OPTIMAL DESIGNS FOR THE HILL MODEL

WITH THREE PARAMETERS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Travis Jon Dockter

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

May 2012

Fargo, North Dakota

North Dakota State University
Graduate School

Title

OPTIMAL DESIGN FOR THE HILL MODEL WITH 3 PARAMETERS

By

Travis Dockter

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Seung Won Hyun
Chair

Dr. Rhonda Magel

Dr. Ronald Degges

Dr. Sean Sather-Wagstaff

Approved:

05/02/2012
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

Optimal designs specify design points to use and how to distribute subjects over these design points in the most efficient manner. The Hill model with three parameters is often used to describe sigmoid dose response functions. In our paper, we study optimal designs under the Hill model. The first is D-optimal design, which works best to study the model to fit the data. Next is c-optimal design, which works best to study a target dose level, such as ED50 - the dose level with 50% maximum treatment effect. The third is a two-stage optimal design, which considers both D-optimality and c-optimality. In order to compare the optimal designs, their design efficiencies are compared.

ACKNOWLEDGEMENTS

I owe my deepest thanks to Dr. Seung Won Hyun for his constant guidance throughout this study. Without his support and encouragement, this research paper would not have been possible.

I would like to thank Dr. Rhonda Magel and Dr. Ronald Degges for their help and advice during the past few semesters and for being on my committee. I would also like to thank Dr. Sean Sather-Wagstaff for being on my committee and for his time in this process.

I would like to thank my colleague and friend Alfred Ndungu. I am very grateful for his support throughout this study and throughout my graduate career. He has been willing and ready to offer advice and guidance at any point during my study.

Finally, I would like to thank my family. Without their constant support during the past few years, I may never have gotten through some of the rough times. I am very thankful for their love and prayers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Experimental design is one of the most important branches in Statistics. Its applications for research are seen throughout the world. Experimental design is useful because it can help experimenters minimize costs while still obtaining valid results. Various forms of experiments exist for researchers to use in their studies. Each type of experimental design has different assumptions and restrictions applied to it. Hence, there is no single design that can be applied to every single experiment. However, there are many specialized designs that can be used in various situations. For example, if an experimenter is interested in testing the effects of a certain type of medicine, he may obtain a random sample of people and randomly apply each treatment to a set number of individuals. Such an experiment is known as a randomized complete design, or RCD (Montgomery, 2009). There are numerous other designs that can be used in any area of research.

There is one area of research that experimental design is frequently applied to: biostatistics. Biostatistics is the branch of statistics that deals with data relating to living organisms. Within this branch fall various topics, such as clinical trials and survival analysis. Pharmaceutical companies are especially interested in experimental design as it applies to biostatistics because they frequently conduct studies to assess and compare the efficacy and the toxicity of new drugs. And since they are businesses, one of their main goals is to collect results that are valid while lowering costs as much as possible.

Dose response studies are used to study the efficacy and toxicity of drugs. One purpose is to determine if there is some relationship between the dose level and the response. A researcher or doctor may be interested in finding a pattern between dose level and response. Or, a target dose level may be of interest. For example, we may want to find the minimum effective

dose, or the smallest dose that produces a clinically important response and can be declared statistically significantly different compared to a placebo effect. In dose response studies, the following questions are fundamental (Ruberg, 1995):

- Does the drug have an evident effect?
- Which doses produce significantly different responses from the control dose?
- What is the nature of the dose-response relationship?
- What dose level is optimal?

To study these questions efficiently, we can use optimal designs (Bretz et al, 2009, Dette et al., 2008, Dragalin et al, 2007, Miller et al, 2007, Leonov and Miller, 2009, Hyun et al, 2011).

Optimal designs specify which treatments to use and how to distribute samples over the treatments to study the goal of the experiment in the most efficient manner. These designs can be used in all areas of research, not just biostatistics. There are different types of optimality criterion based on the goal of the experiment. For example, D-optimality is applied when a researcher is interested in estimating the value of parameters for a model. Overall, optimal designs allow a researcher to minimize the variance of estimating interesting features of the study.

In this paper, we examine two types of optimality criterion, and another type that is a hybrid of the two others. The first type considered is D-optimality. D-optimal criterion is focused on estimating the parameters of a model accurately, which will allow us to obtain the overall information about the dose response. To find the D-optimal design, we look for a design that minimizes the determinant of the inverse of our information matrix. The next type of design considered is a c-optimal design. C-optimal criteria are used when a researcher wants to estimate an interesting particular dosage level efficiently, which will be denoted as ED_p, where "p" gives

the 100*p% of the maximum response. For example, a doctor may be interested in finding a treatment that gives ED_{50}, which is the dose producing 50% of the maximum response. In this case, a c-optimal design would be appropriate. The criterion of a c-optimal design is to minimize the variance of estimating ED_p. The variance can be expressed as a function of model parameters in general. The final design is a two-stage optimal design that is a hybrid of D- and c-optimality criteria. First, the number of subjects is split in half, and each half is assigned to each stage. In stage one, a D-optimal design is used to estimate model parameters using the first half of our sample. Then, a c-optimal design is used to estimate a target dose level ED_p using the second half of our sample in stage two. This type of design will cover both types of criteria, which is a benefit.

For this paper, the Hill model is used as the model of interest. The Hill model gives the average activity of an enzyme per catalytic site as a function of the total substrate concentration. Li et al (2004) adopted the hill model to study the effects of serine-13 phosphorylation on human systolic tymidine kinase. The paper provides the design space and parameter values to study D-, c- and two-stage optimal designs. To obtain our designs, a numerical approach is used. The numerical algorithm used to obtain our designs is called the V-algorithm, which is a very popular algorithm for these types of problems. Chapter 3 will discuss the model under consideration for our optimal designs and in chapter 4 we will find each type of design, including a traditional uniform design. In chapter 5, we will discuss the efficiency of each optimal design and the uniform design with respect to D-optimality criteria and c-optimality criteria. The next chapter will further explore the background of the different optimality criteria and the methods used to find optimal designs.

2. BACKGROUND

When a researcher is conducting an experiment, often the goal is to estimate parameters and then, using these estimates, to fit a model for prediction. To help ensure valid results, one must minimize the variance of the parameters of these estimated parameters and the predicted values. We can minimize these variances by determining what levels of treatments we should use and how our samples should be distributed over our treatments. This information must be determined beforehand, and they depend on both the number of samples that are available and the range of our treatments. To help us make our decisions on these issues, we can use optimal designs.

Optimal designs are used to specify how a researcher can distribute the resources for an experiment in the most efficient way. These designs will also find the best locations to obtain our observations, given that we have a response surface. Optimal designs are experimental designs that are powerful, flexible and efficient. They can provide us with accurate statistical results while lowering research costs. They can also be used to estimate unbiased parameters for a model that also have the smallest variance for almost any model. Optimal designs have different criteria that can be used, depending on the goal of a particular experiment. To find such optimal designs, we need to find a design that minimizes optimality criteria Ψ.

2.1 Criteria of Optimal Design

For a given model, let Θ denote the vector of our parameters, x_i will be the i^{th} dosage level, n_i will be the number of subjects allocated to the i^{th} dosage level, and n will be the total number of subjects, where $n = \sum_{i=1}^{K} n_i$. Also, let $M(\xi;\Theta)$ denote the Fisher information matrix for each subject with our parameter vector θ and our design $\xi = \{(x_i, w_i), i = 1, \dots, K\}$, where $w_i = \frac{n_i}{n}$ denotes the weights for our design points. So, our problem now is to find ξ such that we

can minimize our optimality criteria $\Psi\{M(\xi;\Theta)\}$. Next we shall define the criteria for A, D and c-optimality.

2.1.1 A-Optimality

When our goal is to estimate linear combinations of model parameters, we use a design that is A-optimal. A-optimality criterion is to minimize the sum of variances for the parameter estimates. The criterion here is given below:

$$\Psi = tr\{M(\xi;\ \Theta)^{-1}\}.$$

2.1.2 D-Optimality

D-optimal designs are used when our goal is to estimate parameters in the model. In this design, the criterion is to minimize the determinant of the inverse of the Fisher information matrix. The criterion for this design is

$$\Psi = |M(\xi;\Theta)^{-1}|.$$

2.1.3 c-Optimality

A c-optimal design is used when a researcher is interested in estimating a function of parameters in the model, and is a special case of A-optimality. The c-optimal design minimizes the variance of the function of parameters, denoted as $c^T\Theta$, where c is a *m x 1* vector of constants and m is the number of parameters in the model. The criterion is

$$\Psi = c^T M(\xi;\Theta)^{-1}c.$$

2.2 Caratheodory's Theorem

When we are finding a D-optimal design, Caratheodory's Theorem gives us an upper bound on the number of dosage levels in our design. According to this theorem, we will have no more than $\frac{m(m+1)}{2} + 1$ dosage levels, where m is the number of parameters in our model.

2.3 The General Equivalence Theorem

The General Equivalence Theorem is essential to find and verify optimal designs (Kiefer, 1958). What it does is provides us with a method for constructing our optimal design and then checking to see if they are in fact optimal with respect to some criterion. It can be viewed as a consequence of the result that derivatives are zero at the minimum of a continuous function. However, the function depends on the measure of our design ξ through the information matrix M(ξ; Θ). Let the measure $\bar{\xi}$ put unit mass at the point x and let the measure put unit mass at the point x and let the measure ξ' be given by

$$\xi' = (1-\alpha)\xi + \alpha\bar{\xi}.$$

Then,

$$M(\xi';\Theta) = (1-\alpha)M(\xi;\Theta) + \alpha M\big(\bar{\xi};\Theta\big).$$

Accordingly, the derivative of Ψ in the direction of $\bar{\xi}$ is

$$\phi(x,\xi) = \lim_{a \to 0^+} \frac{1}{\alpha}\big[\Psi\big\{(1-\alpha)M(\xi;\Theta) + \alpha M\big(\bar{\xi};\Theta\big)\big\} - \Psi\{M(\xi;\Theta)\}\big]$$

The General Equivalence Theorem states that, for a given design ξ*,

1. ξ* minimizes Ψ{M(ξ;Θ)}
2. ξ* maximizes the minimum over our design space of $\phi(x,\xi)$
3. The minimum over our design space $\phi(x,\xi) = 0$ only when we have points that support our design ξ* (Atkinson, 2007).

Thus, we can use this for each of our designs. It has a generalized structure for any type of design. Let ξ* be the optimal design under some ϕ_t-optimality criterion. Suppose that the goal of our design ξ* is interested in finding s ($s \leq m$) linear combinations of our parameters $A^T\Theta$, where A is an m by s matrix of rank s. Then, our information matrix for $A^T\Theta$ for our model, which will be discussed in chapter 3, is $(A^T M^-(\xi;\Theta)A)^{-1}$, where $M^-(\xi;\Theta)$ is the generalized

inverse for our information matrix M(ξ;Θ). The Generalized Equivalence Theorem states that ξ* is the ϕ_t-optimal design for $A^T\Theta$ if and only if there is a generalized inverse $M^-(\xi^*;\Theta)$ of M(ξ*;Θ) such that

$$f(x)Cf(x)^T \leq tr((A^T M^-(\xi^*;\Theta)A)^{-t}), \quad (1)$$

where

$$C = M^-(\xi^*;\Theta)A(A^T M^-(\xi^*;\Theta)A)^{-(t+1)}A^T\left(M^-(\xi^*;\Theta)\right)^T. \quad (2)$$

If $M^-(\xi^*;\Theta)$ is constructed from any ϕ_t-optimal design ξ* for $A^T\Theta$, then equality will be obtained in (1) whenever x is a support point for ξ* (Hyun et al, 2011). When t = 0, ϕ_t-optimality becomes D-optimality and when t = 1, it becomes A-optimality.

2.4 The V-algorithm

To search for our optimal designs for each criterion, the V-algorithm will be used. The V-algorithm is a very popular tool for this purpose. It is a quick, efficient algorithm that allows a researcher to obtain the design points for various optimal designs. However, the V-algorithm does not work efficiently to find the weights for optimal designs points. Because of this, we employ a Newton-Raphson algorithm to find the optimal weights for identified optimal design points from the V-algorithm.

