

ROBUST D-OPTIMAL DESIGN FOR MULTIPLE NOMINAL PARAMETER VALUES
UNDER THE 5PL-1P MODEL

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

By
Cuiping Liang

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

Major Department:
Statistics

April 2018

Fargo, North Dakota

North Dakota State University
Graduate School

Title

ROBUST D-OPTIMAL DESIGN FOR MULTIPLE NOMINAL
PARAMETER VALUES UNDER THE 5PL-1P MODEL

By

Cuiping Liang

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. Gang Shen

Dr. Yongki Choi

Approved:

5/1/2018

Date

Dr. Rhonda Magel

Department Chair

ABSTRACT

A robust D-optimal design that works well for multiple nominal parameter values is presented in this paper. In general, D-optimal design works very well for estimating the model parameters, but it is very sensitive to multiple nominal model parameter values when the response is modeled by nonlinear models. The 5PL-1P model is considered in this study to describe a dose-response function. The sensitivity of the D-optimal design to the model parameter values under the 5PL-1P model is studied. The robust D-optimal design that can reduce the impact of the multiple nominal model parameter values is proposed using the Bayesian technique. Lastly, we compare performances of the proposed design to other well-known designs for estimating the model parameters under the 5PL-1P model.

ACKNOWLEDGEMENTS

This work would not have been possible without the financial support of the workforce center in Alexandria, MN. I am especially thankful to Kris, who has been working actively to support my career goals and given me courage to finish master's degree.

I am grateful to all of the teachers and students from whom I have had the pleasure to learn. They have helped me jump into Statistics and develop solid an academic background.

I would like to thank Gang Shen and Yongki Choi who were willing to participate as my defense committee. I would like to express my deepest gratitude to my advisor Seung Won Hyun who has provided me professional guidance and taught me a great deal about both scientific research and attitude towards work.

I also would like to thank Rollie Finke from Alexandria Adult Basic Education and Gail Morris from Alexandria Literacy Project for their valuable comments and suggestions.

Finally, I would like to thank my husband and kids for encouraging me in all my pursuits and inspiring me to follow my dreams.

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
LIST OF ABBREVIATIONS.....	viii
1. INTRODUCTION.....	1
2. BACKGROUND.....	4
2.1. Optimal Design.....	4
2.2. Fisher Information Matrix.....	4
2.3. The General Equivalence Theorem.....	5
2.4. V-Algorithm.....	6
3. MODEL.....	7
3.1. The 5PL-1P Model.....	7
3.2. Multiple Nominal Parameter Values.....	7
3.3. Checking the Model Fit.....	9
4. OPTIMAL DESIGN.....	12
4.1. D-optimal Design.....	12
4.2. Robust D-optimal Design.....	18
5. EFFICIENCY.....	24
6. CONCLUSION.....	28
REFERENCES.....	29
APPENDIX A. R CODES FOR NINE SETS OF PARAMETER VALUES.....	31
APPENDIX B. R CODES FOR ROBUST D-OPTIMAL DESIGN.....	33
APPENDIX C. R CODES FOR EFFICIENCIES.....	39

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Nine sets of parameter values for compound BRAN	8
2. Nine sets of parameter values for compound CLAN	8
3. D-optimal designs of nine sets of parameter values for BRAN	14
4. D-optimal designs of nine sets of parameter values for CLAN	15
5. Robust D-optimal design of randomly selected parameter values for BRAN	21
6. Robust D-optimal design of randomly selected parameter values for CLAN	21
7. Original designs for the Two Compounds	24
8. Efficiencies for nine sets of parameter values among Robust D-optimal design, Original design and D-optimal designs for BRAN	25
9. Efficiencies for nine sets of parameter values among Robust D-optimal design, Original design and D-optimal designs for CLAN	26

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Dose-response curves of nine sets of parameter values for BRAN	10
2. Dose-response curves of nine sets of parameter values for CLAN.....	11
3. Sensitivity function plot Verifying the D-optimal designs of nine sets of parameter values for BRAN.....	16
4. Sensitivity function plot verifying the D-optimal designs of nine sets of parameter values for CLAN	17
5. Plot of the sensitivity function of the robust D-optimal design for BRAN using Bayesian technique when uniform weight is assumed.....	22
6. Plot of the sensitivity function of the robust D-optimal design for CLAN using Bayesian technique when uniform weight is assumed.....	23

LIST OF ABBREVIATIONS

5PL-1P.....	Five parameter model minus one parameter
5PL.....	Five-parameter logistic
4PL.....	Four-parameter logistic
3PL.....	Three-parameter logistic
BRAN.....	Bromoacetonitrile
CLAN.....	Chloroacetonitrile

1. INTRODUCTION

Experimental design plays an important role in scientific and medical research because it can help experimenters minimize costs while still obtaining valid results (Montgomery, 2009). Experimental designs in biopharmaceutical and toxicology research often involve investigating dose-response relationships. Logistic models are often used to study dose-response relationships. Each model has slightly different number of parameters: the three-parametric model (3PL), a knowledge-based parametric model that fixes either the upper asymptote to 100% or the lower asymptote to 0% (Birnbaum, 1968); the four-parametric model (4PL), a parametric model that assumes symmetry (Rodbard, 1975); and the five-parametric model (5PL), a parametric model that allows asymmetry (Prentice, 1976; Rodbard et al., 1978).

The five-parameter logistic minus one parameter (5PL-1P) model (Dawson et al., 2012) is the focus of this study. The 5PL-1P model is a hybrid model of the five-parameter model (5PL) and the four-parameter model (4PL). The 4PL model includes minimum dose, median effective dose concentration, slope, and maximum dose values for each curve. The 5PL model adds a fifth parameter to the 4PL model to allow the curve to be asymmetric. However, when the model has this additional parameter, it becomes much more difficult to fit the model to the data. The 5PL-1P model is similar to the 4PL model, but it replaces the minimum effect parameter with the asymmetric parameter. This helps the model more easily fit to the asymmetric dose-response data.

Dawson et al. (2012) used the 5PL-1P model to fit asymmetric sigmoid curves into concentration-effect data for 72 binary chemical combinations. In the study, two chemical compounds bromoacetonitrile, abbreviated as BRAN, and chloroacetonitrile, abbreviated as CLAN were included. These chemicals are by-products from disinfecting water and are known

to influence DNA (Bromoacetonitrile, 2016). The tests were conducted to determine the effects of the BRAN and CLAN compounds using bioluminescent bacteria with the Microtox® machine (Dawson et al. 2012). Dawson et al.'s (2012) study indicated that the dose-response relationship between percentage of bioluminescence inhibition in the marine bacterium and dose level of each chemical compound (BRAN and CLAN) follows the 5PL-1P model.

Dawson et al. (2012) used these two chemicals together with other chemicals to analyze dose-response relationships. The study assessed the performance of both the 4PL model and 5PL-1P model including an asymmetry parameter into curve-fitting. With higher mean coefficients determination (r^2) and lower sum-of-squares of the residuals, the 5PL-1P model provided better fitting than the 4PL model. The study indicated that the 5PL-1P model with this additional asymmetric parameter can provide more accurate results than the 4PL model.

An experimental design of dose–response studies is implemented by determining the number and location of dose levels, and the allocation of sample size at each dose level. The 5PL-1P model is a relatively new model with very little research in the selection of experimental design. MacDonald's (2016) study on the D-optimal designs under the 5PL-1P model showed that the D-optimal designs performed better than the original designs by providing higher efficiencies and lower Mean Squared Errors (MSE) for the model parameters.

This paper studies a robust D-optimal design for multiple nominal parameter values under the 5PL-1P model. According to MacDonald (2016), D-optimal designs under the 5PL-1P model perform well for a given parameter value. However, when a study has multiple nominal parameter values, there is no guarantee that the D-optimal design obtained for one set of parameter value works well for other parameter values. This brings us to the problem of how to construct optimal designs that work robustly for multiple nominal parameter values. One

approach for solving this problem is to use the Bayesian technique which accounts for the prior uncertainty of the parameters. The idea behind the Bayesian optimal design is to utilize a prior distribution on the unknown parameters. There have been various studies using the Bayesian optimal design technique. Chaloner and Larntz (1989) investigated Bayesian optimal design for the one-variable logistic regression model. Both Chaloner and Verdinelli (1995) and DasGupta (1995) presented a detailed summary review of the Bayesian approach to experimental design. Later, several Bayesian approaches for the dose-response models were presented in studies, such as Mukhopadhyay and Haines (1995), Dette (1996), and Lauer et al. (1997). Dette et al. (2007) performed the study of constructing Bayesian optimal designs for dose findings under several regression models. Also, Abebe et al. (2014) calculated the Bayesian D-optimal design for the logistic models with two parameters. In this study, the Bayesian technique was applied to construct robust D-optimal design under the 5PL-1P model.

A robust D-optimal design was found for multiple nominal parameter values under the 5PL-1P model using the Bayesian technique for each compound (BRAN and CLAN). Based on previous research on the 5PL-1P model, various parameter values were generated to be used to construct the robust D-optimal design. Then, the performance of the robust D-optimal designs under the 5PL-1P model were compared to that of other well-known designs.

Chapter 2 covers the basic concepts of optimal designs. Chapter 3 describes the models of interest. Chapter 4 discusses the D-optimal designs and the robust D-optimal design for various parameter values under the 5PL-1P model. Chapter 5 compares the performances of the robust D-optimal designs to other well-known traditional designs by comparing their efficiencies. All conclusions will be addressed in Chapter 6.

2. BACKGROUND

This section discusses the background of optimal designs, specifically D-optimal designs, which will be used later in this study. In the study, we are interested in estimating the parameters for the SPL-1P model that describes the relationship between dose and response. Optimal designs allow a researcher to select the dose levels and distribute samples over those dose levels in the most efficient way. This is important because the sample size can be reduced through the optimal design process and ultimately, the efficient optimal experiment designs lead to cost reduction.

2.1. Optimal Design

Optimal design theory uses prior information of the model and parameter estimates to optimize a convex(concave) function of the Fisher information matrix to meet specified objectives (Aarons and Ogunbenro, 2010). There are different optimal design criteria depending on different study objectives. The optimal design is the design that minimizes or maximizes the optimality criterion (Atkinson et al., 2007): often the criterion is determined by the convex or concave function of the Fisher information matrix: $\psi\{M(\xi, \Theta)\}$, where $M(\xi, \Theta)$ denotes the Fisher information matrix for the model parameters Θ .

2.2. Fisher Information Matrix

The optimal design is obtained through minimizing or maximizing a function of the Fisher information matrix, which can minimize variance–covariance matrix of the parameter estimation in some sense (Atkinson et al., 2007). The Fisher information matrix is the fundamental component needed for finding the optimal design. We can find the Fisher information matrix $M(\xi, \Theta)$ using the gradient of the statistical model mean function $f(x, \Theta)$:

$$M(\xi, \Theta) = \sum w_i g(x_i) g(x_i)^T$$

where $g(x_i, \Theta)$ is the gradient of the mean function $f(x_i, \Theta)$:

$g(x_i, \Theta) = \left(\frac{f(x_i, \Theta)}{d\theta_1} \frac{f(x_i, \Theta)}{d\theta_2} \frac{f(x_i, \Theta)}{d\theta_3} \frac{f(x_i, \Theta)}{d\theta_4} \dots \frac{f(x_i, \Theta)}{d\theta_s} \right)^T$, s represents the number of parameters in the model, and $\theta_1, \theta_2, \dots$, and θ_s are parameters in the mean function. $\xi = (x_i, w_i)_{i=1}^k$ is an approximate design: x_i refers to the i_{th} dose level, and $w_i = n_i/N$ is the weight for each of those doses. In real studies, the closest integer to $w_i N$ will be the replication size n_i for the i_{th} dose level.

