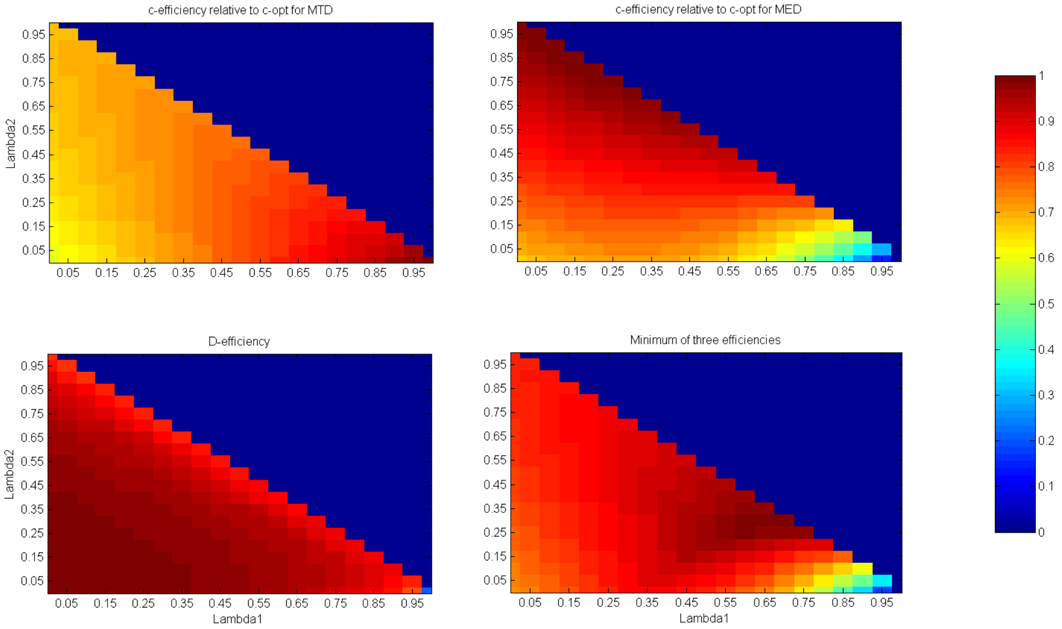


Figure 4.5: Different efficiency plots of compound optimal designs for the CR model with nominal values: $(a_1 = -1, b_1 = 0.5, a_2 = 2, b_2 = 1)$.

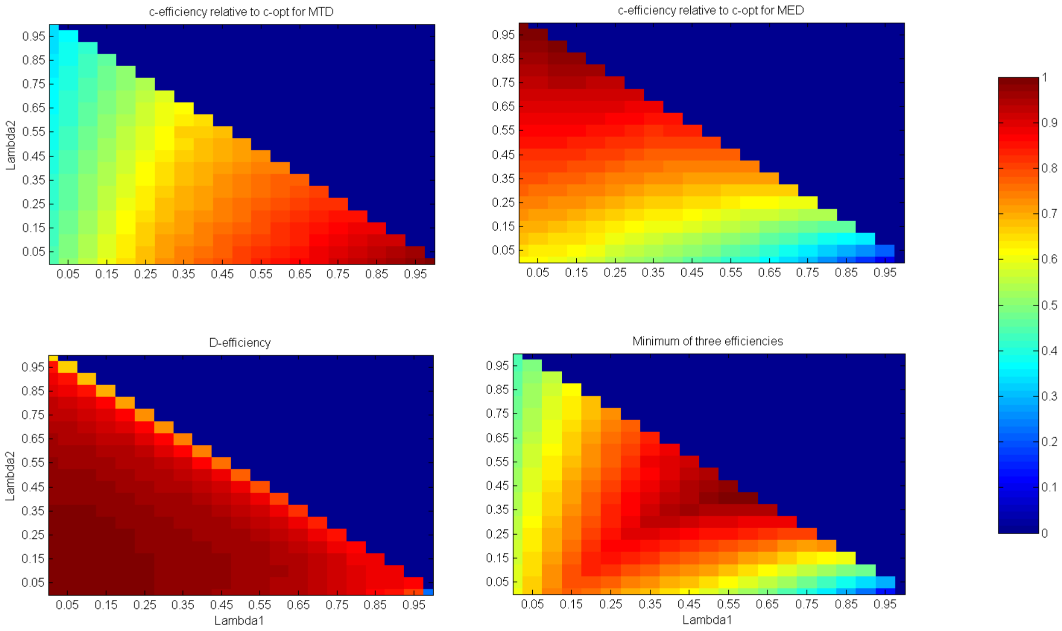


Figure 4.6: Different efficiency plots of compound optimal designs for the CR model with nominal values: $(a_1 = 0.4, b_1 = 0.2, a_2 = 2, b_2 = 1)$.

CHAPTER 5

Comparison of Repair Mechanisms in PSO and Comparisons of Competitive Algorithms

In previous chapters, I discussed using PSO to find locally D- and c-optimal designs for univariate models, multivariable models, and a multivariate model with different nominal values of parameters. The computational experience I have with these problems are similar to what is reported in the literature. Many parameters such as γ_1,γ_2, and inertia weight ω in the PSO do not seem to matter much in terms of searching efficiency. Following convention, I use default parameters of the standard PSO algorithm in all the examples. The two parameters that I change from problem to problem are the number of iterations and the flock size. For complicated problems such as multivariable Poisson and Exponential models, large flock size and iteration numbers are required to ensure the generated design has a high efficiency. In addition to these two parameters, I find another factor that may influence the searching efficiency of PSO in the optimal design problems we are working with here, is the repair mechanism. This refers to how PSO handles particles that wander outside the design space and prevent generating infeasible solutions. In the next section, I discuss two main types of repair mechanisms in detail and compare the performance of PSO using these two mechanisms to find the locally D- and c-optimal designs for a variety of models.

There is a famous impossibility theorem in the optimization area called No Free Lunch (NFL) Theorem, proposed by Wolpert and Macready [1997]. It states that there is no algorithm that performs universally better than any other algorithms. In other words, any algorithm's average performance on the space of all possible problems is equivalent to or no better than a random search, despite some algorithms do perform better on some specific problems. In the second part of this chapter, I compare the performance of PSO with other competing algorithms. I am particularly interested in two popular algorithms: the Cocktail algorithm (CA) by Yu [2011] and the Differential Evolution (DE) algorithm by Storn and Price [1997]. CA is a deterministic algorithm while DE is a metaheuristic algorithm like PSO. CA works fast except that it can only find locally D-optimal designs. Yu [2011] showed CA performs much faster than each of its component algorithms for several univariable models and also for a bivariable linear model. DE is typically viewed as a competitor to PSO, and seems to be a popular optimization tool for solving engineering problems. I compare the performance of three algorithms in terms of CPU time to find locally D- and c-optimal designs for different models. Moreover, since CA works on a discretized design space, rather than a continuous space that PSO and DE work on, the number n of discretized points (grid size) also plays an important role in the quality of CA generated design and the CPU times. Therefore I also compare the performance of CA using different grid sizes to see if we obtain better designs by using larger grid sizes.

I remind readers that all algorithms are coded in MATLAB 2013a, and all are implemented on a workstation with Intel i7-4770 and 16GB ram.

5.1 Repair mechanisms comparison in PSO

Conventionally, there are two types of repair mechanisms employed in literature: random repair and boundary repair (Zhang et al. [2004]). Assume PSO searches a J-dimensional design space $\prod_{j=1}^{J}[lb_j, ub_j]$, where ub_j and lb_j are the upper and lower bounds of the j^{th} dimension. The i^{th} particle's position at time t is a vector $z_i^{(t)} = (\ z_{i,1}^{(t)} \ \ z_{i,2}^{(t)} \ \ ... \ \ z_{i,J}^{(t)} \ )^T$. The random repair, as the name indicates, randomly assigns a position within the design space when a particle wanders outside of the design space,

$$z_{i,j}^{(t)} = \begin{cases} rand(lb_j, ub_j) & \text{if } z_{i,j}^{(t)} > ub_j \\ rand(lb_j, ub_j) & \text{if } z_{i,j}^{(t)} < lb_j \\ z_{i,j}^{(t)} & \text{otherwise} \end{cases}$$

where $z_{i,j}^{(t)}$ is the j^{th} component of the i^{th} particle's position at time t, and $rand(lb_j, ub_j)$ is a random variable from a uniform distribution on $[lb_j, ub_j]$. This repair mechanism has two direct effects on the swarm movements: a) increasing the i^{th} particle's velocity $v_i^{(t)}$ at time t; and b) increasing $|p_{i,j} - z_{i,j}^{(t)}|$ and $|p_{g,j} - z_{i,j}^{(t)}|$. Both effects increase the energy of swarm, and "disturb the swarm into chaos state", thereby slow down the convergence speed to the global optimum (Zhang et al. [2004]).

Another type of repair strategy in PSO is boundary repair. This strategy pulls errant particles back to the nearest boundary of the design space,

$$z_{i,j}^{(t)} = \begin{cases} ub_j & \text{if } z_{i,j}^{(t)} > ub_j \\ lb_j & \text{if } z_{i,j}^{(t)} < lb_j \\ z_{i,j}^{(t)} & \text{otherwise} \end{cases}$$

The effects of boundary repair on swarm movements are opposite to what random repair does: a) decreasing the velocity $v_i^{(t)}$; and b) decreasing $|p_{i,j} - z_{i,j}^{(t)}|$ and $|p_{g,j} - z_{i,j}^{(t)}|$. Both effects directly decrease the energy of swarm. The boundary of the design space acts as a quasi-gravity center that attracts particles current and following positions until the population best position is no longer at the boundary. Such repair mechanism "accelerates the swarm into equilibrium state and may lead to the premature convergence." (Zhang et al. [2004]). In our specific optimal design problems such as finding locally D- and c-optimal designs, optimal designs frequently have support points at or near the boundary of the design space, see examples in Li and Majumdar [2009] and my dissertation. Thereby boundary repair may provide desirable effects on swarm movements in finding optimal designs.

To compare the performance of the two repair mechanisms, I conduct simulation studies to find locally D-optimal designs for the following models.