The V-algorithm finds optimal design points based on the General Equivalence Theorem. We begin with some initial design ξ^0 with associated information matrix M(ξ^0;Θ). Each point in ξ^0 will have equal weight. At each point in our design space, we calculate the standardized variance, which is given by the left side of (1) and is denoted as d_n, where n denotes the n^{th} step of the V-algorithm:

$$d_n = f(x)Cf(x)^T,$$

where C is given by (2). Then, we choose a point x from our design space that maximizes d_n, denoted as

$$\bar{d}_n = \max_x d_n$$

For each step, the Fisher information matrix is calculated as:

$$M_{n+1} = (1 - \alpha_{n+1})M_n + \alpha_{n+1} f(x)_{(n+1)} f(x)^T_{(n+1)},$$

where α_{n+1} is our optimum step length. We calculate α_{n+1} by using the General Equivalence Theorem. Define a function $F_{\phi_t}(M(\xi;\Theta), f(x)f(x)^T)$ such that

$$F_{\phi_t}(M(\xi;\Theta), f(x)f(x)^T) = f(x)Cf(x)^T - tr((A^T M^-(\xi;\Theta)A)^{-t}).$$

Thus, we find α_{n+1} by setting

$$F_{\phi_t}\left(M_{n+1}(\xi;\Theta), f(x)_{(n+1)} f(x)^T_{(n+1)}\right) = \bar{d}_n - tr((A^T M^-_{n+1}(\xi;\Theta)A)^{-t}) = 0$$

and solving for α_{n+1}. Or, we can simply set $\alpha_{n+1} = \frac{1}{n+1}$. We continue in this stepwise fashion until $F_{\phi_t}\left(M_{n+1}(\xi;\Theta), f(x)_{(n+1)} f(x)^T_{(n+1)}\right)$ is very close to 0. When this stopping point is reached, we have our optimal design points (Federov and Hackl, 1997).

2.5 Newton-Raphson Algorithm

To obtain our optimal weights, we must use the Newton-Raphson approach. Before we introduce this approach, we must first discuss the 2nd order Taylor approximation. Let f(x) be a function that is at least twice differentiable on an open interval I. For any two points x and x+h, the 2nd order Taylor approximation of f at x is:

$$f(x+h) \approx f(x) + f'(x)h + \frac{1}{2} f''(x)h^2$$

We can rewrite this as:

$$f(x+h) \approx a + bh + \frac{1}{2} ch^2$$

Here, a = f(x), b = f'(x) and c = f''(x). Thus, the 2nd order Taylor approximation is a 2nd order polynomial in h.

We are interested in maximizing this function with respect to x. Based on the equation above, this implies

$$f'(x+h) \approx b + ch$$

Let $\hat{h}$ be the first condition for the value of h that maximizes f(x+h). Then,

$$0 = b + c\hat{h},$$

and $\hat{h} = \frac{-b}{c}$. So, the value that maximizes the value of our 2nd order Taylor approximation of f at x is:

$$x + \hat{h} = x - \frac{b}{c}$$

$$x + \hat{h} = x - \frac{f'(x)}{f''(x)}.$$

This is the Newton-Raphson algorithm (Quinn, 2001).

Now, we can write this algorithm in terms of Ψ and w. Ψis a function of weights given the design points and parameters and is at least twice differentiable. Hyun, 2011 states that the nonnegative solutions of $\frac{\partial}{\partial w}\Psi = 0$ are the optimal design weights. Let Ψ = f(x), $w_{new} = x + \hat{h}$, and $w_{old} = x$. Then, by using the Newton-Raphson algorithm, we have:

$$w_{new} = w_{old} - \left[\frac{\partial}{\partial w}\Psi\right] \cdot \left[\frac{\partial^2}{\partial w^2}\Psi\right]^{-1}.$$

3. MODEL

In this section, we introduce the model under consideration for this paper. We then find the Fisher information matrix that we will use to obtain D-, c- and two-stage optimal designs.

3.1 The Hill Model

Before we begin discussing the types of optimal designs used in this paper, we shall give a brief background of the Hill model, the model selected for use in this paper. The Hill model is used to give the average activity of an enzyme per catalytic site as a function of the total substrate concentration. Below, we give the equation for our model:

$$k = \frac{k_{max}\left(\frac{[S_T]}{S_{50}}\right)^h}{1+\left(\frac{[S_T]}{S_{50}}\right)^h}$$

In the model above, k is our average enzyme activity, $[S_T]$ is the total substrate concentration, k_{max} is the maximum possible activity obtained for our substrate concentrations, S_{50} is the substrate concentration that gives half of the total activity, or $\frac{1}{2}*k_{max}$, and h is known as the Hill coefficient. If h is greater than 1, the model is said to exhibit positive Hill cooperativity. Adversely, if h is less than 1, the model exhibits negative Hill cooperativity.

For the purposes of simplicity, we decided to reparameterize our model in terms of alpha, beta and gamma. Our reparameterized model is given below:

$$\mu(x_i;\Theta) = \frac{\alpha\left(\frac{x_i}{\beta}\right)^\gamma}{1+\left(\frac{x_i}{\beta}\right)^\gamma} \tag{3}$$

For this model, x_i is our i^{th} dose, μ is our mean response, α is an amplitude scale parameter, β is a concentration scale parameter and γ is a shape parameter (Radivoyevitch, 2009). Figure 1 shows

the dose-response curve for the given values of the parameters.

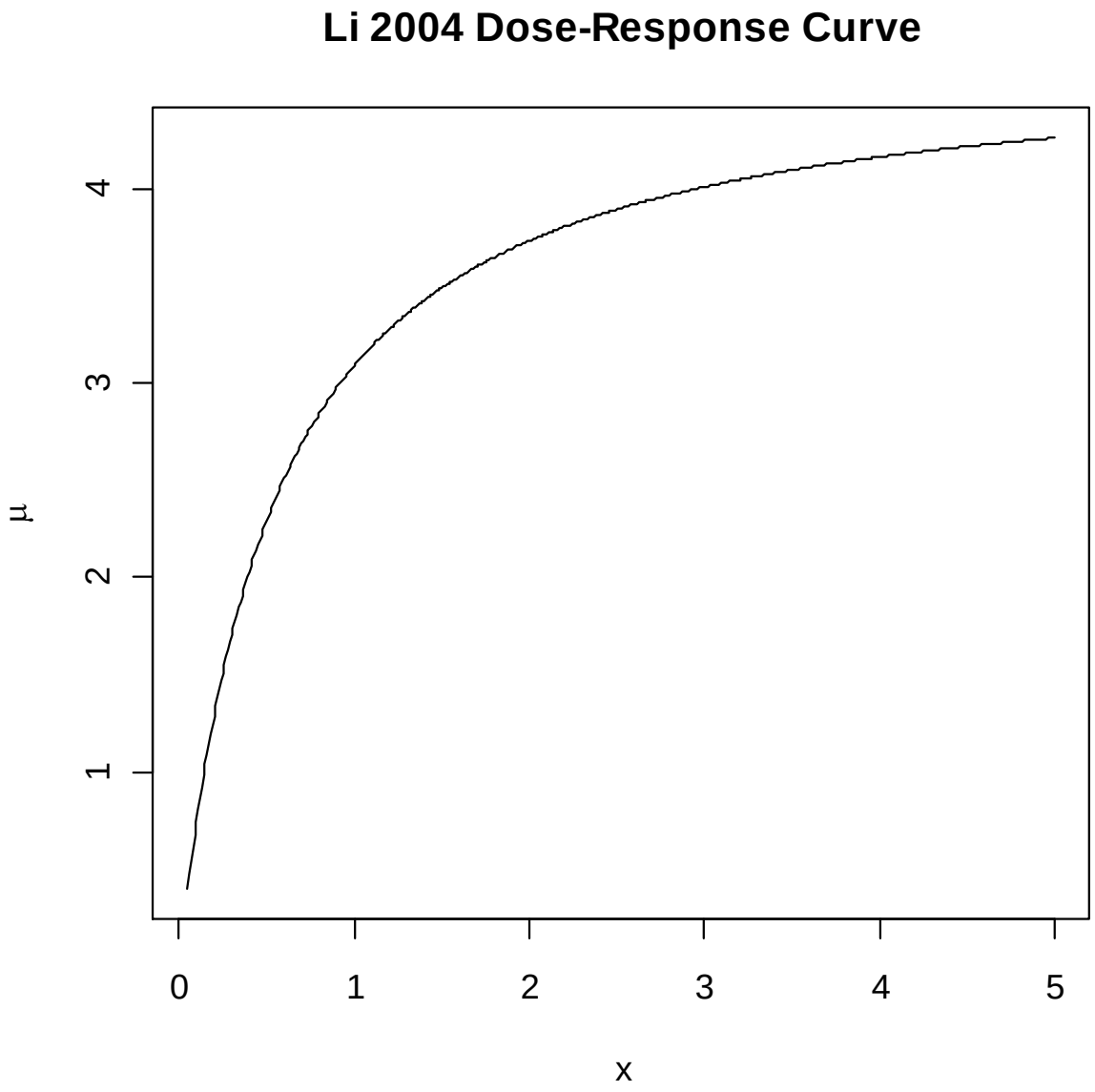

Figure 1: The Hill Model when alpha = 4.7, beta = 0.525, gamma = 1.01

3.2 Model for Observation

To perform our research, we shall consider a standard normal model with a continuous response variable. Our model under consideration is given below:

$$y_{ij} = \mu(x_i;\, \Theta) + \varepsilon_{ij}, \tag{4}$$

where ε_{ij} are iid $N(0, \sigma^2)$, j = 1, 2, …, n_i, i = 1, 2, … k, and σ^2 is assumed to be unknown. Here, we denote Θ as the vector of our parameters α, β and γ. That is,

$$\Theta = (\alpha, \beta, \gamma)'$$

Also, $\mu(x_i; \Theta)$ stands for the Hill model (3) discussed before. In this paper, we are focusing on finding design points x_i's and associated weights w_i's. This implies that we are finding "locally" optimal designs for each criterion, i.e. we assume that the values of the parameters are known and focus only on finding optimal design points and weights. One important note is that our results are not changed by the value of σ^2.

For our research, we also need to construct the Fisher information matrix, which will be denoted as $M(\xi; \Theta)$, where ξ is used to denote a design. To compute our information matrix, we use the following formula:

$$M(\xi; \Theta) = \frac{n}{\sigma^2} \sum_{i=1}^{k} w_i f(x_i)' f(x_i)$$

Here, $f(x_i)$ is a vector of the first-order derivatives of our mean function $\mu(x_i; \Theta)$. That is,

$$f(x) = \left(\frac{\partial \mu(x; \Theta)}{\partial \alpha}, \frac{\partial \mu(x; \Theta)}{\partial \beta}, \frac{\partial \mu(x; \Theta)}{\partial \gamma} \right)'$$

$$= \left(\frac{\left(\frac{x}{\beta}\right)^{\gamma}}{1 + \left(\frac{x}{\beta}\right)^{\gamma}}, \quad \frac{-\alpha\gamma \left(\frac{x}{\beta}\right)^{\gamma}}{\beta \left(1 + \left(\frac{x}{\beta}\right)^{\gamma}\right)^2}, \quad \frac{\alpha \left(\frac{x}{\beta}\right)^{\gamma} \ln \frac{x}{\beta}}{\left(1 + \left(\frac{x}{\beta}\right)^{\gamma}\right)^2} \right)'$$

$$= \frac{\left(\frac{x}{\beta}\right)^{\gamma}}{1 + \left(\frac{x}{\beta}\right)^{\gamma}} \begin{pmatrix} 1 & \frac{-\alpha\gamma}{\beta \left(1 + \left(\frac{x}{\beta}\right)^{\gamma}\right)} & \frac{\alpha \ln \frac{x}{\beta}}{\left(1 + \left(\frac{x}{\beta}\right)^{\gamma}\right)} \end{pmatrix}'.$$

Thus, the Fisher information matrix for our model is given below:

$$M(\xi;\Theta)=\frac{n}{\sigma^2}\sum_{i=1}^{k}w_i\frac{\left(\frac{x_i}{\beta}\right)^{2\gamma}}{\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)^2}\begin{pmatrix}1 & \frac{-\alpha\gamma}{\beta\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)} & \frac{\alpha\ln\frac{x_i}{\beta}}{\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)}\\ \frac{-\alpha\gamma}{\beta\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)} & \frac{\alpha^2\gamma^2}{\beta^2\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)^2} & \frac{-\alpha^2\gamma\ln\frac{x_i}{\beta}}{\beta\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)^2}\\ \frac{\alpha\ln\frac{x_i}{\beta}}{\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)} & \frac{-\alpha^2\gamma\ln\frac{x_i}{\beta}}{\beta\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)^2} & \frac{\alpha^2\left(\ln\frac{x_i}{\beta}\right)^2}{\left(1+\left(\frac{x_i}{\beta}\right)^{\gamma}\right)^2}\end{pmatrix}.$$

Now, this information matrix is used to obtain optimal designs. As shown in the background theory, optimality criteria totally depend on the Fisher information matrix. Thus, the Fisher information matrix is essential in order to obtain optimal designs.

