Summer 2010

Adaptive designs for dose response studies

Yu-Hui Huang Chang
University of Iowa

ADAPTIVE DESIGNS FOR DOSE RESPONSE STUDIES

by

Yu-Hui Huang Chang

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Biostatistics in
the Graduate College of
The University of Iowa

July 2010

Thesis Supervisor: Professor Kathryn Chaloner

ABSTRACT

This thesis is motivated by a study in which healthy volunteers were inoculated with different doses of nontypeable *Haemophilus influenzae*. The goal was to estimate the doses at which 50% and 90% of subjects became colonized, and these doses are denoted as the $HCD50$ and $HCD90$ respectively. A fifteen-subject study was designed in two stages, with the first six subjects allocated sequentially. The design was chosen based on scientific, practical, and statistical arguments, however, due to limited time, heuristic decisions were made for expedience. The design implemented in the study, together with a number of alternative designs based on specific algorithms or criteria, are evaluated in depth, under both Bayesian and frequentist paradigms.

In particular, Bayesian myopic strategies with one-, two- and three-step-look-ahead procedures are investigated. The optimal sequential design strategy is that with minimum expected loss, where the expected loss is defined as the sum of the expected posterior variance of the $HCD50$ and $HCD90$. The higher the expected loss is, the worse is the performance of the design. In addition, a toxicity-response relationship may be appropriate, and can be incorporated into the design by putting a constraint on the posterior probability of toxicity at any dose. A new model considering both colonization (efficacy) and adverse event (toxicity) is proposed, and design procedures are developed incorporating this constraint.

Monte Carlo simulations are used to estimate the expected loss for candidate

design strategies for both univariate and bivariate models, and the results show that it is typically beneficial to look more steps ahead in determining designs, although the benefit may not be large. For the bivariate model, as the restriction becomes more conservative, the expected loss becomes larger and early stopping may occur since there are no acceptable doses available.

Non-sequential designs are also found and examined. Criteria from optimal design theory are used by optimizing a function of the expected Fisher information matrix: the inverse of this matrix corresponds to the asymptotic covariance matrix of the parameters. An A-optimal design criterion is used which minimizes the sum of the asymptotic variances of the $HCD50$ and $HCD90$, or an expectation of the sum over a prior distribution. The corresponding sum of the exact posterior variances are estimated for the non-sequential designs, and compared with estimates from sequential strategies. These comparisons show that using sequential design strategies is better than non-sequential strategies, and the improvement may be large.

Finally, some projects for future research in this area are proposed.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

ADAPTIVE DESIGNS FOR DOSE RESPONSE STUDIES

by

Yu-Hui Huang Chang

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Biostatistics in
the Graduate College of
The University of Iowa

July 2010

Thesis Supervisor: Professor Kathryn Chaloner

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Yu-Hui Huang Chang

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Biostatistics at the July 2010 graduation.

Thesis Committee: _____
Kathryn Chaloner, Thesis Supervisor

_____
Joseph Cavanaugh

_____
Joseph Lang

_____
Brian Smith

_____
Patricia Winokur

To My Parents and Michael

ACKNOWLEDGEMENTS

It is a pleasure to thank the many people who made this thesis possible.

I would like to gratefully acknowledge the enthusiastic supervision of my thesis advisor, Dr. Kathryn Chaloner. With her inspiration and guidance, I had a lot of fun working on my dissertation. Without her good teaching, endless support and encouragement throughout the development of my thesis, I would have been lost.

Furthermore, I am indebted to the committee members, Dr. Joseph Cavanaugh, Dr. Joseph Lang, Dr. Brian Smith, and Dr. Patricia Winokur, for their invaluable suggestions and comments.

I also appreciate the support of the training grant, T32GM077973 from the National Institute of General Medical Sciences, which provides the financial support for my research.

Finally, I would like to thank my parents, Mr. Chin-Tang Huang and Mrs. Yu-Hsia Chang, and my husband, Michael Chang for their never-ending support and love. To them I dedicate this thesis.

ABSTRACT

This thesis is motivated by a study in which healthy volunteers were inoculated with different doses of nontypeable *Haemophilus influenzae*. The goal was to estimate the doses at which 50% and 90% of subjects became colonized, and these doses are denoted as the $HCD50$ and $HCD90$ respectively. A fifteen-subject study was designed in two stages, with the first six subjects allocated sequentially. The design was chosen based on scientific, practical, and statistical arguments, however, due to limited time, heuristic decisions were made for expedience. The design implemented in the study, together with a number of alternative designs based on specific algorithms or criteria, are evaluated in depth, under both Bayesian and frequentist paradigms.

In particular, Bayesian myopic strategies with one-, two- and three-step-look-ahead procedures are investigated. The optimal sequential design strategy is that with minimum expected loss, where the expected loss is defined as the sum of the expected posterior variance of the $HCD50$ and $HCD90$. The higher the expected loss is, the worse is the performance of the design. In addition, a toxicity-response relationship may be appropriate, and can be incorporated into the design by putting a constraint on the posterior probability of toxicity at any dose. A new model considering both colonization (efficacy) and adverse event (toxicity) is proposed, and design procedures are developed incorporating this constraint.

Monte Carlo simulations are used to estimate the expected loss for candidate

design strategies for both univariate and bivariate models, and the results show that it is typically beneficial to look more steps ahead in determining designs, although the benefit may not be large. For the bivariate model, as the restriction becomes more conservative, the expected loss becomes larger and early stopping may occur since there are no acceptable doses available.

Non-sequential designs are also found and examined. Criteria from optimal design theory are used by optimizing a function of the expected Fisher information matrix: the inverse of this matrix corresponds to the asymptotic covariance matrix of the parameters. An A-optimal design criterion is used which minimizes the sum of the asymptotic variances of the $HCD50$ and $HCD90$, or an expectation of the sum over a prior distribution. The corresponding sum of the exact posterior variances are estimated for the non-sequential designs, and compared with estimates from sequential strategies. These comparisons show that using sequential design strategies is better than non-sequential strategies, and the improvement may be large.

Finally, some projects for future research in this area are proposed.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION: ADAPTIVE DOSE SEEKING DESIGN

1.1 Introduction

Dose-seeking experiments occur in early phase clinical trials: phase I and phase II clinical trials. The purpose of early phase clinical trials is sometimes to determine whether a treatment has sufficient promise of efficacy and safety to study further. In cancer research, they may test a new drug on volunteers with life threatening disease. In this chapter, some general approaches to dose-finding designs are described. Then an overview of the thesis is presented, followed by an introduction to Bayesian decision theory.

1.2 Methods for Adaptive Dose-response Designs

The goal for cancer phase I clinical trials is to determine the maximum tolerated dose (MTD). The MTD is defined as the dose that causes unacceptable risk, dose limiting toxicity (DLT), to a specific proportion of subjects [9, 6, 23]. The DLT is determined by severity according to the Common Terminology Criteria for Adverse Event (CTCAE) of the National Cancer Institute [36], and by the relation to treatment [24, 37]. Usually, a DLT is defined as a toxicity of grade 3 or higher from CTCAE and/or at least possibly related to treatment [19, 23]. More specifically, the DLT is a binary outcome. Denote θ be the probability, the relationship between MTD, DLT and θ is defined as follows [6, 23]:

$$\Pr(\text{DLT} \mid \text{MTD}) = \theta.$$

A number of methods have been proposed to find the MTD, and these methods can be divided into two categories: conventional or algorithm-based methods and model-based methods [46, 19]. Both methods assume a monotonic dose-response relationship. The conventional or algorithm-based method is to treat MTD as a statistic, while in model-based designs, the MTD is a parameter to be estimated. A brief introduction to both methods will be described in the following.

1.2.1 Conventional / Algorithm-based Method

Typically, a small group (often three) of subjects are treated at each dose. The dose escalation may follow the "up-and-down" scheme [22, 65]: the increment or decrement in dose depends on the outcomes observed. The value of θ is often defined as 1/3 [52], and a group of three subjects are treated with the same dose level. Other variations of this type of design are described by Storer [52]. Although the conventional method is easy to implement, it usually does not provide good estimation of the MTD [39, 29].

1.2.2 Model-based Methods

Model-based methods define the MTD as a parameter to be estimated. In addition to frequentist approaches, Bayesian approaches have also been applied. Gatsonis and Greenhouse [28], and Babb and Rogatko [6] apply Bayesian methods to estimate the MTD, and to understand the dose-response curve. Different Bayesian approaches in designing such experiments are reviewed by Berry [8], and Babb and Rogatko [6]. Methods using Bayesian approaches include the continual

reassessment method (CRM), dose escalation with overdose control (EWOC), and Bayesian decision theory.

The CRM was proposed by O'Quigley, Pepe and Fisher [40], and was motivated by phase I clinical trial in oncology. A one-parameter model for the dose-response relationship, a prior distribution for the parameter of the model, a set of doses, and target proportion of responses are predefined in the CRM. The idea of this method is to sequentially assign subjects to one of the pre-specified doses by updating the posterior distribution of the parameters in the model. The criterion for dose allocation is based on the dose for which the probability of toxicity is closest to the target dose. Some extensions and improvement of CRM were proposed by O'Quigley and Chevert [39], Korn et al. [35], Goodman et al. [30], Piantadosi and Liu [43], and Braun [12]. O'Quigley and Shen [41] later proposed CRM under the frequentist framework.

Babb, Rogatko and Zacks [5] proposed EWOC. Similar to the original CRM, it is a Bayesian sequential approach, and potential dose levels need to be pre-specified. EWOC utilizes a two-parameter model, and an overdose control: controlling the proportion of overdosed patients to be lower than a specific value. The authors discussed that using this method, the MTD was estimated accurately, and a lower frequency of toxicities occurred compared to CRM [67]. Work extending EWOC can be found by Tighiouart, Rogatko and Babb [56].

Whitehead and Brunier [59] used Bayesian decision theory for dose determining experiments. As in CRM and EWOC, they proposed a sequential and adaptive

method. A two-parameter model, a prior distribution for the parameters, a set of possible action (doses), and a gain or loss function are used. The dose is selected by minimizing the expected loss or by maximizing the expected gain. In their proposal, Bayesian decision theory is incorporated in the design of the study, and the frequentist approach is used in estimation. Other extended work with regard to this approach can be found by Whitehead [58], Whitehead and Williamson [60], Whitehead and Zhou et al. [61], and Whitehead et al. [62].

The CRM, EWOC, and the method proposed by Whitehead and Brunier can be considered as Bayesian myopic strategies since the dose determination procedures involve updating the posterior distribution after sequentially observing the outcomes, and then evaluating the corresponding loss function for the next subject [46]. More precisely, these are one-step-look-ahead myopic procedures (page 367 - 368, [20]). The true optimal design can be found by backward induction, also known as dynamic programming [27, 21]. However, the computational complexity makes backward induction impractical in most cases. A more feasible approach is to extend the number of steps to look ahead in the myopic procedure: the m-step-look-ahead procedure for $m > 1$. Intuitively, as m increases, the closer to optimal the design should be, although this may not necessarily be true. In design with a fixed sample size, when m is equal to the sample size, the m-step-look-ahead procedure is equivalent to backward induction.

1.3 Overview of the Thesis

In this thesis, the algorithms for different myopic strategies are developed, and the performances of the strategies are evaluated. A logistic model for a univariate outcome (efficacy) is applied first, and then a new model for bivariate outcomes incorporating both efficacy and toxicity is proposed.

The thesis is organized as follows. Bayesian decision theory is briefly discussed in the next section of this chapter. In Chapter 2, the motivating study is introduced, and issues related to the study are also addressed and evaluated. In Chapter 3, algorithms for four myopic procedures with a univariate binary outcome are provided, and applied to a logistic model. The results from simulation studies are summarized in Chapter 4. In Chapter 5, several methods for finding designs for bivariate outcomes are reviewed, and a new model incorporating both efficacy and toxicity is proposed. The algorithms and results from simulations with the new model are described in Chapter 6 and Chapter 7. Two optimality criteria, D- and A-optimality, are described and implemented in Chapter 8, and simulations for the new model and the design criteria are shown in this chapter. Finally, some ideas for future work are given in Chapter 9.

1.4 Bayesian Decision Theory

Let $\boldsymbol{\theta}$ be the parameter of interest, $\mathbf{x}$ be the data, and $f(\mathbf{x}|\boldsymbol{\theta})$ be the likelihood. Bayesian decision theory includes the following elements [7, 21, 64]:

1. Prior distribution: This represents the uncertainty about $\boldsymbol{\theta}$, and is selected based on previous experience or knowledge of the investigators of a study.

The density function for $\boldsymbol{\theta}$ is denoted as $p(\boldsymbol{\theta})$.

2. A set of decisions: A collection of decisions are pre-specified, and $\{d_1, d_2, ..., d_d\} = \mathcal{D}$ denotes a set of all possible decisions.

3. Loss or gain function: Let $L(d_j, \boldsymbol{\theta})$ denote the loss of choosing the decision d_j, where the true parameter value is $\boldsymbol{\theta}$. The corresponding gain function is $-L(d_j, \boldsymbol{\theta})$.

To choose a decision, the expected loss is computed. Denotes $EL(d_j, \boldsymbol{\theta})$ be the expected loss for selecting d_j, and $p(\boldsymbol{\theta}|\mathbf{x})$ be the posterior distribution of $\boldsymbol{\theta}$. By Bayes' theorem

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{f(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int f(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})\ d\boldsymbol{\theta}},$$

and

$$EL(d_j, \boldsymbol{\theta}) = E\Big(L(d_j, \boldsymbol{\theta})\Big) = \int L(d_j, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x})\ d\boldsymbol{\theta}.$$

The best decision, d^* is the one with minimal expected loss, hence,

$$d^* = \underset{d_j \in \mathcal{D}}{\operatorname{argmin}} \left\{ EL(d_j, \boldsymbol{\theta}) \right\}.$$

1.5 Discussion

Bayesian design is used in this thesis. For studies with a small sample size, such as animal studies or early phase clinical trials, a benefit of using a Bayesian approach is that it works well with a small sample size. Prior information is formally incorporated, and design is well suited for a decision theoretic approach.

CHAPTER 2
MOTIVATING EXAMPLE: THE H-FLU STUDY

2.1 H-Flu (*Haemophilus influenzae*) Study

Haemophilus influenzae (*H. influenzae*) is a Gram-negative bacteria that colonizes and infects humans. Nontypeable *H. influenzae* (NTHi) are a group of *H. influenzae*. NTHi are responsible for 35% of otitis media which costs approximately $3.5 billion a year [51]. Also 25% of cases of acute sinusutis are thought to be related to NTHi. Approximately 75-80% of healthy adults are colonized with NTHi at any one time, and the relationship of colonization to disease is not well understood. Thus, it is anticipated that developing a model for *H. influenzae* colonization will help in developing a vaccine. A strain of NTHi, NTHi 2019 StrR #1, has been developed which has a point mutation and makes this pathogen resistant to streptomycin but susceptible to all commonly used antibiotics. The H-Flu study motivating this work was designed to estimate the human colonization dose 50 and 90 ($HCD50$ and $HCD90$) [63]. $HCD50$ and $HCD90$ are the doses that 50% and 90% of the subjects are expected to be colonized. The NTHi 2019 StrR #1 will be used in future challenge studies as a control to examine the effectiveness of potential vaccines.

2.2 H-Flu Design

Healthy volunteers were recruited and inoculated nasally with various doses of NTHi 2019 StrR #1. Nasal washes and nasopharyngeal swabs were taken and

analyzed for recovery of this pathogen, and then the colonization status can be determined. This was the response of primary interest in a dose response model. Adverse events, such as sore throat, running nose, and headache, were also monitored after inoculation.

Subjects were enrolled in two stages. At the first stage, there were six subjects, who entered the study sequentially. Dose allocation was determined by the outcomes from previous subjects. At this stage, the purpose was to develop a colonization model and understand the dose-response relationship. After observing the outcomes for the first six subjects, estimates of $HCD50$ and $HCD90$ were computed. At the second stage, nine additional subjects were enrolled. The objective was to gain additional experience at the dose above, below and closest to $HCD90$, and to further refine the estimates of the dose response relationship. This thesis focuses on the first stage of the H-Flu design.

2.3 Outcomes and Dose Selection

In this study, a successful outcome ($y_i = 1$) was said to occur if a subject was colonized with NTHi 2019 StrR #1, and if a subject was not colonized then $y_i = 0$. The potential doses for this study ranged from 1.5 to 5.0 in the log scale. In the colony forming units (CFU) scale, the range was $10^{1.5}$ to 10^5 CFU. The starting dose was chosen to be 3.0 since it was considered as safe dose from other studies. The first stage of the study was a modification of the Up-and-Down Method [22, 65]:

$$x_{i+1} = \begin{cases} x_i + 0.5 & \text{if } y_i = 0 \\ x_i - 0.5 & \text{if } y_i = 1 \end{cases}$$

Under the Up-and-Down framework, the dose for the $(i+1)^{th}$ subject would be 0.5 lower if a successful outcome was observed for the i^{th} subject; if failure outcome was observed, dose for the next subject would be 0.5 higher. Due to the limited time available to plan the study, heuristic decisions were made based on scientific and statistical arguments. To increase the probability of existence of maximum likelihood estimator (MLE), the design was modified so that no more than two subjects were allocated to any one dose. The dose allocation for all possible outcomes in the first stage is shown in Figure 2.1 and 2.2.

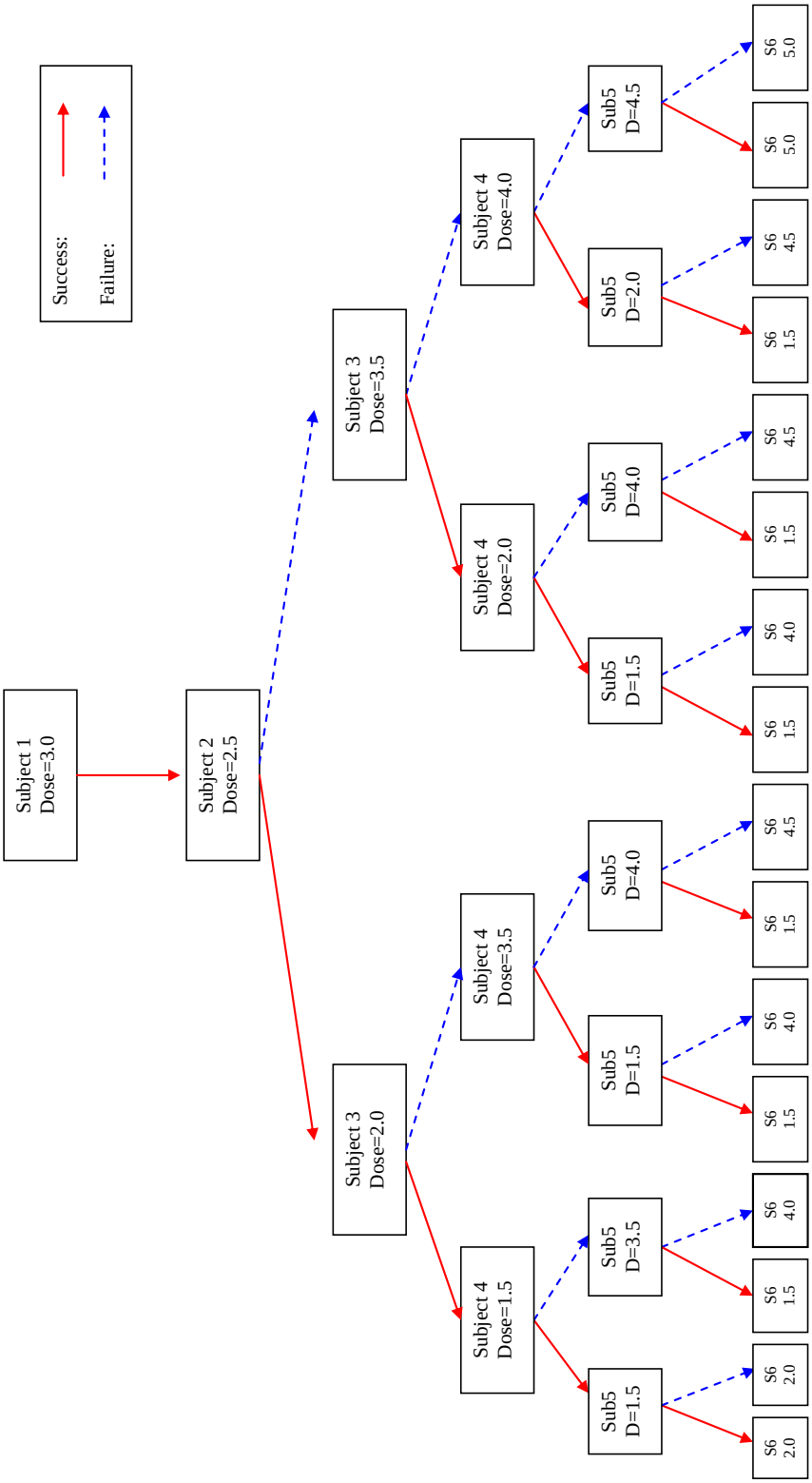

Figure 2.1: Diagram for H-Flu Design (Stage 1), First Subject Success.