2.3. The General Equivalence Theorem

The general equivalence theorem is used to check whether the D-optimal design is optimum or not (Kiefer, 1958). The theorem includes the Fisher information matrix $M(\xi, \Theta)$, the directional derivative of convex function $\psi \{ \}$, and is expressed as:

$$\phi(x_i, \xi) = \lim_{\alpha \rightarrow 0^+} [\psi\{(1 - \alpha)M(\xi, \Theta) + \alpha M(\bar{\xi}, \Theta)\} - \psi\{M(\xi, \Theta)\}]$$

Where ξ denotes the design matrix, Θ is the parameter vector, and $\bar{\xi}$ is the measure that puts on point mass at the point x_i . When iteration begins, design matrix ξ' , $\xi' = (1 - \alpha)\xi + \alpha\bar{\xi}$ is placed into the Fisher information matrix. Then the Fisher information matrix is updated by $M(\xi', \Theta) = (1 - \alpha)M(\xi, \Theta) + \alpha M(\bar{\xi}, \Theta)$ during each iteration. The design problem consists in finding a measure ξ^* that maximizes the criterion function. The theorem indicates that a D-optimal design ξ^* equals the following conditions (Atkinson et al, 2007):

1. The design ξ^* minimizes $\psi\{M(\xi, \Theta)\}$.
2. The design ξ^* maximizes the minimum concentration x_i over the design space $\phi(x_i, \xi)$.
3. The minimum over the design space $\phi(x_i, \xi)$ equals zero at the points support the given design ξ^* .

2.4. V-Algorithm

The V-algorithm was introduced by Fedorov, Klimko, and Studden (1972) to search D-optimal designs numerically. It is an iterative method which starts with initial design ξ_0 and $s + 1$ dose levels, where s represents the number of parameters in the model. Following the General Equivalence theorem, the algorithm maximizes the standardized variance d_n at the n_{th} iteration,

$$d_n(x, \Theta) = g(x)^T M^{-1}(\xi_n, \Theta) g(x)$$

where $x \in \chi$ and $g(x)$ is the gradient of the mean response function.

The V-algorithm identifies the design point x from design space χ by maximizing the standardized variance for $\bar{d}_n(x, \Theta) = \max_x d_n(x, \Theta)$. The design point is then used to update the Fisher information matrix M , $M(\xi_{n+1}, \Theta) = (1 - \alpha_{n+1})M_n(\xi_n, \Theta) + \alpha_{n+1}g(x)g(x)^T$, where $\alpha_{n+1} = \frac{1}{n+1}$. The iteration continues until the condition $d_n(x, \Theta) = g(x)^T M^{-1}(\xi_n, \Theta) g(x) - s \leq \varepsilon$ is satisfied, where $\varepsilon = 10^{-6}$ and s represents the number of parameters in the model.

3. MODEL

In order to obtain D-optimal and robust D-optimal designs for nonlinear models, the parameters of the model need to be estimated first or specified in advance. This section will cover the five-parameter minus one-parameter model and multiple nominal parameter values used to find the optimal designs.

3.1. The 5PL-1P Model

The five-parameter logistic minus one-parameter(5PL-1P) function is different from the four-parameter logistic(4PL) function by removing the minimum parameter from the 4PL and adding an asymmetric parameter (Dawson et al., 2012). The general model with the j_{th} continuous response at the i_{th} dose x_i is represented by

$$Y_{ij} = f(x_i, \Theta) + \varepsilon_{ij}; \quad \varepsilon_{ij} \sim N(0, \sigma^2), j = 1, 2, \dots, n_i, i = 1, 2, \dots, k, n_1 + \dots + n_k = n$$

where $f(x_i, \Theta)$ is the mean response at dose x_i , Θ is the vector of model parameters, and σ^2 is an unknown positive constant. The mean response follows the 5PL-1P model

$$f(x_i, \Theta) = \theta_1 / [1 + \left(\frac{\theta_2}{x_i}\right)^{\theta_3}]^{\theta_4}$$

where θ_1 = maximum response, θ_3 = slope, θ_4 = asymmetric factor, and $x_i = i_{th}$ dose level. The parameter θ_2 represents a function of EC_{50} , where EC_{50} is the dose level corresponding to 50% of the difference between the maximum and minimum effect:

$$\theta_2 = f(EC_{50}) = EC_{50} * 10^{\left[\frac{1}{\theta_3} \log\left(2^{1/\theta_4} - 1\right)\right]}$$

when the asymmetric factor $\theta_4 = 1$, $\theta_2 = EC_{50}$.

3.2. Multiple Nominal Parameter Values

Two chemical compounds, BRAN and CLAN, are used in this paper. To illustrate the multiple nominal parameter values used to determine the robust D-optimal design, nine sets of

parameter values were considered. Six sets of parameter values ($\Theta_1, \Theta_2, \dots, \Theta_6$) were randomly selected within the parameter ranges for BRAN and CLAN. This was done in R software. The remaining three sets of parameter values ($\Theta_7, \Theta_8, \Theta_9$) for each compound were from MacDonald (2016), in which the dose response was recorded at three different exposure times: 15, 30, and 45 minutes. There are nine total sets of parameter values for each compound in Tables 1 and 2.

Table 1

Nine sets of parameter values for compound BRAN

Parameter value	Exposure Time(minutes)	Maximum concentration θ_1	θ_2	Hill slope θ_3	Asymmetric Factor θ_4
Θ_1		100.0000	1.495398	2.965406	0.3353759
Θ_2		100.0000	1.206563	1.631951	2.5835328
Θ_3		100.0000	3.277633	3.493400	0.5118468
Θ_4		100.0000	1.894980	3.923933	0.3128005
Θ_5		100.0000	2.304118	1.222718	0.6942559
Θ_6		100.0000	1.535736	2.840775	1.0558678
Θ_7	BRAN (15)	128.1528	2.3244	0.9791	1.5470
Θ_8	BRAN (30)	103.2062	1.6336	1.5402	0.8235
Θ_9	BRAN (45)	100.97883	1.08130	1.70242	0.71926

Table 2

Nine sets of parameter values for compound CLAN

Parameter value	Exposure Time(minutes)	Maximum concentration θ_1	θ_2	Hill slope θ_3	Asymmetric Factor θ_4
Θ_1		100.0000	85.02277	3.291940	0.6210168
Θ_2		100.0000	83.62907	1.128005	0.8271627
Θ_3		100.0000	166.62151	2.247638	0.6791109
Θ_4		100.0000	105.95462	2.776109	0.7623249
Θ_5		100.0000	194.76003	1.245860	0.5923309
Θ_6		100.0000	96.83994	1.112767	0.7172172
Θ_7	CLAN (15)	105.7901	204.3502	1.5494	0.8279
Θ_8	CLAN (30)	100.78867	119.55175	1.89378	0.56313
Θ_9	CLAN (45)	100.73194	75.21709	1.87647	0.54536

3.3. Checking the Model Fit

After the parameter values were generated, it was essential to check how those values fit the data. Figures 1 and 2 are plots of each compound (BRAN and CLAN): within the corresponding model parameter values $(\theta_1, \theta_2, \dots, \theta_9)$, the dose concentration is represented by x , and the dose response is represented by y . It is clear from the plots that all nine models are reasonable dose-response curve under the 5PL-1P model. One thing to note is the curvature change as the exposure time increases. This curvature is normal because as exposure time increases, the fitted line takes on a more defined curve (MacDonald, 2016). It is assumed that these parameter values are reasonable parameter values for the 5PL-1P model and will be used to find the D-optimal designs and the robust D-optimal designs, which will be discussed later in this paper.