4. OPTIMAL DESIGNS

In this chapter, optimal designs are obtained under the model (4). Here, we find D, c and two-stage optimal designs. To find the design points, the V-algorithm will be used for all three designs. To find the optimal weights, the Newton-Raphson algorithm will be used. We use design space $0 \leq x_i \leq 5$ to obtain our optimal designs. This interval is the same interval used by Li back in 2004. As said before, the focus of this paper is to find locally optimal designs, which is the optimum for the given value of parameters, not to estimate the true values of our parameters. The locally optimal designs are used as bench-markers for the given values of parameters, and all other designs can be compared to the locally optimal designs in order to see their performance. For these designs, we assume the values of the parameters are: $\hat{\alpha} = 4.7, \hat{\beta} = 0.525$ and $\hat{\gamma} = 1.01$. The values of parameters are the same values used by Li, 2004.

4.1 Uniform Design

A uniform design is a traditionally used design to study dose-response functions. Here, a researcher may consider equally spaced design points, with equal weights assigned to each point. Since Li, 2004 used 14 design points in their study, we consider the following uniform design, denoted as ξ^U:

$$\xi^U = \begin{pmatrix} 0.05 & 0.40 & 0.75 & 1.10 & 1.45 & 1.80 & 2.15 & 2.50 & 2.85 & 3.20 & 3.55 & 3.90 & 4.25 & 4.60 \\ \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} & \frac{1}{14} \end{pmatrix}.$$

4.2 D-optimal Design

D-optimal design is used when the goal of an experiment is to estimate model parameters. To obtain a locally D-optimal design, we obtain design points and associated weights that will minimize the determinant of the inverse of the Fisher information matrix $M(\xi; \Theta)$. Thus, D-optimality is minimizing

$$\Psi = |M^{-1}(\xi; \Theta)| \qquad (5)$$

As mentioned before, the D-optimal design is obtained using numerical algorithms. Also, as proven by Hyun et. al, 2011, the weights for a D-optimal design are equal when the number of design points is equal to the number of parameters. The obtained D-optimal design is verified by the General Equivalence Theorem. According to the theorem, a design ξ* is a D-optimal design if and only if

$$f(x_i)'M^{-1}(\xi^*;\Theta)f(x_i) \leq m, \forall x_i \qquad (6).$$

Here, m is the number of parameters in our model. The left side of (6) is a standardized variance of predicted response. In this case, the standardized variance for any x_i will be less than or equal to 3, and equality will hold if and only if x_i is a D-optimal design point. Also, $f(x_i)$ is the vector of the first-order derivatives of our model.

Based on Caratheodory's theorem, the initial design used here is given below:

$$\xi_0 = \begin{pmatrix} 0.05 & 0.5 & 1 & 2.3 & 4.2 & 5 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}$$

Based on the numerical algorithms, the D-optimal design is obtained below:

$$\xi^D = \begin{pmatrix} 0.15 & 0.95 & 4.95 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

So, the D-optimal design is to assign about 33% of the subjects to each of 0.15, 0.95 and 4.95. This will minimize the criterion (5) under the given values of the parameters. The D-optimal design is verified by the General Equivalence Theorem (the left side of (6) is plotted over the design space [0, 5]).

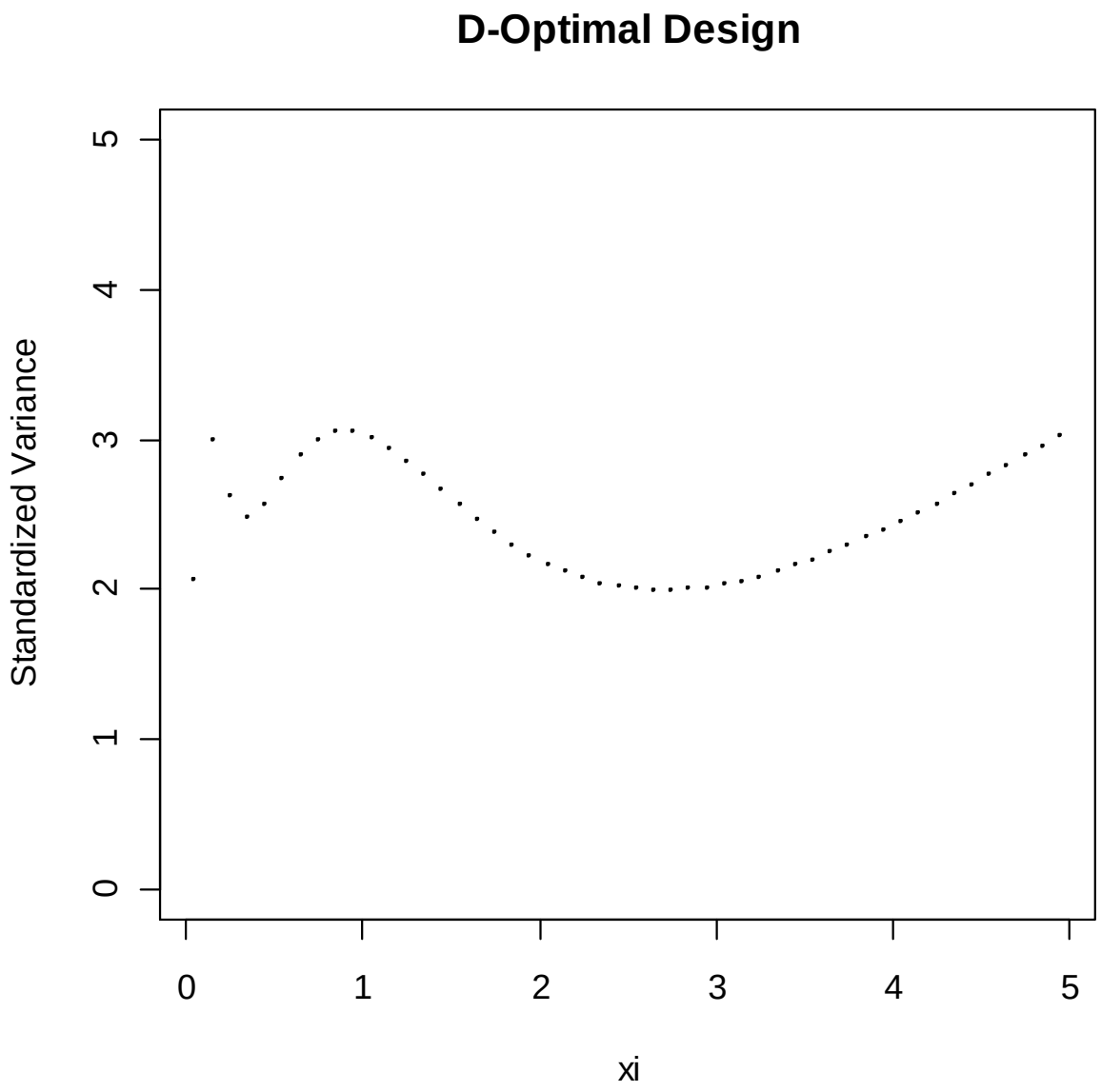

Figure 2: D-optimal Design: Plotting a standardized variance of predicted response over the design space [0, 5]

As stated in the General Equivalence Theorem, the plot hits the maximum when the design points are D-optimal design points. Otherwise, the plots are always less than the maximum.

4.3 c-optimal Design

The goal of a c-optimal design is to estimate a particular effective dose level, which can only be expressed as a function of parameters. For instance, a researcher may be interested in

finding the dosage level that produces the 50% of the maximum response, denoted as ED_{50}. Now we focus on finding a c-optimal design, which has design points and corresponding weights that estimate a particular dosage level ED_p precisely.

To find a c-optimal design to estimate ED_p, we must solve the following equation for x:

$$p = \frac{f(x;\theta) - \theta_1}{\theta_2 - \theta_1}$$

Here, p is 100*p% of the maximum response, f(x; θ) is our mean function, θ_1 is a minimum effective dose response level and θ_2 is a maximum effective dose response level. So, we replace the variables above with our information. Here, $\mu(x_i;\theta) = f(x;\theta)$, θ_1 will be assumed to be equal to 0, and $\theta_2 = \alpha$. Hence, we have:

$$p = \frac{\mu(x_i;\Theta)}{\alpha}$$

$$p = \frac{\left(\frac{x_i}{\beta}\right)^{\gamma}}{1 + \left(\frac{x_i}{\beta}\right)^{\gamma}}$$

Solving this equation with respect to x_i yields the following formula for ED_p:

$$x_i = ED_p = \beta\left(\frac{p}{1-p}\right)^{\frac{1}{\gamma}}$$

Then, we take the derivatives with respect to α, β and γ to obtain ED_p', which will be used to calculate our standardized variance.

$$ED_p' = \begin{pmatrix} 0 & \left(\frac{p}{1-p}\right)^{\frac{1}{\gamma}} & \frac{-\beta\left(\frac{p}{1-p}\right)^{\frac{1}{\gamma}} \ln\frac{p}{1-p}}{\gamma^2} \end{pmatrix}.$$

To obtain our c-optimal design, we need to find a design that minimizes the variance of estimating ED_p:

$$Var\left(\widehat{ED}_p\right) \approx ED_p' M^{-1}(\xi;\Theta)\left[ED_p'\right]^T \quad (7)$$

To find the design points for this model, we use the V-algorithm. Then, to find the optimal weights for our points, we use the Newton-Raphson algorithm. After accomplishing this, we use the General Equivalence Theorem to verify that the points we find are indeed the correct c-optimal points. In this case, a design ξ* is c-optimal if and only if:

$$tr\left\{\mu(x_i,\theta)M^{-1}(\xi^*;\Theta)\left[ED_p'\right]^T ED_p' M^{-1}(\xi^*;\Theta)\right\} \leq tr\left\{\left[ED_p'\right]^T ED_p' M^{-1}(\xi^*;\Theta)\right\}, \forall x_i \quad (8)$$

Here, $\mu(x_i,\theta)$ is our mean function from the model (2). The left side of this inequality is the standardized variance of our predicted response. Again, we will only have equality if a point x_i is a c-optimal point. For this design, set p = 0.7, $\hat{\alpha} = 4.7, \hat{\beta} = 0.525$ and $\hat{\gamma} = 1.01$.

Here, we can choose an arbitrary initial design. Without loss of generality, we can assume that we have equal weights for our design. The initial c-optimal design is given below:

$$\xi_0 = \begin{pmatrix} 0.5 & 1 & 3 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

After using the V-algorithm to find our c-optimal points and the Newton-Rhapson algorithm to find the optimal weights, we obtain our c-optimal design for ED_{70}:

$$\xi^C = \begin{pmatrix} 0.075 & 0.92 & 5 \\ 0.222 & 0.474 & 0.304 \end{pmatrix}$$

The c-optimal design is to assign 22.2% of the subjects to 0.075, 47.4% of the subjects to 0.92, and 30.4% of the subjects to 5. This will minimize the criterion (7) under the given value of the parameters and the given value of p. Thus, this is the best design to estimate ED_{70} under the given value of the parameters. The c-optimal design is verified using the General Equivalence Theorem (the left side of (8) is plotted over the design space [0,5]).