Univariable models:

a) Compartmental model

$$\mathrm{E}y = \theta_3(\exp(-\theta_2 x) - \exp(-\theta_1 x)), \;\; x \in \chi = [0, 20]$$

with nominal values of $\theta_1 = 4.298, \theta_2 = 0.05884, \theta_3 = 21.8$.

b) Logistic quadratic model

$$\log(\frac{\mathrm{E}y}{1-\mathrm{E}y}) = \alpha + \beta(x-\mu)^2, \;\; x \in \chi = [-1, 1]$$

with nominal values of $\alpha = 3, \beta = -5, \mu = 0$.

c) Heteroscedastic 4-parameter Hill model

$$\mathrm{E}y = \frac{(E_{con} - B)(\frac{x}{IC_{50}})^m}{1 + (\frac{x}{IC_{50}})^m} + B + \varepsilon, \;\; \varepsilon \sim N(0, \sigma_0 y^{2\lambda}) \text{ and } x \in \chi = [0, 453]$$

with nominal values of $E_{con} = 1.7, B = 0.137, IC_{50} = 0.453, m = -0.825, \lambda = 3$. Without loss of generality, I set $\sigma_0 = 1$.

Multivariable models:

d) Bivariable linear model

$$\mathrm{E}y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{12} x_1 x_2, \;\; x \in \chi = [-1, 1] \times [0, 1]$$

with nominal values of β_i's =1 on $\chi = [-1, 1] \times [0, 1]$. Note this model is linear therefore locally D-optimal design is independent of β's.

e) 3-variable Poisson model

$$\mathrm{E}y_k = \exp(\beta_0 + \sum_{m=1}^{3} \beta_m x_{km} + \sum_{m=1}^{2} \sum_{m'>m}^{3} \beta_{mm'} x_{km} x_{km'})$$

with all nominal values of $r_{mm'} = \frac{\beta_m \beta_{m'}}{\beta_{mm'}} = r = 0$ or $r_{mm'} = r = -5$ on IED space where $q_{km} = \exp(\beta_m x_{km}) \in [0.01, 1]$ or $[0.1, 1]$.

Multivariate model:

f) CR model

$$\begin{aligned} \log(\pi_3(x)/(1-\pi_3(x))) =& a_1 + b_1 x \\ \log(\pi_2(x)/\pi_1(x)) =& a_2 + b_2 x, \;\; x \in \chi = [-10, 10] \end{aligned}$$

with constant slope $a_1 = 0, b_1 = 1, a_2 = 5, b_2 = 1$; or non-constant slopes $a_1 = -3.3, b_1 = 0.5, a_2 = 3.8, b_2 = 1$.

I also compare the performance of the two repair mechanisms for finding c-optimal designs for the following quantities:

g) Most effective dose (MED) of non-constant slope CR model in $f)$ where MED is the solution of the following equation

$$b_2(1 + \exp(-a_1 - b_1 x)) - b_1(1 + \exp(a_2 + b_2 x)) = 0.$$

h) Area under the curve (AUC) of compartmental model in $a)$, where

$$AUC = \frac{1}{\theta_1} - \frac{1}{\theta_2}.$$

i) Time to maximum concentration t_{max} of compartmental model in $a)$, where

$$t_{max}(\theta) = \frac{\log\theta_1 - \log\theta_2}{\theta_1 - \theta_2}.$$

In the simulation studies, for locally D-optimal designs I set the fitness value as $\log(I(\xi, \theta))$;and for c-optimal designs, the fitness value is $-\log(-\nabla^T I^{-1}(\xi, \theta)\nabla)$. For PSO with random or boundary repair, I repeat their searching for locally D- and c-optimal designs for models $a)$ to $i)$ for 20 times. In each replicate, I record the best fitness values found by the entire flock in every iteration. The average best fitness values over 20 replicates are plotted against the number of iterations for the two repair mechanisms. In all the figures in this section, the solid curves are generated by PSO with boundary repair and the dotted curves are generated by PSO with random repair.

For finding locally D-optimal designs for univariable models, Figures 5.1, 5.2 and 5.3 show that PSO with boundary repair consistently coverages to the optimum faster than PSO with random repair. For multivariable models including bivariable linear model d) and 3-variable Poisson model e), the difference of performance of the two types of PSO is even greater than that in univariable models (Figure 5.4, 5.5 and 5.6). For these complicated multivariable models, my experience is that PSO with random repair is even unable to find the optimal design regardless how many iterations and flock size I choose.

When searching for c-optimal designs for estimating the AUC and t_{max} in compartmental model in a), both Figures 5.9 and 5.10 show that PSO with boundary repair consistently finds better fitness values faster than PSO with random repair.

For CR models, by the comparisons of locally D-optimal designs for constant (Figure 5.7) and non-constant slope model (Figure 5.8), and c-optimal design for MED (Figure 5.11), I observe that although PSO with boundary repair still outperforms the other, the difference in their performance seems to be increasing smaller as PSO progresses.

5.2 Comparisons of algorithms

Our results in the previous section suggests generally PSO with boundary repair performs better than PSO with random repair in searching for locally D- and c-optimal designs. In the following subsections, I compare the performance between PSO with boundary repair with two of its main competitors: DE and CA. DE is a metaheuristic algorithm like PSO, which is able to find both locally D- and c-optimal designs. Therefore, in section 5.2.1 I compare PSO with DE for finding

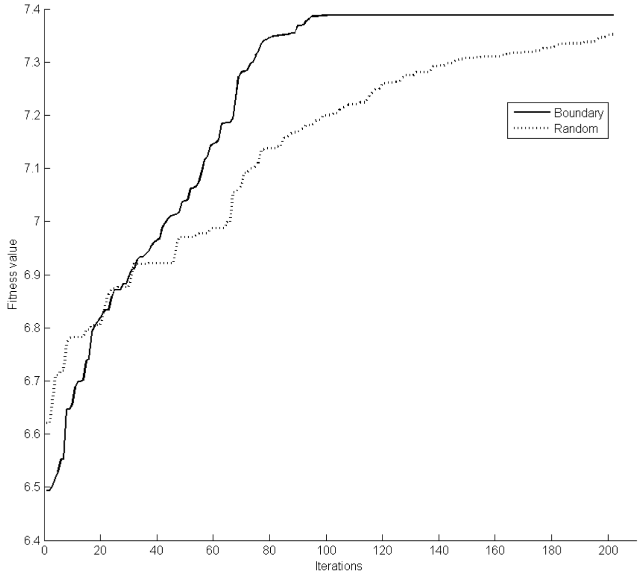


Figure 5.1: Repair mechanisms comparison of the locally D-optimal design for the compartmental model with $a = 4.298, b = 0.05884, c = 21.8$ on $\chi = [0, 20]$.

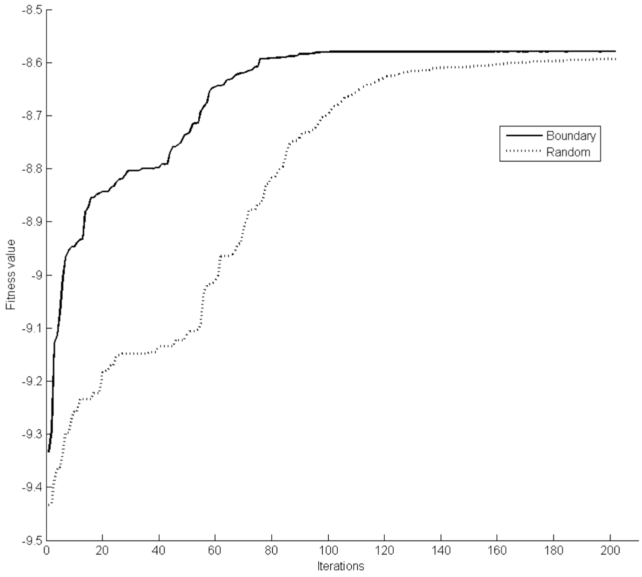


Figure 5.2: Repair mechanisms comparison of the locally D-optimal design for the logistic quadratic model with $\alpha = 3, \beta = -5, \mu = 0$ on $\chi = [-1, 1]$.

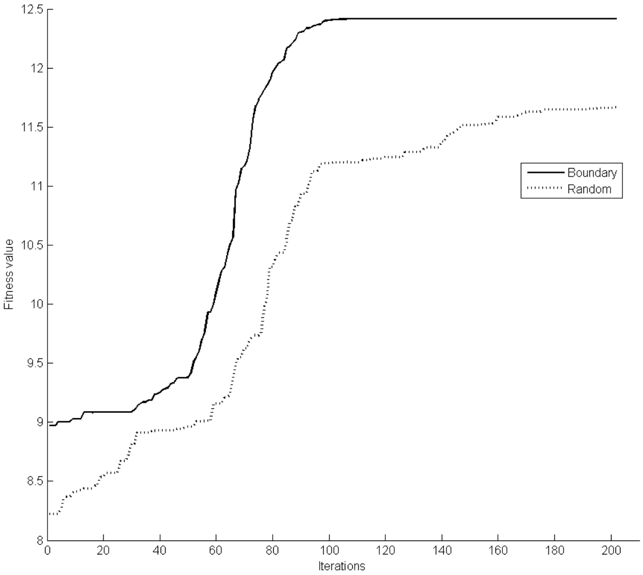


Figure 5.3: Repair mechanisms comparison of the locally D-optimal design for the 4-parameter Hill model with $Ec = 1.7, b = 0.137, IC = 0.453, m = -0.825, \lambda = 3$ on $\chi = [0, 453]$.

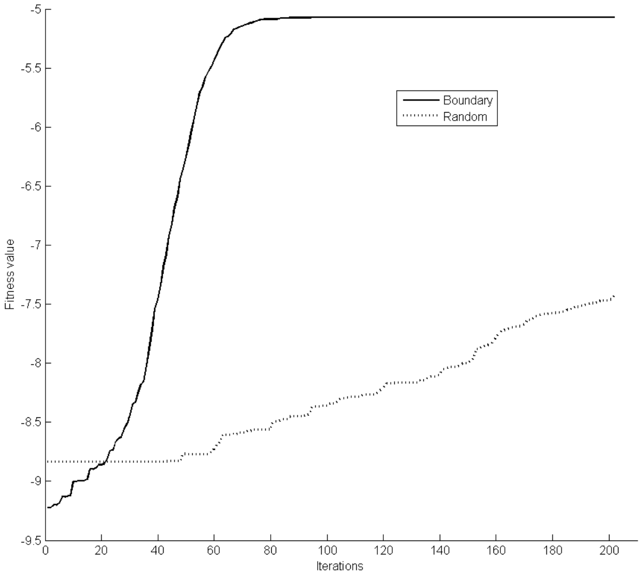


Figure 5.4: Repair mechanisms comparison of the locally D-optimal design for the 2-variable linear model on $\chi = [-1, 1] \times [0, 1]$.