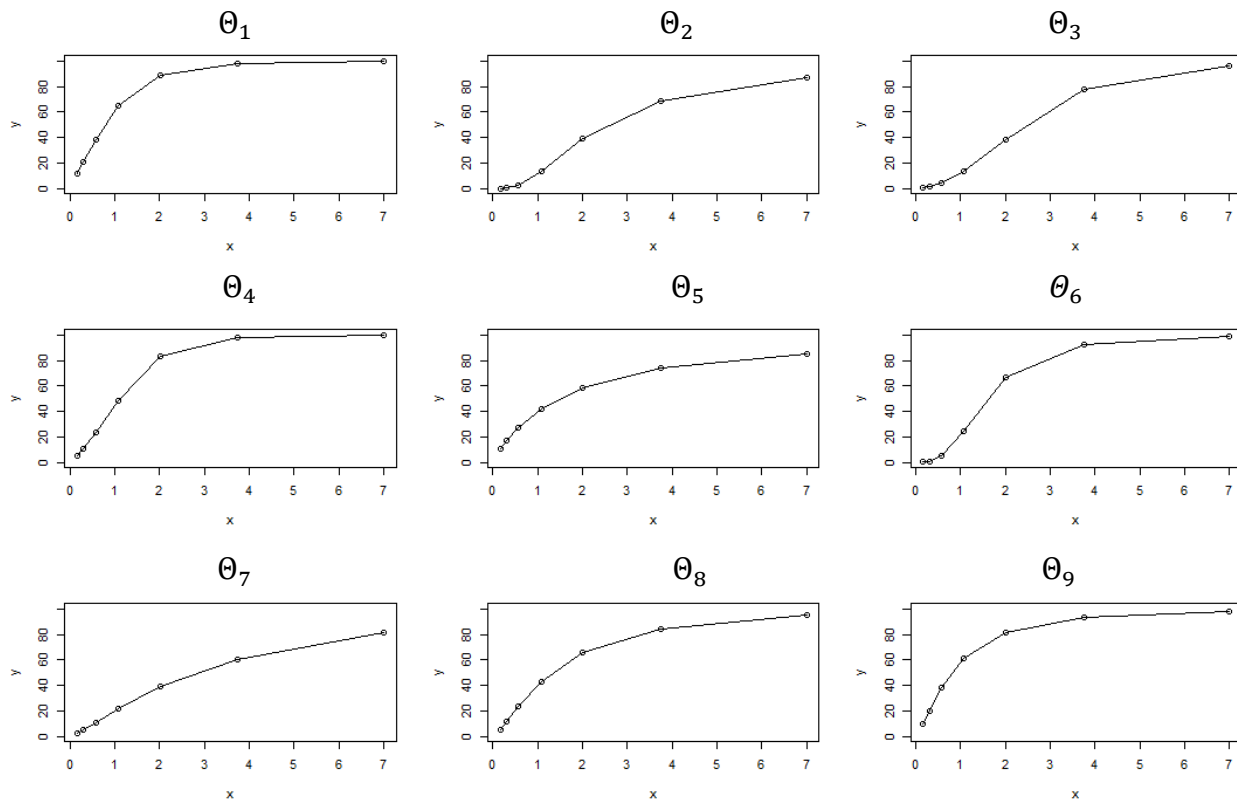


Figure 1. Dose-response curves of nine sets of parameter values for BRAN

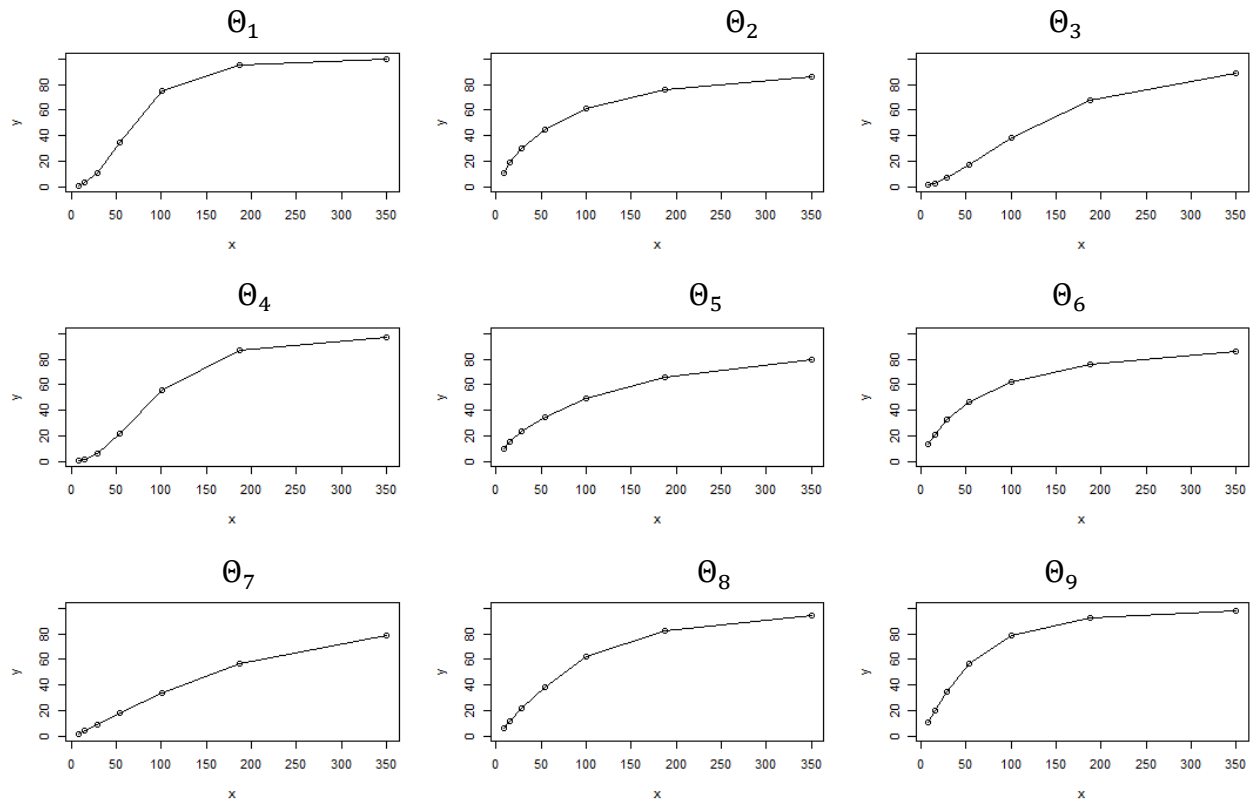


Figure 2. Dose-response curves of nine sets of parameter values for CLAN

4. OPTIMAL DESIGN

Optimal design is a field of statistics that is used to optimize design experiments in the most efficient way. In dose-response studies, optimal designs are used to determine treatment levels and the samples across the treatment levels efficiently. The optimal design criteria are developed based on different study objectives. The D-optimality criterion is one commonly used criterion for achieving efficient parameter estimation. This section will find the D-optimal designs and robust D-optimal designs for multiple nominal parameter values under the 5PL-1P model for each compound (BRAN and CLAN).

4.1. D-optimal Design

The D-optimality criterion is a commonly used criterion for obtaining optimal designs when the goal is to estimate the parameters of the model (Atkinson et al., 2007). The D-optimality criterion, by definition, maximizes the determinant of the Fisher information matrix: $\max_{\xi} |M(\xi, \Theta)|$. In the General Equivalence Theorem, which verifies whether the obtained D-optimal design is optimal or not, the equivalence of D-optimality under the 5PL-1P model was established under the condition:

$$\frac{g(x)^T M^{-1}(\xi_D, \Theta) g(x)}{4} \leq 1$$

where D-optimal design $\xi_D = \max_{\xi} |M(\xi, \Theta)|$, the equality holds when x is one of the dose levels in ξ_D .

This equivalence theorem is implemented in an algorithm to obtain the D-optimal design numerically. In this study, the efficient algorithm in Hyun, Wang, and Yang (2018) was adopted to search the optimal designs. The efficient algorithm utilizing the V-algorithm and the Newton

Raphson Method was obtained by modifying the YBT algorithm created by Yang, Biedermann and Tang (2013).

Here is a brief introduction of the efficient algorithm. The first step is to run the V-algorithm r times and select the last $s + 1$ generated dose levels as the initial dose levels, where r is a prespecified number, $r > s + 1$, and s represents the number of parameters. The next step is Newton Raphson method, which is used to determine the optimal weight corresponding to dose levels. Then a value of x_* is found that maximizes $\frac{g(x)^T M^{-1}(\xi, \theta) g(x)}{4}$, where $g(x)$ represents the gradient of the mean response function and $M^{-1}(\xi, \theta)$ is the inverse of the Fisher information matrix using the initial design. The final step is to observe if the general equivalence theorem, $\frac{g(x_*)^T M^{-1}(\xi, \theta) g(x_*)}{4} - 1 \leq \varepsilon$, $\varepsilon = 10^{-6}$, is satisfied; If so, the result design ξ will be the D-optimal design ξ_D . However, if this is not satisfied, return to the Newton Raphson method step with the x^* value added to the design, find another dose level value to maximize the sensitivity function until the condition is satisfied. More detail is given in Hyun, Wang and Yang(2018).

For the 5PL-1P Model, the Fisher information matrix is obtained using the first derivative $g(x_i, \theta)$ of model mean function $f(x, \theta)$ with respect to parameters $\theta_1, \theta_2, \theta_3$ and θ_4 according to Fisher information matrix equation in section 2.2. MacDonald's (2016) study provided the detailed Fisher information matrix formula under the 5PL-1P model. As we can see from the definition of Fisher information matrix, it depends on the parameter values θ for the 5PL-1P model. In order to find the D-optimal designs, the model parameter values θ must be known or specified in advance. As mentioned in Chapter 3, we chose nine sets of parameter values for each compound representing multiple nominal parameter values. We assume that the parameter values from the nine scenarios are the true values of model parameters.

Under the nine sets of random-selected parameter values $(\theta_1, \theta_2, \dots, \theta_9)$ for each compound in Tables 1 and 2, the resulting D-optimal designs under the 5PL-1P model in Tables 3 and 4 have four dose levels with equal weight, where the top row corresponds to the dose levels and the bottom row corresponds to their given weight. We found D-optimal designs for six sets of randomly selected parameter values. D-optimal designs generated under the given parameter values $(\theta_7, \theta_8, \theta_9)$ in Tables 1 and 2 for each compound at different exposure times: 15, 30 and 45 minutes were from MacDonald (2016).

Table 3

D-optimal designs of nine sets of parameter values for BRAN

$\xi_D^{BRAN-\theta_1} = \begin{pmatrix} 0.25 & 0.90 & 1.96 & 7.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{BRAN-\theta_2} = \begin{pmatrix} 0.78 & 1.67 & 3.55 & 7.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{BRAN-\theta_3} = \begin{pmatrix} 1.04 & 2.33 & 4.04 & 7.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{BRAN-\theta_4} = \begin{pmatrix} 0.45 & 1.27 & 2.32 & 7.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{BRAN-\theta_5} = \begin{pmatrix} 0.13 & 0.86 & 3.00 & 7.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{BRAN-\theta_6} = \begin{pmatrix} 0.74 & 1.43 & 2.66 & 7.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{BRAN-\theta_7} = \begin{pmatrix} 0.33 & 1.33 & 3.78 & 7 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{BRAN-\theta_8} = \begin{pmatrix} 0.26 & 1.01 & 2.84 & 7 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{BRAN-\theta_9} = \begin{pmatrix} 0.18 & 0.70 & 2.03 & 7 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	

Table 4

D-optimal designs of nine sets of parameter values for CLAN

$\xi_D^{CLAN-\theta_1} = \begin{pmatrix} 31.06 & 67.05 & 122.74 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-\theta_2} = \begin{pmatrix} 8 & 39.32 & 139.51 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{CLAN-\theta_3} = \begin{pmatrix} 36.07 & 101.48 & 206.56 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-\theta_4} = \begin{pmatrix} 37.71 & 83.79 & 160.07 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{CLAN-\theta_5} = \begin{pmatrix} 8.0 & 53.70 & 176.54 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-\theta_6} = \begin{pmatrix} 8.0 & 39.23 & 142.40 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{CLAN-\theta_7} = \begin{pmatrix} 24.0 & 90.8 & 212.7 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	$\xi_D^{CLAN-\theta_8} = \begin{pmatrix} 14.7 & 63.9 & 161.7 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$
$\xi_D^{CLAN-\theta_9} = \begin{pmatrix} 9.8 & 42.1 & 116.8 & 350.0 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$	

In order to verify the *D*-optimal designs follow the *D*-optimality criteria, their sensitivity functions $d(x_i, \theta) = \frac{g(x_i)^T M^{-1}(\xi, \theta) g(x_i)}{4}$ were plotted in Figures 3 and 4. The maxima of $d(x_i, \theta)$ in $x_i \in \chi$ were bounded horizontally below one. The curve showed that sensitivity reached their peaks at the dose levels x_i in their *D*-optimal design.