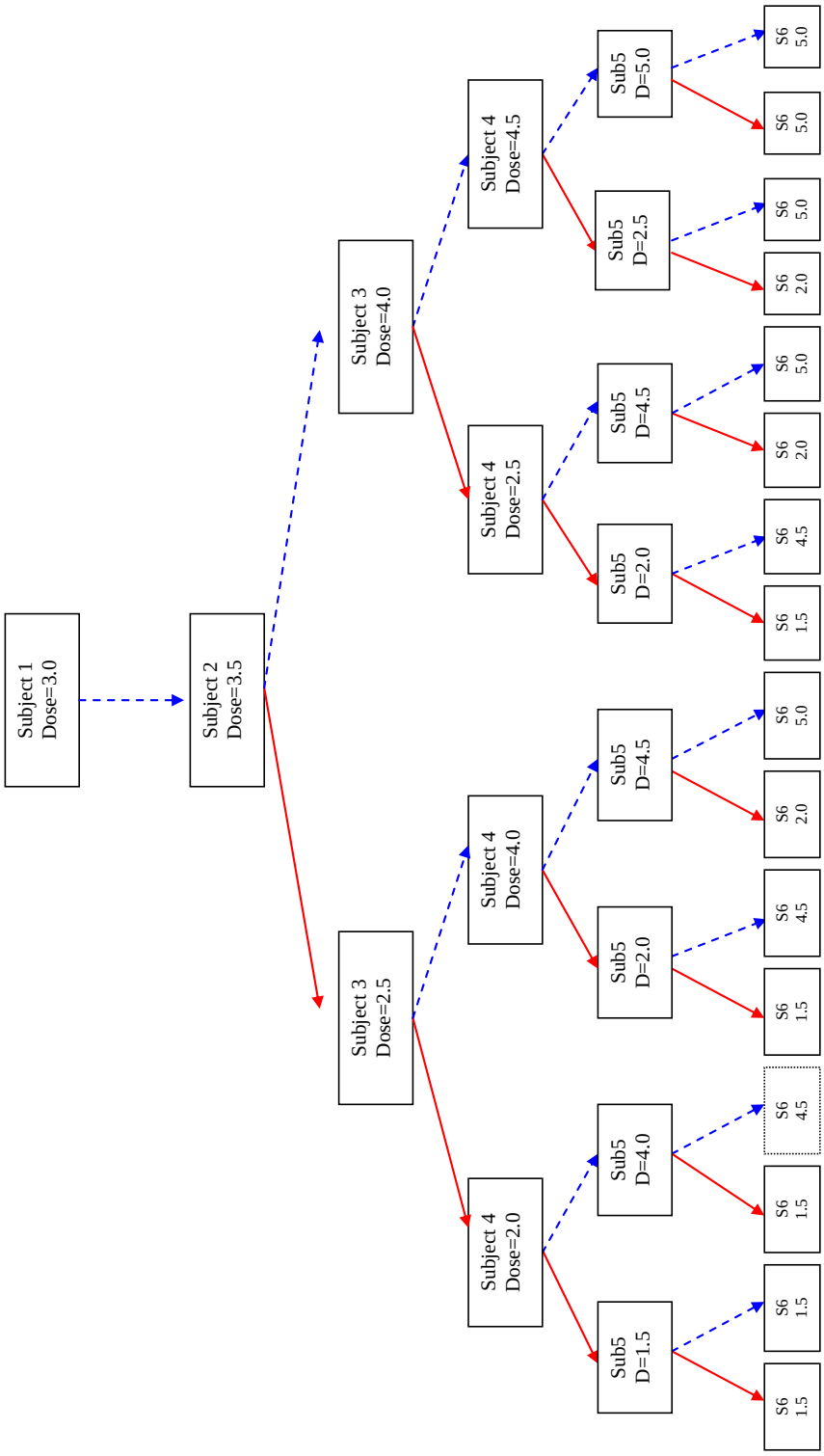

Figure 2.2: Diagram for H-Flu Design (Stage 1, Continued), First Subject Failure.

2.4 Issues Related to the Design

Since healthy subjects were inoculated with bacteria, dose escalation was treated cautiously by allowing the maximum $(i+1)^{th}$ dose to be at most 0.5 higher than the maximum of the first i subjects. There were only six subjects in the first stage of the study, and the small sample size may lead to data with no MLE. Let x^+_{min} and x^+_{max} be the minimum and maximum doses for those with successful outcomes, and x^-_{min} and x^-_{max} be the minimum and maximum doses for those with failure outcomes. To evaluate whether the MLE exists, the dose range from both groups can be examined by the following [49, 65, 1]:

$$(x^+_{min}, x^+_{max}) \cap (x^-_{min}, x^-_{max}) \text{ is empty,}$$

$$\text{or } \; x^+_{min} < x^-_{min} = x^-_{max} < x^+_{max},$$

$$\text{or } \; x^-_{min} < x^+_{min} = x^+_{max} < x^-_{max}.$$

From the expression above, if the dose range for those with successful outcomes does not overlap with dose range for those with failure outcomes, the MLE does not exist.

2.5 Evaluation of the H-Flu Design

To evaluate the design, a simulation study is performed, and both Bayesian and frequentist analysis are implemented. For several fixed parameter values, the sampling distribution of MLE and Bayes estimates are examined. Denote x_i and y_i be dose and outcome $(i = 1, .., 6)$ and the probability of success (colonization) is $\Pr(Y_i = 1) = p_i$. Since the outcome is dichotomous, $Y_i \sim \text{Binomial}(1, p_i)$, and dose-

response relationship can be modeled by a logistic regression, $\text{logit}(p_i) = \alpha + \beta x_i$; hence, $HCD50 = -\alpha/\beta$ and $HCD90 = HCD50 + \log 9/\beta$.

2.5.1 Maximum Likelihood and Bayesian Approaches

For frequentist analysis, the MLEs for α and β are obtained by the Newton-Raphson method, together with estimates and asymptotic standard errors for the $HCD50$ and $HCD90$. For Bayesian analysis, define $\mu = HCD50$ ($\mu \in [1.5, 5]$) and $\delta = HCD90 - HCD50$ ($\delta \in [0.1, 5.1]$), and assume the prior distributions for $\frac{\mu-1.5}{3.5}$ and $\frac{\delta-0.1}{5}$ are beta distributions. More specifically,

$$\frac{\mu - 1.5}{3.5} \sim \text{Beta}(1, 1), \quad \frac{\delta - 0.1}{5} \sim \text{Beta}(0.25, 1), \quad \mu, \delta \text{ independent.}$$

The prior distribution reflects what is believed: the $HCD50$ is somewhere between 1.5 and 5.0, and the $HCD90$ is at least 0.1 log higher, and is at most 10.1. Bayes estimates are the posterior means of $HCD50$ and $HCD90$.

2.5.2 Simulation and Results

Nineteen combinations of true values $(HCD50, HCD90)$ are chosen (Table 2.5.2), and for each combination $1,000$ of (x_i, y_i) are simulated based on the H-Flu and Up-and-Down design. Figure 2.3 and 2.4 illustrate the results from simulation. The probability of no MLE is very high (> 0.4) for H-Flu design, but is even higher in Up-and-Down design (> 0.8). Using a Bayesian approach, posterior means for $HCD50$ and $HCD90$ are always available. The MSEs from Bayesian analysis for most of the combinations are lower than those from frequentist analysis. In some cases, the MSEs from frequentist analysis are lower; however, the MSEs from

frequentist approach are conditional on MLEs being well defined.

Table 2.1: Nineteen Combinations of True $(HCD50, HCD90)$

True $HCD50$	True $HCD90$				
	3.0	3.5	4.0	4.5	5.0
1.5	1	2	3	4	5
2.0	6	7	8	9	10
2.5	11	12	13	14	15
3.0		17	18	19	20

2.6 Conclusions

The simulation results show that a normal approximation to the distribution of the MLE does not hold, which is not surprising because of the small sample size. They also show that Bayesian estimates, incorporating prior information and prior restrictions, are very appealing.

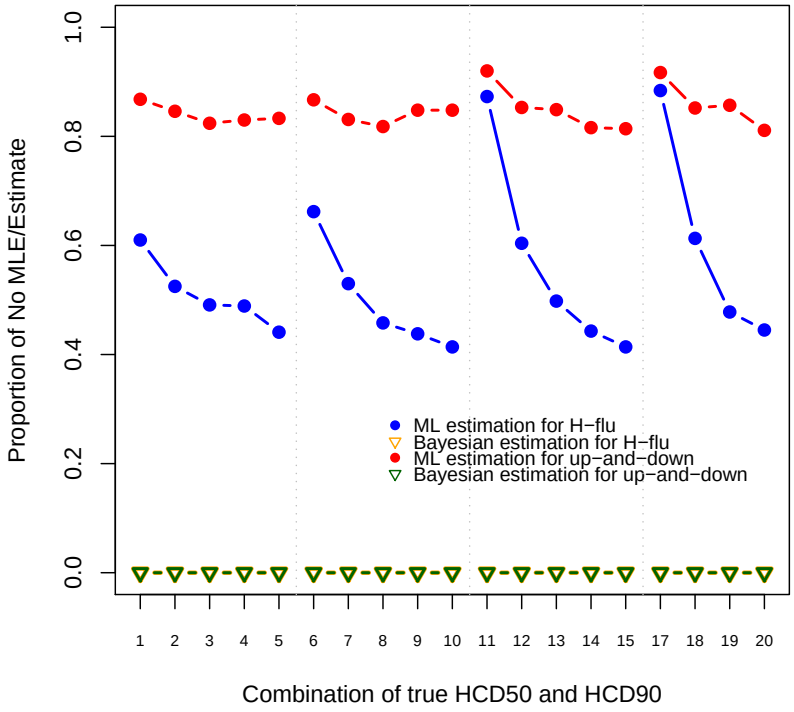

Figure 2.3: Proportion of No MLE/Estimate for H-Flu and Up-and-Down Design

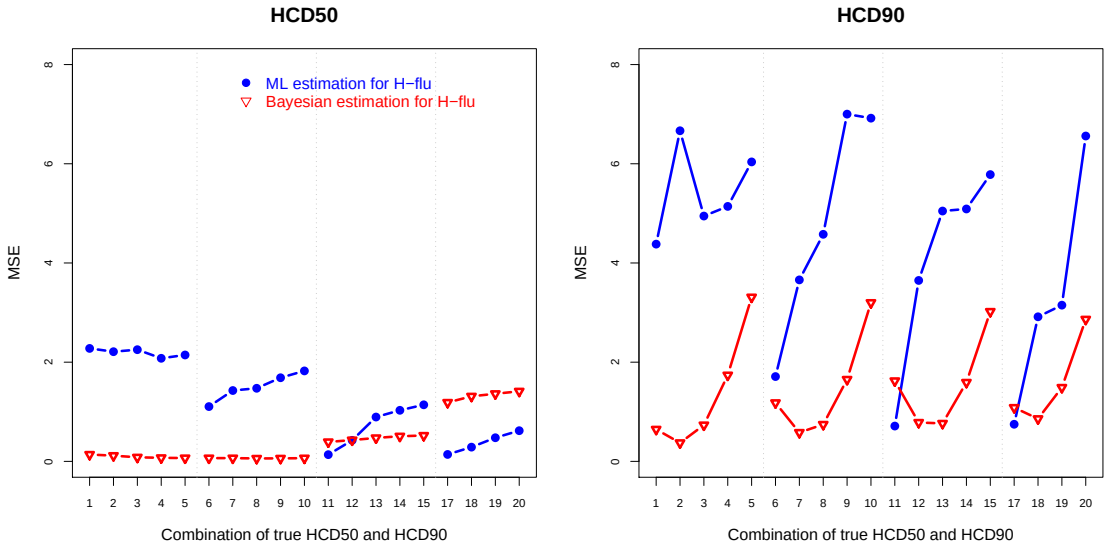

Figure 2.4: MSE for $HCD50$ and $HCD90$ from H-Flu Design (MSE for MLE Conditional on Having MLE Well Defined)

CHAPTER 3
BAYESIAN MYOPIC PROCEDURES FOR MODEL CONSIDERING EFFICACY

3.1 Optimal Bayesian Sequential Designs

Finding the optimal Bayesian sequential design requires backward induction, also known as dynamic programming [21]. However, it is not computationally practical. As the number of subjects increases, the number of scenarios increases exponentially. A more feasible approach is to use a myopic procedure, also called an m-step-look-ahead strategy, to select the optimal dose for each subject. The m-step-look-ahead procedure is to look m steps ahead, and choose the next dose that minimizes the expected loss, assuming there is/are m step(s) left.

We consider one-, two- and three-step look ahead approaches to design. A variation in myopic procedure is m-step-look-ahead with a batch sequential strategy. Here, for a two-step-look-ahead batch design, a batch of two subjects is used and assumed two subjects remaining. A dose strategy for both subjects is implemented using the best two-step sequential procedure.

3.2 Definitions

1. $HCD50$ and $HCD90$: Denote $\mu = HCD50$ and $\gamma = HCD90$, and also denote $\delta = \gamma - \mu$ be the difference between $HCD90$ and $HCD50$.

2. Bounds on $HCD50$ and $HCD9050$ be positive, and at least c_2 and at most $c_1 + c_2$ $(c_1, c_2 > 0)$: Denote L and U are the bounds where $L \leq \mu \leq U$.

3. Dose: $\{d_1, d_2, ..., d_d\} = \mathcal{D}$ denotes the complete set of possible doses.

4. Design point: Denote $\mathbf{x^{(i)}} = (x_1, ..., x_i)$ as a possible design sequence for the first i subjects, $i = 1, ..., I$.

5. Outcomes: $\mathbf{y^{(i)}} = (y_1, ..., y_i)$ indicates a set of outcomes for the first i subjects, where $y_j \in \{0, 1\}$ for $j = 1, ..., i$, $i = 1, ..., I$.

6. History:
$\mathbf{h^{(i)}} = \begin{pmatrix} \mathbf{x^{(i)^T}} \\ \mathbf{y^{(i)^T}} \end{pmatrix}$ denotes a possible design sequence and outcome history up to the i^{th} stage, $i = 1, ..., I$.

7. Allowable doses at stage i conditional on $\mathbf{h^{(i-1)}}$: Denote $\mathcal{D}(\mathbf{h^{(i-1)}})$ to be the subset of $\mathcal{D}$ that is allowable at stage i given $\mathbf{h^{(i-1)}}$, $i = 1, ..., I$.

8. Predictive probability of success and failure at dose d conditional on history up to time $(i-1)$: Denote $p^{(i)}(d, \mathbf{h^{(i-1)}})$ and $q^{(i)}(d, \mathbf{h^{(i-1)}})$ as probabilities of success and failure for the i^{th} subject at dose d given $\mathbf{h^{(i-1)}}$, $i = 1, ..., I$:

$$p^{(i)}(d, \mathbf{h^{(i-1)}}) = \Pr(y_i = 1 | \mathbf{h^{(i-1)}}, x_i = d),$$

$$q^{(i)}(d, \mathbf{h^{(i-1)}}) = \Pr(y_i = 0 | \mathbf{h^{(i-1)}}, x_i = d).$$

9. Loss function conditional on $\mathbf{h^{(i)}}$: Denote $L^{(i)}(\mathbf{h^{(i)}})$ as the loss at the i^{th} stage conditional on $\mathbf{h^{(i)}}$, $i = 1, ..., I$:

$$L^{(i)}(\mathbf{h^{(i)}}) = Var(\mu \mid \mathbf{h^{(i)}}) + Var(\gamma \mid \mathbf{h^{(i)}}).$$

10. Expected loss conditional on $\mathbf{h^{(i-1)}}$ and d_i: Denote $EL^{(i)}(\mathbf{h^{(i-1)}}, d_i)$, $i = 1, ..., I$:

$$\begin{aligned} EL^{(i)}(\mathbf{h^{(i-1)}}, d_i) &= E_{y_i|\mathbf{y^{(i-1)}}} \left\{ L^{(i)}(\mathbf{h^{(i)}}) \right\} \\ &= p^{(i)}(d_i, \mathbf{h^{(i-1)}}) L^{(i)}(\mathbf{h^{(i)}}, \text{ if } y_i = 1) \\ &\quad + q^{(i)}(d_i, \mathbf{h^{(i-1)}}) L^{(i)}(\mathbf{h^{(i)}}, \text{ if } y_i = 0). \end{aligned}$$

11. Overall expected loss conditional on $\mathbf{h^{(I)}}$: $OEL = E_{\mathbf{y^{(I)}}} \left\{ L^{(I)}(\mathbf{h^{(I)}}) \right\}$, is the expected loss conditional on $\mathbf{h^{(I)}}$.

3.3 The Constraint

A constraint is introduced to make dose escalation cautious. The maximum dose increases slowly by constraining to 0.5 higher than the previous maximum. The range of the candidate doses for the $(i+1)^{th}$ subject is:

$$\left[1.5, 2.0, .., \max_{j \leq i} \{x_i\} + 0.5 \right].$$

3.4 Loss Function and the Optimal Dose

Since $HCD50$ and $HCD90$ are the parameters of interest, the loss function is chosen to be the sum of posterior variance of the $HCD50$ and posterior variance of the $HCD90$. Monte Carlo simulations are used to estimate the expected loss. When implementing a m-step-look-ahead procedure, using $m = 1$ as an example, after $(i-1)$ observations, $\mathbf{x^{(i-1)}}$ and $\mathbf{y^{(i-1)}}$ are fixed. By looking one step ahead, all possible outcomes are considered (e.g. in our example, the outcome is colonization status) for each of the candidate dose. For each outcome, the posterior loss is computed. For each candidate dose, the expected loss is computed, and this is to take

the expectation over the distribution of $y_i|\mathbf{y^{(i-1)}}$. The optimal dose for the i^{th} subject conditional on $\mathbf{h^{(i-1)}}$ is the one minimizing the expected loss, $EL^{(i)}(\mathbf{h^{(i-1)}}, d_i)$. That is, $\underset{d_i \in \mathcal{D}(\mathbf{h^{(i-1)}})}{\operatorname{argmin}} \left\{EL^{(i)}(\mathbf{h^{(i-1)}}, d_i)\right\}$. After the optimal dose for the last subject is chosen, OEL is computed by taking expectation on the loss at the I^{th} stage over the distribution of $\mathbf{y^I}$. The lower the OEL, the better the performance of a design. Recall that the optimal design can be found by backward induction, and the OEL from backward induction should be the lowest OEL compare to the myopic procedures.

3.5 The Model

Since the outcome of interest is a binary variable (1 = colonized; 0 = not colonized) and the dose information is available. Logistic regression is used to model the dose-response relationship:

$$\text{logit}\Big[\Pr(Y_i = 1)\Big] = \alpha + \beta x_i.$$

To solve α and β:

$$\begin{cases} \text{logit}(0.5) = \alpha + \beta\mu \\ \text{logit}(0.9) = \alpha + \beta(\gamma). \end{cases}$$

Then

$$\begin{cases} \alpha = -\left(\dfrac{\log 9}{\gamma - \mu}\right)\mu = -\left(\dfrac{\log 9}{\delta}\right)\mu \\ \\ \beta = \dfrac{\log 9}{\gamma - \mu} = \dfrac{\log 9}{\gamma - \mu}. \end{cases}$$

Thus the model can be expressed in terms of dose:

$$\text{logit}\Big[\Pr(Y_i = 1)\Big] = \left(\frac{x_i - \mu}{\delta}\right)\log 9.$$

3.6 Algorithm

3.6.1 Generating the Prior Sample

Define the difference between $HCD50$ and $HCD90$ to be δ, and $\delta = HCD_{90} - HCD_{50} = \gamma - \mu$. Denote $S_i = S_i(\mathbf{h^{(i-1)}}) \in \mathcal{D}_i$ is the set of allowable doses at the i^{th} stage given $\mathbf{h^{(i-1)}}$.

Step 1: Before observing any data, select a random sample of (μ_k, γ_k), $k = 1, 2, ..., N$ from the prior distribution.

Step 2: The prior probabilities of success and failure if the first dose is chosen to be d_j, using Monte Carlo; that is for each (μ_k, γ_k) combination denote

$$p_{jk} = \frac{9^{\frac{d_j - \mu_k}{\delta_k}}}{1 + 9^{\frac{d_j - \mu_k}{\delta_k}}}$$

$$q_{jk} = 1 - p_{jk}.$$

Then the expected prior probability of success at dose d_j is just the average of p_{jk} over $k = 1, ..., N$. Equivalently, let the initial weight, w_{0k} be 1, for all k.

$$p^{(1)}(d_j, h_{j1}) = Pr(y_1 = 1 |\ h_0) = \frac{\sum_k p_{jk}\ w_{0k}}{\sum_k w_{0k}}$$

$$q^{(1)}(d_j, h_{j0}) = Pr(y_1 = 0 |\ h_0) = \frac{\sum_k q_{jk}\ w_{0k}}{\sum_k w_{0k}}$$

3.6.2 One-step-look-ahead Procedure

When selecting the optimal dose for each stage, assuming there is only one step left and the optimal dose is the one that minimize expected loss. For the first stage ($i = 1$), since there is no previous stage, the initial weights, w_{0k}, are identical and equal to 1. At the i^{th} stage ($i \geq 2$), suppose the optimal dose from

previous stage is $x_{i-1} = d_{j^*}$ and $\mathcal{D}(\mathbf{h}^{(i-1)})$ represent possible doses at the i^{th} stage. $S_i = \{s_{il}; l = 1, ..., L_i\}$.