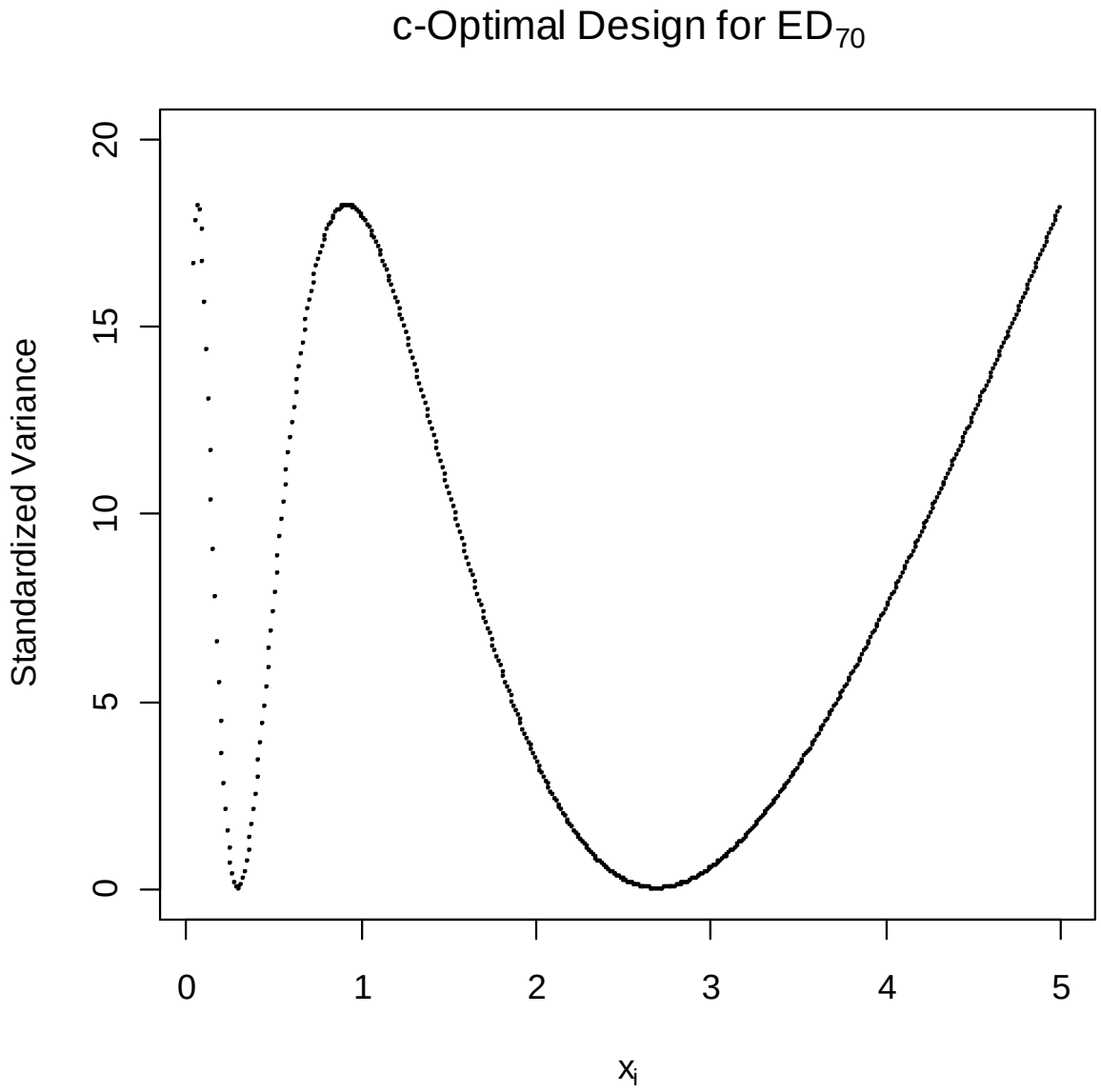

Figure 3: c-optimal Design: Plotting a standardized variance of an effective dosage level over the design space [0,5]

As stated in the General Equivalence Theorem, the plot hits the maximum when the design points are c-optimal design points. Otherwise, the plots are always less than the maximum.

4.4 Two-Stage Design

When conducting a dose-response study, researchers sometimes have multiple objectives, such as selecting a target dose, estimating dose-response, identifying clinical relevance, etc. An

optimal design can be obtained for each of these goals individually. As demonstrated earlier, we can find a D-optimal design to estimate the dose response, and a c-optimal design for estimating a particular effective dose level. However, if we consider these designs for different objectives, they may be extremely inefficient. Therefore, designs that can efficiently address several objectives are more advantageous. With that in mind, we introduce a two-stage design.

Here, we discuss a two-stage design that addresses both D-optimality and c-optimality criteria at the same time. This type of design will allow researchers to study the shape of the dose response curve and study a particular dosage level. The process of obtaining a two-stage optimal design is as follows. In the first stage of the study, a D-optimal design is used to estimate the dose-response curve. Then, after this has been established, the focus is shifted to finding a target dose level. A question that arises with this type of design is the timing of the switch from D-optimality to c-optimality. For simplicity, we use the first half of subjects and assign them to the D-optimal design to learn about the dose-response curve. The second half of the subjects is assigned according to the c-optimal design to estimate our dosage level of interest. However, the c-optimal design obtained during the second stage must take the existing allocation of subjects at the first stage into account. Because of this, the c-optimal design at the second stage is not a true optimal design but an augmented optimal design, taking into account the existing D-optimal design from the first stage (see Atkinson et al., 2007 and Padmanabhan and Dragalin, 2010). Let ξ_1 denote the D-optimal design from the first stage, and $M(\xi_1;\Theta)$ is the Fisher information matrix for the first stage using ξ_1. Then, our augmented c-optimality criterion to be maximized is

$$\Psi = tr\left\{\mu(x;\Theta)\widetilde{M}^{-1}(\xi;\Theta)\left[ED_p'\right]^T ED_p'\widetilde{M}^{-1}(\xi;\Theta)\right\} \quad (9)$$

where

$$\widetilde{M}(\xi;\Theta) = \frac{1}{2}(M(\xi_1;\Theta) + M(\xi;\Theta)).$$

A design ξ that minimizes (9) is the augmented c-optimal design that is needed for the second stage of the design.

Next, we shall obtain a two-stage optimal design to study dose-response function and a target dose level effectively at the same time. For the sake of comparison, we shall once again estimate ED_{70}. There is a slight difference in terms of starting our design. Here, the points we found above in our D-optimal design will not change, and the weights will be halved. So, we start out with an initial design for the second stage for finding an augmented c-optimal design. Like before, we specify an initial design ξ^0, which is given below. One can notice that the first three points are from our D-optimal design and the next three are random points.

$$\xi_0 = \begin{pmatrix} 0.15 & 0.95 & 4.95 & 0.5 & 1 & 3 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{pmatrix}.$$

After using the V-algorithm and the Newton-Raphson algorithm, we obtain the following augmented c-optimal design:

$$\xi^{ac} = \begin{pmatrix} 0.075 & 0.92 & 5 \\ 0.128 & 0.611 & 0.261 \end{pmatrix}.$$

. A design ξ^* is an augmented c-optimal design if and only if:

$$tr\left\{\mu(x;\Theta)\widetilde{M}^{-1}(\xi;\Theta)\left[ED_p'\right]^T ED_p'\widetilde{M}^{-1}(\xi;\Theta)\right\} \leq tr\left\{\left[ED_p'\right]^T ED_p'\widetilde{M}^{-1}(\xi;\Theta)\right\}.$$

The equality will hold if a point x_i is an augmented c-optimal design point. The General Equivalence Theorem is used to verify the augmented c-optimal design. As stated in the General Equivalence Theorem, the graph hits its maximum when the design points are augmented c-optimal design points.

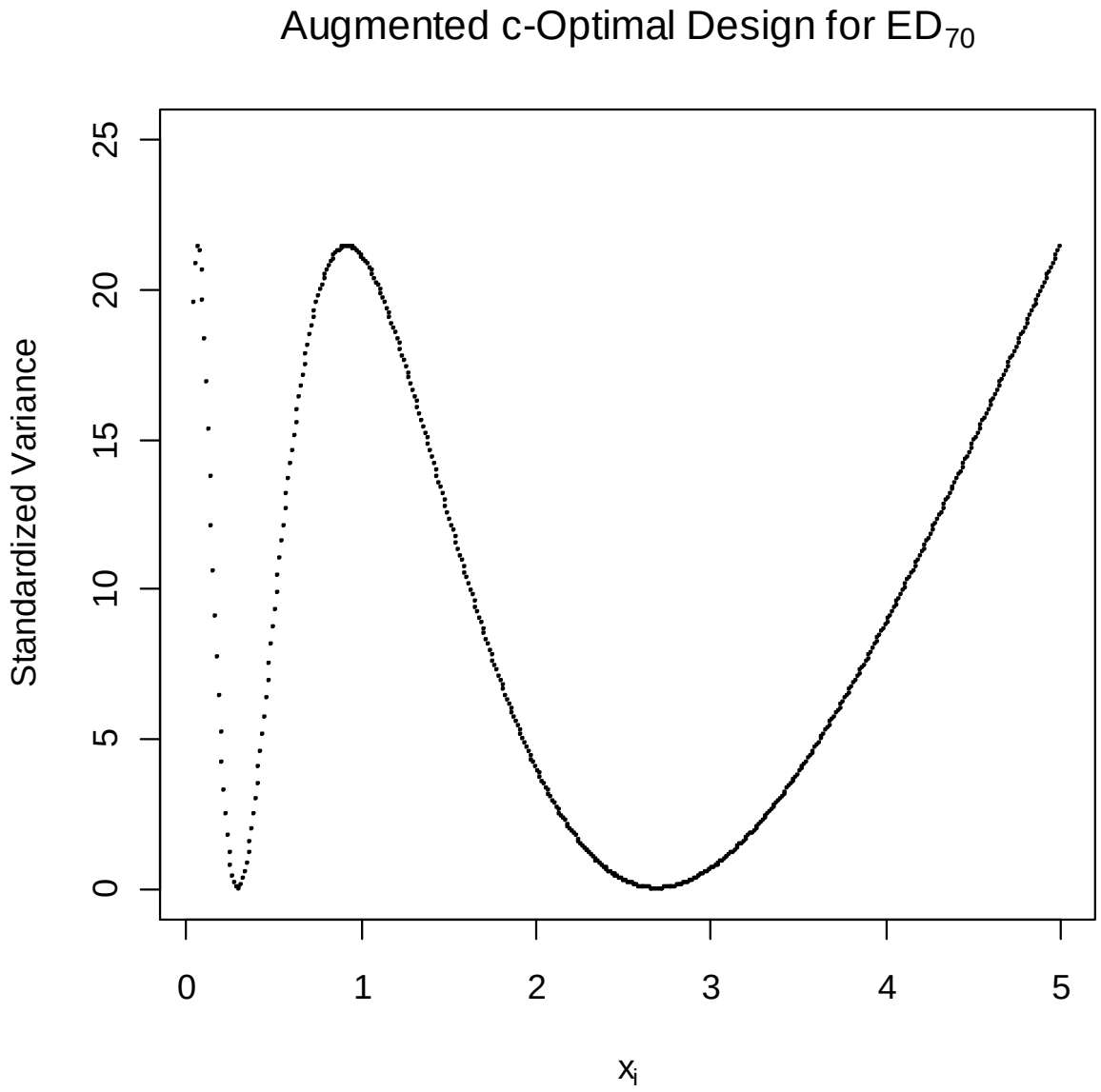

Figure 4: Two-Stage Design: Plotting a standardized variance of the augmented c-optimal design over [0,5]

Then, the two-stage optimal design uses ξ^D for the first half of the subjects and ξ^{ac} for the second half. The two-stage optimal design is:

$$\xi^2 = \begin{pmatrix} 0.075 & 0.15 & 0.92 & 0.95 & 4.95 & 5 \\ 0.064 & 0.167 & 0.306 & 0.167 & 0.166 & 0.130 \end{pmatrix}.$$

Our two-stage optimal design is to assign 6.4% of our subjects to 0.075, 16.7% of the subjects to 0.15, 30.6% of the subjects to 0.92, 16.7% of the subjects to 0.95, 16.6% of the subjects to 4.95, and the remaining 13% of the subjects to 5.