Verifying D-optimal Designs for BRAN

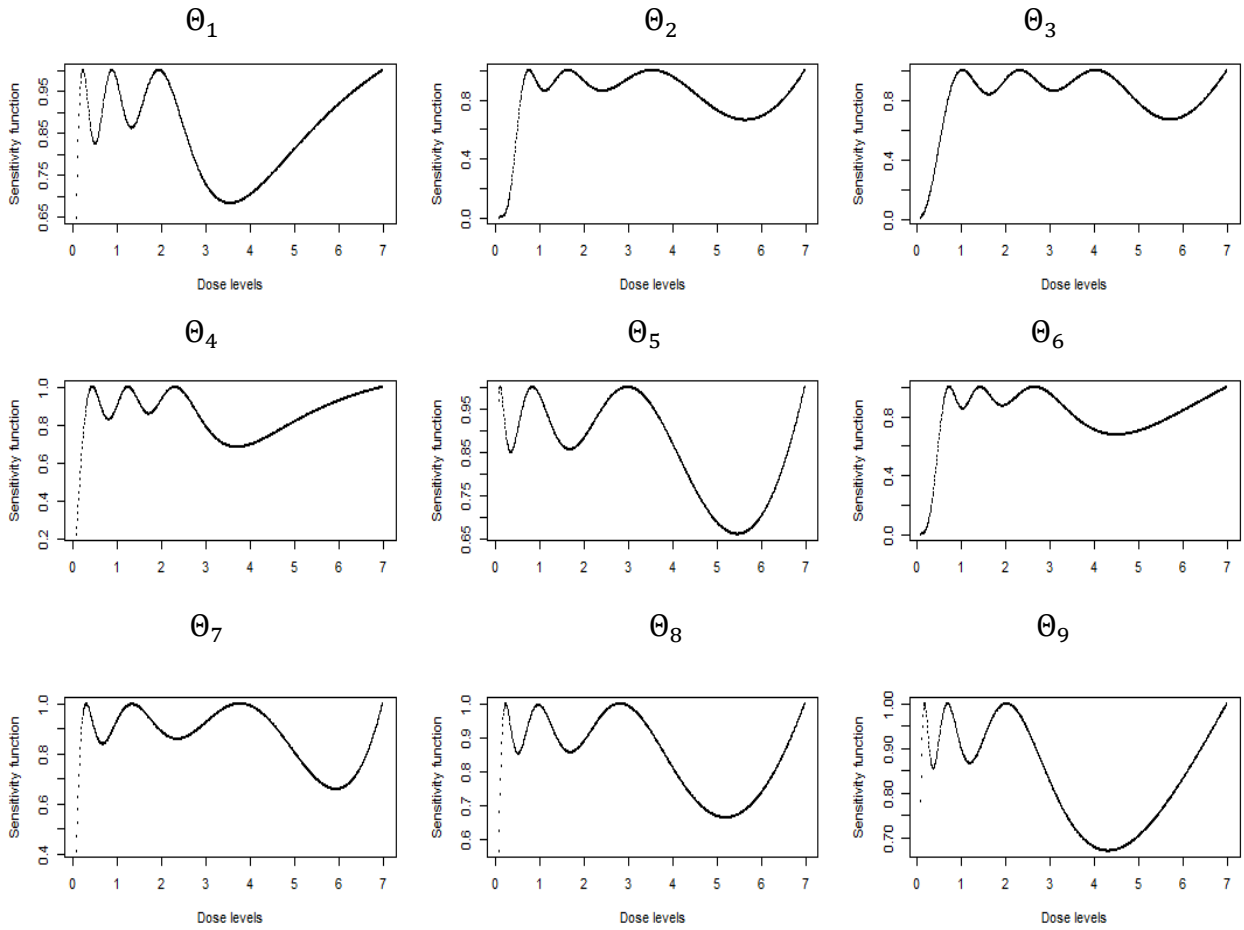


Figure 3. Sensitivity function plot Verifying the D-optimal designs of nine sets of parameter values for BRAN

Verifying D-optimal Designs for CLAN

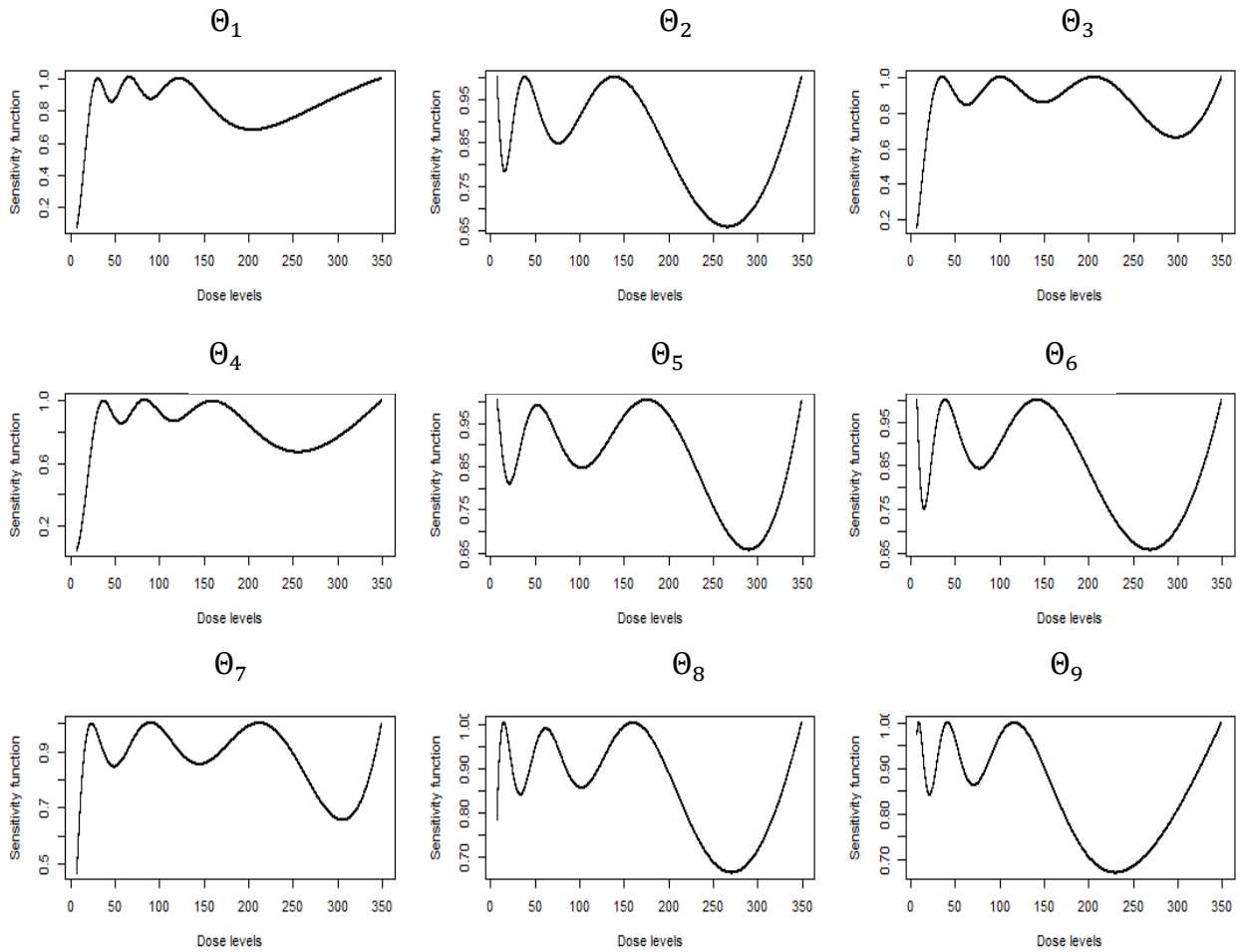


Figure 4. Sensitivity function plot verifying the D-optimal designs of nine sets of parameter values for CLAN

4.2. Robust D-optimal Design

The purpose of robust D-optimal design is to determine the design that works well for studying multiple nominal parameter values under the 5PL-1P model. Because the D-optimal design for one parameter value will likely not be optimal for different parameter values, the implemented robust D-optimal design must be selected carefully to provide satisfactory efficiencies for all considered parameter values simultaneously. A common approach for finding the robust D-optimal design is to use a compound design criterion that combines the various optimality criteria using their efficiencies and determine a design that maximizes the various efficiencies at the same time (Atkinson et al., 2007, Chapter 21). Here, the Bayesian technique was used to build the compound design criterion by utilizing the prior distribution to account for the uncertainty of multiple nominal parameter values.

For a model with s parameters in the mean function, the relative D-efficiency of a design ξ is defined as

$$Eff_D(\xi) = \left(\frac{|M_{(\xi; \Theta)}|}{|M_{(\xi_D; \Theta)}|} \right)^{\frac{1}{s}}$$

where Θ represents the multiple nominal parameter values and ξ_D represents the D-optimal design for Θ . The value of D-efficiency is between 0 and 1, which can be interpreted as: if ξ has $Eff(\xi) = e$, it then needs $100(1/e - 1)\%$ additional samples to perform as well as the D-optimal design. Here $Eff_{D_i}(\xi)$ measures the D-efficiency under the given parameter values θ_i , $i = 1, \dots, 9$.

As mentioned in Chapter 3, we considered nine sets of parameter values ($\theta_1, \theta_2, \dots, \theta_9$) under the 5PL-1P model to represent the multiple nominal parameter values. The robust D-optimal design $\xi_{RB, \lambda}$ is obtained to maximize the D-efficiencies under the nine sets of parameter

values and this can be constructed by using the Bayesian technique maximizing an weighted average of the nine efficiencies through a given prior distribution λ_i on Θ_i , $i = 1, \dots, 9$. λ_i is the prior weight representing the relative plausibility of each Θ_i . The weight vector λ allows us to specify a weight of interest for each set of parameter values by setting a value between 0 and 1. This identifies which parameter value is more important, or that all the parameter values have equal importance. In general, the robust D-optimal design $\xi_{RB,\lambda}$ using the Bayesian technique can be obtained through following equation:

$$\begin{aligned}\xi_{RB,\lambda} &= \arg \max_{\xi} \prod_{i=1}^9 \lambda_i \log \left(Eff_{D_i}(\xi) \right) \\ &= \arg \max_{\xi} \left\{ \lambda_1 \log \left(Eff_{D_1}(\xi) \right) + \lambda_2 \log \left(Eff_{D_2}(\xi) \right) + \dots + \lambda_9 \log \left(Eff_{D_9}(\xi) \right) \right\} \\ &= \arg \max_{\xi} \left\{ \frac{\lambda_1}{s} \log(|M_1(\xi, \Theta_1)|) + \frac{\lambda_2}{s} \log(|M_2(\xi, \Theta_2)|) + \dots + \frac{\lambda_9}{s} \log(|M_9(\xi, \Theta_9)|) \right\}\end{aligned}$$

where Fisher information matrix M_i for parameter value Θ_i under the 5PL-1P model is defined as $M_i(\xi, \Theta_i) = \sum_{j=1}^k w_j g(x_j) g(x_j)^T$ and the approximate design ξ takes w_j proportion of the total subjects at x_j , $j = 1, \dots, k$.

The derivative function of the above formula is shown below. This puts the criterion $\frac{\lambda_1}{s} g_1(x)^T M_1^{-1}(\xi_{RB,\lambda}, \Theta_1) g_1(x) + \frac{\lambda_2}{s} g_2(x)^T M_2^{-1}(\xi_{RB,\lambda}, \Theta_2) g_2(x) + \dots + \frac{\lambda_9}{s} g_9(x)^T M_9^{-1}(\xi_{RB,\lambda}, \Theta_9) g_9(x) \leq 1$, where $i = 1, 2, \dots, 9$. $M_i^{-1}(\xi, \theta)$ is the inverse of the Fisher information matrix of Θ_i .

A direct calculation shows the sensitivity function is:

$$\begin{aligned}d(x, \xi) &= \frac{\lambda_1}{s} g_1(x)^T M_1^{-1}(\xi_{RB,\lambda}, \Theta_1) g_1(x) + \frac{\lambda_2}{s} g_2(x)^T M_2^{-1}(\xi_{RB,\lambda}, \Theta_2) g_2(x) + \dots \\ &\quad + \frac{\lambda_9}{s} g_9(x)^T M_9^{-1}(\xi_{RB,\lambda}, \Theta_9) g_9(x)\end{aligned}$$

If all but one of the weights is nonzero, the above sensitivity function becomes the Equivalence Theorem for the one set of parameter value D-optimal design. Under the given

weight vector λ , the Equivalence Theorem for the design $d(x, \xi_{RB, \lambda}) - 1 \leq \varepsilon$ ($\varepsilon = 10^{-6}$) shows $\xi_{RB, \lambda}$ is the robust D-optimal design and the equality holds when x is one of the dose levels in $\xi_{RB, \lambda}$.