Step 1: If the dose is chosen to be s_{il}, the average probability of success and failure at does l given $\mathbf{h^{(i-1)}}$ are:

$$p^{(i)}(s_{il}, \mathbf{h^{(i-1)}}) = \Pr(y_i = 1 |\ \mathbf{h^{(i-1)}}) = \sum_k p_{lk}\ w_k(h_{r1}(\mathbf{h^{(i-1)}}))$$

$$q^{(i)}(s_{il}, \mathbf{h^{(i-1)}}) = \Pr(y_i = 0 |\ \mathbf{h^{(i-1)}}) = \sum_k q_{lk}\ w_k(h_{r0}(\mathbf{h^{(i-1)}})).$$

Then calculate posterior weights, $w_k(h_{l1}(\mathbf{h^{(i-1)}}))$ for $y_i = 1$ and $w_k(h_{l0}(\mathbf{h^{(i-1)}}))$ for $y_i = 0$.

For $y_i = 1$, let

$$w_k(h_{l1}(\mathbf{h^{(i-1)}})) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{lk'}} \times N.$$

For $y_i = 0$, let

$$w_k(h_{l0}(\mathbf{h^{(i-1)}})) = \frac{w_k(\mathbf{h^{(i-1)}})\ q_{lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ q_{lk'}} \times N.$$

Note that these posterior probabilities are normalized to sum to N over k, to avoid underflow.

Step 2: The posterior means for μ and γ given $\mathbf{h^{(i)}}$ corresponding to either $y_i = 1$ or $y_i = 0$ are

$$E(\mu \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}})\ \mu_k}{\sum_k w_k(\mathbf{h^{(i)}})}$$

$$E(\gamma \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}})\ \gamma_k}{\sum_k w_k(\mathbf{h^{(i)}})},$$

where $\mathbf{h^{(1)}}$ is either h_{l1} or h_{l0} for $d_l \in S_1(h_0) = S_1$. The posterior variances for μ and γ and for $y_i = 1$ or $y_i = 0$ are

$$Var(\mu \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}}) \ \left\{\mu_k - E(\mu \mid \mathbf{h^{(i)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i)}})}$$

$$Var(\gamma \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}}) \ \left\{\gamma_k - E(\gamma \mid \mathbf{h^{(i)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i)}})}.$$

Step 3: The loss if $x_i = s_{il}$ is then

$$L^{(i)}(\mathbf{h}^{(i)}) = Var(\mu \mid \mathbf{h^{(i)}}) + Var(\gamma \mid \mathbf{h^{(i)}}).$$

The expected loss at dose l, looking at one step ahead, is

$$EL^{(i)}(\mathbf{h}^{(i-1)}, s_{il}) = p^{(i)}(s_{il}, \mathbf{h^{(i-1)}})L^{(i)}(\mathbf{h}^{(i-1)}, \text{ if } y_i = 1)$$
$$+ q^{(i)}(s_{il}, \mathbf{h^{(i-1)}})L^{(i)}(\mathbf{h}^{(i-1)}, \text{ if } y_i = 0).$$

Step 4: The best one-step-look-ahead dose for stage i is $x_i = d_{l^*}$, where

$$l^* = \underset{s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})}{\operatorname{argmin}} \left\{EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il})\right\}.$$

3.6.3 Two-step-look-ahead Procedure: Batch of Two and Moving Window

There are two methods examined for implementing a two-step-look-ahead procedure, a batch of two strategy and a moving window strategy. The two-step-look-ahead procedure with batch of two is to select doses for two stages at one time as if there are two more steps left. For example, at the i^{th} stage, by looking two steps ahead, a combination of possible doses and outcome for stage i and $(i+1)$ are obtained. For each batch, the optimal doses are selected using the concept of

backward induction. First calculate the expected loss for stage $(i+1)$ and choose the dose with minimum expected loss. Then move back to the i^{th} stage and calculate the expected loss. The doses for stage i and $(i+1)$ are then selected. For the moving window strategy, the dose for each stage is selected as if there are two more steps left; only the optimal dose for the first of two stages used. Suppose the dose from the $(i-1)^{th}$ is $x_{i-1} = d_{j^*}$, and suppose $s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})$ and $s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})$ represent candidate doses for stage i and $i+1$.

At the i^{th} stage For $y_{i-1} = 1$ and $y_{i-1} = 0$:

Step 1: If the dose is chosen to be s_{il}, the average of probabilities of success and failure at dose l given $\mathbf{h^{(i-1)}}$:

$$p^{(i)}(s_{il}, \mathbf{h^{(i-1)}}) = \Pr(y_i = 1 |\ \mathbf{h^{(i-1)}}) = \sum_k p_{lk}\ w_k(h_{l1}(\mathbf{h^{(i-1)}}))$$

$$q^{(i)}(s_{il}, \mathbf{h^{(i-1)}}) = \Pr(y_i = 0 |\ \mathbf{h^{(i-1)}}) = \sum_k q_{lk}\ w_k(h_{l0}(\mathbf{h^{(i-1)}})).$$

Then calculate posterior weight $w_k(h_{l1}(\mathbf{h^{(i-1)}}))$ for $y_i = 1$ and $w_k(h_{l0}(\mathbf{h^{(i-1)}}))$ for $y_i = 0$.

For $y_i = 1$,

$$w_k(h_{l1}(\mathbf{h^{(i-1)}})) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{lk}}{\sum_k w_{k'}(\mathbf{h^{(i-1)}})\ p_{lk'}} \times N$$

For $y_i = 0$,

$$w_k(h_{l0}(\mathbf{h^{(i-1)}})) = \frac{w_k(\mathbf{h^{(i-1)}})\ q_{lk}}{\sum_k w_{k'}(\mathbf{h^{(i-1)}})\ q_{lk'}} \times N$$

At the $(i+1)^{th}$ stage

Step 2: If the dose is chosen to be $s_{(i+1)g}$, the average of probabilities of success and failure at dose g given $\mathbf{h^{(i)}}$:

$$p^{(i+1)}(s_{(i+1)g}, \mathbf{h^{(i)}}) = \Pr(y_{i+1} = 1 |\ \mathbf{h^{(i)}}) = \sum_k p_{gk}\ w_k(\mathbf{h^{(i)}})$$

$$q^{(i+1)}(s_{(i+1)g}, \mathbf{h^{(i)}}) = \Pr(y_{i+1} = 0 |\ \mathbf{h^{(i)}}) = \sum_k q_{gk}\ w_k(\mathbf{h^{(i)}}).$$

Then calculate posterior weights $w_k(h_{g1}(\mathbf{h^{(i)}}))$ for $y_{i+1} = 1$ and $w_k(h_{g0}(\mathbf{h^{(i)}}))$ for $y_{i+1} = 0$.

For $y_{i+1} = 1$,

$$w_k(h_{g1}(\mathbf{h^{(i)}})) = \frac{w_k(\mathbf{h^{(i)}})\ p_{gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{gk'}} \times N$$

For $y_{i+1} = 0$,

$$w_k(h_{g0}(\mathbf{h^{(i)}})) = \frac{w_k(\mathbf{h^{(i)}})\ q_{gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ q_{gk'}} \times N$$

Step 3: The posterior means for μ and γ given $\mathbf{h^{(i+1)}} = h_{g1}(\mathbf{h^{(i)}})$ for $y_{i+1} = 1$, or $h_{g0}(\mathbf{h^{(i)}})$ for $y_{i+1} = 0$ are:

$$E(\mu \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \mu_k}{\sum_k w_k(\mathbf{h^{(i+1)}})}$$

$$E(\gamma \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \gamma_k}{\sum_k w_k(\mathbf{h^{(i+1)}})},$$

and posterior variances for μ and γ for $y_{i+1} = 1$ or $y_{i+1} = 0$ are:

$$Var(\mu \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \left\{\mu_k - E(\mu \mid \mathbf{h^{(i+1)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+1)}})}$$

$$Var(\gamma \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \left\{\gamma_k - E(\gamma \mid \mathbf{h^{(i+1)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+1)}})}$$

Step 4: The loss if $x_{i+1} = s_{(i+1)g}$ for each possible outcome is:

$$L^{(i+1)}(\mathbf{h}^{(i+1)}) = Var(\mu\ |\mathbf{h}^{(\mathbf{i+1})}) + Var(\gamma\ |\mathbf{h}^{(\mathbf{i+1})})$$

for $y_{i+1} = 1$ or $y_{i+1} = 0$.

Step 5: The expected loss at dose g is then

$$EL^{(i+1)}(\mathbf{h}^{(i+1)}, s_{(i+1)g}) = p^{(i+1)}(s_{(i+1)g}, \mathbf{h}^{(\mathbf{i})})\ L^{(i+1)}(\mathbf{h}^{(i+1)},\ \text{if } y_{i+1} = 1)$$
$$+ q^{(i+1)}(s_{(i+1)g}, \mathbf{h}^{(\mathbf{i})})\ L^{(i+1)}(\mathbf{h}^{(i+1)},\ \text{if } y_{i+1} = 0).$$

Selecting the best dose

Step 6: At stage $(i+1)$:

For each s_{il}, the minimum expected loss at stage $(i+1)$ is:

For $y_i = 1$

$$g_1^* = g_1^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h}^{(\mathbf{i})})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h}^{(\mathbf{i})}, s_{(i+1)g}) \right\}.$$

For $y_i = 0$

$$g_0^* = g_0^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h}^{(\mathbf{i})})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h}^{(\mathbf{i})}, s_{(i+1)g}) \right\}.$$

Step 7: At stage i:

For each s_{il} the expected loss is

$$EL^{(i)}(\mathbf{h}^{(\mathbf{i-1})}, s_{il}) = p^{(i)}(s_{il}, \mathbf{h}^{(\mathbf{i-1})})\ EL^{(i+1)}(\mathbf{h}^{(\mathbf{i})}, s_{(i+1)g_1^*})$$
$$+ q^{(i)}(s_{il}, \mathbf{h}^{(\mathbf{i-1})})\ EL^{(i+1)}(\mathbf{h}^{(\mathbf{i})}, s_{(i+1)g_0^*}).$$

The optimal two-step-look-ahead dose for the stage i is $x_i = d_{l^*}$, where

$$l^* = \underset{s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})}{\operatorname{argmin}} \left\{ EL^{(i)}(\mathbf{h^{(i-1)}} s_{il}) \right\}.$$

For batch method, the doses for stage $(i+1)$ are $d_{g_1^*}$ and $d_{g_0^*}$, and the dose for stage i is d_{l^*}. For moving-window method, dose for stage i is d_{l^*}, after which a new two step calculation is done for $(i+1)$ and $(i+2)$, and the dose for stage $(i+1)$ is implemented.

3.6.4 Three-step-look-ahead Procedure: Moving Window

Three-step-look-ahead procedure is the one that for each stage, it is treated as if there are three steps left. The moving window method is applied. For each stage, only one optimal dose, $x_i = d_{l^*}$, is selected. Suppose the dose from the $(i-1)^{th}$ is $x_{i-1} = d_{j^*}$, and suppose $s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})$, $s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})$ and $s_{(i+2)f} \in \mathcal{D}(\mathbf{h^{(i+1)}})$ represent candidate doses for stage i, $(i+1)$ and $(i+2)$.

At the i^{th} stage

Step 1: If the dose is chosen to be s_{il}, the average of probabilities of success and failure at dose l given $\mathbf{h^{(i-1)}}$:

$$p^{(i)}(s_{il}, \mathbf{h^{(i-1)}}) = \Pr(y_i = 1 |\ \mathbf{h^{(i-1)}}) = \sum_k p_{lk}\ w_k(h_{l1}(\mathbf{h^{(i-1)}}))$$

$$q^{(i)}(s_{il}, \mathbf{h^{(i-1)}}) = \Pr(y_i = 0 |\ \mathbf{h^{(i-1)}}) = \sum_k q_{lk}\ w_k(h_{l0}(\mathbf{h^{(i-1)}})).$$

Then calculate posterior weights $w_k(\mathbf{h^{(i)}})$ for $y_i = 1$ and $w_k(\mathbf{h^{(i)}})$ for $y_i = 0$.

For $y_i = 1$,

$$w_k(h_{l1}(\mathbf{h^{(i-1)}})) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{lk'}} \times N$$

For $y_i = 0$,

$$w_k(h_{l0}(\mathbf{h^{(i-1)}})) = \frac{w_k(\mathbf{h^{(i-1)}})\ q_{lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ q_{lk'}} \times N$$

At the $(i+1)^{th}$ stage

Step 2: If the dose is chosen to be $s_{(i+1)g}$, the average of probabilities of success and failure at dose g given $\mathbf{h^{(i)}}$:

$$p^{(i+1)}(s_{(i+1)g}, \mathbf{h^{(i)}}) = \Pr(y_{i+1} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{gk}\ w_k(h_{g1}(\mathbf{h^{(i)}}))$$

$$q^{(i+1)}(s_{(i+1)g}, \mathbf{h^{(i)}}) = \Pr(y_{i+1} = 0|\ \mathbf{h^{(i)}}) = \sum_k q_{gk}\ w_k(h_{g0}(\mathbf{h^{(i)}})).$$

Then calculate posterior weights $w_k(h_{g1}(\mathbf{h^{(i)}}))$ for $y_{i+1} = 1$ and $w_k(h_{g0}(\mathbf{h^{(i)}}))$ for $y_{i+1} = 0$.

For $y_{i+1} = 1$,

$$w_k(h_{g1}(\mathbf{h^{(i)}})) = \frac{w_k(\mathbf{h^{(i)}})\ p_{gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{gk'}} \times N$$

For $y_{i+1} = 0$,

$$w_k(h_{g0}(\mathbf{h^{(i)}})) = \frac{w_k(\mathbf{h^{(i)}})\ q_{gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ q_{gk'}} \times N$$

At the $(i+2)^{th}$ stage

Step 3: If the dose is chosen to be $s_{(i+2)f}$, the average of probabilities of success and failure at dose f given $\mathbf{h^{(i+1)}}$:

$$p^{(i+2)}(s_{(i+2)f}, \mathbf{h^{(i+1)}}) = \Pr(y_{i+2} = 1|\ \mathbf{h^{(i+1)}}) = \sum_k p_{fk}\ w_k(h_{f1}(\mathbf{h^{(i+1)}}))$$

$$q^{(i+2)}(s_{(i+2)f}, \mathbf{h^{(i+1)}}) = \Pr(y_{i+2} = 0|\ \mathbf{h^{(i+1)}}) = \sum_k q_{fk}\ w_k(h_{f0}(\mathbf{h^{(i+1)}})).$$

Then calculate posterior weights $w_k(h_{f1}(\mathbf{h^{(i+1)}}))$ for $y_{i+2} = 1$ and $w_k(h_{f0}(\mathbf{h^{(i+1)}}))$ for $y_{i+2} = 0$.

For $y_{i+2} = 1$,

$$w_k(h_{f1}(\mathbf{h^{(i+1)}})) = \frac{w_k(\mathbf{h^{(i+1)}})\ p_{fk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i+1)}})\ p_{fk'}} \times N$$

For $y_{i+2} = 0$,

$$w_k(h_{f1}(\mathbf{h^{(i+1)}})) = \frac{w_k(\mathbf{h^{(i+1)}})\ q_{fk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i+1)}})\ q_{fk'}} \times N$$

Step 4: The posterior means for μ and γ given $\mathbf{h^{(i+2)}} = h_{f1}(\mathbf{h^{(i+1)}})$ or $h_{f0}(\mathbf{h^{(i+1)}})$ for $y_{i+2} = 1$ or $y_{i+2} = 0$ are:

$$E(\mu \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \mu_k}{\sum_k w_k(\mathbf{h^{(i+2)}})}$$

$$E(\gamma \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \gamma_k}{\sum_k w_k(\mathbf{h^{(i+2)}})},$$

and posterior variances for μ and γ for $y_{i+2} = 1$ or $y_{i+2} = 0$ are:

$$Var(\mu \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \left\{\mu_k - E(\mu \mid \mathbf{h^{(i+2)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+2)}})}$$

$$Var(\gamma \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \left\{\gamma_k - E(\gamma \mid \mathbf{h^{(i+2)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+2)}})}$$

Step 5: The loss if $x_{i+2} = s_{(i+2)f}$ for each possible outcome is:

$$L^{(i+2)}(\mathbf{h}^{(i+2)}) = Var(\mu\ |\mathbf{h^{(i+2)}}) + Var(\gamma\ |\mathbf{h^{(i+2)}})$$

for $y_{i+2} = 1$ or $y_{i+2} = 0$.

Step 6: The expected loss at dose f is then

$$EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f}) = p^{(i+2)}(s_{(i+2)f}, \mathbf{h^{(i+1)}})\ L^{(i+2)}(\mathbf{h^{(i+2)}},\ \text{if } y_{i+2}=1)$$
$$+ q^{(i+2)}(s_{(i+2)f}, \mathbf{h^{(i+1)}})\ L^{(i+2)}(\mathbf{h^{(i+2)}},\ \text{if } y_{i+2}=0)$$

Selecting the best dose

Step 7: At stage $(i+2)$:

For each s_{1l} and $s_{(i+1)g}$, the minimum expected loss is: For $y_{i+1}=1$

$$f_1^* = f_1^*(s_{il}, s_{(i+1)g}) = \underset{s_{(i+2)f}\in\mathcal{D}(\mathbf{h^{(i+1)}})}{\operatorname{argmin}} \left\{EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f})\right\}.$$

For $y_{i+1}=0$

$$f_0^* = f_0^*(s_{il}, s_{(i+1)g}) = \underset{s_{(i+2)f}\in\mathcal{D}(\mathbf{h^{(i+1)}})}{\operatorname{argmin}} \left\{EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f})\right\}.$$

Step 8: At stage $(i+1)$:

For each s_{1l}, the expected loss for $y_{i+1}=1$ or $y_{i+1}=0$ is

$$EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) = p^{(i+1)}(s_{(i+1)g}, \mathbf{h^{(i)}})EL^{(i+2)}(\mathbf{h^{(i)}}, s_{(i+2)f_1^*})$$
$$+ q^{(i+1)}(s_{(i+1)g}, \mathbf{h^{(i)}})EL^{(i+2)}(\mathbf{h^{(i)}}, s_{(i+2)f_0^*}).$$

The minimum expected loss are:

For $y_i=1$:

$$g_1^* = g_1^*(s_{il}) = \underset{s_{(i+1)g}\in\mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g})\right\}.$$

For $y_i=0$:

$$g_0^* = g_0^*(s_{il}) = \underset{s_{(i+1)g}\in\mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g})\right\}.$$

Step 9: At stage i: For each s_{il} the expected loss is

$$EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il}) = p^{(i)}(s_{il}, \mathbf{h^{(i-1)}})\ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{i+1,g_1^*}) + q^{(i)}(s_{il}, \mathbf{h^{(i-1)}})\ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{i+1,g_0^*}).$$

The best three-step-look-ahead dose for the stage i is $x_i = d_{l^*}$, where

$$l^* = \underset{s_{il} \in \mathcal{D}(\mathbf{h}^{(i-1)})}{\operatorname{argmin}} \left\{ EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il}) \right\}.$$

3.7 Summary

These algorithms will be applied in Chapter 4.

CHAPTER 4
SIMULATION RESULTS FROM MODEL CONSIDERING EFFICACY

4.1 Settings for Simulation

Designs are simulated based on the four strategies (one-, two-, and three-step-look-ahead procedures), and on different sets of prior distribution. A sample of a thousand prior values are generated for the simulation. The overall expected loss is the measure of performance of a design. To compare the performance of different strategies, the ratio of overall expected loss is computed using one-step-look-ahead procedure as a benchmark. If the ratio is less than one, the strategy performs better than the one-step-look-ahead procedure. If the ratio is greater than one, the strategy performs worse than the one-step-look-ahead procedure. Since the optimal design can be only found by backward induction, if it could be calculated, the ratio of the optimal design to the one-step-look-ahead procedure should be less than one, and is the lowest among all possible procedures.

4.1.1 Prior Distributions

Let v and z be the linear transformation of μ and δ. Assume that μ and δ are bounded. Define c_1 and c_2 so that γ is greater than μ by at least c_2 and is also greater than μ by at most $c_1 + c_2$. Then $\delta = c_1 z + c_2$, $(c_1, c_2 > 0)$, and if $L \leq \mu \leq U$,

$$\mu = (U - L)v + L,$$

$$\gamma = \delta + \mu = (c_1 z + c_2) + \mu,$$

and v and z are assumed to follow independent Beta distributions. Three prior distributions will be applied in the simulations.

1. Primary prior distribution (PP):

$$v \sim \text{Beta}(5, 5),\ z \sim \text{Beta}(1, 6)$$

This reflects the approximate knowledge of the investigators.

2. More precise prior distribution (MP):

$$v \sim \text{Beta}(22, 22),\ z \sim \text{Beta}(4, 26)$$

3. Less precise prior distribution (LP):

$$v \sim \text{Beta}(7/8, 7/8),\ z \sim \text{Beta}(0.15, 0.85)$$

The expectation of v and z are identical across prior distributions, and the standard deviation from MP is half of PP, and the standard deviation from LP is twice the standard deviation from PP. The graphical presentation of these prior distributions are shown in Figure 4.1.