5. EFFICIENCY

In this section, we discuss the efficiencies of our optimal designs with respect to D and c-optimality criteria. Efficiency can tell us a performance of a design under a specific goal of an experiment. For a design ξ, if an efficiency of ξ is ℓ, the design ξ needs $100(1/\ell - 1)\%$ more subjects to provide the same accuracy for estimating interesting features as the optimal design provides. We will find the relative efficiencies for our designs. These will tell us how designs work with respect to some criteria. Here, we discuss the criterion for D-efficiency and c-efficiency. Then, we compare the efficiencies of the designs obtained in chapter 4.

5.1 D-Efficiency

To find the efficiency of a design with respect to D-optimality criterion, we compare the determinants of the inverses of the Fisher information matrix for D-optimal design ξ^D and another design ξ. The criterion for D-efficiency is denoted as Eff_D and is given below:

$$Eff_D(\xi) = \left(\frac{|M^{-1}(\xi^D;\Theta)|}{|M^{-1}(\xi;\Theta)|}\right)^{\frac{1}{m}}.$$

Here, m is the number of parameters in our model.

5.2 c-Efficiency

To find the efficiency of a design with respect to c-optimality criterion, we compare the variance of some dosage level EDp for some design ξ to the variance of c-optimal design ξ^c for the same dosage level. We use ED_{70} again to compute c-efficiency. The criterion for c-efficiency is denoted as Eff_c and is given as follows:

$$Eff_c(\xi) = \frac{ED'_{70}M^{-1}(\xi^c;\Theta)[ED'_{70}]^T}{ED'_{70}M^{-1}(\xi;\Theta)[ED'_{70}]^T}.$$

5.3 Efficiencies of Optimal Designs

Using the results from chapter 4, we compare the efficiencies of the designs using c-optimality criteria and D-optimality criteria. We expect to see that the D-optimal design does not perform well under c-optimality criteria, and vice versa. However, we expect to see that the two-stage design performs well under both criteria. The table below gives the values of these efficiencies for each of our designs:

	D-optimality	**c-optimality (ED_{70})**
D-Optimal Design	1	0.8931948
c-Optimal Design	0.7803313	1
Two-Stage	0.9418978	0.9222931
Uniform	0.01578803	0.4017052

Table 1: Efficiencies of Optimal Designs

Clearly, we see that the D-optimal design works best for estimating model parameters, but does not hold up well under c-optimality criteria, and the c-optimal design works best for estimating ED_{70}, but doesn't perform very well under D-optimality criterion. We also observe that the two-stage design, while not performing as well as a single criterion design, still performs fairly well for both criteria. The uniform design chosen performed very poorly with each criterion. One note to make is that our D-optimal design did not have a very low efficiency for c-optimality criteria, and the same can be said for our c-optimal design under D-optimality criteria. However, this is most likely due to the fact that each design had very similar design points.

6. CONCLUSION

Optimal designs can be used in all areas of statistics to help a researcher specify which treatments to use and how to distribute subjects in the most specific manner. A D-optimal design to study a dose-response curve and a c-optimal design to estimate a target dose ED_p are obtained. A two-stage design that combines the criteria for D- and c-optimality was also obtained, and the efficiencies of each design were compared to each other, and to a traditional uniform design.

After comparing the efficiencies, it was found that the D-optimal design to estimate model parameters does not perform well when looking at c-optimality criteria, and the c-optimal design to estimate the target dose ED_{70} does not work well under D-optimality criteria. However, the two-stage design performed fairly well under both criteria. While not as efficient as the D-optimal design with D-optimality criteria or the c-optimal design with c-optimality criteria, it performed better than the D-optimal design under c-optimality criteria and the c-optimal design under D-optimality criteria. Also, as expected, the traditional uniform design performed poorly with respect to each optimality criteria. Hence, a two-stage design combining D- and c-optimality could be a valid option for a researcher that is interested in studying both a dose-response curve and a target dose ED_p at the same time.

The designs obtained in this paper were locally optimal designs. As stated before, we assumed the values of the parameters to be true and focused on finding optimal design points and optimal weights for each design on the design space [0,5]. Future work for this study is to estimate the true values of the parameters for the Hill model using sequential updated optimal designs. Also, different types of two-stage optimal designs can be considered with respect to different criteria. There are many optimality criteria that can be considered, and finding different combinations can prove advantageous to researchers who are looking to conduct experiments

with varying research goals. Also, future work could include finding optimal designs for models with 4 or more parameters.

REFERENCES

Atkinson, Anthony C., "Examples of the Use of an Equivalence Theorem in Constructing Optimum Experimental Designs for Random-Effects Nonlinear Regression Models," London School of Economics, 2007.

Bretz, F., Koenig, F., Brannath, W., Glimm, E. and Posch, M., Ädaptive designs for confirmatory clinical trials," *Statistics in Medicine,* 28 (2009), 1181-1217.

Dette, H., Pepelyshev, A. and Zhigljavsky A., "Improving updating rules in multiplicative algorithms for computing D-optimal designs," *Computational Statistics & Data Analysis,* 53 (2008), 312–320.

Dragalin, V., Hsuan, F. and Padmanabhan, S.K., "Adaptive Designs for dose-finding studies based on the sigmoid Emax model," *Journal of Biopharmaceutical Statistics,* 17 (2007), 1051–1070.

Federov, Veleri V. and Hackl, Peter, *Model-Oriented Design in Experiments,* 1997.

Hyun, Seung Won, Yang, Min and Flournoy, Nancy, "A Procedure for Finding an Improved Upper Bound on the Number of Optimal Design Points," 2011.

Hyun, Seung Won, Yang, Min and Flournoy, Nancy, "Optimal designs for response functions with a downturn," *Journal of Statistical Planning and Inference,* 141 (2011), 559-575.

Kiefer, J., "On the nonrandomized optimality and randomized nonoptimality of symmetrical designs," *Annals of Mathematical Statistics,* 29 (1958), 675-699.

Leonov, S. and Miller, S., "An adaptive optimal design for the Emax model and its application in clinical trials," *Journal of Biopharmaceutical Statistics,* 19 (2009) (2), 360 – 385.

Li CL, Lu CY, ke PY and Chang ZF, "Perturbation of ATP-induced tetramerization of human cytosolic thymidine kinase by substitution of serine-13 with aspartic acid at the mitotic phosphorylation site," *Biochem Biophys Res Commun,* 313(3) (2004), 587-593.

Miller, F., Guilbaud, O. and Dette, H., "Optimal designs for estimating the interesting part of a dose-effect curve," *Journal of Biopharmaceutical Statistics,* 17 (2007), 1097 – 1115.

Montgomery, Douglas C., *Design and Analysis of Experiments, Sixth Edition,* 2009.

Padmanabhan, S. Krishna and Dragalin, Vladimir, "Adaptive Dc-optimal designs for dose finding based on a continuous efficacy endpoint," *Biometric Journal,* 52 (2010) 6, 836-862.

Quinn, Kevin, "The Newton Raphson Algorithm for Function Optimization," University of Washington, 2001.

Radivoyevitch, Tomas, "Mass action models versus the Hill model: An analysis of tetrameric human thymidine kinase I positive cooperativity," *Biology Direct*, 2009, 4:49.

Ruberg, Stephen, "Dose Response Studies. I. Some Design Considerations," *Journal of Biopharmaceutical Statistics,* 5(1) (1995), 1-14.

APPENDIX A. R CODE FOR D-OPTIMAL DESIGN

```
###D-optimal Design###

#Number of Parameters
k=3

#Value of Parameters
alpha=4.7
beta=0.525
gamma=1.01

#Initial value
x0=c(.05, .5, 1, 2.3, 4.2, 5)
n0=length(x0)
w=rep(1/n0,n0)
D=rbind(x0,w)
#Initial Information matrix
A1<-rep(0,n0)
A2<-rep(0,n0)
A3<-rep(0,n0)
A4<-rep(0,n0)
A5<-rep(0,n0)
A6<-rep(0,n0)
for (i in 1:n0)
{
A1[i]=w[i]*((x0[i]/beta)^gamma / (1+(x0[i]/beta)^gamma))^2
A2[i]=w[i]*(-alpha)*gamma*(x0[i]/beta)^(2*gamma) /
(beta*(1+(x0[i]/beta)^gamma)^3)
A3[i]=w[i]*alpha*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
((1+(x0[i]/beta)^gamma)^3)
A4[i]=w[i]*alpha^2*gamma^2*(x0[i]/beta)^(2*gamma) /
(beta^2*(1+(x0[i]/beta)^gamma)^4)
A5[i]=w[i]*(-alpha^2)*gamma*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
(beta*(1+(x0[i]/beta)^gamma)^4)
A6[i]=w[i]*alpha^2*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta)^2 /
(1+(x0[i]/beta)^gamma)^4
}
M0=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
IM0=solve(M0)


#Find dn,

f<-function(x)
{matrix(c((x/beta)^gamma / (1+(x/beta)^gamma), -alpha*gamma*(x/beta)^gamma
/ (beta*(1+(x/beta)^gamma)^2), alpha*(x/beta)^(gamma)*log(x/beta) /
(1+(x/beta)^gamma)^2),nrow=3,ncol=1,byrow=F)}
p=1
while(p>.05){
x1=seq(0.05,5,.1)
n1=length(x1)
```

```
dn=rep(0,n1)
for (j in 1:n1)
{dn[j]=t(f(x1[j]))%*%solve(M0)%*%f(x1[j])}
for (j in 1:n1)
{if(max(dn)==dn[j])x1[j]=x1[j] else x1[j]=NA}
newX=na.omit(x1)
newdn=max(dn)

#Find alpha(n+1)
an=(newdn-k)/(k*(newdn-1))
p<-newdn-k
#Get M(n+1)
newM=(1-an)*M0+an*f(newX)%*%t(f(newX))
M0<-newM
newW=(1-an)*D[2,]
W=c(newW,an)
X=c(D[1,],newX)
newD=rbind(X,W)
D=newD
print(p)}

#Verify D-optimal design
#number of parameter
k=3
#value of parameter
alpha=4.7
beta=0.525
gamma=1.01

#D-optimal design
x=c(0.15, 0.95, 4.95)
n=length(x)
w=rep(1/3,n)
D=rbind(x,w)
#Information matrix
A1<-rep(0,n)
A2<-rep(0,n)
A3<-rep(0,n)
A4<-rep(0,n)
A5<-rep(0,n)
A6<-rep(0,n)
for (i in 1:n)
{
A1[i]=w[i]*((x[i]/beta)^gamma / (1+(x[i]/beta)^gamma))^2
A2[i]=w[i]*(-alpha)*gamma*(x[i]/beta)^(2*gamma) /
(beta*(1+(x[i]/beta)^gamma)^3)
A3[i]=w[i]*alpha*(x[i]/beta)^(2*gamma)*log(x[i]/beta) /
((1+(x[i]/beta)^gamma)^3)
A4[i]=w[i]*alpha^2*gamma^2*(x[i]/beta)^(2*gamma) /
(beta^2*(1+(x[i]/beta)^gamma)^4)
A5[i]=w[i]*(-alpha^2)*gamma*(x[i]/beta)^(2*gamma)*log(x[i]/beta) /
(beta*(1+(x[i]/beta)^gamma)^4)
A6[i]=w[i]*alpha^2*(x[i]/beta)^(2*gamma)*log(x[i]/beta)^2 /
(1+(x[i]/beta)^gamma)^4
```

```
}
M=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
IM=solve(M)


#Find dn

f<-function(x)
{matrix(c((x/beta)^gamma / (1+(x/beta)^gamma), -alpha*gamma*(x/beta)^gamma
/ (beta*(1+(x/beta)^gamma)^2), alpha*(x/beta)^(gamma)*log(x/beta) /
(1+(x/beta)^gamma)^2),nrow=3,ncol=1,byrow=F)}

phi.1 <- function(x){
      matrix(c(0, (x/(1-x))^(1/gamma), -beta*(x/(1-x))^(1/gamma)*log(x/(1-
x))/gamma^2), nrow=3, ncol=1, byrow=F)
}

p1=0.7
p=1
x1=seq(0.05,5,.1)
n1=length(x1)
dn=rep(0,n1)
for (j in 1:n1)
{dn[j]=t(f(x1[j]))%*%solve(M0)%*%f(x1[j])}
plot(x1,dn, ylim = c(0, 5), cex = 0.1, xlab = quote(x [i]), ylab =
"Standardized Variance", main="D-Optimal Design")

### Uniform Design
#Initial value
x0=c(0.05, 0.4, 0.75, 1.1, 1.45, 1.8, 2.15, 2.5, 2.85, 3.2, 3.55, 3.9,
4.25, 4.6)
n0=length(x0)
w=rep(1/n0,n0)
D=rbind(x0,w)
#Initial Information matrix
A1<-rep(0,n0)
A2<-rep(0,n0)
A3<-rep(0,n0)
A4<-rep(0,n0)
A5<-rep(0,n0)
A6<-rep(0,n0)
for (i in 1:n0)
{
A1[i]=w[i]*((x0[i]/beta)^gamma / (1+(x0[i]/beta)^gamma))^2
A2[i]=w[i]*(-alpha)*gamma*(x0[i]/beta)^(2*gamma) /
(beta*(1+(x0[i]/beta)^gamma)^3)
A3[i]=w[i]*alpha*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
((1+(x0[i]/beta)^gamma)^3)
A4[i]=w[i]*alpha^2*gamma^2*(x0[i]/beta)^(2*gamma) /
(beta^2*(1+(x0[i]/beta)^gamma)^4)
A5[i]=w[i]*(-alpha^2)*gamma*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
(beta*(1+(x0[i]/beta)^gamma)^4)
```

```
A6[i]=w[i]*alpha^2*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta)^2 /
(1+(x0[i]/beta)^gamma)^4
}
M0=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
IM0=solve(M0)

# D-optimal efficiency calculations
e.d1 <- det(M)^(1/3)
e.d2 <- (t(phi.1(0.7)) %*% solve(M) %*% phi.1(0.7))

# e.d1 <- 0.4731286
# e.d2 <- 23.34466

# Uniform efficiency calculations:
e.u1 <- det(M0) ^ 1/3
e.u2 <- (t(phi.1(0.7)) %*% solve(M0) %*% phi.1(0.7))

# e.u1 = 0.007469769
# e.u2 = 45.34811
```