The Bayesian technique averages the criterion over the multiple nominal parameter values and optimizes the resulting criterion. According to the plausibility of each set of parameter values, prior weights are assigned to different sets of nominal parameter values; otherwise if nominal parameter values come from equally good previous studies, one may use a uniform prior to average out the uncertainty. For this paper, we only evaluated the robust D-optimal design with equal weight of interest for the nine sets of parameter values, using a uniform prior with $\lambda_1 = \lambda_2 = \dots = \lambda_9 = 1/9$.

The robust D-optimal design was accomplished by modifying the V-algorithm to work with multiple nominal parameter values and defining the robust D-optimality criterion in the algorithm. We found robust D-optimal design using Bayesian technique by modifying the algorithm provided by Hyun, Wang, and Yang (2018) in R software. For the nine sets of parameters values, we used uniform prior $\lambda_1 = \lambda_2 = \dots = \lambda_9 = 1/9$ and model parameters $s = 4$. The algorithm used to search robust D-optimal design followed the same steps as the efficient algorithm used to find D-optimal design except that different Fisher information matrix and sensitivity function were used. For the nine sets of parameter values, the robust D-optimal design was searched through maximizing the function

$$\frac{\lambda_1}{s} g_1(x)^T M_1^{-1}(\xi, \Theta_1) g_1(x) + \frac{\lambda_2}{s} g_2(x)^T M_2^{-1}(\xi, \Theta_2) g_2(x) + \dots + \frac{\lambda_9}{s} g_9(x)^T M_9^{-1}(\xi, \Theta_9) g_9(x)$$

until the inequality $d(x, \xi) - 1 \leq \varepsilon$, $\varepsilon = 10^{-6}$ is satisfied. In addition, during each iteration M_i is updated through the formula $M_i = [(1 - \alpha)M_i + \alpha g_i(x)g_i(x)^T]$. The detailed program was attached in appendix B.

The robust D-optimal designs using the Bayesian technique found from the modified algorithm are 7-point designs shown in Tables 5 and 6: ξ_{BD}^{BRAN} is the robust D-optimal design for BRAN and ξ_{BD}^{CLAN} is the robust D-optimal design for CLAN, where the top row corresponds to the concentration levels and the bottom row corresponds to their given weight.

Table 5

Robust D-optimal design of randomly selected parameter values for BRAN

$$\xi_{BD}^{BRAN} = \begin{pmatrix} 0.25 & 0.71 & 0.89 & 1.38 & 2.33 & 3.84 & 7 \\ 0.1401622 & 0.1477032 & 0.04025987 & 0.1492074 & 0.1292279 & 0.1626288 & 0.2308106 \end{pmatrix}$$

Table 6

Robust D-optimal design of randomly selected parameter values for CLAN

$$\xi_{BD}^{CLAN} = \begin{pmatrix} 9.43 & 34.83 & 72.86 & 101.26 & 130.34 & 177.78 & 350.00 \\ 0.1414436 & 0.1885416 & 0.1415009 & 0.06637699 & 0.03851655 & 0.1806724 & 0.242948 \end{pmatrix}$$

Figure 5 and 6 showed that the sensitivity function of the robust D-optimal design has a maximum value of one at the optimal dose levels over the dose interval and confirmed the optimality of the generated design.

Verifying the robust D-optimal Design for BRAN

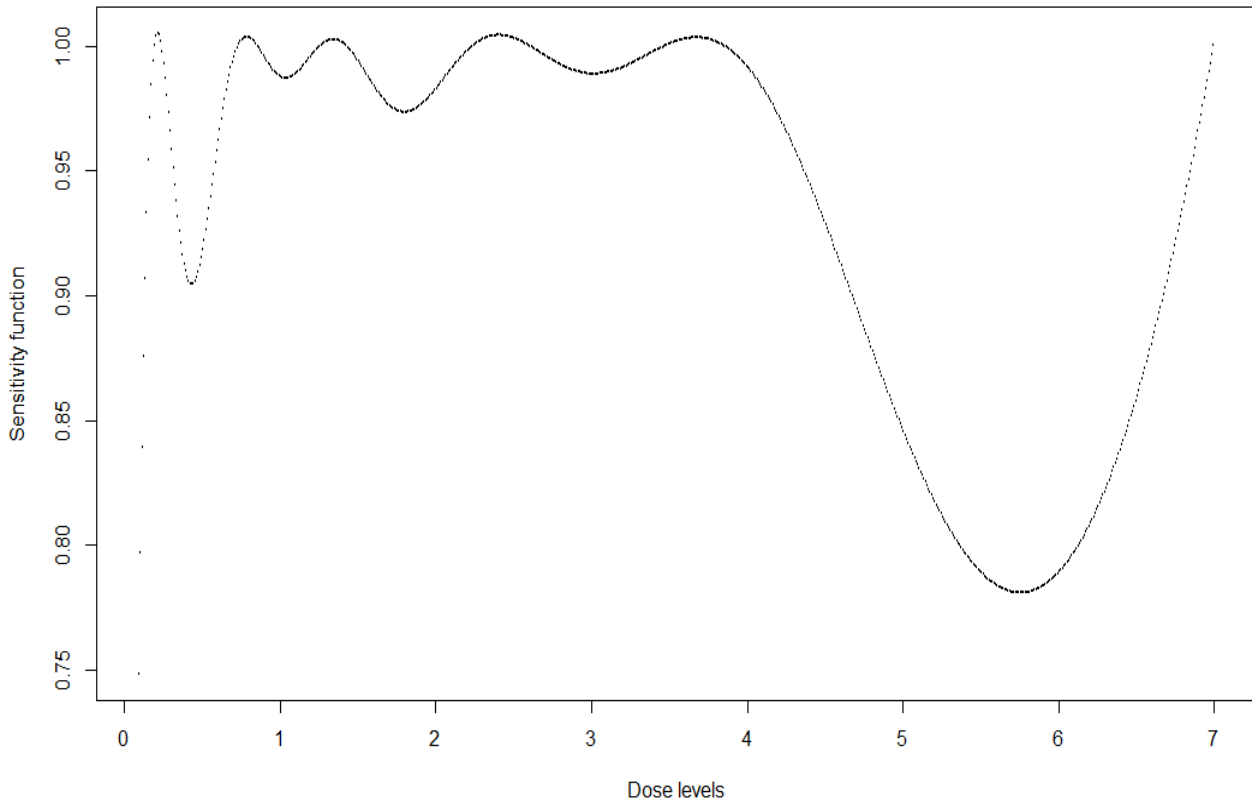


Figure 5. Plot of the sensitivity function of the robust D-optimal design for BRAN using Bayesian technique when uniform weight is assumed

Verifying the robust D-optimal Design for CLAN

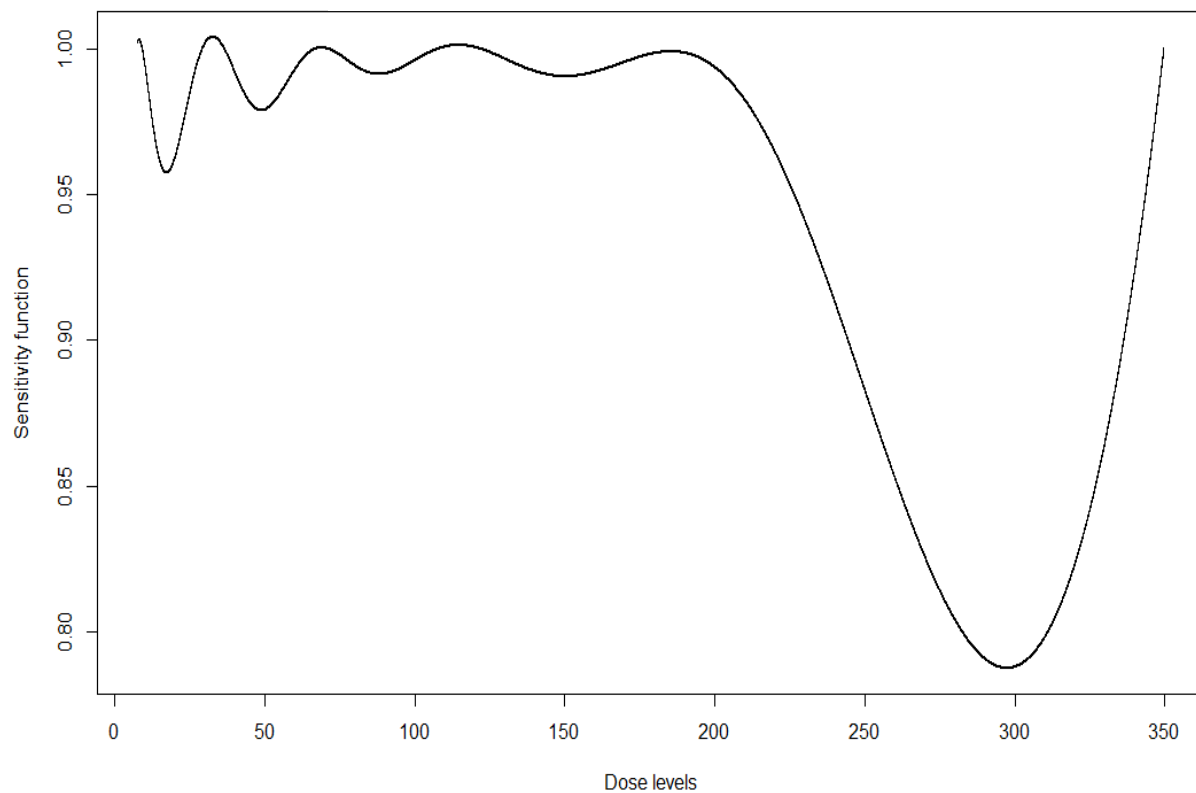


Figure 6. Plot of the sensitivity function of the robust D-optimal design for CLAN using Bayesian technique when uniform weight is assumed

5. EFFICIENCY

Since the parameter values of the chemical compounds BRAN and CLAN were used to obtain the robust D-optimal design, we checked the efficiency matrix by comparing robust D-optimal design to D-optimal designs, and original designs for each compound (BRAN and CLAN).

As mentioned in Chapter 4, different parameter value at different exposure time provided different dose-response function. Thus, D-optimal designs were obtained differently for parameter values ($\Theta_7, \Theta_8, \Theta_9$) at exposure times: 15, 30 and 45 minutes. For original designs from Dawson (2007), Table 7 included seven concentration levels for two compounds: BRAN and CLAN. The top row represents the concentration levels for each compound and the bottom row represents the weight for each of those levels. In the original study, the same design was used for different exposure time, i.e., the same design was always used disregarding the model parameter values.