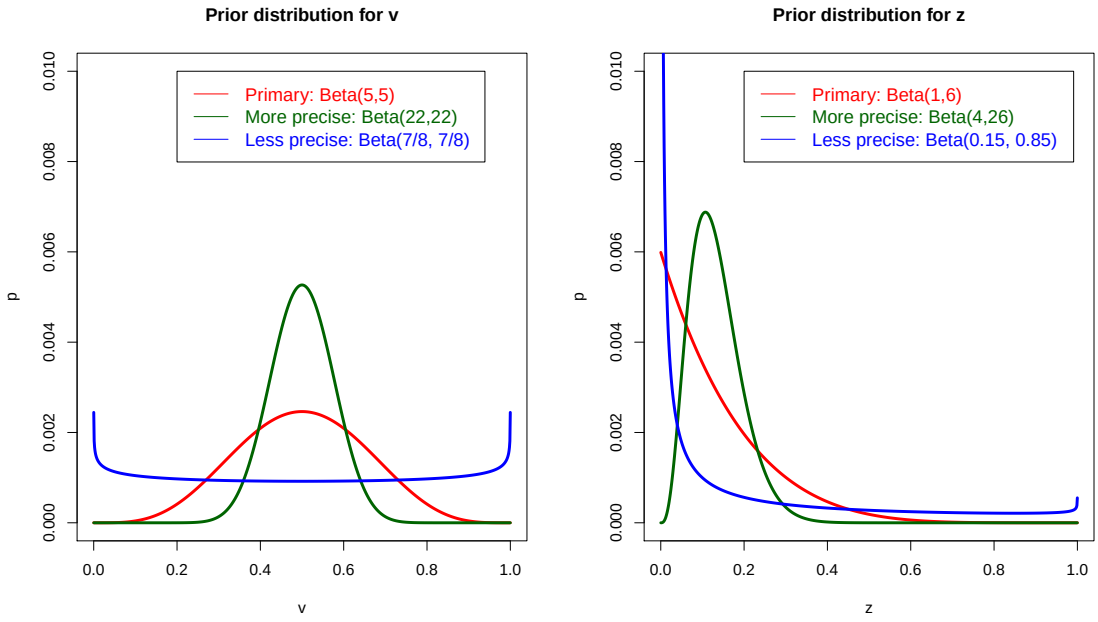

Figure 4.1: Three Sets of Prior Distribution for v and z

4.1.2 Generating the Prior Sample

Select a sample of $1,000$ independent and identically distributed values (v_k, z_k) $(k = 1, ..., N)$ from the prior distribution. Recall that in the H-Flu study, the candidate doses range from 1.5 to 5.0, and $HCD90$ should be at least 0.1 greater than $HCD50$ and the largest possible value of $HCD90$ should not be greater than 10.1. Therefore,

$$\mu_k = 3.5v_k + 1.5, \qquad \delta_k = 5z_k + 0.1.$$

4.2 Results

The overall expected loss for each strategy and for each prior distribution is shown in Figure 4.2. As the number of subjects increases, the overall expected loss decreases. Furthermore, the more precise the prior distribution, the lower the expected loss. The results from the PP and MP show that there is not much difference in the performance of these strategies. Figure 4.3 presents the ratio of overall expected loss. It is observed that using the less precise prior distribution, LP, two-step batch sequential, two-step- and three-step-look-ahead procedure perform noticeably better than one-step strategy.

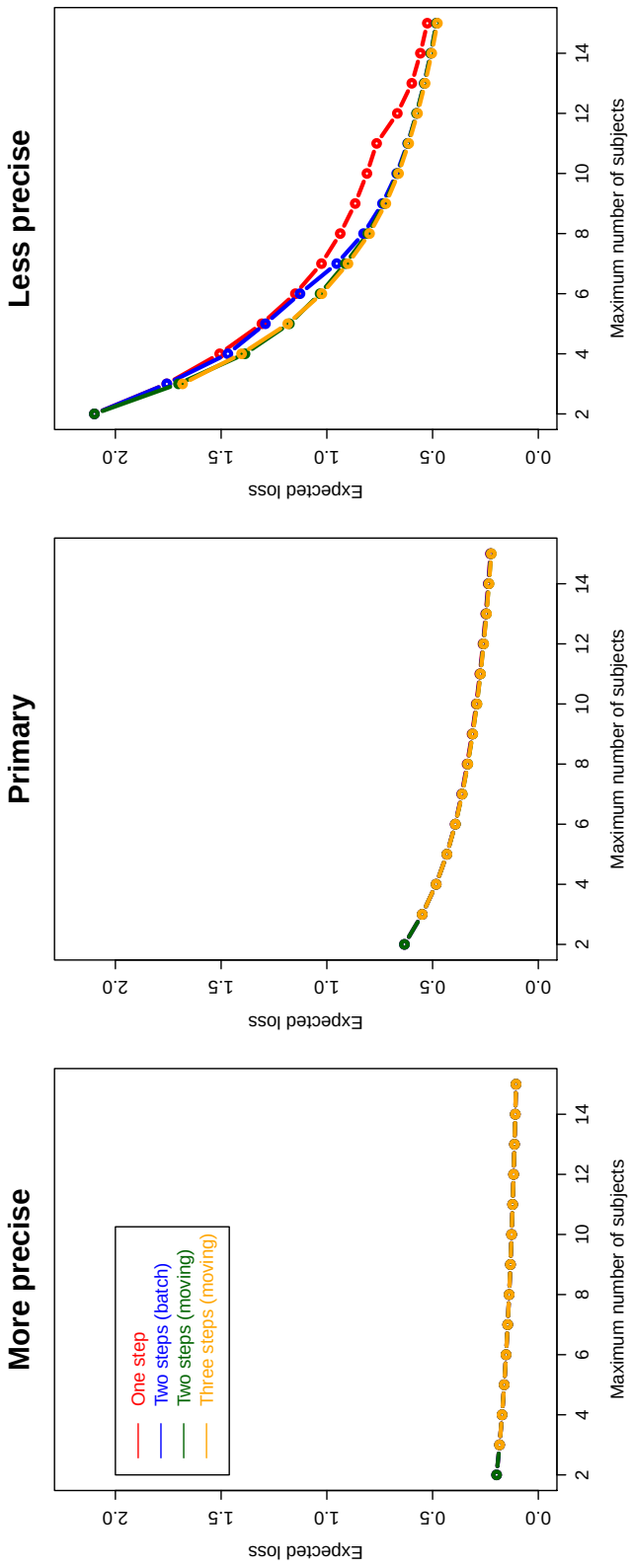

Figure 4.2: Expected Loss for Different Strategies and Prior Distributions

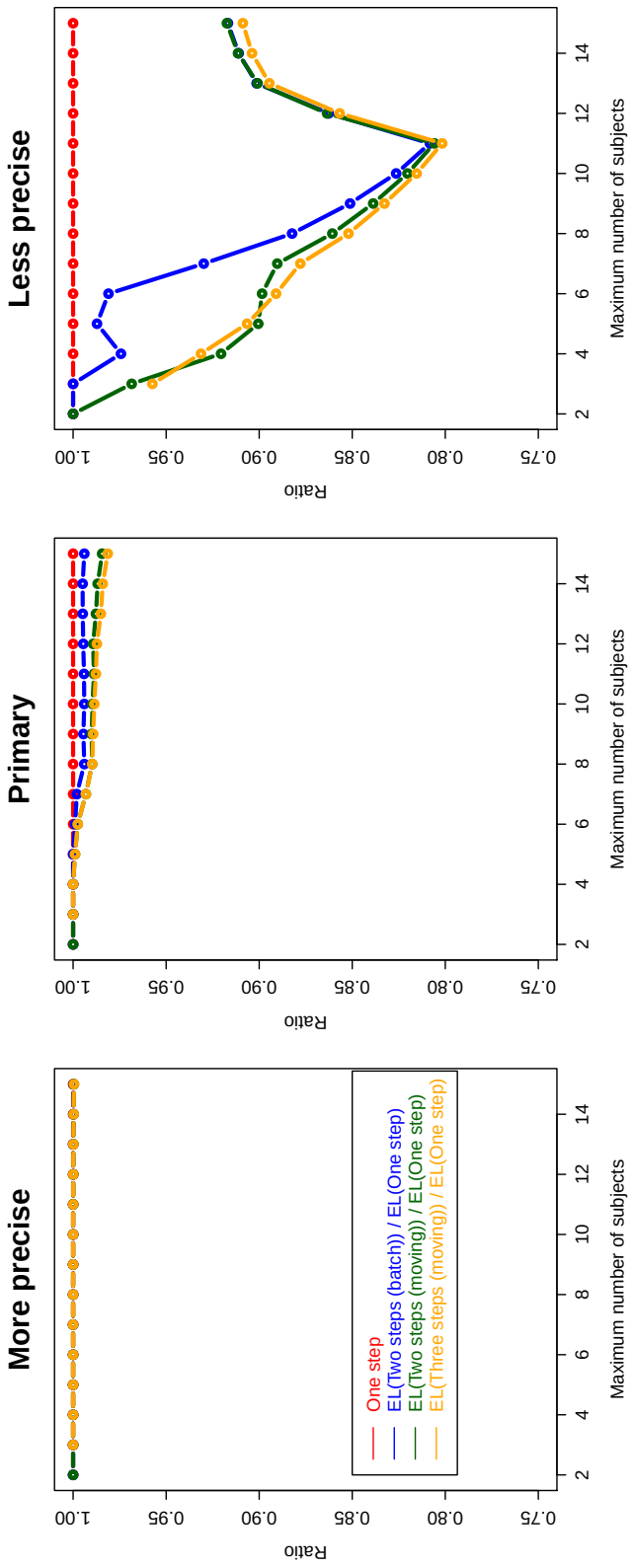

Figure 4.3: Expected Loss for Different Strategies and Prior Distributions

4.3 Discussion

For the more precise prior distribution, MP, the improvement in using the m-step-look-ahead over one-step-look-ahead strategy is very small. In contrast, it appears that there is much more to gain when the prior distribution is less precise, as in LP.

CHAPTER 5
THE NEW MODEL CONSIDERING EFFICACY AND TOXICITY

5.1 Introduction

Several approaches to design have been proposed for models with a multinomial response. Heise and Myers [32] apply a Gumbel bivariate logistic model allowing correlation between efficacy and toxicity. For modeling ordinal responses, proportional odds model are discussed by Thall and Russell [55]. Fan and Chaloner [26] present optimal design for trinomial responses with a continuation-ratio model. The advantage of continuation-ratio model is that the assumption of proportional odds is relaxed, and a continuation-ratio model often has a better fit compared to the proportional odds model [26]. Thall and Cook [53] apply both proportional odds models and continuation-ratio models for trinomial responses considering trade-off between efficacy and toxicity.

In the setting of the H-Flu study, not only the dose-response relationship, but also the toxicity-response relationship is of interest. In addition, the toxicity-response relationship may differ between colonized and non-colonized subjects, and it is reasonable to assume that colonization increases the probability of toxicity. Considering both efficacy (yes/no) and toxicity (yes/no), the outcome can be thought of as a multinomial variable with four responses. A new model taking dose-response and toxicity-response relationship into account is proposed here.

5.2 The New Model for Multinomial Responses

Denote C be colonization status (yes=1, no=0) and A be adverse event (yes=1, no=0), and $\mathbf{y_i} = (y_{i1}, y_{i2}, y_{i3}, y_{i4})$ be the indicator for the i^{th} subject with the following colonization-toxicity outcome:

$$\begin{cases} y_{i1} = 1 & \text{if } A = 0 \text{ and } C = 0, \\ y_{i2} = 1 & \text{if } A = 1 \text{ and } C = 0, \\ y_{i3} = 1 & \text{if } A = 0 \text{ and } C = 1, \\ y_{i4} = 1 & \text{if } A = 1 \text{ and } C = 1. \end{cases}$$

The outcome follows a multinomial distribution, $\mathbf{y_i} \sim \text{Multinomial}(1, p_{i1}, p_{i2}, p_{i3}, p_{i4})$, $\sum_{j=1}^{4} p_{ij} = 1$ and

$$\Pr(\mathbf{y_1}, ..., \mathbf{y_n}) = \prod_{i=1}^{n} \prod_{j=1}^{4} p_{ij}^{y_{ij}}.$$

Also, let

$$p_1(x_i) = \Pr(C = 0, A = 0|x_i),\ p_2(x_i) = \Pr(C = 0, A = 1|x_i),$$

$$p_3(x_i) = \Pr(C = 1, A = 0|x_i),\ p_4(x_i) = \Pr(C = 1, A = 1|x_i),$$

and $p_5(x_i) = \Pr(C = 1|x_i)$. A new model based on dose-response and toxicity-response relationship is given below, subjects are assumed to be independent:

$$\text{logit}\Big\{ \Pr(C = 1|x_i) \Big\} = \log\left(\frac{p_5(x_i)}{1 - p_5(x_i)} \right) = a_1 + b_1 x_i \tag{5.1}$$

$$\text{logit}\Big\{ \Pr(A = 1|C = 0, x_i) \Big\} = \log\left(\frac{p_2(x_i)}{p_1(x_i)} \right) = a_2 + b_2 x_i \tag{5.2}$$

$$\text{logit}\Big\{ \Pr(A = 1|C = 1, x_i) \Big\} = \log\left(\frac{p_4(x_i)}{p_3(x_i)} \right) = a_3 + b_3 x_i. \tag{5.3}$$

Equation 5.1 models the dose-response relationship, and equations 5.2 and 5.3 model the toxicity-response relationship for non-colonized and colonized groups, respectively. From this model, $p_1(x_i), p_2(x_i), p_3(x_i)$, and $p_4(x_i)$ can be calculated as follows:

$$p_1(x_i) = \Pr(A = 0|C = 0, x_i) \Pr(C = 0|x_i) = \left(\frac{1}{1 + \exp(a_2 + b_2 x_i)}\right)(1 - p_5(x_i)),$$

$$p_2(x_i) = \Pr(A = 1|C = 0, x_i) \Pr(C = 0|x_i) = \left(\frac{\exp(a_2 + b_2 x_i)}{1 + \exp(a_2 + b_2 x_i)}\right)(1 - p_5(x_i)),$$

$$p_3(x_i) = \Pr(A = 0|C = 1, x_i) \Pr(C = 1|x_i) = \left(\frac{1}{1 + \exp(a_3 + b_3 x_i)}\right) p_5(x_i),$$

$$p_4(x_i) = \Pr(A = 1|C = 1, x_i) \Pr(C = 1|x_i) = \left(\frac{\exp(a_3 + b_3 x_i)}{1 + \exp(a_3 + b_3 x_i)}\right) p_5(x_i),$$

and $p_5(x_i) = \Pr(C = 1|x_i) = \dfrac{\exp(a_1 + b_1 x_i)}{1 + \exp(a_1 + b_1 x_i)}$.

5.3 Parametrization on the Model

Sometimes it is easier to interpret the model in terms of dose rather than the intercept and slope, $a_i, b_i, i = 1, 2, 3$. Let $DANC50$ and $DANC90$ be the dose that will cause 50% and 90% of non-colonized subjects to experience adverse events, and denote $\mu_2 = DANC50$, $\gamma_2 = DANC90$ and $\delta_2 = \gamma_2 - \mu_2$. Let $DAC50$ and $DAC90$ be the dose responsible for 50% and 90% of colonized subjects with adverse events, and denote $\mu_3 = DAC50$, $\gamma_3 = DAC90$ and $\delta_3 = \gamma_3 - \mu_3$. The model can

be expressed in terms of these doses as follows:

$$\log\left(\frac{p_5(x_i)}{1-p_5(x_i)}\right) = a_1 + b_1 x_i = \left(\frac{x_i-\mu_1}{\delta_1}\right)\log 9 \tag{5.4}$$

$$\log\left(\frac{p_2(x_i)}{p_1(x_i)}\right) = a_2 + b_2 x_i = \left(\frac{x_i-\mu_2}{\delta_2}\right)\log 9 \tag{5.5}$$

$$\log\left(\frac{p_4(x_i)}{p_3(x_i)}\right) = a_3 + b_3 x_i = \left(\frac{x_i-\mu_3}{\delta_3}\right)\log 9. \tag{5.6}$$

The relationship between dose, intercept and slope is:

$$\mu_1 = -\left(\frac{a_1\delta_1}{\log 9}\right),\ \delta_1 = \frac{\log 9}{b_1},$$

$$\mu_2 = -\left(\frac{a_2\delta_2}{\log 9}\right),\ \delta_2 = \frac{\log 9}{b_2},$$

$$\mu_3 = -\left(\frac{a_3\delta_3}{\log 9}\right),\ \delta_3 = \frac{\log 9}{b_3}.$$

5.4 Assumptions

There are three assumptions to this model:

A1: Monotone increasing dose-response relationship: $b_1 > 0$.

It is assumed that as dose increases, the probability of colonization increases, that is $\Pr(C = 1|x) = p_5(x)$ is a monotone increasing function. This is achieved by the assumption $b_1 > 0$, which is equivalent to the assumption $\delta_1 > 0$. Since the dose, $HCD50$, should be a positive value, $\mu_1 > 0$, and this is equivalent to $a_1 < 0$.

A2: Monotone increasing conditional toxicity-response relationship: $b_2, b_3 > 0$.

The conditional probability of toxicity given no colonization, $\Pr(A = 1|C =$

$0, x)$, and the conditional probability of toxicity given colonization, $\Pr(A = 1|C = 1, x)$, are assumed to increase with dose. This is achieved by the assumptions $b_2, b_3 > 0$, which is equivalent to $\delta_2, \delta_3 > 0$. Since $DANC50$ and $DAC50$ should be positive, and $a_2, a_3 < 0$.

A3: For any dose, the conditional probability of toxicity given non colonization is less than or equal to the conditional probability of toxicity given colonization: $a_2 \leq a_3$, $b_2 \leq b_3$.

$$\Pr(A = 1|C = 0, x) \leq \Pr(A = 1|C = 1, x) \text{ for all } x.$$

That is

$$\frac{\exp(a_2 + b_2 x)}{1 + \exp(a_2 + b_2 x)} \leq \frac{\exp(a_3 + b_3 x)}{1 + \exp(a_3 + b_3 x)} \qquad \text{for all } x$$

This implies, $\exp(a_2 + b_2 x) \leq \exp(a_3 + b_3 x)$ for all x. For $x = 0$, this implies $a_2 \leq a_3$ and for x large, this implies $b_2 \leq b_3$. Conversely if $a_2 \leq a_3$ and $b_2 \leq b_3$ then the relationship holds. Note that this assumption also implies $\delta_3 \leq \delta_2$.

A consequence of these three assumptions is that there is a non-decreasing in marginal toxicity-response relationship:

From A2, it can be shown that the probability of toxicity, $\Pr(A = 1|x)$, is a non-decreasing function since

$$\begin{aligned}
\Pr(A = 1|x) &= \Pr(A = 1, C = 0|x) + \Pr(A = 1, C = 1|x) \\
&= \left(\frac{\exp(a_2 + b_2 x)}{1 + \exp(a_2 + b_2 x)}\right)\left(\frac{1}{1 + \exp(a_1 + b_1 x)}\right) \\
&+ \left(\frac{\exp(a_3 + b_3 x)}{1 + \exp(a_3 + b_3 x)}\right)\left(\frac{\exp(a_1 + b_1 x)}{1 + \exp(a_1 + b_1 x)}\right)
\end{aligned}$$

Taking the derivative of $\Pr(A=1|x)$ with regard to x, this is non-negative as shown below:

$$
\begin{aligned}
&\frac{\partial \Pr(A=1|x)}{\partial x} \\
&= \left\{ \left(\frac{\exp(a_3+b_3x)}{1+\exp(a_3+b_3x)} \right) \left(\frac{\exp(a_1+b_1x)}{1+\exp(a_1+b_1x)} \right) \right. \\
&\left. \times \left(\frac{b_3}{1+\exp(a_3+b_3x)} + \frac{b_1}{1+\exp(a_1+b_1x)} \right) \right\} \\
&+ \left\{ \left(\frac{\exp(a_2+b_2x)}{1+\exp(a_2+b_2x)} \right) \left(\frac{1}{1+\exp(a_1+b_1x)} \right) \right. \\
&\left. \times \left(\frac{b_2}{1+\exp(a_2+b_2x)} - \frac{b_1(\exp(a_1+b_1x)}{1+\exp(a_1+b_1x)} \right) \right\} \\
&= \left(\frac{1}{1+\exp(a_1+b_1x)} \right) \left\{ b_3 \left(\frac{\exp(a_3+b_3x)\exp(a_1+b_1x)}{(1+\exp(a_3+b_3x))^2} \right) \right. \\
&+ b_2 \left(\frac{\exp(a_2+b_2x)}{(1+\exp(a_2+b_2x))^2} \right) \\
&\left. + b_1 \left(\frac{\exp(a_1+b_1x)}{1+\exp(a_1+b_1x)} \right) \left(\frac{\exp(a_3+b_3x)}{1+\exp(a_3+b_3x)} - \frac{\exp(a_2+b_2x)}{1+\exp(a_2+b_2x)} \right) \right\}
\end{aligned}
$$

From A3, $\frac{\exp(a_2+b_2x)}{1+\exp(a_2+b_2x)} \leq \frac{\exp(a_3+b_3x)}{1+\exp(a_3+b_3x)}$, therefore, the derivative of $\Pr(A=1|x)$ with regard to x, $\frac{\partial \Pr(A=1|x)}{\partial x}$, is non-negative.