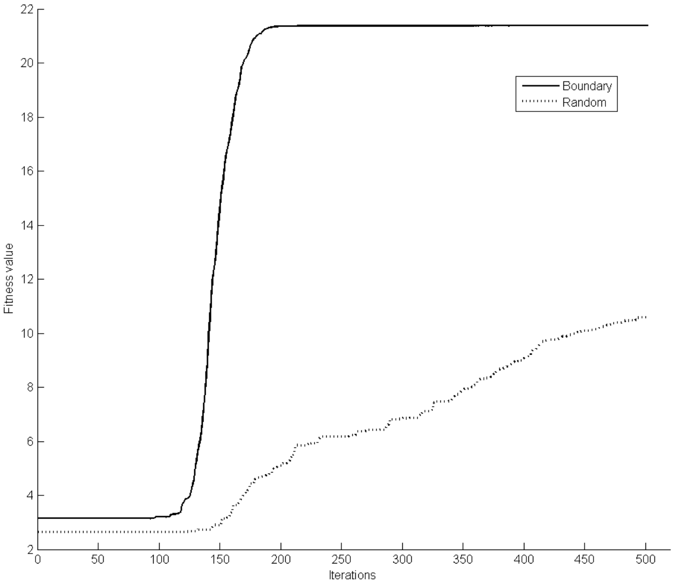


Figure 5.5: Repair mechanisms comparison of the locally D-optimal design for the 3-variable Poisson model with $r = 0$ on IED space $[0.01, 1]^3$.

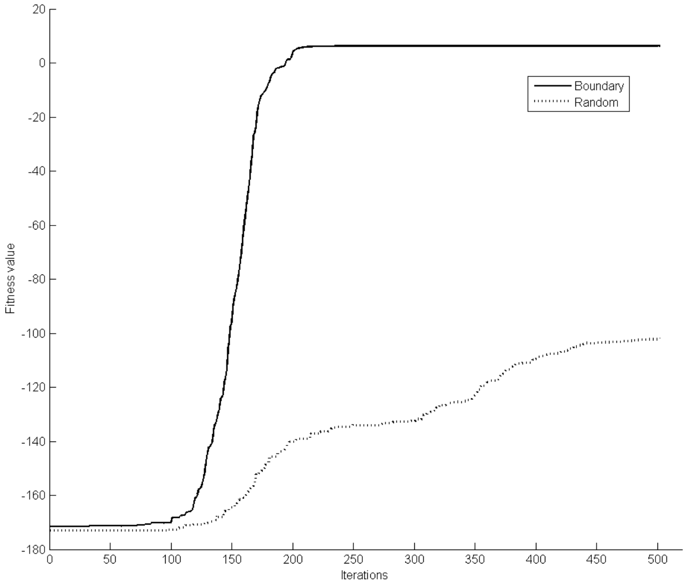


Figure 5.6: Repair mechanisms comparison of the locally D-optimal design for the 3-variable Poisson model with $r = -5$ on IED space $[0.01, 1]^3$.

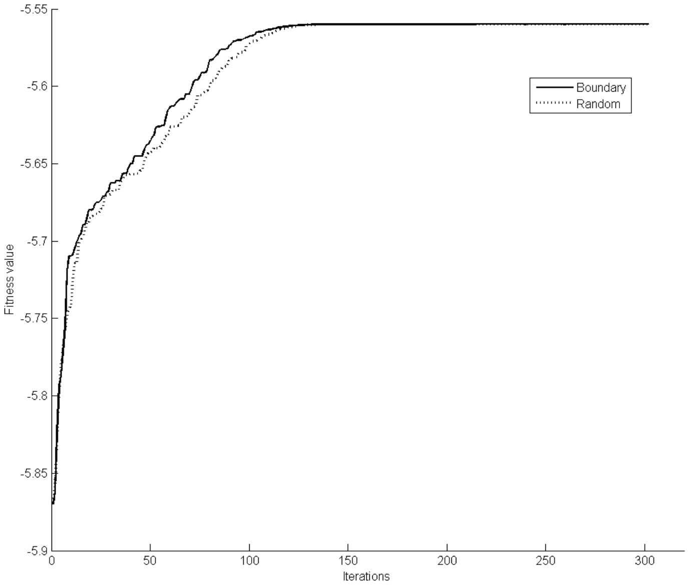


Figure 5.7: Repair mechanisms comparison of the locally D-optimal design for the CR model with $a_1 = 0, b_1 = 1, a_2 = 5, b_2 = 1$ on $\chi = [-10, 10]$.

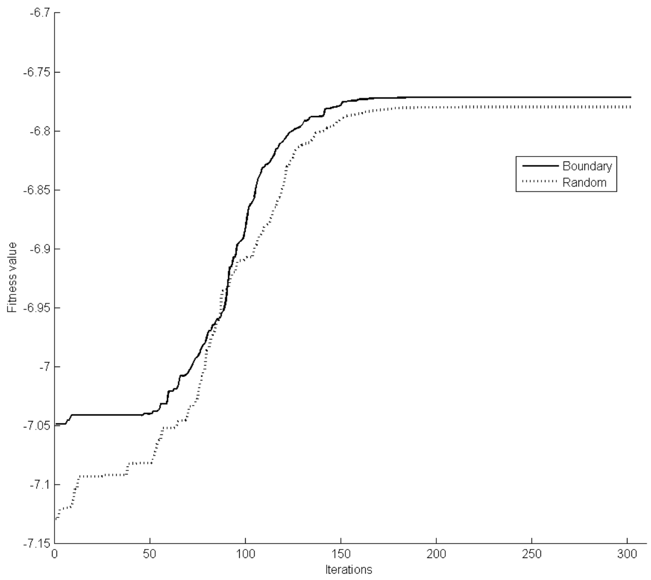


Figure 5.8: Repair mechanisms comparison of the locally D-optimal design for the CR model with $a_1 = -3.3, b_1 = 0.5, a_2 = 3.8, b_2 = 1$ on $\chi = [-10, 10]$.

locally D-optimal design for models a) to f) in section 5.1, and c-optimal designs for three quantities g) to i). Here the generated design is considered to be optimal when all its design points and their weights are the same as the known optimal design rounded to the 4 decimal places.

CA searches for locally D-optimal designs only. In section 5.2.2, I compare PSO with the boundary repair to CA in terms of CPU time for locally D-optimal design for models a) to f). It is worth noting that CA only works on a finite design space, which can be used to approximate a continuous space. This approach limits the ability of CA to find the exact D-optimal design (i.e. design points and corresponding weights are the same as the known optimal design rounded to the 4 decimal places), even though the generated designs by CA may be highly D-efficient (with D-efficiency greater than 0.99). In addition to the comparison of CPU time, I hereby compare the number of support points in the sought designs by PSO and CA, since practitioners usually prefer a simple design with fewer

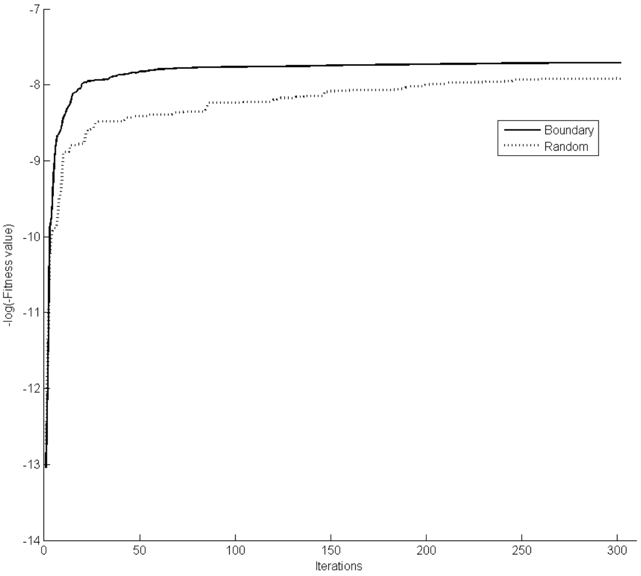


Figure 5.9: Repair mechanisms comparison of the locally c-optimal design for estimating the AUC in the compartmental model with $a = 4.298, b = 0.05884, c = 21.8$ on $\chi = [0, 20]$.

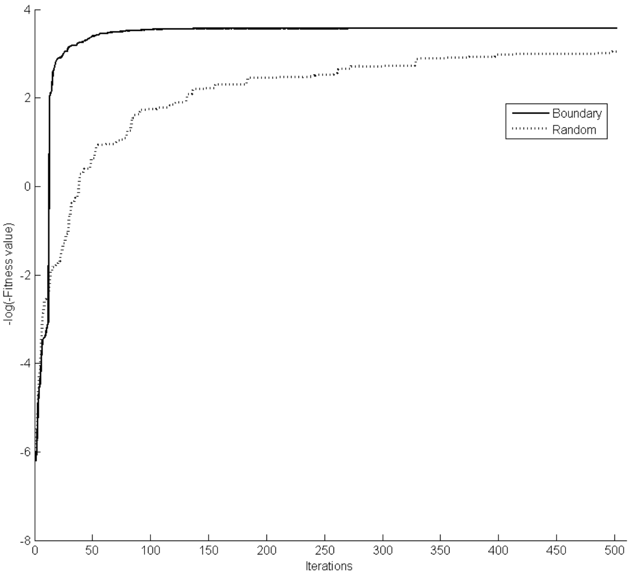


Figure 5.10: Repair mechanisms comparison of the locally c-optimal design for estimating the t_{max} in the compartmental model with $a = 4.298, b = 0.05884, c = 21.8$ on $\chi = [0, 20]$.