Table 7

Original designs for the Two Compounds

$\xi_o^{BRAN} = \begin{pmatrix} 0.1655 & 0.3089 & 0.5765 & 1.0762 & 2.0089 & 3.75 & 7 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \end{pmatrix}$
$\xi_o^{CLAN} = \begin{pmatrix} 8.273 & 15.44 & 28.83 & 53.81 & 100.5 & 187.5 & 350 \\ 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 & 1/7 \end{pmatrix}$

For the 5PL-1P model, the relative efficiency of a design ξ is defined as

$$Eff_D(\xi) = \left(\frac{|M(\xi; \Theta)|}{|M(\xi_D; \Theta)|} \right)^{\frac{1}{4}}, \text{ Where } \xi_D \text{ represents the D-optimal design for } \Theta.$$

The calculated efficiencies are in Tables 8 and 9. The efficiency values in each row represents D-efficiency values of the specific design under nine sets of parameter values

$(\theta_1, \theta_2, \dots, \theta_9)$. These specific designs are the D-optimal designs for $\theta_1, \theta_2, \dots, \theta_9$, original design, and robust D-optimal design. D_{θ_i} represents the D-optimal design for θ_i , OD represents the original design and RD represents the robust D-optimal design.

Table 8

Efficiencies for nine sets of parameter values among Robust D-optimal design, Original design and D-optimal designs for BRAN

efficiency	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
D_{θ_1}	1.0000000	0.2757944	0.1615699	0.8204908	0.8146586	0.3894918	0.7298630	0.9079198	0.9410515
D_{θ_2}	0.4183381	1.0000000	0.8317284	0.6673696	0.5307401	0.8823614	0.7707420	0.6375839	0.3679331
D_{θ_3}	0.1797781	0.8379141	1.0000000	0.3405153	0.3603431	0.5709841	0.6106347	0.4047461	0.1833943
D_{θ_4}	0.8355679	0.6658238	0.3782309	1.0000000	0.6767360	0.8036195	0.7627900	0.8363158	0.6838382
D_{θ_5}	0.7554422	0.1151200	0.1857017	0.5666733	1.0000000	0.1683297	0.8153189	0.9033906	0.8685271
D_{θ_6}	0.5782200	0.9015655	0.5652835	0.8842945	0.5031776	1.0000000	0.6614716	0.6459496	0.4346067
D_{θ_7}	0.6154262	0.5369646	0.5113570	0.6550546	0.8573792	0.5033713	1.0000000	0.9148853	0.6706239
D_{θ_8}	0.8647521	0.3667484	0.3122816	0.7938814	0.9317136	0.4333085	0.9227092	1.0000000	0.8888468
D_{θ_9}	0.9346766	0.1430663	0.1121683	0.6526198	0.8875624	0.2141712	0.7044136	0.8930882	1.0000000
OD	0.8708354	0.6556210	0.6448561	0.7856552	0.8726632	0.6199144	0.8662679	0.8870162	0.8880933
RD	0.8779131	0.8135749	0.7932608	0.8672779	0.8628626	0.8071052	0.9106979	0.9196935	0.8724754

Table 9

Efficiencies for nine sets of parameter values among Robust D-optimal design, Original design and D-optimal designs for CLAN

efficiency	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
D_{θ_1}	1.0000000	0.5179422	0.6419795	0.8679358	0.5726877	0.4930651	0.6542145	0.7777875	0.6457712
D_{θ_2}	0.3818216	1.0000000	0.3889221	0.3079162	0.9435750	0.9997070	0.6050850	0.8624480	0.9622498
D_{θ_3}	0.5811793	0.4454663	1.0000000	0.8707370	0.6334741	0.4330159	0.9506096	0.7493008	0.4266294
D_{θ_4}	0.8563980	0.4595196	0.8800861	1.0000000	0.5906214	0.4413793	0.8170795	0.7803874	0.5120179
D_{θ_5}	0.3864148	0.9436109	0.5199689	0.3711641	1.0000000	0.9487149	0.7484936	0.9101426	0.8567118
D_{θ_6}	0.3754397	0.9997410	0.3923979	0.3064925	0.9478857	1.0000000	0.6103472	0.8625402	0.9559425
D_{θ_7}	0.5676319	0.5908879	0.9419353	0.7710476	0.7787242	0.5767079	1.0000000	0.8595591	0.5567532
D_{θ_8}	0.6885040	0.8357235	0.7171904	0.6705545	0.9114695	0.8151263	0.8664755	1.0000000	0.8652198
D_{θ_9}	0.5079356	0.9499982	0.3905223	0.3711162	0.8682957	0.9353424	0.5819659	0.8671031	1.0000000
OD	0.6911575	0.8737852	0.7334303	0.6775043	0.8687276	0.8664666	0.8011590	0.8687979	0.8871318
RD	0.7738370	0.9033385	0.8242321	0.8365960	0.9144814	0.8959483	0.8593645	0.9198618	0.9008115

In tables 8 and 9, the design in each row is used to check the D-efficiency under the 9 sets of parameter values. For example, in Table 8 for the first row the design D_{θ_1} under the second column θ_2 is 0.2757944. This means that it would take approximately

$100 \left(\frac{1}{\text{Eff}(\xi_D)} - 1 \right) = 265\%$ more samples for the design D_{θ_1} to reach the same accuracy of the design D_{θ_1} when the design D_{θ_1} is used under the model parameter value θ_2 .

Based on these efficiency values, it appeared that the robust D-optimal design using Bayesian technique performed better than other designs across the nine sets of parameter values. As we saw, D-optimal designs for each compound has an efficiency of one under their own parameter values but don't perform well for other parameter values. For the original designs and

robust D-optimal designs, efficiencies are evenly distributed across nine sets of parameter values, however, most of the efficiencies of the robust D-optimal designs are higher than that of the original designs. The higher efficiencies of the robust D-optimal design tell that it would require more samples for the original design to provide the same accuracy as the robust D-optimal design. Though the efficiencies indicated that the original designs used to fit the 5PL-1P model were acceptable, it was apparent that the accuracy can be increased by using the robust D-optimal designs when there are various parameter values are possibly considered.

6. CONCLUSION

In this study, I present a robust D-optimal design that works well for various parameter values under the 5PL-1P model.

Various parameter values are considered to describe various shapes of dose-response curves for the 5PL-1P model. The robust D-optimal design is obtained using the Bayesian techniques and is compared to other competitor designs. For searching the optimal designs effectively, the modified algorithm in Hyun, Wang, and Yang (2018) is used.

It is observed that the robust D-optimal design works better than other competitor designs under the 5PL-1P model by comparing their relative efficiencies. The relative efficiencies show that the robust D-optimal design performs evenly well across the various parameter values, while the D-optimal design performs well under its own parameter value but performs badly under other parameter values. Moreover, it also indicates that more samples are needed for the original design to reach the accuracy of the robust D-optimal design.

In this paper, the robust D-optimal design is searched for various nominal parameter values with equal weights using the Bayesian technique. The larger the numbers of multiple nominal parameter values are, the longer time it takes to find the robust D-optimal design. In the future, when there are larger numbers of multiple nominal parameter values under the 5PL-1P model, this study can be extended by implementing clustering methods, so larger numbers of parameter values can be clustering into small numbers of groups. Then, the robust D-optimal design can be obtained based on these clustered sets of parameter values efficiently.

REFERENCES

- Aarons, L., Ogungbenro, K. (2010). Optimal Design of Pharmacokinetic Studies, *Basic & Clinical Pharmacology & Toxicology* 106(3), 250-255. [Peer Reviewed Journal]
- Abebe, H. T., Tan, F. E. S., van Breukelen, G. J. P., & Berger, M. P. F. (2014). Bayesian D-optimal designs for the two-parameter logistic mixed effects model, *Computational Statistics & Data Analysis* 71, 1066–1076.
- Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007). Optimum Experimental Designs with SAS, *Oxford: Oxford University Press*.
- Bromoacetonitrile. (2016). *National Center for Biotechnology Information, PubChem Compound Database*. Retrieved from <https://pubchem.ncbi.nlm.nih.gov/compound/11534>
- Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability, *In F. M. Lord and M. R. Novick (Eds.), Statistical theories of mental test scores* (chaps. 17–20). Reading, MA: AddisonWesley.
- Chaloner, K., Larntz K. (1989). Optimal bayesian designs applied to logistic regression experiments, *Journal of Statistical Planning and Inference* 21, 191-208.
- Chaloner, K. and Verdinelli, I. (1995). Bayesian Experimental Design, *A Review, Statistical Science* 10(3), 273-304.
- DasGupta, A. (1995). Review of optimal Bayes designs, *Purdue Univ*.
- Dette, H. (1996). A note on Bayesian C- and D-optimal designs in nonlinear regression models, *Ann Statist* 3, 1225–1234.
- Dette, H., Haines, L. M. and Imhof, L. A. (2007). Maximin and bayesian optimal designs for regression models, *Statistica Sinica* 17, 463–480.
- Dawson, D. A., Genco, N., Bensinger, H. M., Quinn, D., Il'Giovine, Z. J., T. Wayne, S. T., and Pöch, G. (2012). Evaluation of an asymmetry parameter for curve-fitting in single chemical and mixture toxicity assessment, *Toxicology* 292(2-3), 156-161.
- Dawson, D. A., Jeyaratnam, J., Mooneyham, T., Pöch, G., and Schultz, T. W. (2010). Mixture Toxicity of *Sn*²-Reactive Soft Electrophiles: 1. Evaluation of Mixtures Containing α -Halogenated Acetonitriles, *Arch Environ Contam Toxicol* 59(4), 532-541.
- Fedorov, V. V., Klimko, E. M., and Studden, W. J. (1972). Theory of Optimal Experiments. *Academic press*, New York.

Hyun, S.W., Wong, W. K., and Yang, Y. (2018). VNM: An R package for finding multiple objective optimal designs for the 4-parameter logistic model, *Journal of Statistical Software* 83(5).

Hyun, S. W., and Wong, W. K. (2015). Multiple-objective optimal designs for studying the dose response function and interesting dose levels, *The Internal Journal of Biostatistics* 11(2), 253-271.

Kiefer, J. (1958). On the nonrandomized optimality and randomized nonoptimality of symmetrical designs, *Journal of Statistical Planning and Inference* 141, 559-575.

Lauer, S., Timmer, J., van Calker, D., Maier, D., Honerkamp, J. (1997). Optimal weighted Bayesian design applied to dose response-curve analysis, *Comm Statist TheoryMethods* 26, 2879–2903.

Macdonald, J. (2016). D-optimal design for the 5PL-1P model in chemical toxicity assessment. *ProQuest Dissertations and Theses*.

Montgomery, D.C. (2009). *Design and Analysis of Experiments (7th Edition)*, Wiley & Sons, Inc.

Mukhopadhyay, S., Haines, L. M. (1995). Bayesian D-optimal designs for the exponential growth model, *J Statist Plan Inference* 44, 385–397.

Prentice, R. (1976). A generalization of the probit and logit methods for dose response curves, *Biometrics* 32, 761–768. [PubMed]

Rodbard, D., Frazier, G. (1975). Statistical analysis of radioligand assay data, *Methods in Enzymology* 37, 3–22. [PubMed]

Rodbard, D., Munson, P., DeLean, A. (1978). Improved curve-fitting, parallelism testing, characterization of sensitivity and specificity, validation, and optimization for radioligand assays, *Radioimmunoassay and Related Procedures in Medicine* 1, 469–504.