For given parameter values satisfying the assumptions, for example, $(\mu_1, \gamma_1) = (3.2, 4.8)$, $(\mu_2, \gamma_2) = (3.6, 5.6)$, and $(\mu_3, \gamma_3) = (3.9, 6.0)$, the joint and marginal probabilities are shown in Figure 5.1. From the three graphs, the probability of nothing happening, $\Pr(A=0, C=0)$, decreases as dose increases, and the probability of both events happening, $\Pr(A=1, C=1)$, increases by dose. The probabilities of exactly one of the two happening (e.g. $\Pr(A=1, C=0)$, $\Pr(A=0, C=1)$) first increases and decreases after certain dose.

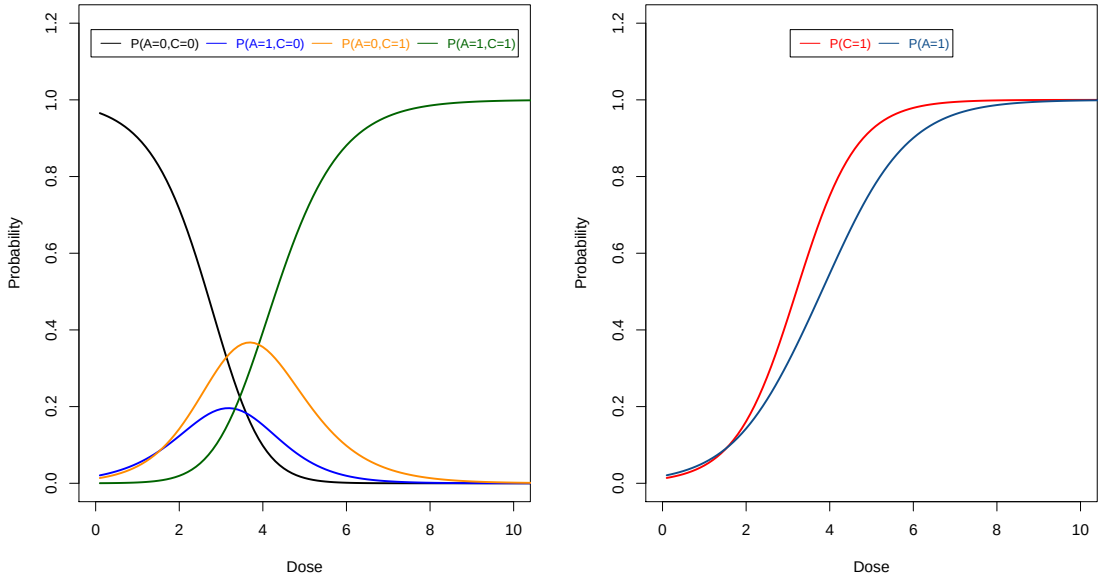

Figure 5.1: An Example for the New Model of the Dose Response of the Four Multinomial Probabilities (left) and the Marginal Probabilities (right)

5.5 Application: Analysis on H-Flu Data Using the New Model

Data from H-Flu study are fit to the new model. An adverse event is defined as those who have moderate reactogenicity reactions. Nine out of 15 (60%) are colonized and approximately 27% experienced adverse events. Tables 5.1 and Figure 5.2 show the data from the H-Flu study. The analysis is performed using both a maximum likelihood and a Bayesian approach.

Table 5.1: Summary of H-Flu Data

Colonization	Adverse Event		Total
	Yes	No	
Yes	3	6	9
No	1	5	6
Total	4	11	15

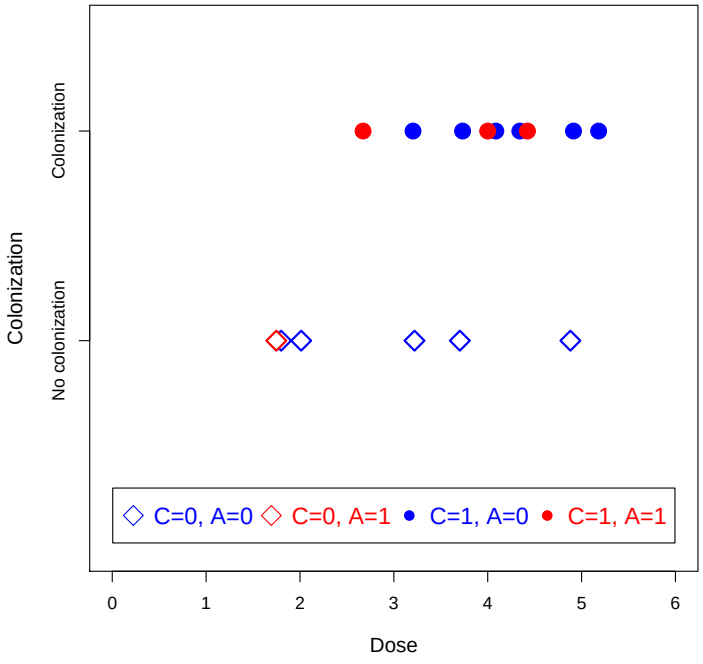

Figure 5.2: Graphical Presentation of H-Flu Data

5.5.1 Maximum Likelihood Analysis

The MLE of $(a_1, b_1, a_2, b_2, a_3, b_3)$ are denotes as $(\hat{a}_1, \hat{b}_1, \hat{a}_2, \hat{b}_2, \hat{a}_3, \hat{b}_3)$, can be obtained by Newton-Raphson method:

$$(\hat{a}_1, \hat{b}_1, \hat{a}_2, \hat{b}_2, \hat{a}_3, \hat{b}_3) = (-3.9680, 1.2404, 18.3595, -10.6325, -6.0007, 1.6531).$$

The corresponding dose estimates are:

$$(\hat{HCD}50, \hat{HCD}90, \hat{DANC}50, \hat{DANC}90, \hat{DAC}50, \hat{DAC}90)$$

$$= (3.1963, 4.9681, 1.7267, 1.5201, 3.6300, 4.9593).$$

The dose-and-toxicity response curves from this fitted model are presented in Figure 5.3. From this graph, the dose-response curve is an monotone increasing function, but toxicity-response is not a non-decreasing function. The estimated values, $\hat{DANC}50$, $\hat{DANC}90$ are such that $\hat{DAC}50 > \hat{DAC}90$. also confirm this finding.

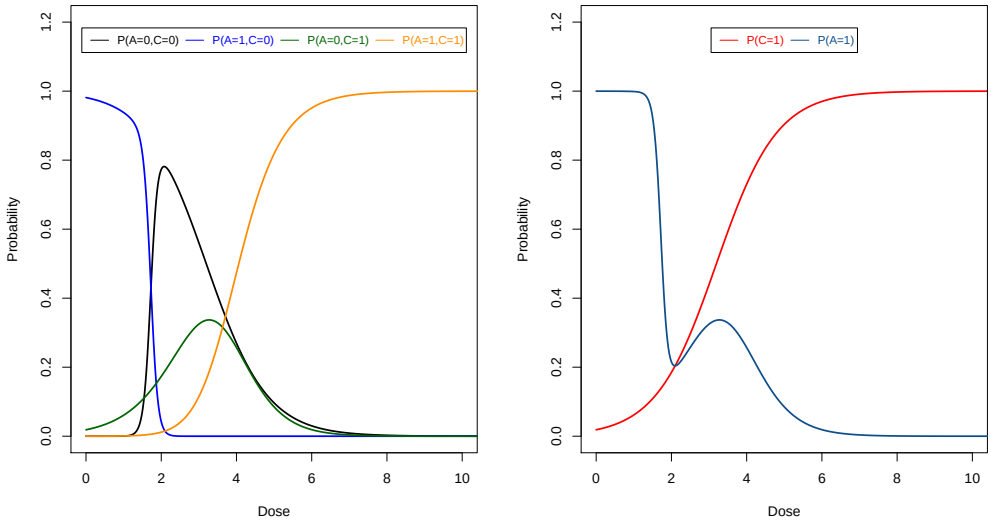

Figure 5.3: Dose- and Toxicity-response Curves for H-Flu Data (Unconstrained Maximum Likelihood Estimates)

5.5.2 Bayesian Analysis

Prior distributions To meet the assumptions of this new model, the prior distributions are constructed in a way that $HCD90$, $DANC90$ and $DAC90$ are always greater than $HCD50$, $DANC50$ and $DAC50$ respectively. The prior distributions for μ_1, δ_1 are, as in PP, and are independent with

$$\frac{\mu_1 - 1.5}{3.5} \sim \text{Beta}(5, 5),$$
$$\frac{\delta_1 - 0.1}{5} \sim \text{Beta}(1, 6)$$

Two sets of prior distribution for $(DANC50, DANC90, DAC50, DAC90)$ are considered:

1. Less informative prior distribution: uniform distributions for μ_2, δ_2, μ_3 and δ_3 are used,
$$\frac{\mu_2 - 1.5}{3.5}, \frac{\mu_3 - 1.5}{3.5} \sim \text{Beta}(1, 1),$$
$$\frac{\delta_2 - 0.1}{5}, \frac{\delta_3 - 0.1}{5} \sim \text{Beta}(1, 1),$$
and μ_2, δ_2, μ_3 and δ_3 are mutually independent, and independent of (μ_1, δ_1).

2. More informative prior distribution: beta distributions for for μ_2, δ_2, μ_3 and δ_3 are used,
$$\frac{\mu_2 - 1.5}{3.5}, \frac{\mu_3 - 1.5}{3.5} \sim \text{Beta}(5, 5),$$
$$\frac{\delta_2 - 0.1}{5}, \frac{\delta_3 - 0.1}{5} \sim \text{Beta}(1, 6).$$
Again, μ_2, δ_2, μ_3 and δ_3 are mutually independent, and independent of (μ_1, δ_1).

The graphical display for these prior distributions is presented in Figure 5.4.

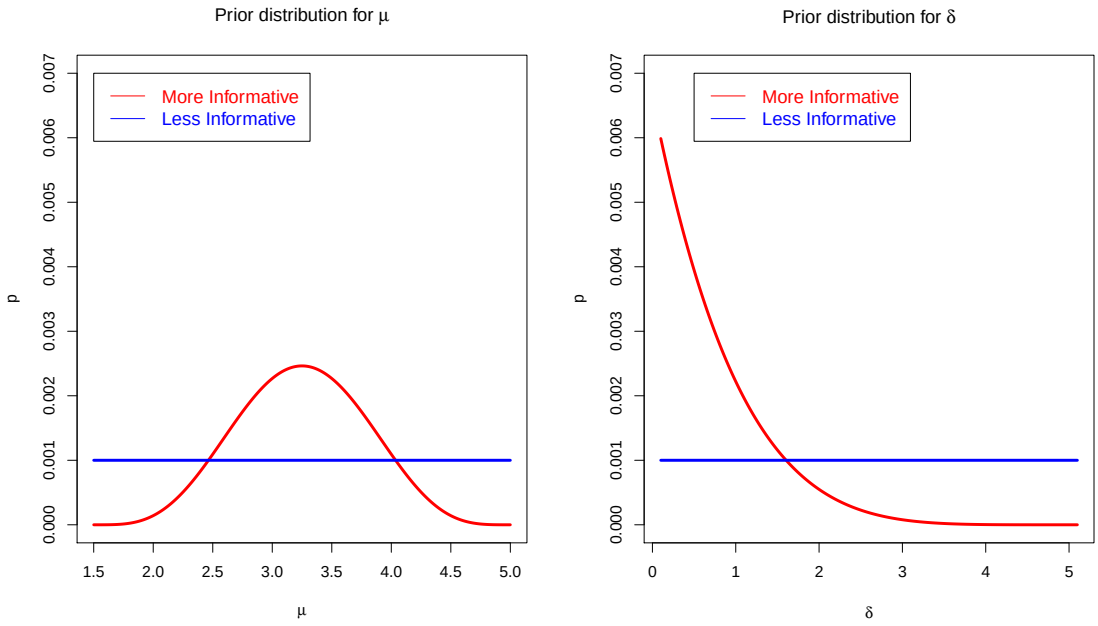

Figure 5.4: Prior Distributions for μ and δ

Results The dose- and toxicity-response curves from the model using the posterior means from the Bayesian analysis can be found in Figure 5.5. The posterior means of doses are listed below:

1. Posterior means of $(HCD50, HCD90, DANC50, DANC90, DAC50, DAC90)$ using less informative prior distributions:

   $$(3.234, 4.843, 4.007, 7.667, 4.160, 7.902).$$

   Denote the values as $\boldsymbol{\theta_1}$.

2. Posterior means of $(HCD50, HCD90, DANC50, DANC90, DAC50, DAC90)$ using more informative prior distributions:

   $$(3.235, 4.847, 3.598, 5.647, 3.818, 6.011).$$

   Denote the values as $\boldsymbol{\theta_2}$.

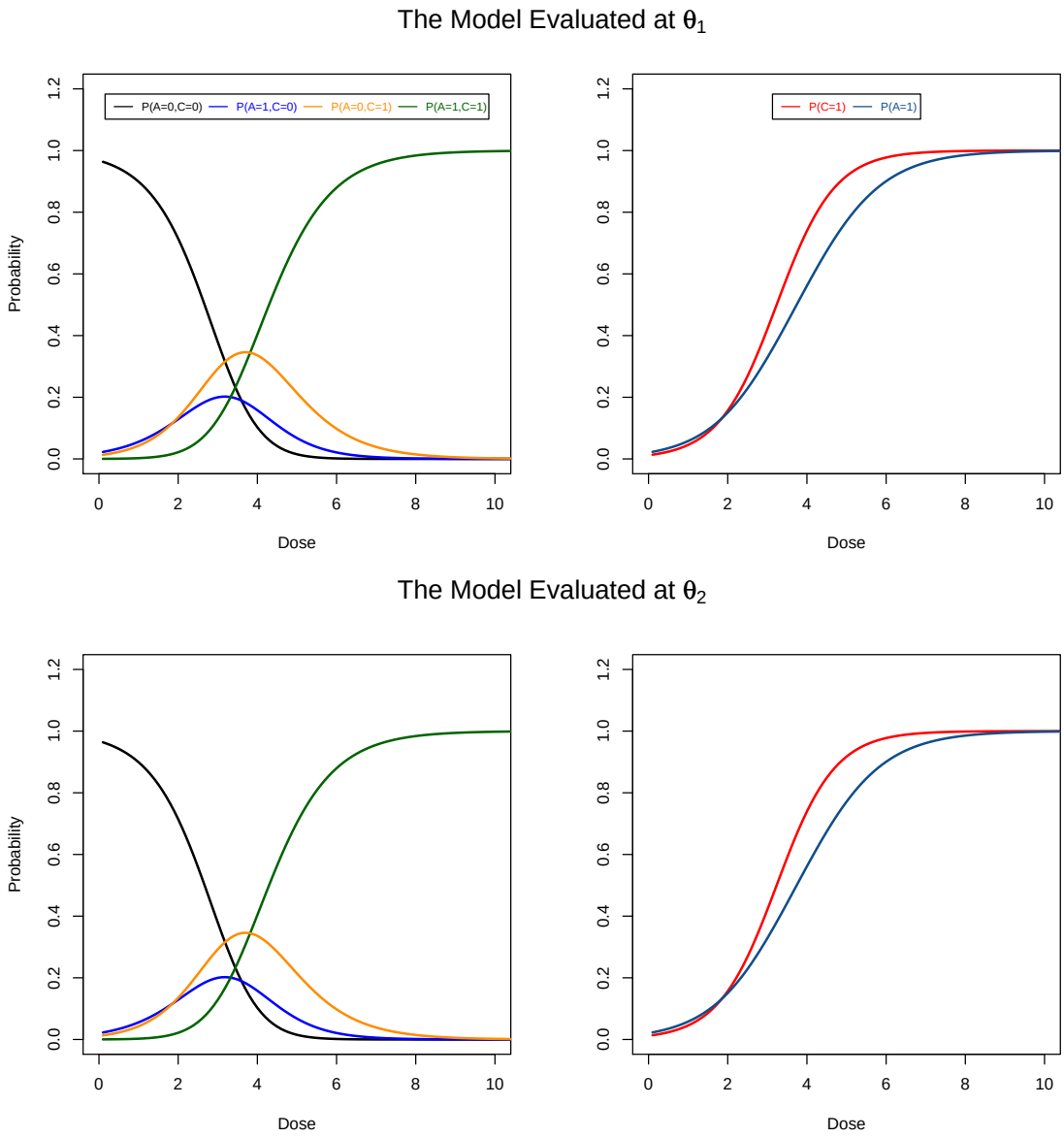

Figure 5.5: Dose- and Toxicity-response Curves for H-Flu data (Bayesian approach) Evaluated at Posterior Means from Less Informative Prior Distribution $\theta_{\mathbf{1}}$, and Evaluated at Posterior Means from More Informative Prior Distribution $\theta_{\mathbf{2}}$.

5.6 Discussion

From the Figures, substituting the posterior means for the parameters, the marginal distributions for colonization and toxicity are monotone increasing functions which is consistent with the assumptions. Doses for colonization and toxicity meet the assumptions for this model. Using a Bayesian approach to analyze this data is appealing as there are only 15 subjects and prior information is helpful. The two prior distributions lead to very similar estimates. Asymptotic standard error from maximum likelihood analysis are likely to be unreliable with a sample sample size. Designs for this model will be investigated further in the following two chapters.

CHAPTER 6
BAYESIAN MYOPIC PROCEDURES FOR THE NEW MODEL

6.1 Optimal Bayesian Sequential Designs

One-, two- and three-step look ahead procedures are implemented to obtain the dose for each stage. As in the model with univariate outcome (see Chapter 3), the loss function is chosen to be the sum of the posterior variance of $HCD50$ and posterior variance of $HCD90$, and Monte Carlo simulations are used to estimate the expected loss. The best dose is one that minimizes the expected loss under the one-, two-, or three-step strategy. The overall expected loss for each strategy is also computed, and this is the measure of performance.

6.2 The Constraints

The candidate doses for each stage should satisfy some constraints to make dose escalation cautious. In the H-Flu study example, there are two constraints.

1. Constraint on dose escalation:

   As in the univariate model, the first constraint is to increase the maximum dose slowly by restricting to 0.5 higher than the previous maximum. The range of the candidate doses for the $(i+1)^{th}$ subject is:

   $$\left[1.5, 2.0, .., \max_{j \le i} \{x_i\} + 0.5\right].$$

2. Constraint on toxicity:

   Since it is important to minimize toxicity, the second constraint is on the

posterior probability of adverse event. For each stage, the posterior probability of adverse event given dose should satisfy the following:

$$\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i) < \epsilon \qquad 0 < \epsilon \leq 1.$$

When $\epsilon = 1$, there is no restriction on $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i)$.

6.3 Algorithm

6.3.1 Generating the Prior Sample

Define $\mu_1 = HCD50$, $\gamma_1 = HCD90$, and $\delta_1 = HCD90 - HCD50 = \gamma_1 - \mu_1$, $\mu_2 = DANC50$, $\gamma_2 = DANC90$, and $\delta_2 = DANC90 - DANC50 = \gamma_2 - \mu_2$, $\mu_3 = DAC50$, $\gamma_3 = DAC50$, and $\delta_3 = DAC90 - DAC50 = \gamma_3 - \mu_3$. Denote $S_i = S_i(\mathbf{h^{(i-1)}}) \in \mathcal{D}_i$ is the set of allowable doses at the i^{th} stage given $\mathbf{h^{(i-1)}}$.

Step 1: Before observing any data, select a random sample of $(\mu_{1k}, \gamma_{1k}, \mu_{2k}, \gamma_{2k}, \mu_{3k}, \gamma_{3k})$, $k = 1, 2, ..., N$ from the prior distribution.

Step 2: Compute the prior probabilities of each outcome if the first dose is chosen to be d_j, using Monte Carlo; that is for each $(\mu_{1k}, \gamma_{1k}, \mu_{2k}, \gamma_{2k}, \mu_{3k}, \gamma_{3k})$ combination denote

$$p_{5jk} = \frac{9^{\frac{d_j - \mu_{1k}}{\delta_{1k}}}}{1 + 9^{\frac{d_j - \mu_{1k}}{\delta_{1k}}}},$$

$$p_{1jk} = \frac{1}{1 + 9^{\frac{d_j - \mu_{2k}}{\delta_{2k}}}}(1 - p_{5jk}), \qquad p_{2jk} = \frac{9^{\frac{d_j - \mu_{2k}}{\delta_{2k}}}}{1 + 9^{\frac{d_j - \mu_{2k}}{\delta_{2k}}}}(1 - p_{5jk}),$$

$$p_{3jk} = \frac{1}{1 + 9^{\frac{d_j - \mu_{3k}}{\delta_{3k}}}}(p_{5jk}), \qquad p_{4jk} = \frac{9^{\frac{d_j - \mu_{3k}}{\delta_{3k}}}}{1 + 9^{\frac{d_j - \mu_{3k}}{\delta_{3k}}}}(p_{5jk}).$$

Then the prior probabilities of each outcome at dose d_j are just the average of p_{1jk}, p_{2jk}, p_{3jk} and p_{4jk} over $k = 1, ..., N$. Equivalently, let the initial weight, w_{0k} be 1, for all k.

$$\Pr(y_{11} = 1|\ h_0) = p^{(1)}(d_j, h_{j1}) = \frac{\sum_k p_{1jk}\ w_{0k}}{\sum_k w_{0k}},$$

$$\Pr(y_{12} = 1|\ h_0) = p^{(1)}(d_j, h_{j2}) = \frac{\sum_k p_{2jk}\ w_{0k}}{\sum_k w_{0k}},$$

$$\Pr(y_{13} = 1|\ h_0) = p^{(1)}(d_j, h_{j3}) = \frac{\sum_k p_{3jk}\ w_{0k}}{\sum_k w_{0k}},$$

$$\Pr(y_{14} = 1|\ h_0) = p^{(1)}(d_j, h_{j4}) = \frac{\sum_k p_{4jk}\ w_{0k}}{\sum_k w_{0k}}.$$

6.3.2 One-step-look-ahead Procedure

For the first stage ($i = 1$), since there is no previous stage, the initial weights, w_{0k}, are identical and equal to 1. At the i^{th} stage ($i \geq 2$), suppose the dose from previous stage is $x_{i-1} = d_{j^*}$ and $\mathcal{D}_i(\mathbf{h^{(i-1)}})$ represent possible doses at the i^{th} stage: $S_i = \{s_{il}; l = 1, ..., L_i\}$.