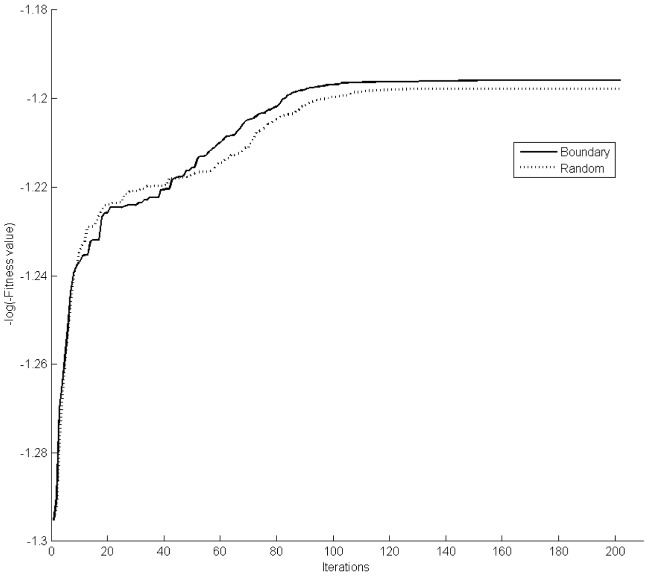


Figure 5.11: Repair mechanisms comparison of the locally c-optimal design for estimating the MED in the CR model with $a_1 = -3.3, b_1 = 0.5, a_2 = 3.8, b_2 = 1$ on $\chi = [-10, 10]$.

support points to a complicated design with a lot of points on the condition that both designs provide almost the same efficiency.

5.2.1 Comparison between PSO and DE

In general, comparing algorithms that use different strategies and with different numbers of tuning parameters can be challenging because one may argue that tuning parameters in the two algorithms were not properly selected. The key tuning parameters in DE are the crossover constant $CR \in [0, 1]$ and the amplification factor $CF \in [0, 2]$. I use the default values that Storn and Price [1997] provided i.e. $CR = 0.5$ and $CF = 1$, which show the best performance in all the tested functions in their paper. My experience is that for the types of problems we want to optimize, only two key variables: flock size and maximum number of iterations in the two algorithms seem to matter, a finding not too unlike others reported in the literature. For the comparisons here, I did preliminary runs with various tuning parameters to first estimate the minimal flock size and iteration number that would generate the optimal design with 90% chance over 10 replicates. I informally call this the "90% rule". If a particular tuning parameter setting fails, I repeat the search by increasing the flock size by 20 and the iteration numbers by 100 sequentially. Therefore the chosen iteration numbers flock size are minimum satisfying the 90% rule. All reported CPU times are averaged over 10 replicates. In the tables below, the numbers in the parenthesis after PSO and CA are flock size and maximal iteration number.

Table 5.1 reports the CPU time of finding locally D-optimal designs for all the models I used in section 1 by the two algorithms. For univariable models, PSO outperforms DE and sometimes by as much as 41(=57.4/1.4) times in terms of CPU times as in the case for finding locally D-optimal design for the quadratic logistic model. For multivariable models, I observe consistently faster performance of PSO over DE for both the bivariable linear model and the 3-variable Poisson regression models with different nominal values of interactions. For example I consider the locally D-optimal design for Poisson model with 3 variables and all pairwise interaction terms plus an intercept in IED space [0.01,1] with $r = 0$. This model has 7 parameters and its locally D-optimal design is minimally supported at 7 points, which is equivalent to solving a 27 (=3*7+6) dimensional constrained optimization problem. PSO took 2.8 seconds but DE took 113.7 seconds to converge to the D-optimal design, showing that PSO again outperforms DE by 40 times. Additionally I find that DE cannot find locally D-optimal design for the 4- or 5-variable Poisson and Exponential models under the 90% rule, regardless what flock size or iteration numbers I use.

In the comparisons of finding c-optimal designs to estimate quantities t_{max} and AUC of the compartmental model, and MED of the non-constant slope CR model, PSO and DE provide very close results in terms of CPU time. (Table 5.2). The biggest difference is in the searching for c-optimal design for estimating the MED of the non-constant slope CR model, in which PSO takes 1.2 seconds and DE takes 7.4 seconds to converge to the locally D-optimal design.

5.2.2 Comparison between PSO and CA

Cocktail algorithm (CA) is a deterministic algorithm which always produces the same output given one input. For PSO I still use the same tuning parameters in section 5.2.1 such that the algorithm produces the locally D-optimal at 90% chance. My experience with CA is that although its generated designs can be very efficient even when the grid size n is as small as 10, they may have many more support points than the true optimal design does. For example, I compare the locally D-optimal designs for bivariable linear model d) generated by PSO and CA in Table 5.3. The numbers in the parenthesis after PSO are flock size and maximum iteration number; the numbers in the parenthesis after CA are the grid size used in different optimization problems. PSO is able to find the true optimal

Table 5.1: CPU time of locally D-optimal designs by PSO and DE.

Model	Algorithm	CPU time
a) Compartmental model	PSO(20,200)	0.4
	DE(20,200)	2.9
b) Logistic quadratic model	PSO(100,200)	1.4
	DE(80,400)	57.4
c) Hill model	PSO(20,200)	0.3
	DE(20,200)	1.4
d) Bivariable linear model	PSO(80,300)	1.2
	DE(60,200)	11.0
e) 3-variable Poisson model ($r = 0$)	PSO(60,1500)	8.2
	DE(60,500)	113.7
e) 3-variable Poisson model ($r = -5$)	PSO(100,1000)	6.2
	DE(60,700)	215.3
f) CR model ($b_1 = b_2$)	PSO(20,200)	0.1
	DE(20,200)	1.4
f) CR model ($b_1 \neq b_2$)	PSO(100,300)	2.5
	DE(40,300)	6.5

The numbers in parenthesis are flocks size and maximal iterations number.

Table 5.2: CPU time of locally c-optimal designs by PSO and DE.

Model	Algorithm	CPU time
g) t_{max}	PSO(60,100)	4.5
	DE(20,200)	2.3
h) AUC	PSO(40,1000)	2.8
	DE(40,200)	4.6
i) MED in CR($b_1 \neq b_2$)	PSO(80,200)	1.2
	DE(40,200)	7.4

Table 5.3: PSO and CA generated locally D-optimal designs for *d*) bivariable linear model.

Algorithm	CPU time	D_{eff}	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
PSO(80,300)	1.2	1	-1	-1	0	0	1	1		
			0	1	0	1	0	1		
weight			0.1875	0.1875	0.1275	0.1275	0.1875	0.1875		
CA (10)	0.1	0.997	-1	-1	-0.1111	-0.1111	0.1111	0.1111	1	1
			0	1	0	1	0	1	0	1
weight			0.1870	0.1870	0.6300	0.6300	0.6300	0.6300	0.1870	0.1870
CA (1000)	94.6	1	-1	-1	-0.0101	-0.0101	0.0101	0.0101	1	1
			0	1	0	1	0	1	0	1
weight			0.1875	0.1875	0.6250	0.6250	0.6250	0.6250	0.1875	0.1875

design supported at 6 points with D-efficiency 1. Both of the CA generated designs have D-efficiency greater than 0.99, but they have two more support points near $(0,0)^T$ and $(0,1)^T$ due to the discretized design space. In biomedical researches such as a dose response study, practitioners usually prefer a simple design with fewer support points to a complicated design with a lot of points on the condition that both designs provide almost the same efficiency. Therefore in addition to the average CPU time, I use the number of support points in the generated design as another comparison criterion to measure the accuracy of CA generated design.

As shown in Table 5.4, for univariable models such as compartmental model, logistic quadratic model, CA with a grid size of 10000 converges at about the same speed as PSO with 100 (or less) particles does. For the compartmental model *a*), CA finds the exact same design as PSO does, which means it has no difficulty allocating weights at adjacent points. For logistic quadratic model with $\alpha = 3, \beta = -5, \mu = 0$ on $\chi = [-1, 1]$, CA allocates weights in adjacent points -0.9218 and -0.9216, and the sought design has one more point than the real optimal design. When I increase the grid size of 100000, CA successfully finds the same optimal design as PSO does, but takes much longer CPU time. A similar situation is observed in the case of Hill model, where CA has problem allocating weights at adjacent points even when I increase the grid size to 100000.

For both the 3 and 4-variable Poisson models (Table 5.5, 5.6), PSO consistently produces the minimally supported D-optimal designs regardless how IED space is chosen or what nominal values of interaction $r_{mm'}$ I pick. Note that PSO requires more iterations and particle numbers to distinguish the boundary 0.1 and the design point 0.1353. In contrast, CA converges fast, usually less than a second, when the grid size of each dimension of the design space is as small as 10, but the generated design usually has support points not close to the true optimal points. When the grid size is increased, the CPU time increases dramatically, especially for models with more than 3 variables. Therefore I limit the largest grid size 50 for 4-variable model. For such problems, I observe that when the grid size increases, the CA generated design structure becomes inconsistent. For example, when searching for locally D-optimal designs for the 4-variable Poisson model with $r = 0$ on IED $\chi = [0.1, 1]^4$, the number of support points in CA generated design varies from 11 to 17 when grid size increases from 10 to 50. The reason is that when having more points close to each other, CA has difficulty allocating their weights. Therefore it is impracticable to increase the precision of design by increasing the grid size in finding locally D-optimal designs for multivariable models.

Moreover, because CA finds locally D-optimal design on the discretized design space, I am concerned if the boundaries of design space affects CA generated design, especially for multivariable models. To investigate possible impact of changing design space on the performance of CA, I choose two design spaces $[0.01, 1]^M$ and $[0.1, 1]^M$ for 3 and 4-variable Poisson models. I have previously verified in Chapter 3 that the locally D-optimal design for the 3-variable Poisson model is minimally supported at 7 points, and for the 4-variable model is minimally supported at 11 points located at 0.1353 or 1 in each dimension. Therefore, both of the design spaces should give the same locally D-optimal design since the support

Table 5.4: Comparisons of the locally D-optimal design for univariable models between PSO and CA.