Yang, M., Biedermann, S., and Tang, E. (2013). On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm, *Journal of American Statistical Association* 108, 1411-1420.

APPENDIX A. R CODES FOR NINE SETS OF PARAMETER VALUES

```
##### nine sets of parameter values for BRAN #####
rm(list=ls())
# 0.5<t2<4  0.5<t3<4  0.3<t4<4
size <- 200          #length of random number vectors
set.seed(5)
T1 = rep(100,200)
t2<- runif(size,min = 0.5, max = 4) # generate samples from uniform
distribution (0.5, 4)
t3<-runif(size,min = 0.5, max = 4)
t4<-runif(size,min = 0.3, max = 4)
new<-cbind(T1,t2,t3,t4)

T15b = c(128.1528,2.3244,0.9791,1.5470)
T30b = c(103.2062,1.6336,1.5402,0.8235)
T45b = c(100.97883,1.08130,1.70242,0.71926)

new_df<-rbind(new[c(4,16,49,106,112,114),],T15b,T30b,T45b)
#Parameters
new_df
x<- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
ff<-NULL
fb<-NULL

for(i in 1:dim(new_df)[1]){
  ff<-NULL
  for (j in 1:length(x)){
    f<-new_df[i,1]/(1+(new_df[i,2]/x[j])^(new_df[i,3]))^new_df[i,4]
    ff<-cbind(ff,f)
  }
  fb<-rbind(fb,ff)
}
fb
par(mfrow=c(3,3))
for(i in 1:9){
  plot (x,fb[i,], ylim=c(0, 100))
  lines(x,fb[i,],ylim=c(0, 100))
}
##### nine sets of parameter values for CLAN #####
#cleanup
rm(list=ls())
size <- 450          #length of random number vectors
set.seed(1000)
T12 = rep(100,450)
t22<- runif(size,min = 75, max = 204) # generate samples from uniform
distribution (75, 204)
t32<-runif(size,min = 1, max = 4)
t42<-runif(size,min = 0.5, max = 4)
new2<-cbind(T12,t22,t32,t42)

size <- 450
set.seed(88)
T1 = rep(100,450)
t2<- runif(size,min = 75, max = 204) # generate samples from uniform
distribution (75, 204)
```

```

t3<-runif(size,min = 1, max = 4)
t4<-runif(size,min = 0.5, max = 4)
new<-cbind(T1,t2,t3,t4)

T15c = c(105.7901,204.3503,1.5294,0.8279)
T30c = c(100.78867,119.55175,1.89378,0.56313)
T45c = c(100.73194,75.21709,1.87647,0.54536)
new_df<-rbind(new[c(55,408,413),],new2[c(65,91,137),],T15c,T30c,T45c)
new_df
x <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
ff<-NULL
fb<-NULL
for(i in 1:dim(new_df)[1]){
  ff<-NULL
  for (j in 1:length(x)){
    f<-new_df[i,1]/(1+(new_df[i,2]/x[j])^(new_df[i,3]))^new_df[i,4]
    ff<-cbind(ff,f)
  }
  fb<-rbind(fb,ff)
}
fb
par(mfrow=c(3,3))
for(i in 1:9){
  plot (x,fb[i,], ylim=c(0, 100))
  lines(x,fb[i,],ylim=c(0, 100))
}

```


APPENDIX B. R CODES FOR ROBUST D-OPTIMAL DESIGN

```

#cleanup
rm(list=ls())
MOPTR <- function(LB, UB,P,r=10,grid=0.01, epsilon=0.001, epsilon_w=10^-6) {

  X = c(LB, LB+(UB-LB)/4, LB+2*(UB-LB)/4, LB+3*(UB-LB)/4, UB)
  T=P
  k1 = length(X)
  W = rep(1/5,k1-1)
  e1 = epsilon_w
  e2 = epsilon
  nit=r
  n=1
  gr = grid
  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])
  }

  infor <- function(T,x) {
    f1 <- 1 / ((1+(T[2]/x)^T[3])^T[4])
    f2 <- -1 * T[1] * T[3] * T[4] * ((T[2]/x)^T[3]) * (((T[2]/x)^T[3])+1)^(-
1*T[4]-1))
    f3 <- -1 * T[1] * T[4] * ((T[2]/x)^T[3]) * log(T[2]/x) *
(((T[2]/x)^T[3])+1)^(-1*T[4]-1))
    f4 <- -1 * T[1] * (((T[2]/x)^T[3])+1)^(-1*T[4])) *
log(1+((T[2]/x)^T[3]))
    f = matrix(cbind(f1,f2,f3,f4))
    f%*%t(f)
  }

  upinfor <- function(W,T,x) {
    k = length(x)
    last_infor=infor(T,x[k])
    infor=(1-sum(W))* last_infor
    for (i in 1:(k-1)) {
      infor = infor + W[i] * infor(T,x[i])
    }
    infor
  }

  d1 <- function(T,x,xL,inv) {
    D1<-NULL
    for (i in 1:9){
      D1[i]=sum(diag(inv[[i]]%*%(infor(T[i,],x)-infor(T[i,],xL))))
    }
    sum(D1)/9
  }
}

```

```

ddl <- function(T,x1,x2,xL,inv) {
  DD1<-NULL
  for (i in 1:9){
    DD1[i]=sum(diag(-inv[[i]]%*%(infor(T[i,],x2)-
infor(T[i,],xL))%*%(infor(T[i,],x1)-infor(T[i,],xL))))
  }
  sum(DD1)/9
}
D_weight <- function(W,T,X,d) {
  p = length(W)
  k = length(X)

  inv<-list()
  for (i in 1:9){
    inv[[i]] = ginv(upinfor(W,T[i,],X))
  }

  f1 = rep(0,p)
  f2 = matrix(c(rep(f1,p)),nrow=p,ncol=p,byrow=F)
  for(i in 1:p) {
    f1[i] = d1(T,X[i],X[k],inv)
  }
  for(i in 1:p) {
    for(j in 1:p) {
      f2[i,j] = ddl(T,X[i],X[j],X[k],inv)
    }
  }
  newweight = W - d*(f1%*%ginv(f2))
  newweight
}

S_weight <- function(X,T) {
  diff = 10
  k = length(X)
  W = rep(1/k,k-1)
  while(diff>e1) {
    d = 1
    NW = D_weight(W,T,X,d)
    minW = min(min(NW),1-sum(NW))
    while(minW<0 & d>0.0001) {
      d = d/2
      NW = D_weight(W,T,X,d)
      minW=min(min(NW),1-sum(NW))
    }
    NW = c(NW,1-sum(NW))
    n = length(NW)
    minW = min(NW)
    diff = max(abs(W-NW[1:n-1]))
    if(abs(minW)<0.01 || minW<0) {
      for(i in 1:n) {
        if(NW[i]==minW)NW[i]=0
      }
    }
  }
  D = rbind(x=X,NW)
  for(i in 1:n) {
    if(D[2,i]==0)D[,i]= NA
  }
}

```

```

    }
    X=D[1,]
    W=D[2,]
    X=na.omit(X)
    W=na.omit(W)
    k=length(X)
    W=W[1:k-1]
  }
  W = c(W,1-sum(W))
  D = rbind(X,W)
  D
}
f = function(T,x) {
  f1 <- 1 / ((1+(T[2]/x)^T[3])^T[4])
  f2 <- -1 * T[1] * T[3] * T[4] * ((T[2]/x)^T[3]) * (((T[2]/x)^T[3])+1)^(-
1*T[4]-1))
  f3 <- -1 * T[1] * T[4] * ((T[2]/x)^T[3]) * log(T[2]/x) *
(((T[2]/x)^T[3])+1)^(-1*T[4]-1))
  f4 <- -1 * T[1] * (((T[2]/x)^T[3])+1)^(-1*T[4]) *
log(1+((T[2]/x)^T[3]))
  matrix(cbind(f1,f2,f3,f4))
}

ds1 = function(T,x,inv) {
  Ds<-NULL
  for (i in 1:9){
    Ds[i]=(t(f(T[i,],x))%*%inv[[i]]%*%f(T[i,],x))/4
  }
  sum(Ds)/9
}

M<-list()
for (i in 1:9){
  M[[i]]= upinfor(W,T[i,],X)
}

while(n<nit){
  x=seq(LB,UB,gr)
  n1=length(x)
  ds=rep(0,n1)

  inv<-list()
  for (i in 1:9){
    inv[[i]]<-ginv(M[[i]])
  }

  for (i in 1:n1) {
    ds[i]=ds1(T,x[i],inv)
  }
  for (i in 1:n1) {
    if(max(ds)==ds[i])x[i]=x[i] else x[i]=NA
  }

  newX=na.omit(x)
  newX=round(newX[1],2)
  newds=max(ds)
  an=1/(n+1)
}

```

```

p<-abs(newds-1)

newM<-list()
ff<-list()
for (i in 1:9){
  ff[[i]]= f(T[i,],newX)%*%t(f(T[i,],newX))
  newM[[i]]=(1-an)*M[[i]]+an*ff[[i]]
}
M=newM

X=c(X,newX)
n=n+1
}
s=length(X)
X=unique(X[(s-4):s])
X=sort(X,decreasing=F)
p=1
it=1

while(p>e2) {
  x = seq(LB,UB,gr)
  n1 = length(x)
  ds = rep(0,n1)
  D = S_weight(X,T)
  X = D[1,]
  k = length(X)
  W = D[2,1:k-1]

  inv<-list()
  for (i in 1:9){
    inv[[i]] = ginv(upinfor(W,T[i,],X))
  }

  for(i in 1:n1) {
    ds[i] = ds1(T,x[i],inv)
  }
  newX = x[which.max(ds)]
  newds = max(ds)
  X = c(X,newX)
  X = sort(X,decreasing=F)
  X = unique(X)
  newp <- abs(newds-1)
  if(abs(newp-p)<0.0000001) {
    newp = 10^-20
  }
  if(it>20) {
    newp = 10^-20
  }
  p = newp
  it = it+1
}

X = D[1,]
n = length(X)
W = D[2,1:n-1]
x = seq(LB,UB,gr)
n1 = length(x)

```

```

ds = rep(0,n1)

inv<-list()
for (i in 1:9){
  inv[[i]] = ginv(upinfor(W,T[i,],X))
}

for(i in 1:n1) {
  ds[i] = ds1(T,x[i],inv)
}
plot(x,ds,cex=.3,main="Verify D-optimal design", ylab="Sensitivity
function",xlab="Dose levels")
D
}
#fisher matrix#
infor <- function(T,x) {
  f1 <- 1 / ((1+(T[2]/x)^T[3])^T[4])
  f2 <- -1 * T[1] * T[3] * T[4] * ((T[2]/x)^T[3]) * (((T[2]/x)^T[3])+1)^(-
1*T[4]-1))
  f3 <- -1 * T[1] * T[4] * ((T[2]/x)^T[3]) * log(T[2]/x) *
(((T[2]/x)^T[3])+1)^(-1*T[4]-1))
  f4 <- -1 * T[1] * (((T[2]/x)^T[3])+1)^(-1*T[4])) * log(1+((T[2]/x)^T[3]))
  f = matrix(cbind(f1,f2,f3,f4))
  f%*%t(f)
}
upinfor <- function(W,T,x) {
  k = length(x)
  last_infor=infor(T,x[k])
  infor=(1-sum(W))* last_infor
  for (i in 1:(k-1)) {
    infor = infor + W[i] * infor(T,x[i])
  }
  infor
}
####Robust D-optimal design for nine sets of parameter values for BRAN###
size <- 200
set.seed(5)
T1 = rep(100,200)
t2<- runif(size,min = 0.5, max = 4) # generate samples from uniform
distribution (0.0, 4)
t3<-runif(size,min = 0.5, max = 4)
t4<-runif(size,min = 0.3, max = 4)
new<-cbind(T1,t2,t3,t4)

T15b = c(128.1528,2.3244,0.9791,1.5470)
T30b = c(103.2062,1.6336,1.5402,0.8235)
T45b = c(100.97883,1.08130,1.70242,0.71926)

new_df<-rbind(new[c(4,16,49,106,112,114),],T15b,T30b,T45b)

TT<-new_df

LB = 0.1
UB = 7
OD<-NULL
WEIGHT<-NULL
MOPTR(LB, UB,TT,r=10,grid=0.01, epsilon=0.001, epsilon_w=10^-6)