Step 1: If the dose is chosen to be s_{il}, the average probability of each outcome at dose l given $\mathbf{h^{(i-1)}}$ are:

$$\Pr(y_{i1} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{1lk}\ w_k(h_{l1}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i2} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{2lk}\ w_k(h_{l2}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i3} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{3lk}\ w_k(h_{l3}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i4} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{4lk}\ w_k(h_{l4}(\mathbf{h^{(i-1)}})).$$

Then calculate posterior the weight, $w_k(\mathbf{h^{(i)}})$, for each possible outcome:

For $y_{i1} = 1$, let

$$w_k(h_{l1}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{1lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{1lk'}} \times N;$$

for $y_{i2} = 1$, let

$$w_k(h_{l2}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{2lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{2lk'}} \times N;$$

for $y_{i3} = 1$, let

$$w_k(h_{l3}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{3lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{3lk'}} \times N;$$

for $y_{i4} = 1$, let

$$w_k(h_{l4}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{4lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{4lk'}} \times N.$$

Note that these posterior probabilities, and those in all subsequent steps, are normalized to sum to N over k, to avoid underflow.

Step 2: The posterior means for μ_1 and γ_1 given $\mathbf{h^{(i)}}$ corresponding to each outcome are

$$E(\mu_1 \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}})\ \mu_{1k}}{\sum_k w_k(\mathbf{h^{(i)}})},$$
$$E(\gamma_1 \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}})\ \gamma_{1k}}{\sum_k w_k(\mathbf{h^{(i)}})},$$

where $\mathbf{h^{(1)}}$ is h_{l1}, h_{l2}, h_{l3} or h_{l4} for $d_l \in S_1(h_0) = S_1$.

The posterior variances for μ_1 and γ_1 and for each outcome are

$$Var(\mu_1 \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}})\ \left\{\mu_{1k} - E(\mu_1 \mid \mathbf{h^{(i)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i)}})},$$
$$Var(\gamma_1 \mid \mathbf{h^{(i)}}) = \frac{\sum_k w_k(\mathbf{h^{(i)}})\ \left\{\gamma_{1k} - E(\gamma_1 \mid \mathbf{h^{(i)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i)}})}.$$

Step 3: The loss if $x_i = s_{1l}$ is then

$$L^{(i)}(\mathbf{h^{(i)}}) = Var(\mu_1 \mid \mathbf{h^{(i)}}) + Var(\gamma_1 \mid \mathbf{h^{(i)}}).$$

The expected loss at dose l, looking only one step ahead, is

$$\begin{aligned} EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il}) = \Pr(y_{i1} = 1 \mid \mathbf{h^{(i-1)}}) L^{(i)}(\mathbf{h^{(i-1)}},\ \text{if } y_{i1} = 1) \\ + \Pr(y_{i2} = 1 \mid \mathbf{h^{(i-1)}}) L^{(i)}(\mathbf{h^{(i-1)}},\ \text{if } y_{i2} = 1) \\ + \Pr(y_{i3} = 1 \mid \mathbf{h^{(i-1)}}) L^{(i)}(\mathbf{h^{(i-1)}},\ \text{if } y_{i3} = 1) \\ + \Pr(y_{i4} = 1 \mid \mathbf{h^{(i-1)}}) L^{(i)}(\mathbf{h^{(i-1)}},\ \text{if } y_{i4} = 1). \end{aligned}$$

Step 4: The best one-step-look-ahead dose for stage i is $x_i = d_{l^*}$, where

$$l^* = \underset{s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})}{\operatorname{argmin}} \left\{ EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il}) \right\}.$$

6.3.3 Two-step-look-ahead Procedure: Batch of Two and Moving Window

Suppose the dose from the $(i-1)^{th}$ stage is $x_{i-1} = d_{j^*}$, and suppose $s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})$ and $s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})$ represent candidate doses for stage i and $(i+1)$. The two-step-look-ahead procedure with a batch of two strategy is to assume only two step remain, and doses for stage i and $(i+1)$ are allocated optimally using best two-step sequential procedure; that is the dose at stage $(i+1)$ depends on the response at stage i. On the other hand, the moving window method is to select the optimal dose for the i^{th} stage assuming there are two steps remaining.

At the i^{th} stage

Step 1: If the dose is chosen to be s_{il}, the average probabilities of each possible outcome

at dose l given $\mathbf{h^{(i-1)}}$:

$$\Pr(y_{i1} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{1lk}\ w_k(h_{l1}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i2} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{2lk}\ w_k(h_{l2}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i3} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{3lk}\ w_k(h_{l3}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i4} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{4lk}\ w_k(h_{l4}(\mathbf{h^{(i-1)}})).$$

Then calculate the posterior weight, $w_k(\mathbf{h^{(i)}})$, for each outcome:

For $y_{i1} = 1$, let

$$w_k(h_{l1}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{1lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{1lk'}} \times N;$$

for $y_{i2} = 1$, let

$$w_k(h_{l2}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{2lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{2lk'}} \times N;$$

for $y_{i3} = 1$, let

$$w_k(h_{l3}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{3lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{3lk'}} \times N;$$

for $y_{i4} = 1$, let

$$w_k(h_{l4}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{4lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{4lk'}} \times N.$$

At the $(i+1)^{th}$ stage

Step 2: If the dose is chosen to be $s_{(i+1)g}$, the average of probabilities of each outcome at dose g given $\mathbf{h^{(i)}}$:

$$\Pr(y_{(i+1)1} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{1gk}\ w_k(h_{g1}(\mathbf{h^{(i)}})),$$

$$\Pr(y_{(i+1)2} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{2gk}\ w_k(h_{g2}(\mathbf{h^{(i)}})),$$

$$\Pr(y_{(i+1)3} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{3gk}\ w_k(h_{g3}(\mathbf{h^{(i)}})),$$

$$\Pr(y_{(i+1)4} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{4gk}\ w_k(h_{g4}(\mathbf{h^{(i)}})).$$

Then calculate the posterior weight, $w_k(\mathbf{h^{(i+1)}})$, for each outcome:

For $y_{(i+1)1} = 1$, let

$$w_k(h_{g1}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{1gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{1gk'}} \times N;$$

for $y_{(i+1)2} = 1$, let

$$w_k(h_{g2}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{2gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{2gk'}} \times N;$$

for $y_{(i+1)3} = 1$, let

$$w_k(h_{g3}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{3gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{3gk'}} \times N;$$

for $y_{(i+1)4} = 1$, let

$$w_k(h_{g4}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{4gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{4gk'}} \times N.$$

Step 3: The posterior means for μ_1 and γ_1 given $\mathbf{h^{(i+1)}} = h_{g1}(\mathbf{h^{(i)}})$ for each outcome are:

$$E(\mu_1 \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \mu_{1k}}{\sum_k w_k(\mathbf{h^{(i+1)}})},$$

$$E(\gamma_1 \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \gamma_{1k}}{\sum_k w_k(\mathbf{h^{(i+1)}})},$$

and posterior variances for μ_1 and γ_1 for each outcome are:

$$Var(\mu_1 \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \left\{\mu_{1k} - E(\mu_1 \mid \mathbf{h^{(i+1)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+1)}})},$$

$$Var(\gamma_1 \mid \mathbf{h^{(i+1)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+1)}})\ \left\{\gamma_{1k} - E(\gamma_1 \mid \mathbf{h^{(i+1)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+1)}})}.$$

Step 4: The loss if $x_{i+1} = s_{(i+1)g}$ for each possible outcome is:

$$L^{(i+1)}(\mathbf{h^{(i+1)}}) = Var(\mu_1 \mid \mathbf{h^{(i+1)}}) + Var(\gamma_1 \mid \mathbf{h^{(i+1)}}).$$

Step 5: The expected loss at dose g is then

$$\begin{aligned} EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) = {} & \Pr(y_{(i+1)1} = 1 \mid \mathbf{h^{(i)}})\ L^{(i+1)}(s_{(i+1)g},\ \text{if } y_{(i+1)1} = 1) \\ & + \Pr(y_{(i+1)2} = 1 \mid \mathbf{h^{(i)}})\ L^{(i+1)}(\mathbf{h^{(i)}},\ \text{if } y_{(i+1)2} = 1) \\ & + \Pr(y_{(i+1)3} = 1 \mid \mathbf{h^{(i)}})\ L^{(i+1)}(\mathbf{h^{(i)}},\ \text{if } y_{(i+1)3} = 1) \\ & + \Pr(y_{(i+1)4} = 1 \mid \mathbf{h^{(i)}})\ L^{(i+1)}(\mathbf{h^{(i)}},\ \text{if } y_{(i+1)4} = 1). \end{aligned}$$

Selecting the optimal dose

Step 6: At stage $(i+1)$:

For each s_{il}, the minimum expected loss at stage $(i+1)$ is:

For $y_{i1} = 1$

$$g_1^* = g_1^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

for $y_{i2} = 1$

$$g_2^* = g_2^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

for $y_{i3} = 1$

$$g_3^* = g_3^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

for $y_{i4} = 1$

$$g_4^* = g_4^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\}.$$

Step 7: At stage i:

For each s_{il} the expected loss is

$$\begin{aligned} EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il}) = {} & \Pr(y_{i1} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g_1^*}) \\ & + \Pr(y_{i2} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g_2^*}) \\ & + \Pr(y_{i3} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g_3^*}) \\ & + \Pr(y_{i4} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g_4^*}). \end{aligned}$$

The optimal two-step-look-ahead dose for the stage i is $x_i = d_{l^*}$, where

$$l^* = \underset{s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})}{\operatorname{argmin}} \left\{ EL^{(i)}(s_{il}) \right\}.$$

For the batch method, the optimal doses for stage $(i+1)$ are $d_{g_1^*}$ and $d_{g_0^*}$, and the optimal dose for stage i is d_{l^*}. For moving-window method, optimal dose for stage i is d_{l^*}.

6.3.4 Three-step-look-ahead Procedure: Moving Window

Three-step-look-ahead procedure is the one that for each stage, it is treated as if there are three steps left. The moving window method is applied. For each stage, only one optimal dose, $x_i = d_{l^*}$, is selected.

Suppose the optimal dose from the $(i-1)^{th}$ is $x_{i-1} = d_{j^*}$, and suppose $s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})$, $s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})$ and $s_{(i+2)f} \in \mathcal{D}(\mathbf{h^{(i+1)}})$ represent candidate doses for stage i, $i+1$ and $i+2$.

At the i^{th} stage

Step 1: If the dose is chosen to be s_{il}, calculate the average probabilities of each outcome at dose l given $\mathbf{h^{(i-1)}}$ are:

$$\Pr(y_{i1} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{1lk}\ w_k(h_{l1}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i2} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{2lk}\ w_k(h_{l2}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i3} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{3lk}\ w_k(h_{l3}(\mathbf{h^{(i-1)}})),$$

$$\Pr(y_{i4} = 1|\ \mathbf{h^{(i-1)}}) = \sum_k p_{4lk}\ w_k(h_{l4}(\mathbf{h^{(i-1)}})).$$

Then calculate the posterior weight, $w_k(\mathbf{h^{(i)}})$, for each outcome:

For $y_{i1} = 1$, let

$$w_k(h_{l1}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{1lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{1lk'}} \times N;$$

for $y_{i2} = 1$, let

$$w_k(h_{l2}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{2lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{2lk'}} \times N;$$

for $y_{i3} = 1$, let

$$w_k(h_{l3}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{3lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{3lk'}} \times N;$$

for $y_{i4} = 1$, let

$$w_k(h_{l4}(\mathbf{h^{(i-1)}})) = w_k(\mathbf{h^{(i)}}) = \frac{w_k(\mathbf{h^{(i-1)}})\ p_{4lk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i-1)}})\ p_{4lk'}} \times N.$$

At the $(i+1)^{th}$ stage

Step 2: If the dose is chosen to be $s_{(i+1)g}$, calculate the average probabilities of each outcome at dose g given $\mathbf{h^{(i)}}$:

$$\Pr(y_{(i+1)1} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{1gk}\ w_k(h_{g1}(\mathbf{h^{(i)}})),$$

$$\Pr(y_{(i+1)2} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{2gk}\ w_k(h_{g2}(\mathbf{h^{(i)}})),$$

$$\Pr(y_{(i+1)3} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{3gk}\ w_k(h_{g3}(\mathbf{h^{(i)}})),$$

$$\Pr(y_{(i+1)4} = 1|\ \mathbf{h^{(i)}}) = \sum_k p_{4gk}\ w_k(h_{g4}(\mathbf{h^{(i)}})).$$

Then calculate posterior weight, $w_k(\mathbf{h^{(i+1)}})$, for each outcome:

For $y_{(i+1)1} = 1$, let

$$w_k(h_{g1}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{1gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{1gk'}} \times N,$$

For $y_{(i+1)2} = 1$, let

$$w_k(h_{g2}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{2gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{2gk'}} \times N,$$

For $y_{(i+1)3} = 1$, let

$$w_k(h_{g3}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{3gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{3gk'}} \times N,$$

For $y_{(i+1)4} = 1$, let

$$w_k(h_{g4}(\mathbf{h^{(i)}})) = w_k(\mathbf{h^{(i+1)}}) = \frac{w_k(\mathbf{h^{(i)}})\ p_{4gk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i)}})\ p_{4gk'}} \times N.$$

At the $(i+2)^{th}$ stage

Step 3: If the dose is chosen to be $s_{(i+2)f}$, calculate the average probabilities of each outcome at dose g given $\mathbf{h^{(i+1)}}$:

$$\Pr(y_{(i+2)1} = 1|\ \mathbf{h^{(i+1)}}) = \sum_k p_{1fk}\ w_k(h_{f1}(\mathbf{h^{(i+1)}})),$$

$$\Pr(y_{(i+2)2} = 1|\ \mathbf{h^{(i+1)}}) = \sum_k p_{2fk}\ w_k(h_{f2}(\mathbf{h^{(i+1)}})),$$

$$\Pr(y_{(i+2)3} = 1|\ \mathbf{h^{(i+1)}}) = \sum_k p_{3fk}\ w_k(h_{f3}(\mathbf{h^{(i+1)}})),$$

$$\Pr(y_{(i+2)4} = 1|\ \mathbf{h^{(i+1)}}) = \sum_k p_{4fk}\ w_k(h_{f4}(\mathbf{h^{(i+1)}})).$$

Then calculate posterior weight, $w_k(\mathbf{h^{(i+2)}})$, for each outcome:

For $y_{(i+2)1} = 1$, let

$$w_k(h_{f1}(\mathbf{h^{(i+1)}})) = w_k(\mathbf{h^{(i+2)}}) = \frac{w_k(\mathbf{h^{(i+1)}})\ p_{1fk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i+1)}})\ p_{1fk'}} \times N;$$

for $y_{(i+2)2} = 1$, let

$$w_k(h_{f2}(\mathbf{h^{(i+1)}})) = w_k(\mathbf{h^{(i+2)}}) = \frac{w_k(\mathbf{h^{(i+1)}})\ p_{2fk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i+1)}})\ p_{2fk'}} \times N;$$

for $y_{(i+2)3} = 1$, let

$$w_k(h_{f3}(\mathbf{h^{(i+1)}})) = w_k(\mathbf{h^{(i+2)}}) = \frac{w_k(\mathbf{h^{(i+1)}})\ p_{3fk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i+1)}})\ p_{3fk'}} \times N;$$

for $y_{(i+2)4} = 1$, let

$$w_k(h_{f4}(\mathbf{h^{(i+1)}})) = w_k(\mathbf{h^{(i+2)}}) = \frac{w_k(\mathbf{h^{(i+1)}})\ p_{4fk}}{\sum_{k'} w_{k'}(\mathbf{h^{(i+1)}})\ p_{4fk'}} \times N.$$

Note that these posterior probabilities are normalized to sum to N over k, to avoid underflow.

Step 4: The posterior means for μ_1 and γ_1 given $\mathbf{h^{(i+2)}}$ for each outcome are:

$$E(\mu_1 \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \mu_k}{\sum_k w_k(\mathbf{h^{(i+2)}})},$$

$$E(\gamma_1 \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \gamma_k}{\sum_k w_k(\mathbf{h^{(i+2)}})},$$

and posterior variances for μ_1 and γ_1 for each outcome are:

$$Var(\mu_1 \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \left\{\mu_{1k} - E(\mu_1 \mid \mathbf{h^{(i+2)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+2)}})},$$

$$Var(\gamma_1 \mid \mathbf{h^{(i+2)}}) = \frac{\sum_k w_k(\mathbf{h^{(i+2)}})\ \left\{\gamma_{1k} - E(\gamma_1 \mid \mathbf{h^{(i+2)}})\right\}^2}{\sum_k w_k(\mathbf{h^{(i+2)}})}.$$

Step 5: The loss if $x_{i+2} = s_{(i+2)f}$ for each possible outcome is:

$$L^{(i+2)}(\mathbf{h^{(i+2)}}) = Var(\mu_1 \mid \mathbf{h^{(i+2)}}) + Var(\gamma_1 \mid \mathbf{h^{(i+2)}}).$$

Step 6: The expected loss at dose f is then

$$\begin{aligned} EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f}) = {} & \Pr(y_{(i+2)1} = 1 \mid \mathbf{h^{(i+1)}})\ L^{(i+2)}(\mathbf{h^{(i+2)}},\ \text{if } y_{(i+2)1} = 1) \\ & + \Pr(y_{(i+2)2} = 1 \mid \mathbf{h^{(i+1)}})\ L^{(i+2)}(\mathbf{h^{(i+2)}},\ \text{if } y_{(i+2)2} = 1) \\ & + \Pr(y_{(i+2)3} = 1 \mid \mathbf{h^{(i+1)}})\ L^{(i+2)}(\mathbf{h^{(i+2)}},\ \text{if } y_{(i+2)3} = 1) \\ & + \Pr(y_{(i+2)4} = 1 \mid \mathbf{h^{(i+1)}})\ L^{(i+2)}(\mathbf{h^{(i+2)}},\ \text{if } y_{(i+2)4} = 1). \end{aligned}$$

Selecting the optimal dose

Step 7: At stage $(i+2)$:

For each s_{1l} and $s_{(i+1)g}$, the minimum expected loss is: For $y_{(i+1)1}=1$

$$f_1^* = f_1^*(s_{il}, s_{(i+1)g}) = \underset{s_{(i+2)f}\in\mathcal{D}(\mathbf{h^{(i+1)}})}{\operatorname{argmin}} \left\{EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f})\right\};$$

for $y_{(i+1)2}=1$

$$f_2^* = f_2^*(s_{il}, s_{(i+1)g}) = \underset{s_{(i+2)f}\in\mathcal{D}(\mathbf{h^{(i+1)}})}{\operatorname{argmin}} \left\{EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f})\right\};$$

for $y_{(i+1)3}=1$

$$f_3^* = f_3^*(s_{il}, s_{(i+1)g}) = \underset{s_{(i+2)f}\in\mathcal{D}(\mathbf{h^{(i+1)}})}{\operatorname{argmin}} \left\{EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f})\right\};$$

for $y_{(i+1)4}=1$

$$f_4^* = f_4^*(s_{il}, s_{(i+1)g}) = \underset{s_{(i+2)f}\in\mathcal{D}(\mathbf{h^{(i+1)}})}{\operatorname{argmin}} \left\{EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f})\right\}.$$

Step 8: At stage $(i+1)$:

For each s_{1l}, the expected loss for each outcome is

$$\begin{aligned} EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) = {} & \Pr(y_{(i+1)1}=1|\ \mathbf{h^{(i)}}) EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f_1^*}) \\ & + \Pr(y_{(i+1)2}=1|\ \mathbf{h^{(i)}}) EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f_2^*}) \\ & + \Pr(y_{(i+1)3}=1|\ \mathbf{h^{(i)}}) EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f_3^*}) \\ & + \Pr(y_{(i+1)4}=1|\ \mathbf{h^{(i)}}) EL^{(i+2)}(\mathbf{h^{(i+1)}}, s_{(i+2)f_4^*}) \end{aligned}$$

The minimum expected loss are:

For $y_{i1} = 1$:

$$g_1^* = g_1^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

for $y_{i2} = 0$:

$$g_2^* = g_2^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

for $y_{i3} = 0$:

$$g_3^* = g_3^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

for $y_{i4} = 0$:

$$g_4^* = g_4^*(s_{il}) = \underset{s_{(i+1)g} \in \mathcal{D}(\mathbf{h^{(i)}})}{\operatorname{argmin}} \left\{ EL^{(i+1)}(\mathbf{h^{(i)}}, s_{(i+1)g}) \right\};$$

Step 9: At stage i: For each s_{il} the expected loss is

$$\begin{aligned} EL^{(i)}(\mathbf{h^{(i-1)}}, s_{il}) = {} & \Pr(y_{i1} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{i+1,g_1^*}) \\ & + \Pr(y_{i2} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{i+1,g_2^*}) \\ & + \Pr(y_{i3} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{i+1,g_3^*}) \\ & + \Pr(y_{i4} = 1 |\ \mathbf{h^{(i-1)}}) EL^{(i+1)}(\mathbf{h^{(i)}}, s_{i+1,g_4^*}). \end{aligned}$$

The optimal three-step-look-ahead dose for the stage i is $x_i = d_{l^*}$, where

$$l^* = \underset{s_{il} \in \mathcal{D}(\mathbf{h^{(i-1)}})}{\operatorname{argmin}} \left\{ EL^{(i)}(s_{il}) \right\}.$$

6.4 Summary

These algorithms will be implemented in Chapter 7.