APPENDIX B. R CODE FOR C-OPTIMAL DESIGN

```
## Generalized Inverse of a Matrix
ginv<-function(X, tol = sqrt(.Machine$double.eps))
{

  dnx <- dimnames(X)
  if(is.null(dnx)) dnx <- vector("list", 2)
  s <- svd(X)
  nz <- s$d > tol * s$d[1]
  structure(
    if(any(nz)) s$v[, nz] %*% (t(s$u[, nz])/s$d[nz]) else X,
    dimnames = dnx[2:1])
}
###c-optimality for research###
library(matrixcalc)
#Number of Parameters
k=3

#Value of Parameters
alpha=4.7
beta=0.525
gamma=1.01

#Initial value
x0=c(.5,1,3)
n0=length(x0)
w=rep(1/n0,n0)
D=rbind(x0,w)

#Initial Information matrix
A1<-rep(0,n0)
A2<-rep(0,n0)
A3<-rep(0,n0)
A4<-rep(0,n0)
A5<-rep(0,n0)
A6<-rep(0,n0)
for (i in 1:n0){
      A1[i]=w[i]*((x0[i]/beta)^gamma / (1+(x0[i]/beta)^gamma))^2
      A2[i]=-w[i]*alpha*gamma*(x0[i]/beta)^(2*gamma) /
(beta*(1+(x0[i]/beta)^gamma)^3)
      A3[i]=w[i]*alpha*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
((1+(x0[i]/beta)^gamma)^3)
      A4[i]=w[i]*alpha^2*gamma^2*(x0[i]/beta)^(2*gamma) /
(beta^2*(1+(x0[i]/beta)^gamma)^4)
      A5[i]=w[i]*(-alpha^2)*gamma*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
(beta*(1+(x0[i]/beta)^gamma)^4)
      A6[i]=w[i]*alpha^2*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta)^2 /
(1+(x0[i]/beta)^gamma)^4
}

M0=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
```

```
IM0=ginv(M0)


#Find dn,

f<-function(x){
      matrix(c((x/beta)^gamma / (1+(x/beta)^gamma), -
alpha*gamma*(x/beta)^gamma / (beta*(1+(x/beta)^gamma)^2),
alpha*(x/beta)^(gamma)*log(x/beta) /
(1+(x/beta)^gamma)^2),nrow=3,ncol=1,byrow=F)
}

phi.1 <- function(x){
      matrix(c(0, (x/(1-x))^(1/gamma), -beta*(x/(1-x))^(1/gamma)*log(x/(1-
x))/gamma^2), nrow=3, ncol=1, byrow=F)
}

p=1
t=2

while(p>.005){
      x1=seq(0.01,5,.005)
      p1=0.7
      n1=length(x1)
      dn=rep(0,n1)
      for (j in 1:n1)
      {dn[j]=(t(f(x1[j]))%*%ginv(M0)%*%phi.1(p1))^2}
      for (j in 1:n1)
            {if(max(dn)==dn[j])x1[j]=x1[j] else x1[j]=NA}
      newX=na.omit(x1)
      newdn=max(dn)
      k=t(phi.1(p1))%*%ginv(M0)%*%phi.1(p1)

#Find alpha(n+1)
      #an=(newdn-k)/(k*(newdn-1))
      an=1/t
      #p<-abs(newdn-k)
f
#Get M(n+1)
      newM=c(1-an)*M0+c(an)*f(newX)%*%t(f(newX))
      M0<-newM
      p=abs((t(f(newX))%*%ginv(M0)%*%phi.1(p1))^2-
(t(phi.1(p1))%*%ginv(M0)%*%phi.1(p1)))
      newW=(1-an)*D[2,]
      W=c(newW,an)
      X=c(D[1,],newX)
      newD=rbind(X,W)
      D=newD
      print(p)
      t=t+1
}

###ED70 points: 0.075, 0.92, 5)
```

```
#Verify c-optimal design

#number of parameter
k=3

#value of parameter
alpha=4.7
beta=0.525
gamma=1.01

#c-optimal design
x=c(0.075,.92, 5)
n=length(x)
w=c(.222,0.474, 0.304)
D=rbind(x,w)

#Information matrix
A1<-rep(0,n)
A2<-rep(0,n)
A3<-rep(0,n)
A4<-rep(0,n)
A5<-rep(0,n)
A6<-rep(0,n)
for (i in 1:n){
      A1[i]=w[i]*((x[i]/beta)^gamma / (1+(x[i]/beta)^gamma))^2
      A2[i]=w[i]*(-alpha)*gamma*(x[i]/beta)^(2*gamma) /
(beta*(1+(x[i]/beta)^gamma)^3)
      A3[i]=w[i]*alpha*(x[i]/beta)^(2*gamma)*log(x[i]/beta) /
((1+(x[i]/beta)^gamma)^3)
      A4[i]=w[i]*alpha^2*gamma^2*(x[i]/beta)^(2*gamma) /
(beta^2*(1+(x[i]/beta)^gamma)^4)
      A5[i]=w[i]*(-alpha^2)*gamma*(x[i]/beta)^(2*gamma)*log(x[i]/beta) /
(beta*(1+(x[i]/beta)^gamma)^4)
      A6[i]=w[i]*alpha^2*(x[i]/beta)^(2*gamma)*log(x[i]/beta)^2 /
(1+(x[i]/beta)^gamma)^4
}

M=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
IM=solve(M)


#Find dn

f<-function(x){
      matrix(c((x/beta)^gamma / (1+(x/beta)^gamma), -
alpha*gamma*(x/beta)^gamma / (beta*(1+(x/beta)^gamma)^2),
alpha*(x/beta)^(gamma)*log(x/beta) /
(1+(x/beta)^gamma)^2),nrow=3,ncol=1,byrow=F)
}

phi.1 <- function(x){
      matrix(c(0, (x/(1-x))^(1/gamma), -beta*(x/(1-x))^(1/gamma)*log(x/(1-
x))/gamma^2), nrow=3, ncol=1, byrow=F)
```

```
}

p1=0.7
p=1
x1=seq(0.05,5,.01)
n1=length(x1)
dn=rep(0,n1)

for (j in 1:n1){
      dn[j]=(t(f(x1[j]))%*%ginv(M)%*%phi.1(p1))^2
}

plot(x1,dn, ylim = c(0, 20), cex = 0.1, xlab = quote(x [i]), ylab =
"Standardized Variance", main="c-Optimal Design for ED70")

# Efficiency for ED70 c-optimal design
e.c2 <- (t(phi.1(0.7)) %*% solve(M) %*% phi.1(0.7))
e.c1 <- det(M)^(1/3)
# e.c1 = 0.423596
# e.c2 = 18.21657
```