Model	Algorithm	# of support points	CPU time
a) Compartmental model	PSO(20,200)	3	0.4
	CA(10000)	3	0.3
	CA(100000)	3	5.2
b) Logistic quadratic model	PSO(100,200)	4	1.4
	CA(10000)	5	1.5
	CA(100000)	4	17.9
c) Hill model	PSO(20,200)	4	0.3
	CA(10000)	5	0.7
	CA(100000)	6	8.3

points are all located within the design region. However, from Table 5.5 and 5.6 I observe that changing design spaces causes inconsistency of the structure of CA generated design for both 3 and 4-variable Poisson models. Take the 4-variable Poisson with nominal value of $r = -5$ for example. CA with a grid size of 50 finds a design with 17 support points on IED space $[0.01, 1]^4$, but 23 support points on $[0.1, 1]^4$. In contrast, PSO consistently finds the same locally D-optimal design minimally supported at 11 points regardless what design space is chosen.

Lastly, for multivariate CR model as shown in Table 5.7 , regardless of constant or non-constant slope, PSO with 100 particles consistently produce the optimal design at about the same speed of CA with grid size as small as 100. Similarly as multivariable Poisson models, the CPU time that CA takes to converge increases dramatically when the grid size is increased, but the sought design by CA still have problem allocating weights to adjacent points.

Table 5.5: Comparisons of the locally D-optimal design of the 3-variable Poisson models with all nominal values of interactions $r_{mm'} = r$ between PSO and CA.

Model	Algorithm	# of support points	CPU time
$r = 0$ on IED $\chi = [0.01, 1]^3$	PSO(60,1500)	7	8.2
	CA(10)	7	0.1
	CA(100)	19	>10000
	CA(200)	7	413.3
$r = -5$ on IED $\chi = [0.01, 1]^3$	PSO (100,1000)	7	6.6
	CA(10)	7	0.1
	CA(100)	10	52.1
	CA(200)	7	287.6
$r = 0$ on IED $\chi = [0.1, 1]^3$	PSO (100,1000)	7	6.2
	CA(10)	7	0.1
	CA(100)	7	33.5
	CA(200)	7	315.3
$r = -5$ on IED $\chi = [0.1, 1]^3$	PSO (100,2000)	7	11.1
	CA(10)	7	0.1
	CA(100)	7	41.2
	CA(200)	13	634.2

Table 5.6: Comparisons of the locally D-optimal design of the 4-variable Poisson models with all nominal values of interactions $r_{mm'} = r$ between PSO and CA.

Model	Algorithm	# of support points	CPU time
$r = 0$ on IED $\chi = [0.01, 1]^4$	PSO(100,1500)	11	14.7
	CA(10)	11	0.4
	CA(20)	33	600.5
	CA(50)	11	342.0
$r = -5$ on IED $\chi = [0.01, 1]^4$	PSO(100,2000)	11	20.4
	CA(10)	11	0.4
	CA(20)	15	10.9
	CA(50)	17	456.9
$r = 0$ on IED $\chi = [0.1, 1]^4$	PSO(200,2000)	11	36.1
	CA(10)	11	0.4
	CA(20)	11	6.8
	CA(50)	11	315.8
$r = -5$ on IED $\chi = [0.1, 1]^4$	PSO(100,2000)	11	19.4
	CA(10)	11	0.4
	CA(20)	11	7.8
	CA(50)	23	673.1

Table 5.7: Comparisons of the locally D-optimal design of the CR model with constant or non-constant slopes on $\chi = [-10, 10]$ between PSO and CA.

Model	Algorithm	# of support points	CPU time
CR model with $b_1 = b_2 = 1$	PSO(20,200)	3	0.1
	CA(100)	5	1.7
	CA(1000)	5	20.8
	CA(10000)	4	71.3
CR model with $b_1 = 0.5, b_2 = 1$	PSO(100,500)	4	1.5
	CA(100)	5	2.9
	CA(1000)	6	14.3
	CA(10000)	5	122.7

5.3 Summary

In this chapter, I conduct simulation studies to compare searching efficiencies between PSO with boundary repair and random repair mechanisms. I find that the boundary repair mechanism greatly expedites PSO in searching for locally D- and c-optimal designs for a variety of models I investigated in the previous chapters. Furthermore, PSO with boundary repair is compared to two other popular algorithms: Cocktail algorithm (CA) by Yu [2011] and Differential Evolution (DE) algorithm by Storn and Price [1997]. In the simulations using the same set of models, I find PSO outperforms DE in terms of faster speed to converge to locally D- and c-optimal designs. In comparisons between PSO and CA, I find CA usually has difficulty in allocating weights for adjacent points when the grid size is increased, and CA generated designs do not maintain a consistent structure when the grid size or design space changes. In contrast, PSO consistently finds the locally D-optimal designs regardless what design space is chosen. Since I set PSO to round all the design points and weights to 4 decimal places, such precision enables us explore the structure of optimal designs at a microscopic level.

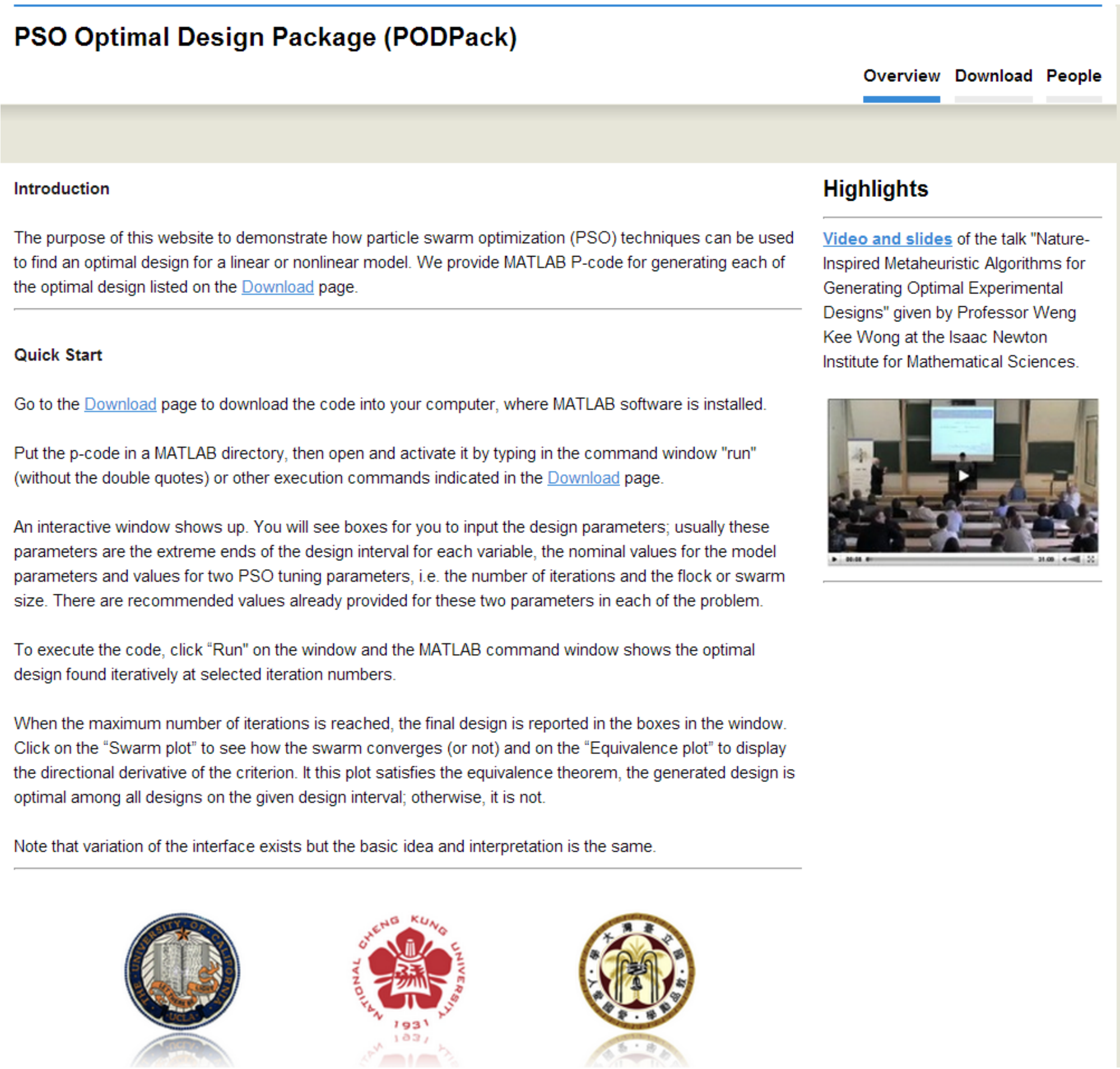
CHAPTER 6

Program Development of PSO

In this chapter, I illustrate some of the PSO codes I developed using MATLAB. To help researchers and practitioners to verify our results and appreciate how PSO works in practice, my collaborators in Taiwan have set up a website with MATLAB P-code for generating each of the optimal design in my dissertation. One mirror site is housed at `http://optimal-design.biostat.ucla.edu/podpack/`. Readers can download the P-code of the model of interest and find optimal designs under different sets of nominal values via PSO/UPSO.

Figure 6.1 displays a snapshot of the homepage. Clicking the “Download” link in the upper-right corner reveals all PSO codes we have developed. The PSO codes to find locally D- and c-optimal designs for univariable models in Chapter 1 are stored in Part C to Part G. The PSO codes for finding locally D-optimal designs for 3, 4 and 5-variable Poisson and Exponential models are stored in Part H. To download MATLAB P-codes, first click the library link of the codes of interest, and then click download in the "File" menu in the top-left part.

In what follows, I provide exemplary descriptions on how to use the PSO codes to find 1) a locally D-optimal design for 3-variable Poisson regression model with pairwise interactions; 2) a locally compound optimal design for estimating MTD, MED and all parameters in the continuation-ratio model. Other PSO codes in our website have the similar user interface and definitions of parameters.



PSO Optimal Design Package (PODPack)

Introduction

The purpose of this website to demonstrate how particle swarm optimization (PSO) techniques can be used to find an optimal design for a linear or nonlinear model. We provide MATLAB P-code for generating each of the optimal design listed on the Download page.

Quick Start

Go to the Download page to download the code into your computer, where MATLAB software is installed.

Put the p-code in a MATLAB directory, then open and activate it by typing in the command window "run" (without the double quotes) or other execution commands indicated in the Download page.

An interactive window shows up. You will see boxes for you to input the design parameters; usually these parameters are the extreme ends of the design interval for each variable, the nominal values for the model parameters and values for two PSO tuning parameters, i.e. the number of iterations and the flock or swarm size. There are recommended values already provided for these two parameters in each of the problem.