CHAPTER 7
SIMULATION RESULTS FROM MODEL CONSIDERING EFFICACY AND TOXICITY

7.1 Settings for Simulation

Using the new model, designs are compared by simulation using different strategies and different prior distributions. A sample of 100 prior values is generated from the prior distribution for each simulation. As in Chapter 4, the ratio of overall expected loss is calculated using one-step-look-ahead procedure as a benchmark. In addition to comparing strategies, comparisons between different prior distribution and different levels of constraint on the posterior probability of toxicity, $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i)$ are provided.

7.1.1 Prior Distributions

Similar to the simulations using the univariate model, the Beta prior distribution is used to construct the prior distribution (see Chapter 4). For $i = 1, 2, 3$, define v_i and z_i such that $0 \leq v_i \leq 1$ and $0 \leq z_i \leq 1$, and

$$v_1, v_2, v_3 \sim \text{Beta}(v_a, v_b), \;\; z_1, z_2, z_3 \sim \text{Beta}(z_a, z_b).$$

Three sets of prior distributions for the bivariate model are considered:

1. Primary prior distribution (BPP):

$$v_i \sim \text{Beta}(5, 5), \;\; z_i \sim \text{Beta}(1, 6) \qquad \text{independently for} \;\; i = 1, 2, 3.$$

This reflects the knowledge from the investigators.

2. More precise prior distribution (BMP):

$$v_i \sim \text{Beta}(22, 22),\ z_i \sim \text{Beta}(4, 26) \qquad \text{independently for } \ i = 1, 2, 3.$$

3. Less precise prior distribution (BLP):

$$v_i \sim \text{Beta}(7/8, 7/8),\ z_i \sim \text{Beta}(0.15, 0.85) \qquad \text{independently for } \ i = 1, 2, 3.$$

The expectation of v_i and z_i are identical across BMP, BPP and BLP, and the standard deviation from BMP is half of BPP, and the standard deviation from BLP is twice the standard deviation from BPP. The plot of these prior distributions can be found in Figure 4.1. In the examples the prior distributions on $v_1, v_2, v_3, z_1, z_2, z_3$ are mutually independent, for simplicity.

7.1.2 Generating the Prior Sample

The range of $HCD50$, μ, is assumed to be bounded by U and L $(U, L > 0)$, $L \leq \mu_1, \mu_2, \mu_3 \leq U$. The independent and identically distributed samples, $(v_{1k}, z_{1k}, v_{2k}, z_{2k}, v_{3k}, z_{3k})$ $(k = 1, ..., N)$, are selected from the prior Beta distributions. Define c_{1i} and c_{2i} so that $\delta_{ik} = c_{1i}z_{ik} + c_{2i}$ $(c_{1i}, c_{2i} > 0)$ so that γ_{ik} is believed to be at least c_{2i} units greater than μ_{ik}, and γ_{ik} is believed to be at most $c_{1i} + c_{2i} + \mu_{ik}$. Then define

$$\mu_{ik} = (U - L)v_{ik} + L,$$

$$\gamma_{ik} = \delta_{ik} + \mu_{ik} = (c_{1i}z_{ik} + c_{2i}) + \mu_{ik}.$$

In the H-Flu example, the candidate doses range from 1.5 to 5.0 with increment of 0.5, on the $\log_{10}$ scale. It is assumed that the value of $HCD90$ should be at

least 0.1 log greater than $HCD50$, and the largest possible value of $HCD90$ should not be greater than 10.1. Hence, in the new model, for $i = 1, 2, 3$ and $k = 1, ..., N$,

$$\mu_{ik} = 3.5v_{ik} + 1.5,$$

$$\delta_{ik} = 5z_{ik} + 0.1,$$

$$\gamma_{ik} = \mu_{ik} + \delta_{ik}.$$

7.1.3 Constraint on Toxicity

There are three levels of restrictions examined on the posterior probability of toxicity:

1. More conservative restriction: $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i) \leq 0.20$
2. Less conservative restriction: $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i) \leq 0.50$
3. No restriction: $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i) \leq 1$

7.2 Results

The results from simulations are presented in Tables 7.1, 7.2 and 7.3. The overall expected loss and the expected number of subjects, $E(N)$, for each strategy and each prior distribution are computed. The overall expected loss decreases as the number of subjects increases since the posterior variance of $HCD50$ and $HCD90$ is lower when more data is obtained. When considering the restrictions on the posterior probability of toxicity, early stopping may occur, and this can impact $E(N)$. At any stage, if there is no dose satisfying the constraint on toxicity, then there is no acceptable dose available, and the trial stops. As the restriction becomes more conservative (e.g., more conservative restriction on $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i)$, see Table 7.2 and 7.3), the prior distribution can impact early stopping as well, and the values of $E(N)$ are less than the maximum numbers of subject in some cases.

Table 7.1: Simulation Results: More Precise Prior Distribution (BMP)

One-step-look-ahead procedure					
Max. No. of Subject	Without restriction Exp Loss	Less Conservative Exp Loss	E(N)	More Conservative Exp Loss	E(N)
2	0.1889	0.1988	2.0000	0.2167	2.0000
3	0.1754	0.1893	3.0000	0.2134	3.0000
4	0.1641	0.1815	4.0000	0.2105	4.0000
5	0.1539	0.1748	5.0000	0.2080	5.0000
6	0.1450	0.1692	6.0000	0.2057	6.0000
7	0.1371	0.1642	7.0000	0.2036	7.0000
8	0.1301	0.1599	8.0000	0.2017	8.0000
9	0.1238	0.1560	9.0000	0.2000	9.0000
Two-step-look-ahead procedure (batch)					
Max. No. of Subject	Without Restriction Exp Loss	Less Conservative) Exp Loss	E(N)	More Conservative Exp Loss	E(N)
2	0.1889	0.1988	2.0000	0.2167	2.0000
3	0.1754	0.1893	3.0000	0.2134	3.0000
4	0.1641	0.1815	4.0000	0.2105	4.0000
5	0.1539	0.1748	5.0000	0.2080	5.0000
6	0.1450	0.1692	6.0000	0.2057	6.0000
7	0.1371	0.1642	7.0000	0.2036	7.0000
8	0.1300	0.1599	8.0000	0.2017	8.0000
9	0.1237	0.1560	9.0000	0.2000	9.0000
Two-step-look-ahead procedure					
Max. No. of Subject	Without Restriction Exp Loss	Less Conservative Exp Loss	E(N)	More Conservative Exp Loss	E(N)
2	0.1889	0.1988	2.0000	0.2167	2.0000
3	0.1754	0.1893	3.0000	0.2134	3.0000
4	0.1641	0.1815	4.0000	0.2105	4.0000
5	0.1539	0.1748	5.0000	0.2080	5.0000
6	0.1449	0.1692	6.0000	0.2057	6.0000
7	0.1370	0.1642	7.0000	0.2036	7.0000
8	0.1300	0.1599	8.0000	0.2017	8.0000
9	0.1237	0.1560	9.0000	0.2000	9.0000
Three-step-look-ahead procedure					
Max. No. of Subject	Without Restriction Exp Loss	Less Conservative Exp Loss	E(N)	More Conservative Exp Loss	E(N)
3	0.1754	0.1893	3.0000	0.2134	3.0000
4	0.1641	0.1815	4.0000	0.2105	4.0000
5	0.1539	0.1748	5.0000	0.2080	5.0000
6	0.1449	0.1692	6.0000	0.2057	6.0000
7	0.1370	0.1642	7.0000	0.2036	7.0000
8	0.1300	0.1598	8.0000	0.2017	8.0000
9	0.1237	0.1559	9.0000	0.2000	9.0000

Table 7.2: Simulation Results: Primary Prior Distribution (BPP)

One-step-look-ahead procedure					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
2	0.5834	0.6663	2.0000	0.8264	2.0000
3	0.5033	0.6165	3.0000	0.7798	3.0000
4	0.4406	0.5800	4.0000	0.7490	3.9844
5	0.3925	0.5462	5.0000	0.7271	4.9842
6	0.3541	0.5191	6.0000	0.7098	5.9920
7	0.3221	0.4962	7.0000	0.6954	6.9953
8	0.2939	0.4765	8.0000	0.6826	7.9968
9	0.2697	0.4568	9.0000	0.6711	8.9980
Two-step-look-ahead procedure (batch)					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
2	0.5834	0.6663	2.0000	0.8264	2.0000
3	0.5033	0.6165	3.0000	0.7798	3.0000
4	0.4406	0.5800	4.0000	0.7490	3.9844
5	0.3924	0.5462	5.0000	0.7271	4.9842
6	0.3525	0.5187	6.0000	0.7097	5.9920
7	0.3199	0.4955	7.0000	0.6951	6.9953
8	0.2912	0.4754	8.0000	0.6821	7.9968
9	0.2660	0.4574	9.0000	0.6703	8.9980
Two-step-look-ahead procedure					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
2	0.5834	0.6663	2.0000	0.8264	2.0000
3	0.5027	0.6165	3.0000	0.7798	3.0000
4	0.4405	0.5800	4.0000	0.7490	3.9844
5	0.3906	0.5462	5.0000	0.7270	4.9842
6	0.3510	0.5186	6.0000	0.7095	5.9920
7	0.3180	0.4945	7.0000	0.6947	6.9953
8	0.2894	0.4741	8.0000	0.6815	7.9969
9	0.2641	0.4542	9.0000	0.6695	8.9980
Three-step-look-ahead procedure					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
3	0.5027	0.6165	3.0000	0.7798	3.0000
4	0.4405	0.5800	4.0000	0.7490	3.9844
5	0.3906	0.5450	5.0000	0.7270	4.9842
6	0.3507	0.5163	6.0000	0.7095	5.9920
7	0.3172	0.4915	7.0000	0.6947	6.9953
8	0.2882	0.4701	8.0000	0.6814	7.9969
9	0.2626	0.4503	9.0000	0.6692	8.9981

Table 7.3: Simulation Results: Less Precise Prior Distribution (BLP)

One-step-look-ahead procedure					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
2	2.4511	2.5969	2.0000	3.5825	1.7500
3	1.9672	2.1358	3.0000	3.0745	2.6923
4	1.5706	1.7902	4.0000	2.6063	3.6512
5	1.2881	1.5462	5.0000	2.3065	4.5683
6	1.0766	1.3688	6.0000	2.1185	5.5131
7	0.9066	1.2273	7.0000	1.9829	6.5004
8	0.7655	1.1115	8.0000	1.8765	7.5009
9	0.6517	1.0154	9.0000	1.7922	8.5166
Two-step-look-ahead procedure (batch)					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
2	2.4505	2.5969	2.0000	3.5825	1.7500
3	1.9415	2.1358	3.0000	3.0745	2.6923
4	1.5419	1.7721	4.0000	2.6063	3.6512
5	1.2494	1.5053	5.0000	2.3065	4.5683
6	1.0252	1.3043	6.0000	2.1180	5.5131
7	0.8526	1.1507	7.0000	1.9811	6.5012
8	0.7077	1.0230	8.0000	1.8709	7.5072
9	0.5903	0.9216	9.0000	1.7848	8.5199
Two-step-look-ahead procedure					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
2	2.4505	2.5969	2.0000	3.5825	1.7500
3	1.9415	2.1358	3.0000	3.0745	2.6923
4	1.5419	1.7721	4.0000	2.6063	3.6512
5	1.2356	1.5006	5.0000	2.3064	4.5683
6	0.9836	1.2981	6.0000	2.1177	5.5178
7	0.7939	1.1381	7.0000	1.9775	6.5100
8	0.6473	1.0121	8.0000	1.8694	7.5116
9	0.5347	0.9119	9.0000	1.7818	8.5227
Three-step-look-ahead procedure					
Max. No. of	Without Restriction	Less Conservative		More Conservative	
Subject	Exp Loss	Exp Loss	E(N)	Exp Loss	E(N)
3	1.9226	2.1358	3.0000	3.0745	2.6923
4	1.5283	1.7721	4.0000	2.6063	3.6512
5	1.2154	1.5006	5.0000	2.3064	4.5683
6	0.9736	1.2965	6.0000	2.1152	5.5202
7	0.7796	1.1350	7.0000	1.9738	6.5143
8	0.6303	1.0063	8.0000	1.8653	7.5151
9	0.5159	0.9036	9.0000	1.7772	8.5248

7.2.1 Comparison between Strategies

The results are shown in Figure 7.1. As the prior distribution becomes less precise, the differences in performance between strategies become more apparent. The results imply that when prior knowledge is limited, it is beneficial to implement the two- or three-step-look-ahead procedure, rather than the one-step-look-ahead or two-step-look-ahead with batch sequential.

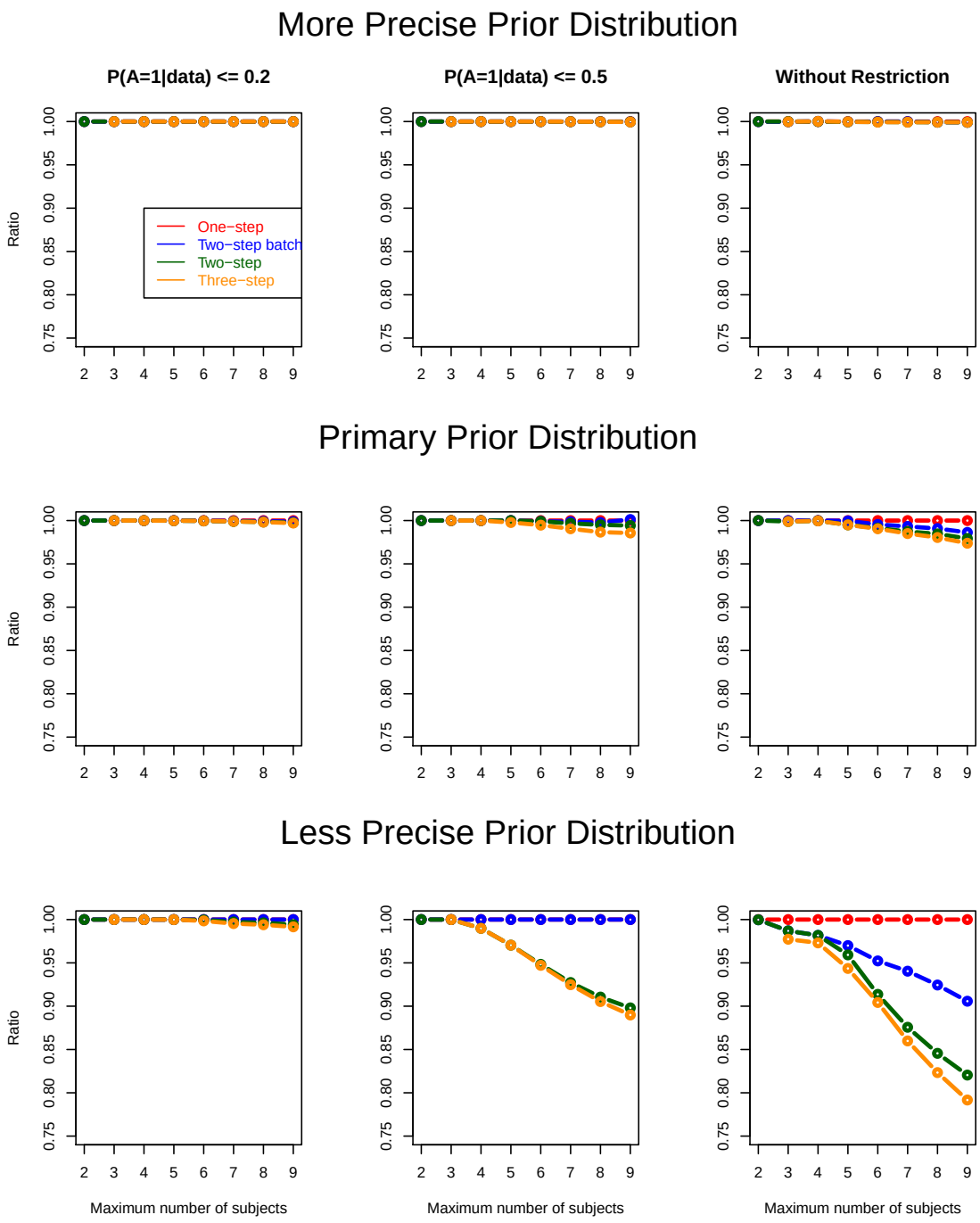

Figure 7.1: Ratio of Overall Expected Loss for Four Strategies

7.2.2 Comparison between Different Levels of Restriction on the Posterior Probability of Adverse Event

To evaluate the performance in designs with and without a restriction on toxicity, designs with the same strategy but different levels of restriction are compared. The results from one-step-look-ahead procedures using BMP, BPP, and BLP are presented in Figure 7.2. For each prior distribution, the more conservative the restriction, the higher the overall expected loss. For example, on the top panel of Figure 7.2 (more precise prior distribution), the overall expected loss for the more conservative restriction is 1.5 times higher than it is for no restriction. This difference is related to the expected number of subjects, because early stopping occurs more often when the more conservative restriction is applied.

The simulation results from two- and three-step-look-ahead can be found in Figures 7.3, 7.4, and 7.5.