APPENDIX C. R CODE FOR TWO-STAGE DESIGN

```
## Generalized Inverse of a Matrix
ginv<-function(X, tol = sqrt(.Machine$double.eps))
{

  dnx <- dimnames(X)
  if(is.null(dnx)) dnx <- vector("list", 2)
  s <- svd(X)
  nz <- s$d > tol * s$d[1]
  structure(
    if(any(nz)) s$v[, nz] %*% (t(s$u[, nz])/s$d[nz]) else X,
    dimnames = dnx[2:1])
}
###2-stage optimality for research###
library(matrixcalc)
#Number of Parameters
k=3

#Value of Parameters
alpha=4.7
beta=0.525
gamma=1.01

#Initial value
x0=c(.5,1,3)
x2=c(0.15, 0.95, 4.95)
n0=length(x0)
n1=length(x2)
w=rep(1/n0,n0)
w1=rep(1/n1, n1)
D=rbind(x0,w)

#Initial Information matrix
A1<-rep(0,n0)
A2<-rep(0,n0)
A3<-rep(0,n0)
A4<-rep(0,n0)
A5<-rep(0,n0)
A6<-rep(0,n0)
for (i in 1:n0){
      A1[i]=w[i]*((x0[i]/beta)^gamma / (1+(x0[i]/beta)^gamma))^2
      A2[i]=-w[i]*alpha*gamma*(x0[i]/beta)^(2*gamma) /
(beta*(1+(x0[i]/beta)^gamma)^3)
      A3[i]=w[i]*alpha*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
((1+(x0[i]/beta)^gamma)^3)
      A4[i]=w[i]*alpha^2*gamma^2*(x0[i]/beta)^(2*gamma) /
(beta^2*(1+(x0[i]/beta)^gamma)^4)
      A5[i]=w[i]*(-alpha^2)*gamma*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta) /
(beta*(1+(x0[i]/beta)^gamma)^4)
      A6[i]=w[i]*alpha^2*(x0[i]/beta)^(2*gamma)*log(x0[i]/beta)^2 /
(1+(x0[i]/beta)^gamma)^4
}
```

```
A1d<-rep(0,n1)
A2d<-rep(0,n1)
A3d<-rep(0,n1)
A4d<-rep(0,n1)
A5d<-rep(0,n1)
A6d<-rep(0,n1)
for (i in 1:n1){
      A1d[i]=w1[i]*((x2[i]/beta)^gamma / (1+(x2[i]/beta)^gamma))^2
      A2d[i]=w1[i]*(-alpha)*gamma*(x2[i]/beta)^(2*gamma) /
(beta*(1+(x2[i]/beta)^gamma)^3)
      A3d[i]=w1[i]*alpha*(x2[i]/beta)^(2*gamma)*log(x2[i]/beta) /
((1+(x2[i]/beta)^gamma)^3)
      A4d[i]=w1[i]*alpha^2*gamma^2*(x2[i]/beta)^(2*gamma) /
(beta^2*(1+(x2[i]/beta)^gamma)^4)
      A5d[i]=w1[i]*(-alpha^2)*gamma*(x2[i]/beta)^(2*gamma)*log(x2[i]/beta)
/ (beta*(1+(x2[i]/beta)^gamma)^4)
      A6d[i]=w1[i]*alpha^2*(x2[i]/beta)^(2*gamma)*log(x2[i]/beta)^2 /
(1+(x2[i]/beta)^gamma)^4
}

M0=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
IM0=ginv(M0)
#new information matrix
#M1=information matrix evaluated using D-optimal design
M1=matrix(c(sum(A1d), sum(A2d), sum(A3d), sum(A2d), sum(A4d), sum(A5d),
sum(A3d), sum(A5d), sum(A6d)), nrow=3, ncol=3, byrow=F)

M0=.5*M1+.5*M0
IM0 = ginv(M0)
#Find dn,

f<-function(x){
      matrix(c((x/beta)^gamma / (1+(x/beta)^gamma), -
alpha*gamma*(x/beta)^gamma / (beta*(1+(x/beta)^gamma)^2),
alpha*(x/beta)^(gamma)*log(x/beta) /
(1+(x/beta)^gamma)^2),nrow=3,ncol=1,byrow=F)
}

phi.1 <- function(x){
      matrix(c(0, (x/(1-x))^(1/gamma), -beta*(x/(1-x))^(1/gamma)*log(x/(1-
x))/gamma^2), nrow=3, ncol=1, byrow=F)
}

p=1
t=2

while(p>.005){
      x1=seq(0.01,5,.005)
      p1=0.7
      n1=length(x1)
      dn=rep(0,n1)
      for (j in 1:n1)
```

```
      {dn[j]=(t(f(x1[j]))%*%ginv(M0)%*%phi.1(p1))^2}
      for (j in 1:n1)
            {if(max(dn)==dn[j])x1[j]=x1[j] else x1[j]=NA}
      newX=na.omit(x1)
      newdn=max(dn)
      k=t(phi.1(p1))%*%ginv(M0)%*%phi.1(p1)

#Find alpha(n+1)
      #an=(newdn-k)/(k*(newdn-1))
      an=1/t
      #p<-abs(newdn-k)

#Get M(n+1)
      newM=c(1-an)*M0+c(an)*f(newX)%*%t(f(newX))
      M0<-newM
      p=abs((t(f(newX))%*%ginv(M0)%*%phi.1(p1))^2-
(t(phi.1(p1))%*%ginv(M0)%*%phi.1(p1)))
      newW=(1-an)*D[2,]
      W=c(newW,an)
      X=c(D[1,],newX)
      newD=rbind(X,W)
      D=newD
      print(p)
      t=t+1
}

###2-stage ED70 points: 0.075, 0.92, 5
###2-stage ED70 weights:  0.127720, 0.611028, 0.2612521

#Verify c-optimal design

#number of parameter
k=3

#value of parameter
alpha=4.7
beta=0.525
gamma=1.01

#c-optimal design
x=c(0.075,.92, 5)
n=length(x)
w=c(0.127720, 0.611028, 0.2612521)
x2=c(0.15, 0.95, 4.95)
n1=length(x2)
w1=rep(1/3, length(x2))
D=rbind(x,w)

#Information matrix
A1<-rep(0,n)
A2<-rep(0,n)
A3<-rep(0,n)
A4<-rep(0,n)
A5<-rep(0,n)
```

```
A6<-rep(0,n)
for (i in 1:n){
      A1[i]=w[i]*((x[i]/beta)^gamma / (1+(x[i]/beta)^gamma))^2
      A2[i]=w[i]*(-alpha)*gamma*(x[i]/beta)^(2*gamma) /
(beta*(1+(x[i]/beta)^gamma)^3)
      A3[i]=w[i]*alpha*(x[i]/beta)^(2*gamma)*log(x[i]/beta) /
((1+(x[i]/beta)^gamma)^3)
      A4[i]=w[i]*alpha^2*gamma^2*(x[i]/beta)^(2*gamma) /
(beta^2*(1+(x[i]/beta)^gamma)^4)
      A5[i]=w[i]*(-alpha^2)*gamma*(x[i]/beta)^(2*gamma)*log(x[i]/beta) /
(beta*(1+(x[i]/beta)^gamma)^4)
      A6[i]=w[i]*alpha^2*(x[i]/beta)^(2*gamma)*log(x[i]/beta)^2 /
(1+(x[i]/beta)^gamma)^4
}

A1d<-rep(0,n1)
A2d<-rep(0,n1)
A3d<-rep(0,n1)
A4d<-rep(0,n1)
A5d<-rep(0,n1)
A6d<-rep(0,n1)
for (i in 1:n1){
      A1d[i]=w1[i]*((x2[i]/beta)^gamma / (1+(x2[i]/beta)^gamma))^2
      A2d[i]=w1[i]*(-alpha)*gamma*(x2[i]/beta)^(2*gamma) /
(beta*(1+(x2[i]/beta)^gamma)^3)
      A3d[i]=w1[i]*alpha*(x2[i]/beta)^(2*gamma)*log(x2[i]/beta) /
((1+(x2[i]/beta)^gamma)^3)
      A4d[i]=w1[i]*alpha^2*gamma^2*(x2[i]/beta)^(2*gamma) /
(beta^2*(1+(x2[i]/beta)^gamma)^4)
      A5d[i]=w1[i]*(-alpha^2)*gamma*(x2[i]/beta)^(2*gamma)*log(x2[i]/beta)
/ (beta*(1+(x2[i]/beta)^gamma)^4)
      A6d[i]=w1[i]*alpha^2*(x2[i]/beta)^(2*gamma)*log(x2[i]/beta)^2 /
(1+(x2[i]/beta)^gamma)^4
}

M=matrix(c(sum(A1), sum(A2), sum(A3), sum(A2), sum(A4), sum(A5), sum(A3),
sum(A5), sum(A6)), nrow=3, ncol=3, byrow=F)
IM=solve(M)
#new information matrix
#M0=information matrix evaluated using D-optimal design
M1=matrix(c(sum(A1d), sum(A2d), sum(A3d), sum(A2d), sum(A4d), sum(A5d),
sum(A3d), sum(A5d), sum(A6d)), nrow=3, ncol=3, byrow=F)

M=.5*M+.5*M1

#Find dn

f<-function(x){
      matrix(c((x/beta)^gamma / (1+(x/beta)^gamma), -
alpha*gamma*(x/beta)^gamma / (beta*(1+(x/beta)^gamma)^2),
alpha*(x/beta)^(gamma)*log(x/beta) /
(1+(x/beta)^gamma)^2),nrow=3,ncol=1,byrow=F)
}
```

```
phi.1 <- function(x){
      matrix(c(0, (x/(1-x))^(1/gamma), -beta*(x/(1-x))^(1/gamma)*log(x/(1-
x))/gamma^2), nrow=3, ncol=1, byrow=F)
}

p1=0.7
p=1
x1=seq(0.05,5,.01)
n1=length(x1)
dn=rep(0,n1)

for (j in 1:n1){
      dn[j]=(t(f(x1[j]))%*%ginv(M)%*%phi.1(p1))^2
}

plot(x1,dn, ylim = c(0, 25), cex = 0.1, xlab = quote(x [i]), ylab =
"Standardized Variance", main = "Two-Stage Optimal ED70")

# Efficiency of 2-stage design
e.21 <- det(M)^(1/3)
e.22 <- (t(phi.1(0.7)) %*% solve(M) %*% phi.1(0.7))

# e.21 <- 0.4465388
# e.22 <- 19.75139

# Comparing efficiences of designs
REdc <- 0.422596 / 0.4731286
REcd <- 18.21657 / 23.34466
RE2d <- 0.4456388 / 0.4731286
RE2c <- 18.21657 / 19.75139
REUd <- 0.007469769 / 0.4731286
REUc <- 18.21657 / 45.34811


# REdc = 0.8931948
# REcd = 0.7803313
# RE2d = 0.9418978
# RE2c = 0.9222931
# REUd = 0.01578803
# REUc = 0.4017052
```