```

```

par(mfrow=c(3,3))

####Robust D-optimal design for nine sets of parameter values for CLAN###
size <- 450           #length of random number vectors
set.seed(1000)
T12 = rep(100,450)
t22<- runif(size,min = 75, max = 204) # generate samples from uniform
distribution (75, 204)
t32<-runif(size,min = 1, max = 4)
t42<-runif(size,min = 0.5, max = 4)
new2<-cbind(T12,t22,t32,t42)

size <- 450           #length of random number vectors
set.seed(88)
T1 = rep(100,450)
t2<- runif(size,min = 75, max = 204) # generate samples from uniform
distribution (75, 204)
t3<-runif(size,min = 1, max = 4)
t4<-runif(size,min = 0.5, max = 4)
new<-cbind(T1,t2,t3,t4)

T15c = c(105.7901,204.3503,1.5294,0.8279)
T30c = c(100.78867,119.55175,1.89378,0.56313)
T45c = c(100.73194,75.21709,1.87647,0.54536)
new_df<-rbind(new[c(55,408,413),],new2[c(65,91,137),],T15c,T30c,T45c)

new_df

TT<-new_df

LB = 8
UB = 350
OD<-NULL
WEIGHT<-NULL

MOPTR(LB, UB,TT,r=10,grid=0.01, epsilon=0.001, epsilon_w=10^-6)

```

APPENDIX C. R CODES FOR EFFICIENCIES

```
#####efficiency for BRAN #####
#D-optimal design function: import MOPT package from the efficient algorithm
in Hyun, Wang, and Yang (2018)
size <- 200
set.seed(5)
T1 = rep(100,200)
t2<- runif(size,min = 0.5, max = 4) # generate samples from uniform
distribution (0.0, 4)
t3<-runif(size,min = 0.5, max = 4)
t4<-runif(size,min = 0.3, max = 4)
new<-cbind(T1,t2,t3,t4)
T15b = c(128.1528,2.3244,0.9791,1.5470)
T30b = c(103.2062,1.6336,1.5402,0.8235)
T45b = c(100.97883,1.08130,1.70242,0.71926)
new_df<-rbind(new[c(4,16,49,106,112,114),],T15b,T30b,T45b)
new_df
LB = 0.1
UB = 7
OD<-NULL
WEIGHT<-NULL
## optima Designs for nine sets of parameter values##
for (i in 1:dim(new_df)[1]){
  TD<-new_df[i,]
  demoni<-MOPT(LB,UB,TD,r=30,epsilon=.01)
  Dx<-demoni[1,]
  weight1<- demoni[2,-length(demoni[2,])]
  OD<-rbind(OD,Dx)
  WEIGHT<-rbind(WEIGHT,weight1)
}
OD
WEIGHT
optimaldesign<-data.frame(OD,WEIGHT)
optimaldesign
detDMM<-NULL
detNMM<-NULL
## Calculate efficiency#
REFF<-NULL
for (i in 1:dim(new_df)[1])
{
  for (j in 1:dim(new_df)[1])
  {
    NM <- upinfor(WEIGHT[i,],new_df[j,],OD[i,])
    detNM <- det(NM)
    detNMM<-cbind(detNMM,detNM)
    DM <- upinfor(WEIGHT[j,],new_df[j,],OD[j,])
    detDM <- det(DM)
    detDMM<-cbind(detDMM,detDM)
    ##Efficiency##
    eff<- (detNM/detDM)^(1/4)
    REFF<-cbind(REFF,eff)
  }
}
REFF
new_df
```

```

optimaldesign

#original design
BRAN <- c(0.1655,0.3089,0.5765,1.0762,2.0089,3.75,7)
weightB <- c(1/7,1/7,1/7,1/7,1/7,1/7)
NMo<-NULL
detNMo<-NULL
detNMMo<-NULL
for (j in 1:dim(new_df)[1])
{
  NMo <- upinfor(weightB,new_df[j,],BRAN)
  detNMo <- det(NMo)
  detNMMo<-cbind(detNMMo,detNMo)
}
detNMMo

#9 sets of parameter values Robust D-optimal design
NBRAN9<-
c(0.2500000,0.7100000,0.8900000,1.3800000,2.3300000,3.8400000,7.0000000)
NW9<-c(0.1401622,0.1477032,0.04025987,0.1492074,0.1626288,0.1292279)
NMN9<-NULL
detNMN9<-NULL
detNMMN9<-NULL
for (j in 1:dim(new_df)[1])
{
  NMN9 <- upinfor(NW9,new_df[j,],NBRAN9)
  detNMN9 <- det(NMN9)
  detNMMN9<-cbind(detNMMN9,detNMN9)
}
detNMMN9

detDMATRIX= matrix(
  detDMM, # the data elements
  nrow=9,          # number of rows
  ncol=9,          # number of columns
  byrow = TRUE)
detDMATRIX

detDMATRIX[9,]
REFFo<-(detNMMo/detDMATRIX[9,])^(1/4)

detDMATRIX[9,]
REFFN9<-(detNMMN9/detDMATRIX[9,])^(1/4)

detNMATRIX= matrix(
  detNMM, # the data elements
  nrow=9,          # number of rows
  ncol=9,          # number of columns
  byrow = TRUE)

detNMATRIX
rbind(detNMATRIX,detNMMo)

effmatrix= matrix(
  REFF, # the data elements
  nrow=9,          # number of rows
  ncol=9,          # number of columns

```



```

byrow = TRUE)

effmatrix

rbind(effmatrix,REFFo,REFFN9)

#####efficiency for CLAN #####
#D-optimal design function: import MOPT package from the efficient algorithm
in Hyun, Wang, and Yang (2018)
size <- 450          #length of random number vectors
set.seed(1000)
T12 = rep(100,450)
t22<- runif(size,min = 75, max = 204) # generate samples from uniform
distribution (75, 204)
t32<-runif(size,min = 1, max = 4)
t42<-runif(size,min = 0.5, max = 4)
new2<-cbind(T12,t22,t32,t42)

size <- 450          #length of random number vectors
set.seed(88)
T1 = rep(100,450)
t2<- runif(size,min = 75, max = 204) # generate samples from uniform
distribution (75, 204)
t3<-runif(size,min = 1, max = 4)
t4<-runif(size,min = 0.5, max = 4)
new<-cbind(T1,t2,t3,t4)
#clan <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
#weightB <- c(1/7,1/7,1/7,1/7,1/7,1/7)
T15c = c(105.7901,204.3503,1.5294,0.8279)
T30c = c(100.78867,119.55175,1.89378,0.56313)
T45c = c(100.73194,75.21709,1.87647,0.54536)
new_df<-rbind(new[c(55,408,413),],new2[c(65,91,137),],T15c,T30c,T45c)
#new_df<-rbind(new[c(55,57,408,413),],new1[c(199,305,338),],T15c,T30c,T45c)
new_df
par(mfrow=c(3,3))

LB = 8
UB = 350
OD<-NULL
WEIGHT<-NULL
## optima Designs for nine sets of parameter values##
for (i in 1:dim(new_df)[1]){
  TD<-new_df[i,]
  demoni<-MOPT(LB,UB,TD,r=30,epsilon=.01)
  OD[[i]]<-demoni[1,]
  WEIGHT[[i]]<- demoni[2,-length(demoni[2,])]
}
OD
WEIGHT
detDMM<-NULL
detNMM<-NULL

## Calculate efficiency#
REFF<-NULL
for (i in 1:dim(new_df)[1])
{
  for (j in 1:dim(new_df)[1])

```

```

{
  NM <- upinfor(WEIGHT[[i]],new_df[j,],OD[[i]])
  detNM <- det(NM)
  detNMM<-cbind(detNMM,detNM)
  DM <- upinfor(WEIGHT[[j]],new_df[j,],OD[[j]])
  detDM <- det(DM)
  detDMM<-cbind(detDMM,detDM)
  ##Efficiency##
  eff<- (detNM/detDM)^(1/4)
  REFF<-cbind(REFF,eff)
}
}

REFF

new_df

clan <- c(8.273,15.44,28.83,53.81,100.5,187.5,350)
weightB <- c(1/7,1/7,1/7,1/7,1/7,1/7)
NMo<-NULL
detNMo<-NULL
detNMMo<-NULL
for (j in 1:dim(new_df)[1])
{
  NMo <- upinfor(weightB,new_df[j,],clan)
  detNMo <- det(NMo)
  detNMMo<-cbind(detNMMo,detNMo)
}
detNMMo

#X W CLAN 9 OPTIMAL DESIGN
#X 9.4300000 34.8300000 72.8600000 101.2600000 130.3400000 177.7800000
350.000000
#W 0.1414436 0.1885416 0.1415009 0.06637699 0.03851655 0.1806724
0.242948
N9BRAN<-
c(9.4300000,34.8300000,72.8600000,101.2600000,130.3400000,177.7800000,350.0
00000)
N9W<-c(0.1414436,0.1885416,0.1415009,0.06637699,0.03851655,0.1806724)
N9MN<-NULL
detNMN9<-NULL
detNMMN9<-NULL
for (j in 1:dim(new_df)[1])
{
  N9MN <- upinfor(N9W,new_df[j,],N9BRAN)
  detNMN9 <- det(N9MN)
  detNMMN9<-cbind(detNMMN9,detNMN9)
}
detNMMN9
detDMATRIX= matrix(
  detDMM, # the data elements
  nrow=9, # number of rows
  ncol=9, # number of columns
  byrow = TRUE)
detDMATRIX

detDMATRIX[9,]

```

```

REFFo<- (detNMMo/detDMATRIX[9,])^(1/4)

detDMATRIX[9,]
REFFN9<- (detNMMN9/detDMATRIX[9,])^(1/4)
detNMATRIX= matrix(
  detNMM, # the data elements
  nrow=9,          # number of rows
  ncol=9,          # number of columns
  byrow = TRUE)

detNMATRIX
rbind(detNMATRIX,detNMMo)

effmatrix= matrix(
  REFF, # the data elements
  nrow=9,          # number of rows
  ncol=9,          # number of columns
  byrow = TRUE)

effmatrix

rbind(effmatrix,REFFo,REFFN9)

```