To execute the code, click “Run" on the window and the MATLAB command window shows the optimal design found iteratively at selected iteration numbers.

When the maximum number of iterations is reached, the final design is reported in the boxes in the window. Click on the “Swarm plot” to see how the swarm converges (or not) and on the “Equivalence plot” to display the directional derivative of the criterion. It this plot satisfies the equivalence theorem, the generated design is optimal among all designs on the given design interval; otherwise, it is not.

Note that variation of the interface exists but the basic idea and interpretation is the same.

Highlights

Video and slides of the talk "Nature-Inspired Metaheuristic Algorithms for Generating Optimal Experimental Designs" given by Professor Weng Kee Wong at the Isaac Newton Institute for Mathematical Sciences.

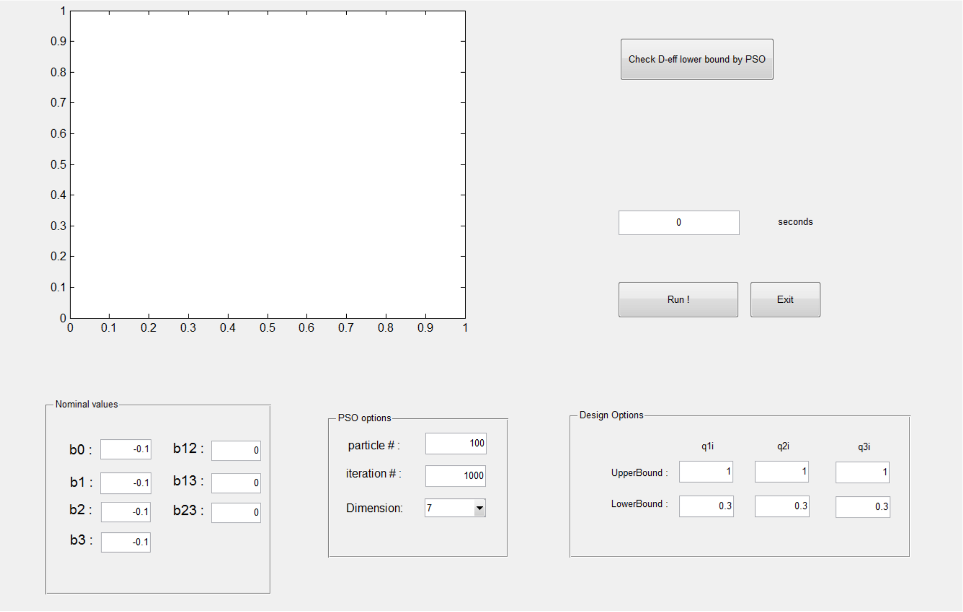
Figure 6.1: Snapshot of PSO website at UCLA.

6.1 Ultra-dimensional PSO (UPSO) for finding locally D-optimal designs for a Poisson model or an Exponential model

This is a manual for using MATLAB (R2013) software to find a locally D-optimal design for a Poisson model or an Exponential model using Ultra-dimensional Particle Swarm Optimization techniques. All regression models have terms for all main effects and all two-factor interaction terms plus an intercept. To fix ideas, I describe input requirements for a Poisson model with 3 toxicants (or variables) and all two-factor interaction terms. This model has a total of 7 parameters in the model including the intercept. The input requirements and explanation for generating a locally D-optimal design for the Exponential model are similar, and so are the cases when we have 4 or 5 toxicants with all two-factor interaction terms and an intercept term in the model.

After downloading the P-code, type **run** in the MATLAB command window and a graphical user interface (GUI) opens up. The UPSO program starts by clicking the **Run!** button and ends by clicking the **Exit** button. For example, if a locally D-optimal design for a Poisson model with 3 toxicants is of interest, upon activation, one sees the interface window (Figure 6.2). Clicking **Run!** will produce the support points and their weights in parentheses of the generated design in the command window as the iteration progresses. All numbers are rounded to 4 decimal places, which is equivalent to having a grid mesh containing as many as 10,000 points. When the UPSO search terminates at the maximum number of iterations, **the equivalence plot** by grid size of 10 is automatically generated and saved in the same folder. A rough D-efficiency lower bound for the generated design found using a grid size of 10 is also given.

INPUT PARAMETERS:



0 seconds

Nominal values

b0 : -0.1 b12 : 0

b1 : -0.1 b13 : 0

b2 : -0.1 b23 : 0

b3 : -0.1

PSO options

particle # : 100

iteration # : 1000

Dimension: 7

Design Options

	q1i	q2i	q3i
UpperBound :	1	1	1
LowerBound :	0.3	0.3	0.3

Figure 6.2: User interface of UPSO for finding locally D-optimal designs for the Poisson model with 3-variable and pairwise interactions.

Nominal values

Supply $b0$ in the box provided. This is the nominal value for the intercept β_0 in the model. Similarly for other parameters.

PSO options

particle #: Allows users to specify the number of particles to employ in the search.

iteration #: Maximum number of iterations allowed.

Dimension: The common number of support points in each particle (design) used in the search. Usually, the number of design point should be equal or greater than the number of parameters in the regression model. For example, we select 7 or more design points for 3-variable models; 11 or more design points for 4-variable models; 16 or more design points for 5-variable models, etc.

Design options

Allows users to specify the design space in terms of IED q_{km} for each agent (or variable). Recall in Chapter 3, we alternatively define the induced design space on the individual effective dose (IED) $q_{km} = \exp(\beta_m x_{km}) \in [c_m, 1]$, $m = 1, 2, ..., M$, or equivalently, $x_{km} \in [0, \frac{1}{\beta_m}\log(c_m)]$, assuming β_m is negative in the background of toxicology. Note that the value entered in the **LowerBound** box should always be positive, and the value entered in the **UpperBound** box should always be equal to 1 by definition of IEDs for the Poisson and Exponential models.

OUTPUT:

The equivalence plot: At the termination of the search, the equivalence plot shows the directional derivatives of objective function $\Psi(I(\xi))$at ξ in the direction of δ_X in a grid size of 10, where δ_X is the one-point design at $X = (x_1, x_2, ..., x_M)^T \in \chi$. The general equivalence theorem states that the generated design ξ is locally D-optimal among all designs on χ if and only if the following checking condition is satisfied

$$\text{trace}[I(\delta_X, \beta)I^{-1}(\xi, \beta)] - P \leq 0 \quad \forall X \in \chi, \tag{6.1}$$

with equality at the support points of ξ. For bivariable Poisson model $P = 4$, and for 3-variable Poisson model $P = 7$.

In bivariable Poisson regression model, the directional derivative (6.1) can be directly displayed in a 3-D equivalence plot. For example, Figure 6.3 shows the equivalence plot of the locally D-optimal design for a 2-variable Poisson model with nominal values of interaction terms equal to 0 on IED design space $[0.01, 1]^2$.

When the number of variables in the model is more than two, however, it is impossible to directly plot the directional derivatives. Consider the 3-variable Poisson for instance. First I construct a N^3 grid is for the 3-variable design space,

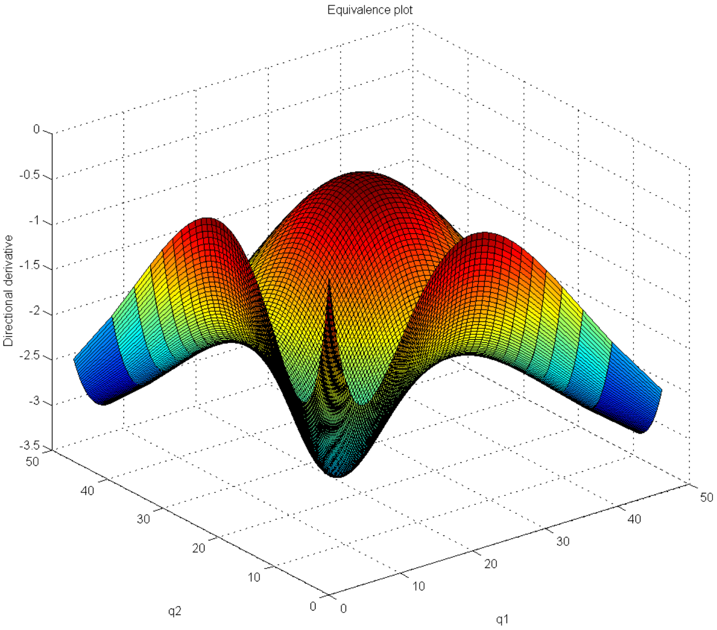


Figure 6.3: Equivalence plot confirming the optimality of the locally D-optimal design for the 2-variable Poisson model with nominal values of interaction terms $r_{mm'} = 0$ on IED design space $[0.01, 1]^2$.

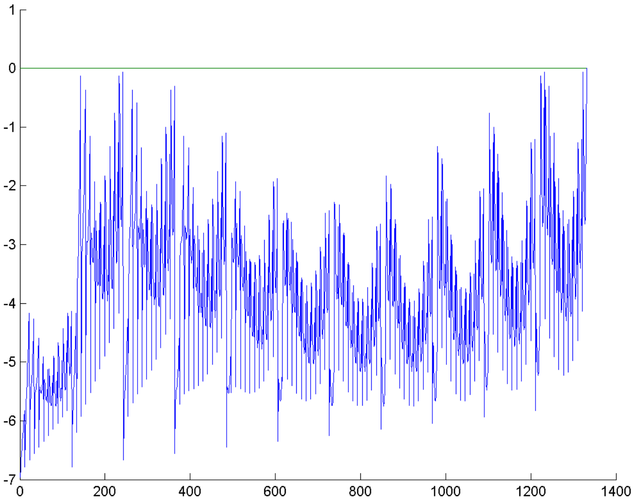


Figure 6.4: Modified equivalence plot against $aN^2 + bN + c$ confirming the optimality of the locally D-optimal design for the 3-variable Poisson model with nominal values of interaction terms $r_{mm'} = 0$ on IED design space $[0.01, 1]^3$.

i.e. $x_{mn} = (\log(c_m)/\beta_m)\frac{n}{N}$, and let

$$X_{aN^2+bN+c} = (x_{1a}, x_{2b}, x_{3c})^T = (\frac{a}{N}\frac{\log(c_1)}{\beta_1}, \frac{b}{N}\frac{\log(c_2)}{\beta_2}, \frac{c}{N}\frac{\log(c_3)}{\beta_3})^T$$

where $a, b, c = 0...N-1$. For example in a coarse grid $N = 10$, $X_{321} = (\frac{3}{10}\frac{\log(c_1)}{\beta_1}, \frac{2}{10}\frac{\log(c_2)}{\beta_2}, \frac{1}{10}\frac{\log(c_3}{\beta_3}$ Therefore we can plot the directional derivative against $aN^2 + bN + c$ regardless of how many variables are there in the design space. From such a plot, we can roughly identify the largest violation of directional derivative upper bounded by 0, and calculate the rough D-efficiency lower bound accordingly. Figure 6.4 shows directional derivative of the locally D-optimal design for three-variable Poisson model with nominal values of interactions as 0 on design space $[0.01, 1]^3$.