APPENDIX D. R CODE FOR C-OPTIMAL WEIGHTS

```
#Parameter Values
alpha=4.7
beta=0.525
gamma=1.01

#Dosage level of interest
D = 0.7

#Derivatives for LD30 (wrt to phi function)
palpha = 0
pbeta = (D/(1-D))^(1/gamma)
pgamma = -beta*(D/(1-D))^(1/gamma)*log(D/(1-D))/gamma^2

## Generalized Inverse of a Matrix
ginv<-function(X, tol = sqrt(.Machine$double.eps))
{

  dnx <- dimnames(X)
  if(is.null(dnx)) dnx <- vector("list", 2)
  s <- svd(X)
  nz <- s$d > tol * s$d[1]
  structure(
    if(any(nz)) s$v[, nz] %*% (t(s$u[, nz])/s$d[nz]) else X,
    dimnames = dnx[2:1])
}

#Find Optimal Weights using Newton Raphson Algorithm for LD 30
n=3
D<-c(0.075,.92, 5)

#Initial weights
w_<-c(.1,.1)
mu<-rep(0,n)
A1<-rep(0,n)
A2<-rep(0,n)
A3<-rep(0,n)
A4<-rep(0,n)
A5<-rep(0,n)
A6<-rep(0,n)

for (i in 1:n){
      A1[i]=((D[i]/beta)^gamma / (1+(D[i]/beta)^gamma))^2
      A2[i]=(-alpha)*gamma*(D[i]/beta)^(2*gamma) /
(beta*(1+(D[i]/beta)^gamma)^3)
      A3[i]=alpha*(D[i]/beta)^(2*gamma)*log(D[i]/beta) /
((1+(D[i]/beta)^gamma)^3)
      A4[i]=alpha^2*gamma^2*(D[i]/beta)^(2*gamma) /
(beta^2*(1+(D[i]/beta)^gamma)^4)
      A5[i]=(-alpha^2)*gamma*(D[i]/beta)^(2*gamma)*log(D[i]/beta) /
(beta*(1+(D[i]/beta)^gamma)^4)
```

```
      A6[i]=alpha^2*(D[i]/beta)^(2*gamma)*log(D[i]/beta)^2 /
(1+(D[i]/beta)^gamma)^4
}

A1L=A1[n]
A2L=A2[n]
A3L=A3[n]
A4L=A4[n]
A5L=A5[n]
A6L=A6[n]
Di<-c(palpha,pbeta,pgamma)

# transpose of D'
Di<-as.matrix(Di)
V<-Di%*%t(Di)
w=matrix(data=0,nrow=n-1,ncol=1,byrow=F)
for(i in 1:(n-1)){
      w[i,1]=w_[i]
}
Dw<-rep(0,n)
p<-1
while(p>.000001){
      k=length(w)

      A1=A1[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A1[i]=NA else A1[i]=A1[i]
      }
      A1=na.omit(A1)

      A2=A2[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A2[i]=NA else A2[i]=A2[i]
      }
      A2=na.omit(A2)

      A3=A3[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A3[i]=NA else A3[i]=A3[i]
      }
      A3=na.omit(A3)

      A4=A4[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A4[i]=NA else A4[i]=A4[i]
      }
      A4=na.omit(A4)

      A5=A5[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A5[i]=NA else A5[i]=A5[i]
      }
      A5=na.omit(A5)
```

```
      A6=A6[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A6[i]=NA else A6[i]=A6[i]
      }
      A6=na.omit(A6)

      for(i in 1:k){
            if(w[i,1]==0) w[i,1]=NA
      }
      w=as.vector(w)
      w=na.omit(w)
      k=length(w)
      w=as.matrix(w)

      sw1=matrix(0,nrow=k,ncol=1)
      sw2=matrix(0,nrow=k,ncol=1)
      sw3=matrix(0,nrow=k,ncol=1)
      sw4=matrix(0,nrow=k,ncol=1)
      sw5=matrix(0,nrow=k,ncol=1)
      sw6=matrix(0,nrow=k,ncol=1)

      for ( i in 1:k){
            sw1[i,1]=w[i,1]*A1[i]
            sw2[i,1]=w[i,1]*A2[i]
            sw3[i,1]=w[i,1]*A3[i]
            sw4[i,1]=w[i,1]*A4[i]
            sw5[i,1]=w[i,1]*A5[i]
            sw6[i,1]=w[i,1]*A6[i]
      }

      p1=sum(sw1)+((1-(sum(w)))*A1L)
      p2=sum(sw2)+((1-(sum(w)))*A2L)
      p3=sum(sw3)+((1-(sum(w)))*A3L)
      p4=sum(sw4)+((1-(sum(w)))*A4L)
      p5=sum(sw5)+((1-(sum(w)))*A5L)
      p6=sum(sw6)+((1-(sum(w)))*A6L)

      I=matrix(c(p1,p2,p3,p2,p4,p5,p3,p5,p6),nrow=3,
      ncol=3,byrow=F)

      IL=matrix(c(A1L,A2L,A3L,A2L,A4L,A5L,A3L,A5L,A6L),
      nrow=3,ncol=3,byrow=F)
      inverseI=ginv(I)

      Min=sum(diag(inverseI%*%V))
      dI_<-array(c(0,0,0,0,0,0,0,0,0), c(3,3,k))

      f1<-matrix(data=0,nrow=k,ncol=1,byrow=F)

      for (i in 1:k){

      dI_[,,i]=matrix(c(A1[i],A2[i],A3[i],A2[i],A4[i],A5[i],A3[i],A5[i],A6
[i]),
            nrow=3,ncol=3,byrow=F)
```

```
            f1[i]=sum(diag(V%*%(-inverseI%*%(dI_[,,i]-IL)%*%inverseI)))
      }
#(w1&w2&w3)

      f2<-matrix(data=0,nrow=k,ncol=k,byrow=F)
      for (i in 1:k){
            for (j in 1:k){
                  f2[i,j]=sum(diag(V%*%((inverseI%*%(dI_[,,j]-
IL)%*%inverseI)%*%(dI_[,,i]-IL)%*%inverseI+
                  inverseI%*%(dI_[,,i]-IL)%*%(inverseI%*%(dI_[,,j]-
IL)%*%inverseI))))
            }
      }

      new.w=w-.01*(ginv(f2)%*%f1)
      for (i in 1:k){
            if(new.w[i,1]<0) new.w[i,1]=0
      }

      p<-max(abs(new.w-w))
      w<-new.w
      for (i in 1:k){
            if(w[i,1]==0){if(Dw[i]==1) Dw[i+1]=1
      }
      if(w[i,1]==0) Dw[i]=1}
}
w
1-sum(w)
Min

###w1 = 0.6969446
###w2 = 0.3030554
```

APPENDIX E. R CODE FOR TWO-STAGE WEIGHTS

```
#Parameter Values
alpha=4.7
beta=0.525
gamma=1.01

#Dosage level of interest
D = 0.7

#Derivatives for LD30 (wrt to phi function)
palpha = 0
pbeta = (D/(1-D))^(1/gamma)
pgamma = -beta*(D/(1-D))^(1/gamma)*log(D/(1-D))/gamma^2

## Generalized Inverse of a Matrix
ginv<-function(X, tol = sqrt(.Machine$double.eps))
{

  dnx <- dimnames(X)
  if(is.null(dnx)) dnx <- vector("list", 2)
  s <- svd(X)
  nz <- s$d > tol * s$d[1]
  structure(
    if(any(nz)) s$v[, nz] %*% (t(s$u[, nz])/s$d[nz]) else X,
    dimnames = dnx[2:1])
}

#Find Optimal Weights using Newton Raphson Algorithm for LD 70
n=3
D<-c(0.075,.92, 5)
D1 <- c(0.15, 0.95, 4.95)
w1 <- rep(1/length(D1), length(D1))

#Initial weights
w_<-c(.1,.1)
mu<-rep(0,n)
A1<-rep(0,n)
A2<-rep(0,n)
A3<-rep(0,n)
A4<-rep(0,n)
A5<-rep(0,n)
A6<-rep(0,n)

for (i in 1:n){
      A1[i]=((D[i]/beta)^gamma / (1+(D[i]/beta)^gamma))^2
      A2[i]=(-alpha)*gamma*(D[i]/beta)^(2*gamma) /
(beta*(1+(D[i]/beta)^gamma)^3)
      A3[i]=alpha*(D[i]/beta)^(2*gamma)*log(D[i]/beta) /
((1+(D[i]/beta)^gamma)^3)
      A4[i]=alpha^2*gamma^2*(D[i]/beta)^(2*gamma) /
(beta^2*(1+(D[i]/beta)^gamma)^4)
```

```
      A5[i]=(-alpha^2)*gamma*(D[i]/beta)^(2*gamma)*log(D[i]/beta) /
(beta*(1+(D[i]/beta)^gamma)^4)
      A6[i]=alpha^2*(D[i]/beta)^(2*gamma)*log(D[i]/beta)^2 /
(1+(D[i]/beta)^gamma)^4
}

A1.d<-rep(0,n)
A2.d<-rep(0,n)
A3.d<-rep(0,n)
A4.d<-rep(0,n)
A5.d<-rep(0,n)
A6.d<-rep(0,n)

for (i in 1:n){
      A1.d[i]=w1[i]*((D1[i]/beta)^gamma / (1+(D1[i]/beta)^gamma))^2
      A2.d[i]=w1[i]*(-alpha)*gamma*(D1[i]/beta)^(2*gamma) /
(beta*(1+(D1[i]/beta)^gamma)^3)
      A3.d[i]=w1[i]*alpha*(D1[i]/beta)^(2*gamma)*log(D1[i]/beta) /
((1+(D1[i]/beta)^gamma)^3)
      A4.d[i]=w1[i]*alpha^2*gamma^2*(D1[i]/beta)^(2*gamma) /
(beta^2*(1+(D1[i]/beta)^gamma)^4)
      A5.d[i]=w1[i]*(-
alpha^2)*gamma*(D1[i]/beta)^(2*gamma)*log(D1[i]/beta) /
(beta*(1+(D1[i]/beta)^gamma)^4)
      A6.d[i]=w1[i]*alpha^2*(D1[i]/beta)^(2*gamma)*log(D1[i]/beta)^2 /
(1+(D1[i]/beta)^gamma)^4
}



A1L=A1[n]
A2L=A2[n]
A3L=A3[n]
A4L=A4[n]
A5L=A5[n]
A6L=A6[n]

A1LD=A1.d[n]
A2LD=A2.d[n]
A3LD=A3.d[n]
A4LD=A4.d[n]
A5LD=A5.d[n]
A6LD=A6.d[n]

Di<-c(palpha,pbeta,pgamma)

# transpose of D'
Di<-as.matrix(Di)
V<-Di%*%t(Di)
w=matrix(data=0,nrow=n-1,ncol=1,byrow=F)
for(i in 1:(n-1)){
      w[i,1]=w_[i]
}
Dw<-rep(0,n)
```

```
p<-1
while(p>.000001){
      k=length(w)

      A1=A1[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A1[i]=NA else A1[i]=A1[i]
      }
      A1=na.omit(A1)

      A2=A2[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A2[i]=NA else A2[i]=A2[i]
      }
      A2=na.omit(A2)

      A3=A3[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A3[i]=NA else A3[i]=A3[i]
      }
      A3=na.omit(A3)

      A4=A4[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A4[i]=NA else A4[i]=A4[i]
      }
      A4=na.omit(A4)

      A5=A5[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A5[i]=NA else A5[i]=A5[i]
      }
      A5=na.omit(A5)

      A6=A6[1:k]
      for(i in 1:k){
            if(w[i,1]==0) A6[i]=NA else A6[i]=A6[i]
      }
      A6=na.omit(A6)

      for(i in 1:k){
            if(w[i,1]==0) w[i,1]=NA
      }
      w=as.vector(w)
      w=na.omit(w)
      k=length(w)
      w=as.matrix(w)

      sw1=matrix(0,nrow=k,ncol=1)
      sw2=matrix(0,nrow=k,ncol=1)
      sw3=matrix(0,nrow=k,ncol=1)
      sw4=matrix(0,nrow=k,ncol=1)
      sw5=matrix(0,nrow=k,ncol=1)
      sw6=matrix(0,nrow=k,ncol=1)
```

```
      for ( i in 1:k){
            sw1[i,1]=w[i,1]*A1[i]
            sw2[i,1]=w[i,1]*A2[i]
            sw3[i,1]=w[i,1]*A3[i]
            sw4[i,1]=w[i,1]*A4[i]
            sw5[i,1]=w[i,1]*A5[i]
            sw6[i,1]=w[i,1]*A6[i]
      }

      p1=sum(sw1)+((1-(sum(w)))*A1L)
      p2=sum(sw2)+((1-(sum(w)))*A2L)
      p3=sum(sw3)+((1-(sum(w)))*A3L)
      p4=sum(sw4)+((1-(sum(w)))*A4L)
      p5=sum(sw5)+((1-(sum(w)))*A5L)
      p6=sum(sw6)+((1-(sum(w)))*A6L)

      I=matrix(c(p1,p2,p3,p2,p4,p5,p3,p5,p6),nrow=3,ncol=3,byrow=F)

      I.d=matrix(c(sum(A1.d), sum(A2.d), sum(A3.d), sum(A2.d), sum(A4.d),
sum(A5.d), sum(A3.d), sum(A5.d), sum(A6.d)), nrow=3, ncol=3, byrow=F)

      I = 0.5*I + 0.5*I.d

      IL=matrix(c(A1L,A2L,A3L,A2L,A4L,A5L,A3L,A5L,A6L),nrow=3,ncol=3,byrow
=F)
      inverseI=ginv(I)

      Min=sum(diag(inverseI%*%V))
      dI_<-array(c(0,0,0,0,0,0,0,0,0), c(3,3,k))

      f1<-matrix(data=0,nrow=k,ncol=1,byrow=F)

      for (i in 1:k){

      dI_[,,i]=matrix(c(A1[i],A2[i],A3[i],A2[i],A4[i],A5[i],A3[i],A5[i],A6
[i]),
            nrow=3,ncol=3,byrow=F)
            f1[i]=sum(diag(-0.5*V%*%(-inverseI%*%(dI_[,,i]-
IL)%*%inverseI)))
      }
#(w1&w2&w3)

      f2<-matrix(data=0,nrow=k,ncol=k,byrow=F)
      for (i in 1:k){
            for (j in 1:k){
                  f2[i,j]=-0.5*sum(diag(V%*%((inverseI%*%(dI_[,,j]-
IL)%*%inverseI)%*%(dI_[,,i]-IL)%*%inverseI+
                  inverseI%*%(dI_[,,i]-IL)%*%(inverseI%*%(dI_[,,j]-
IL)%*%inverseI))))
            }
      }

      new.w=w-.01*(ginv(f2)%*%f1)
```

```
      for (i in 1:k){
            if(new.w[i,1]<0) new.w[i,1]=0
      }

      p<-max(abs(new.w-w))
      w<-new.w
      for (i in 1:k){
            if(w[i,1]==0){if(Dw[i]==1) Dw[i+1]=1
      }
      if(w[i,1]==0) Dw[i]=1}
}
w
1-sum(w)
Min

###Weights for 2-stage ED70:  0.127720, 0.611028, 0.2612521
```