Check D-eff lower bound by PSO: Upon clicking this button at the termination of the run, the lower bound for the D-efficiency of the generated design will be displayed. This number is used to check whether the generated design is locally D-optimal or not. If this number is unity, the generated design is locally

D-optimal among all designs; otherwise it is not. The closer this number is to unity, the closer is the generated design to the locally D-optimal design (without knowing the optimum). This lower bound is obtained using the standard PSO to find the maximum violation in the equivalence theorem by determining the maximum positive value of the function on the left hand side of (2.2) described in the our paper. This is a relatively easy task since the optimization problem now has dimension M, the number of variables. In practice, users need to repeat the search for the maximum violation several times to ensure that PSO gives the same numerical result for the maximum value. The lower bound of D-efficiency is then obtained as described in Pazman [1986].

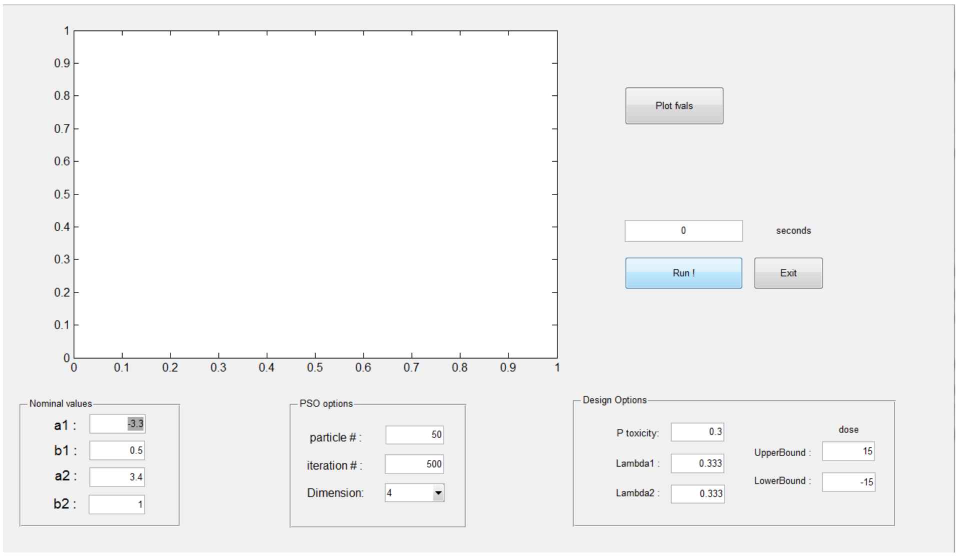
6.2 PSO for finding three-objective compound optimal designs for the non-constant slope CR model

This is a manual for using the MATLAB (R2013) software to find a compound locally D-optimal design for the non-constant slope continuation-ratio model using PSO. There are three objectives in the study. The first objective is locally c-optimal design for estimating the maximum tolerated dose (MTD), the second objective is locally c-optimal design for estimating the most effective dose (MED), and the third one is D-optimal design for estimating all parameters in the model.

To activate the GUI, type **run** in the MATLAB command window as shown in Figure 6.5. The PSO program starts by clicking on the **Run!** button and ends by clicking on the **Exit** button. Clicking **Run!** will first pause the program and plot the probability curves of

$$\pi_1(x) = \frac{1}{(1+e^{a_1+b_1x})(1+e^{a_2+b_2x})} \qquad \text{(no reaction)}$$

$$\pi_2(x) = \frac{e^{a_2+b_2x}}{(1+e^{a_1+b_1x})(1+e^{a_2+b_2x})} \qquad \text{(efficacy without toxicity)}$$



0 seconds

Nominal values
a1 : -3.3
b1 : 0.5
a2 : 3.4
b2 : 1

PSO options
particle # : 50
iteration # : 500
Dimension: 4

Design Options
P toxicity: 0.3
Lambda1 : 0.333
Lambda2 : 0.333
dose
UpperBound : 15
LowerBound : -15

Figure 6.5: User interface of PSO for finding compound optimal designs for the CR model.

and

$$\pi_3(x) = \frac{e^{a_1+b_1x}}{1+e^{a_1+b_1x}} \qquad \text{(toxicity)}.$$

in the user-specified dose range. This plot is constructed based on the input parameters and so it provides guidance or confirmation that the input parameters are reasonable.

Clicking any key resumes the program and generates the support points with their weights in parentheses of the generated design as the iteration progresses. All numbers are rounded to 4 decimal places, which is equivalent to having a grid mesh containing as many as 10,000 points. When the PSO search terminates at the maximum number of iterations, **the equivalence plot** constructed from a grid size of 1000 is automatically generated and saved in the same folder.

INPUT PARAMETERS:

Nominal values

Supply $a1$ which is the nominal value for the intercept a_1 in the model. Similarly for other parameters.

PSO options

particle #: Allows users to specify the number of particles to employ in the search.

iteration #: Maximum number of iterations allowed.

Dimension: The common number of support points in each particle (design) used in the search. Usually, the number of design points should be equal or greater than the number of parameters in the CR model.

Design options

Upperbound; Lowerbound: Allows users to specify the design space for the dose range of interest.

P toxicity: The user-specified proportion ρ of patients that experience dose limiting toxicity (DLT) given MTD.

$$\pi_3(MTD, a_1, b_1) = \rho.$$

Lambda1: Weight λ_1 for the criterion $\Psi_1(\xi)$, the log c-efficiency of the design ξ for estimating MTD.

Lambda2: Weight λ_2 for the criterion $\Psi_2(\xi)$, the log c-efficiency of the design ξ for estimating MED.

Note both λ_1, λ_2 and their sum have to be within [0,1]. This ensures a non-negative weight λ_3 for log D-efficiency to estimate all parameters in the model.

OUTPUT:

The equivalence plot: At the termination of the search, the equivalence plot shows the directional derivative of objective function of the compound optimal design $\Psi(\xi_\lambda)$ being evaluated at ξ_λ in the direction of δ_x in a grid size of 1000.

Bibliography

A. C. Atkinson. The usefulness of optimum experimental designs. *J. R. Statist. Soc. B*, 58:59–76, 1996.

A. C. Atkinson and A. N. Donev. *Optimum Experimental Designs.* Oxford University Press, 1992.

A. C. Atkinson and L. M. Haines. Designs for nonlinear and generalized linear models. *In Handbook of Statistics*, 13:437–475, 1996.

A. C. Atkinson, K. Chaloner, A. M. Herzberg, and J. Juritz. Optimum experimental designs for properties of a compartmental model. *Biometrics*, 49(2): 325–337, 1993.

M. P. F. Berger and W. K. Wong. *An Introduction to Optimal Designs for Social and Biomedical Research.* John Wiley & Sons, Hoboken, NJ Wiley & Sons, Chichester, West Sussex, UK, 2009.

A. Biswas and J. Lopez-Fidalgo. Compound designs for dose-finding in the presence of nondesignable covariates. *Pharmaceutical Statistics*, 12:92–101, 2013.

J. F. Borzelleca and W. Lowenthal. Drug absorption from the rectum. ii. *J. Pharm. Sci.*, 55:151–154, 1966.

K. Chandrasekar and N. V. Ramana. Performance comparison of ga, de, pso and sa approaches in enhancement of total transfer capability using facts devices. *Journal of Electrical Engineering & Technology*, 7:493–500, 2012.

H. Chernoff. Locally optimal designs for estimating parameters. *The Annals of Mathematical Statistics*, 24:586–602, 1953.

T. Chou. Theoretical basis, experimental design, and computerized simulation of synergism and antagonism in drug combination studies. *Pharmacol Rev*, 58: 621–681, 2006.

M. Clyde and K. Chaloner. The equivalence of constrained and weighted designs in multiple objective design problems. *JASA*, 91:1236–1244, 1996.

J. M. Cobby, P. F. Chapman, and D. J. Pike. Design of experiments for estimating inverse quadratic polynomial responses. *Biometrics*, 42(3):659–664, 1986.

R. D. Cook and W. K. Wong. On the equivalence of constrained and compound optimal designs. *Journal of the American Statistical Association*, 89(426):687–692, 1994.

E. Demidenko. *Mixed Model: Theory and Applications.* John Wiley & Sons, Hoboken, NJ, 2004.

H. Dette and S. Beidermann. Robust and efficient designs for the michaelis-menten model. *JASA*, 98:679–686, 2003.

H. Dette, V. B. Melas, and P. Shpilev. Optimal designs for estimating the derivative in nonlinear regression. *Statistica Sinica*, 21:1557–1570, 2011.

R. Dominguez and E. Pomerene. Calculation of the rate of absorption of exogeneous creatinine. *Proc. Soc. Exp. Biol. Med.*, 60:173–181, 1945.

R. C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of IEEE congress evolutionary computation, San Diego, CA*, pages 84–88, 2000.

R. C. Eberhart and Y. Shi. Particle swarm optimization: Developments, applications and resources. *Evolutionary Computation*, 1:81–86, 2001.

S. K. Fan and K. Chaloner. Optimal designs and limiting optimal designs for a trinomial response. *Journal of Statistical Planning and Inference*, 126:347–360, 2004.

V. V. Fedorov. *Theory of optimal experiments.* New York: Academic, 1972.

V. V. Fedorov and Y. Wu. Dose finding designs for continuous responses and binary utility. *Journal of Biopharmaceutical Statistics*, 17:1085–1096, 2007.

V. V. Fedorov, R. C. Gagnon, and S. L. Leonov. Design of experiments with unknown parameters in variance. *Applied Stochastic Models in Business and Industry*, 18:207–218, 2002.

I. Ford. *Ian Ford PhD Thesis.* PhD thesis, University of Glasgow, Scotland, 1976.

E. F. Fornius and H. Nyquist. Using the canonical design space to obtain c-optimal designs for the quadratic logistic model. *Comm. in Statistics-Theory and Methods*, 39:144–157, 2010.

J. Fresen. *Aspects of bioavailability studies. M.Sc. Dissertation.* PhD thesis, Department of Mathematical Statistics, University of Capetown., 1984.

P. Fumoleau, J. M. Trigo, N Isambert, D. Sémiond, S. Gupta, and M. Campone. Phase i dose-finding study of cabazitaxel administered weekly in patients with advanced solid tumours. *BMC Cancer*, 13:1471–2407, 2013.

S. Garnier, J. Gautrais, and G. Thraulaz. The biological principles of swarm intelligence. *Swarm Intell*, 1:3–31, 2007.

S. G. Gilmour and L. A. Trinca. Bayesian l-optimal exact design of experiments for biological kinetic models. *Applied Statistics*, 61:237–251, 2011.

L. M. Haines. The application of the annealing algorithm to the construction of exact optimal designs for linear-regression models. *Technometrics*, 29(4): 439–447, 1987.

L. M. Haines, M. G. Kabera, P. Ndlovu, and T. E. O'Brien. D-optimal designs for logistic regression in two variables. *MODA 8 - Advances in Model-Oriented Design and Analysis*, pages 91–98, 2007.

M. A. Heise and R. H. Myers. Optimal designs for binvariate logistic regression. *Biometrics*, 52:613–624, 1996.

A. V. Hill. The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *The Journal of Physiology*, 40:iv–vii, 1910.

J. Holland. *Adaptation in Natural and Artificial systems.* University of Michigan Press, Ann Anbor, 1975.

V. Kachitvichyanukul. Comparison of three evolutionary algorithms: Ga, pso, and de. *Industrial Engineering and Management Systems*, 11(3):215–223, 2012.

J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 4:1942 – 1948, 1995.

L. A. Khinkis, L. Levasseur, H. Faessel, and W. R. Greco. Optimal design for estimating parameters of the 4-parameter hill model. *Nonlinearity in Biology, Toxicology, and Medicine*, 1(3):363–377, 2003.

J. Kiefer. General equivalence theory for optimum designs (approximate theory). *The Annals of Statistics*, 2:849–879, 1974.

J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, New Series*, 220(4598):671–680, 1983.

C. D. Klaassen. *Principles of toxicology. In: Goodman and Gilman's The Pharmacological Basis of Therapeutics, Gilman, A.G.* Macmillan Company, New York, 1980.

M. Konstantinou, S. Biedermann, and A. Kimber. Optimal designs for two-parameter nonlinear models with application to survival models. *Southampton Statistical Research Institute, University of Southampton*, 2011.

L. M. Levasseur, H. K. Slocum, Y. M. Rustum, and W. R. Greco. Modeling of the time-dependency of in vitro drug cytotoxicity and resistance. *CANCER RESEARCH*, 58:5749–5761, 1998.

G. Li and N. Balakrishnan. Optimal designs for tumor regrowth models. *Journal of Statistical Planning and Inference*, 141:644–654, 2011.

G. Li and D. Majumdar. Some results on d-optimal designs for nonlinear models with applications. *Biometrika*, 96(2):487–493, 2009.

J. Lopez-Fidalgo, M. J. Rivas-Lopez, and R. Del Campo. Optimal designs for cox regression. *Statistica Neerlandica*, 63:135–148, 2009.

W. K. Lutz, O. Tiedge, R. W. Lutz, and H. Stopper. Different types of combination effects for the induction of micronuclei in mouse lymphoma cells by binary mixtures of the genotoxic agents mms, mnu, and genistein. *Toxicological Sciences*, 86:318–323, 2005.

I. C. Marschner. Optimal design of clinical trials comparing several treatments with a control. *Pharmaceutical Statistics*, 6:33, 2007.

N. Metropolis, M. N. Rosenbluth, A. W.and Rosenbluth, A. Teller, and H. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

S. Minkin. Optimal designs for binary data. *JASA*, 82:1098–1103, 1987.

K. Ogungbenro, A. Dokoumetzidis, and L. Aarons. Application of optimal design methodologies in clinical pharmacology experiments. *Pharmaceutical Statistics*, 8:239–252, 2009.

M. A. Panduro, C. A. Brizuela, L. I. Balderas, and D. A. Acosta. A compasrison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. *Progress In Electromagnetics Research B*, 13:171–186, 2009.

K. E. Parsopoulos and M. N. Vrahatis. *Particle Swarm Optimization and Intelligence: Advances and Applications*. IGI Global, 2010.

A. J. Pazman. *Foundations of Optimum Experimental Design*. Reidel Publishing Company, East European series, Dordrecht, Holland, 1986.

R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization: an overview. *Swarm Intell*, 1:33–57, 2007.

J. Qiu, R.B. Chen, W. Wang, and W.K. Wong. Using animal instincts to design efficient biomedical studies via particle swarm optimization. *Swarm Evol Comput*, 18:1–10, 2014.

A. Ratnaweera, S. K. Halgamuge, and H. C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, 2004.

W. Richards, R. Suresh, and B. Belanger. A comparison of crm and polya urn methods in clinical trials. *JSM'97 Abstracts*, page 273, 1997.

S. J. Ruberg. Dose response studies i. some design considerations. *Journal of Biopharmaceutical Statistics*, 5:1–14, 1995.

K. G. Russell, D. C. Woods, S. M. Lewis, and J. A. Eccleston. D-optimal designs for poisson regression models. *Statistica Sinica*, 19:721–730, 2009.

P. Sebastiani and R. Settimi. A note on d-optimal designs for a logistic regression model. *Journal of Statistical Planning and Inference*, 59:359–368, 1997.

S. Z. Selima and K. Alsultanb. A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, 24:1003–1008, 1991.

L. B. Sheiner, S. L. Beal, and N. C. Sambol. Study designs for dose-ranging. *Clinical Pharmacology and Therapeutics*, 46(1):63–77, 1989.

S. D. Silvey. *Optimal Design.* Chapman and Hall, London, 1980.

S. D. Silvey, D. M. Titterington, and B. Torsney. An algorithm for optimal designs on a finite design space. *Commun. Stat. Theory Methods*, 14:1379–1389, 1978.

R. R. Sitter and C. F. J. Wu. Optimal designs for binary response experiments: Fieller, d, and a criteria. *Scandinavian Journal of Statistics*, 20:329–341, 1993a.

R. R. Sitter and C. F. J. Wu. On the accuracy of fieller intervals for binary response data. *JASA*, 88:1021–1025, 1993b.

P. E. Sparrow. The comparison of five response curves for representing the relationship between the annual dry-matter yield of grass herbage and fertilizer nitrogen. *The Journal of Agricultural Science*, 93:513–520, 1979.

R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11: 341–359, 1997.

J. M. G. Taylor. The design of in vivo multifraction experiments to estimate the $\alpha - \beta$ ratio. *Radiation Research*, 121:91–97, 1990.

T. Teorell. Kinetics of distribution of substances administered to body. i. the extravascular modes of administration. *Arch. Int. Pharrnacodyn. Ther.*, 57: 205–225, 1937.

P. F. Thall and K. E. Russell. A strategy for dose-finding and safety monitoring based on efficacy and adverse outcomes in phase i/ii clinical trials. *Biometrics*, 54:251–264, 1998.

J. Y. Tsay. On the sequential construction of d-optimal designs. *JASA*, 71(355): 671–674, 1976.

C. W. Ueberhuber. *Numerical Computation 1: Methods, Software, and Analysis.* Springer, 1997.

Y. Wang. *Optimal experimental designs for the Poisson regression model in toxicity studie.* PhD thesis, Virginia Tech, 2002.

Y. Wang, R. Myers, E. Smith, and K. Ye. D-optimal designs for poisson regression models. *Journal of Statistical Planning and Inference*, 136:2831–2845, 2006.

J. M. Whitacre. Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. *Computing*, 93:121–133, 2011a.

J. M. Whitacre. Survival of the flexible: explaining the recent dominance of nature-inspired optimization within a rapidly evolving world. *Computing*, 93: 135–146, 2011b.

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 1(1):67–82, 1997.

D. C. Woods, S. M. Lewis, J. A. Eccleston, and K. G. Russell. Designs for generalized linear models with several variables and model uncertainty. ;48:. *Technometrics*, 48:284–292, 2006.

C. F. Wu. Some algorithmic aspects of the theory of optimal designs. *Annals of Statistics*, 6:1286–1301, 1978.

H. P. Wynn. Results in the theory and construction of d-optimum experimental designs. *Journal of the Royal Statistical Society: Series B*, 34:133–147, 1972.

M. Yang. On the de la garza phenomenon. *The Annals of Statistics*, 38:2499–2524, 2010a.

M. Yang and J. Stufken. Support points of locally optimal designs for nonlinear models with two parameters. *The Annals of Statistics*, 37:518–541, 2009.

M. Yang, B. Zhang, and S. Huang. Optimal designs for generalized linear models with multiple design variables. *Statistica Sinica*, 21:1415–1430, 2011.

M. Yang, S. Biedermann, and E. Tang. On optimal designs for nonlinear models: a general and efficient algorithm. *JASA*, 108:1411–1420, 2013.

X. S. Yang. *Nature-Inspired Metaheuristic Algorithms.* Luniver Press, 2010b.

X. S. Yang. *Engineering optimization: An introduction with Metaheuristic applications.* John Wiley & Sons, Inc., Hoboken, New Jersey, 2010c.

Y. Yu. D-optimal designs via a cocktail algorithm. *Stat Compute*, 21:475–481, 2011.

W. Zhang, X. Xie, and D. Bi. Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space. *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, 2:2307–2311, 2004.

W. Zhu and W. K. Wong. Multi-objective designs in a dose-response experiment. *Journal of Biopharmaceutical Statistics, 10:1*, 10:1–14, 2000.

S. S. Zocchi and A. C. Atkinson. Optimum experimental designs for multinomial logistic models. *Biometrics*, 55:437–444, 1999.