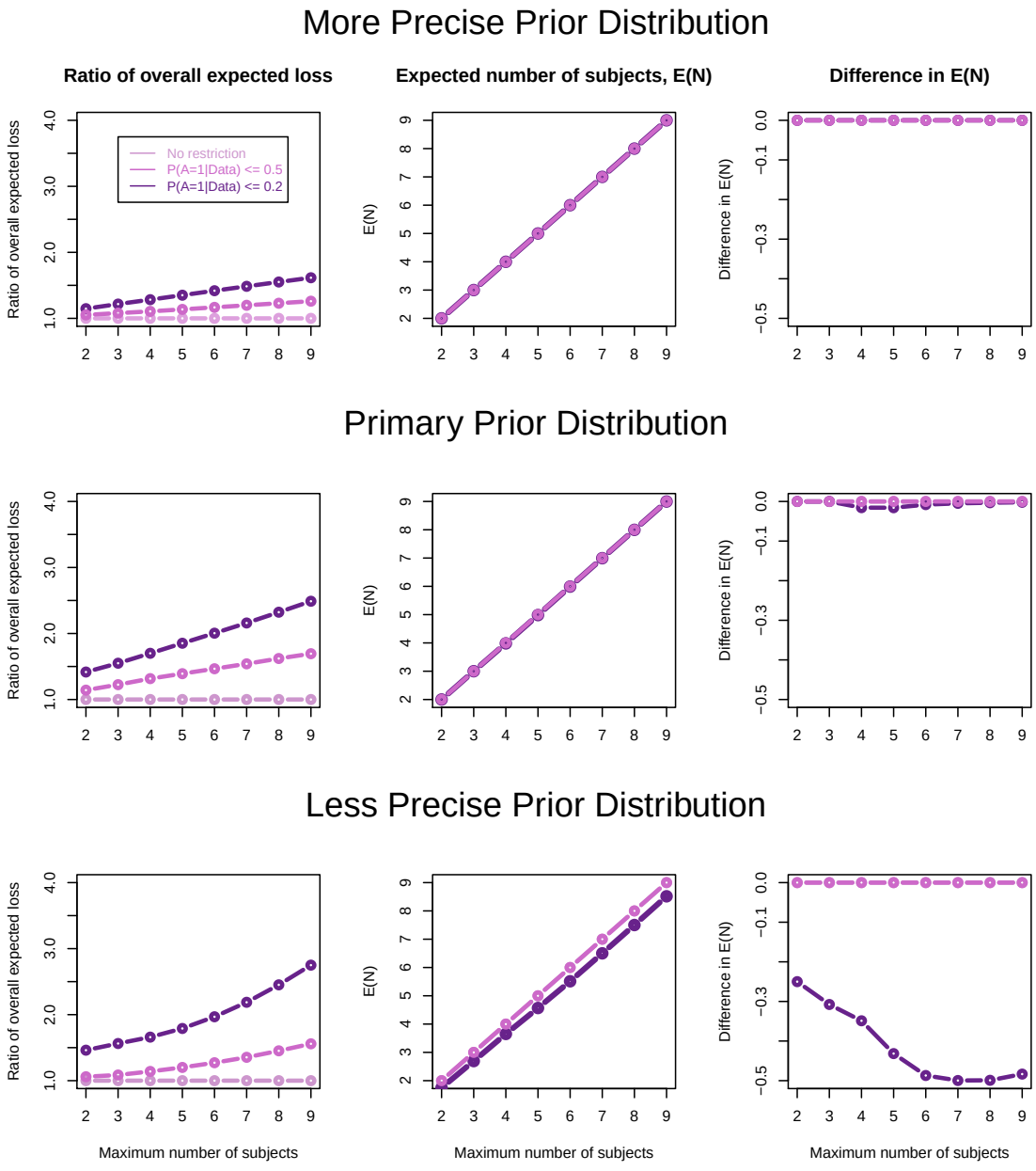

Figure 7.2: Comparison between Different Restriction on $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i)$: One-step-look-ahead Procedure

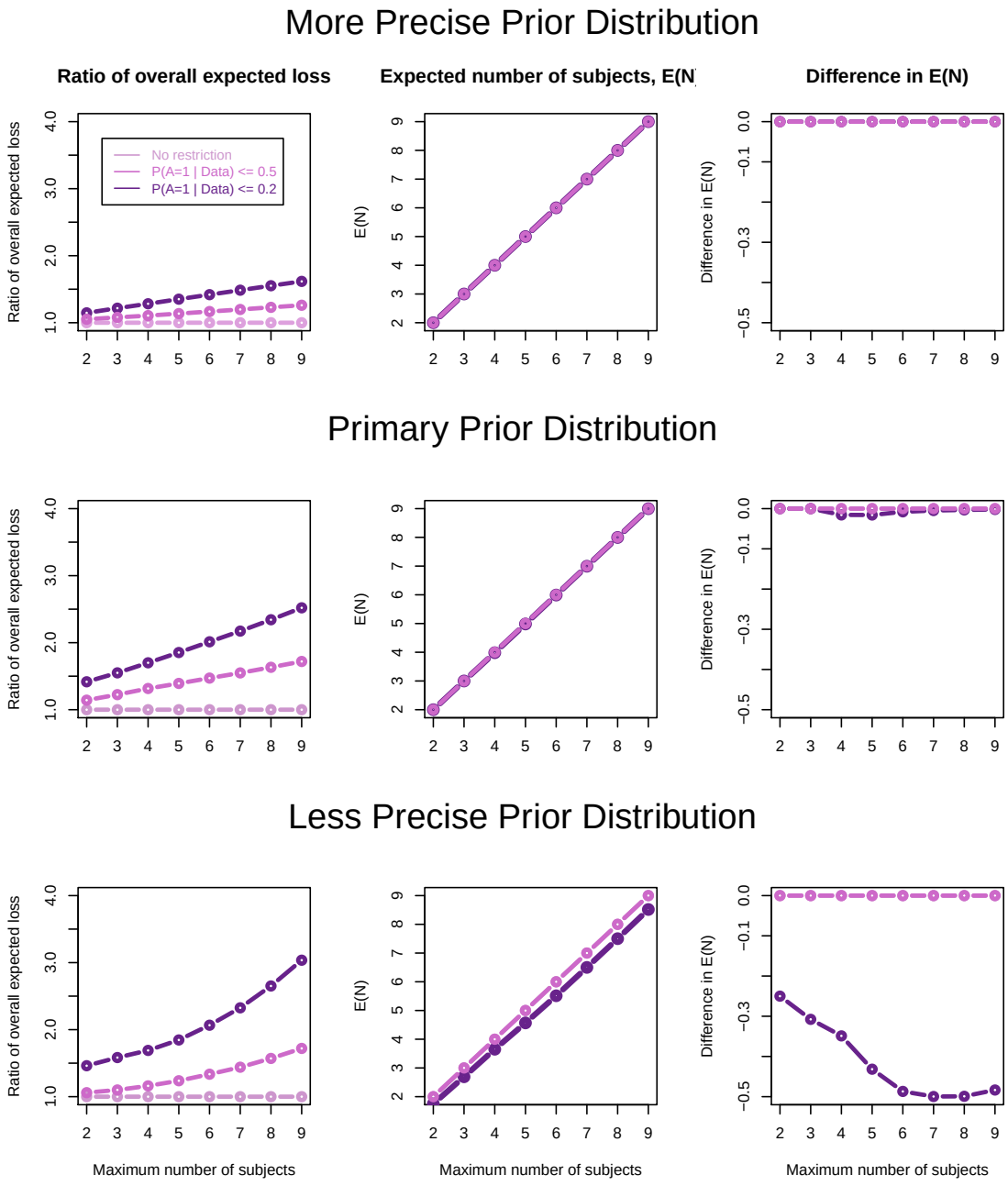

Figure 7.3: Comparison between Different Restriction on $\Pr(A = 1|\mathbf{h}^{(\mathbf{i-1})}, x_i)$: Two-step-look-ahead (Batch) Procedure

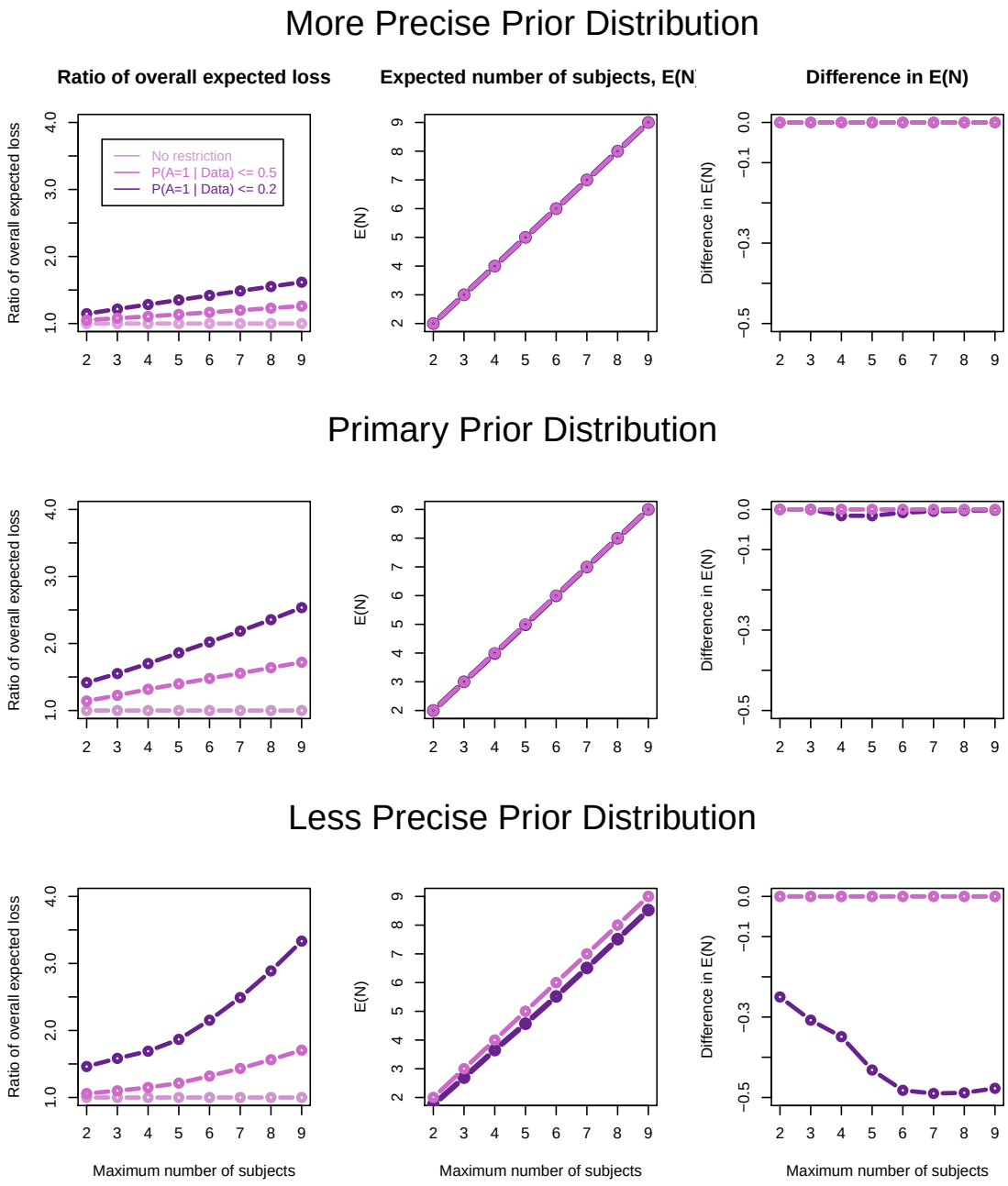

Figure 7.4: Comparison between Different Restriction on $\Pr(A = 1|\mathbf{h}^{(\mathbf{i-1})}, x_i)$: Two-step-look-ahead Procedure

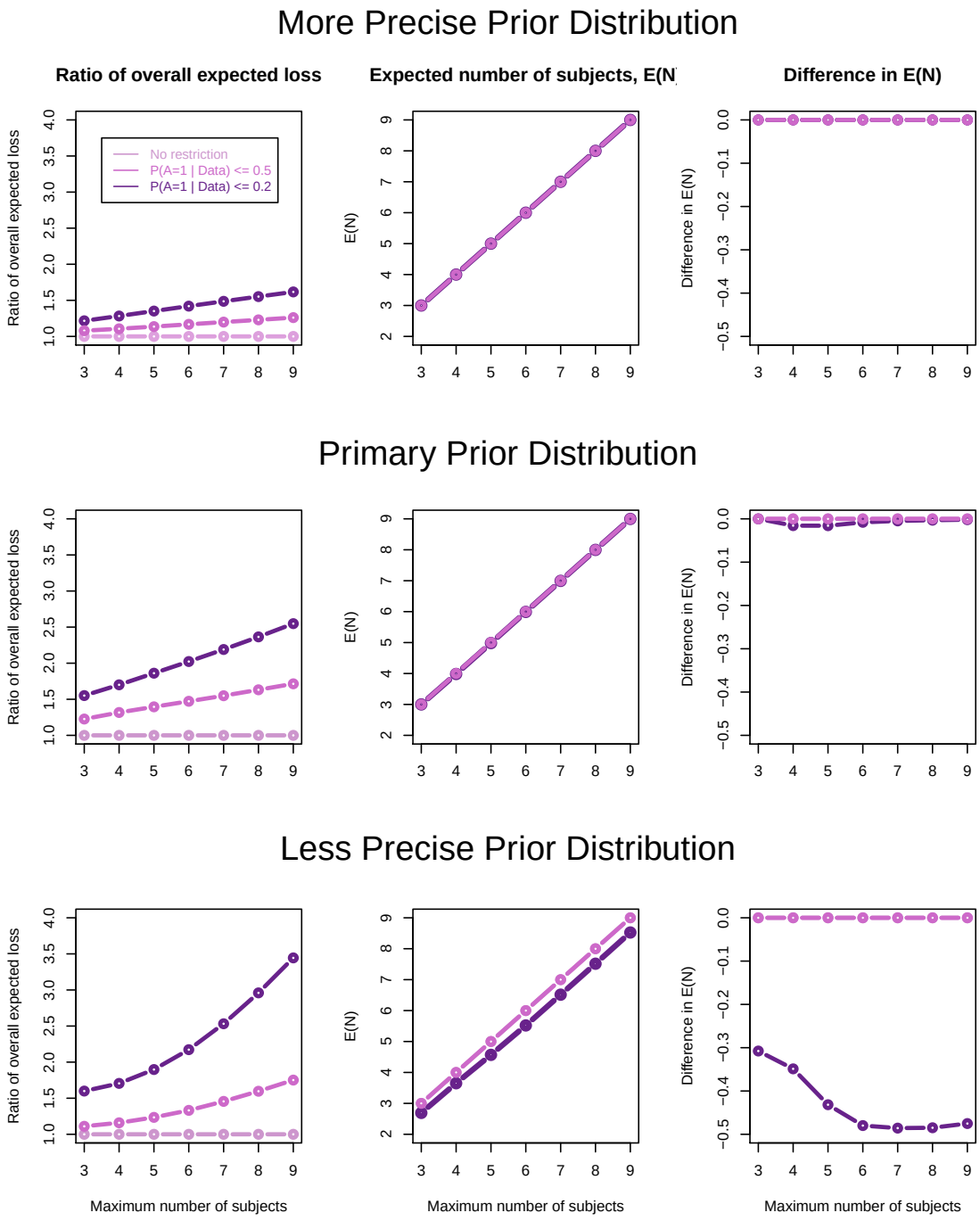

Figure 7.5: Comparison between Different Restriction on $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i)$: Three-step-look-ahead Procedure

7.3 Discussion

It is not surprising to observe that for a strategy, when the restriction on $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i)$ is more relaxed (when ϵ is larger), the overall expected loss is lower. This is due to the fact that when applying the restriction, the number of candidate doses is a subset of the candidate doses from no restriction case, and the expected loss of the optimal dose selected from this subset may be higher than the expected loss of the optimal dose selected from the candidate doses with no restriction. Therefore, decreasing the ϵ reduces the number of candidate doses.

It is also not surprising that when ϵ, is small (in our example, the smallest is $\epsilon = 0.20$), there is no optimal dose selected in some scenarios, and the trial stops. The trend is similar across the three sets of prior distributions. Note that when the vague prior distribution or BLP is used, the difference in the designs with different levels of restriction becomes more obvious.

Toxicity has not been incorporated into the loss, instead it has been constrained. This may be conceptually easier than quantifying the trade-off between toxicity and variance, but should be investigated further.

CHAPTER 8
OPTIMAL DESIGNS FOR THE MODEL INCORPORATING BOTH EFFICACY AND TOXICITY RESPONSES

8.1 Introduction

In this chapter, non-sequential designs are found based on the criteria from optimal design theory: the expected Fisher information matrix is optimized, and the inverse of this matrix corresponds to the asymptotic covariance matrix of the parameters. Two common design criteria, D and A criteria, are used. A locally D-optimal design maximizes the logarithm of the determinant of the Fisher information matrix evaluated at a single value of the unknown parameters. For Bayesian D-optimal designs, the expectation of the logarithm of the determinant of the Fisher information matrix is maximized, and the expectation is taken over the prior distribution of the parameters. Alternatively, the A-optimal design criteria minimize the sum of the asymptotic variances of functions of the parameters evaluated at a particular value, or minimize the expectation of the sum over a prior distribution. Since the $HCD50$ and $HCD90$ are the doses of interest in our motivating example, the A criterion is closely related to the sequential criterion used in the sequential strategies implemented in the H-Flu study (Chapters 3 and 6). The exact posterior variances are estimated for non-sequential strategies, and compared with the estimates from sequential strategies.

The best non-sequential designs are found for the bivariate model incorporating both efficacy and toxicity. A variety of scenarios are also evaluated, and

comparisons to sequential strategies are presented at the end of this chapter.

8.2 Model, Likelihood, and Fisher Information

Recall in Chapter 5 that $a_1, a_2, a_3 < 0$, $b_1, b_2, b_3 > 0$, $a_2 \leq a_3$, and $b_2 \leq b_3$,

$$\mu_k = -\Big(\frac{a_k \delta_k}{\log 9}\Big),\ \delta_k = \frac{\log 9}{b_k},\ k = 1, 2, 3.$$

Then the new model is:

$$\begin{aligned}
\text{logit}\Big\{ \Pr(C = 1|x_i) \Big\} &= \log\left(\frac{p_5(x_i)}{1 - p_5(x_i)}\right) = a_1 + b_1 x_i = \left(\frac{x_i - \mu_1}{\delta_1}\right)\log 9 \\
\text{logit}\Big\{ \Pr(A = 1|C = 0, x_i) \Big\} &= \log\left(\frac{p_2(x_i)}{p_1(x_i)}\right) = a_2 + b_2 x_i = \left(\frac{x_i - \mu_2}{\delta_2}\right)\log 9 \\
\text{logit}\Big\{ \Pr(A = 1|C = 1, x_i) \Big\} &= \log\left(\frac{p_4(x_i)}{p_3(x_i)}\right) = a_3 + b_3 x_i = \left(\frac{x_i - \mu_3}{\delta_3}\right)\log 9
\end{aligned}$$

Denote $\boldsymbol{\theta_1} = (a_1, b_1)^T$, $\boldsymbol{\theta_2} = (a_2, b_2)^T$, $\boldsymbol{\theta_3} = (a_3, b_3)^T$, and $\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{\theta_3})^T$. Let $\mathbf{y_i} = (y_{i1}, y_{i2}, y_{i3}, y_{i4})$, $\mathbf{p_i} = (p_{i1}, p_{i2}, p_{i3}, p_{i4})$, and $\mathbf{y_i} \sim \text{Multinomial}(1, \mathbf{p_i})$, where for N subjects, $\mathbf{y_1}, ..., \mathbf{y_N}$, are independent, $\sum_{j=1}^{4} p_{ij} = 1$ and

$$\Pr(\mathbf{y_1}, ..., \mathbf{y_N}) = \prod_{i=1}^{N}\prod_{j=1}^{4} p_{ij}^{y_{ij}}.$$

The D and A criteria involve the Fisher information matrix [50, 16], and the next section is the derivation of the Fisher information for the new model.

8.2.1 Derivation of the Fisher Information

The likelihood for data from the i^{th} subject is:

$$L_i(\mathbf{y_i}; \mathbf{p_i}) = p_{i1}^{y_{i1}} p_{i2}^{y_{i2}} p_{i3}^{y_{i3}} p_{i4}^{y_{i4}},$$

and the corresponding log likelihood is:

$$
\begin{aligned}
& l_i(\mathbf{y_i}; \mathbf{p_i}) = l_i(\mathbf{y_i}; \boldsymbol{\theta}) \\
& = y_{i1} \log(p_{i1}) + y_{i2} \log(p_{i2}) + y_{i3} \log(p_{i3}) + y_{i4} \log(p_{i4}) \\
& = y_{i1} \log \Big\{ \Big(\frac{1}{1 + \exp(a_1 + b_1 x_i)} \Big) \Big(\frac{1}{1 + \exp(a_2 + b_2 x_i)} \Big) \Big\} \\
& + y_{i2} \log \Big\{ \Big(\frac{1}{1 + \exp(a_1 + b_1 x_i)} \Big) \Big(\frac{\exp(a_2 + b_2 x_i)}{1 + \exp(a_2 + b_2 x_i)} \Big) \Big\} \\
& + y_{i3} \log \Big\{ \Big(\frac{\exp(a_1 + b_1 x_i)}{1 + \exp(a_1 + b_1 x_i)} \Big) \Big(\frac{1}{1 + \exp(a_3 + b_3 x_i)} \Big) \Big\} \\
& + y_{i4} \log \Big\{ \Big(\frac{\exp(a_1 + b_1 x_i)}{1 + \exp(a_1 + b_1 x_i)} \Big) \Big(\frac{\exp(a_3 + b_3 x_i)}{1 + \exp(a_3 + b_3 x_i)} \Big) \Big\} \\
& = y_{i1} \Big\{ - \log \big(1 + \exp(a_1 + b_1 x_i) \big) - \log \big(1 + \exp(a_2 + b_2 x_i) \big) \Big\} \\
& + y_{i2} \Big\{ - \log \big(1 + \exp(a_1 + b_1 x_i) \big) - \log \big(1 + \exp(a_2 + b_2 x_i) \big) + \big(a_2 + b_2 x_i \big) \Big\} \\
& + y_{i3} \Big\{ \big(a_1 + b_1 x_i \big) - \log \big(1 + \exp(a_1 + b_1 x_i) \big) - \log \big(1 + \exp(a_3 + b_3 x_i) \big) \Big\} \\
& + y_{i4} \Big\{ \big(a_1 + b_1 x_i \big) - \log \big(1 + \exp(a_1 + b_1 x_i) \big) \\
& \qquad - \log \big(1 + \exp(a_3 + b_3 x_i) \big) + \big(a_3 + b_3 x_i \big) \Big\}.
\end{aligned}
$$

Since

$$p_5(x_i) = \frac{\exp(a_1 + b_1 x_i)}{1 + \exp(a_1 + b_1 x_i)},$$

$$\frac{p_2(x_i)}{p_1(x_i) + p_2(x_i)} = \frac{\exp(a_2 + b_2 x_i)}{1 + \exp(a_2 + b_2 x_i)}, \text{ and}$$

$$\frac{p_4(x_i)}{p_3(x_i) + p_4(x_i)} = \frac{\exp(a_3 + b_3 x_i)}{1 + \exp(a_3 + b_3 x_i)},$$

the score functions for the i^{th} subject are:

$$\frac{\partial l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_1} = -p_5(x_i)(y_{i1}+y_{i2}) + (1-p_5(x_i))(y_{i3}+y_{i4}),$$

$$\frac{\partial l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial b_1} = x_i\Big[- p_5(x_i)(y_{i1}+y_{i2}) + (1-p_5(x_i))(y_{i3}+y_{i4})\Big],$$

$$\frac{\partial l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_2} = -\left(\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)y_{i1} + \left(1-\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)y_{i2},$$

$$\frac{\partial l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial b_2} = x_i\Big[- \left(\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)y_{i1} + \left(1-\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)y_{i2}\Big],$$

$$\frac{\partial l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_3} = -\left(\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)y_{i3} + \left(1-\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)y_{i4},$$

$$\frac{\partial l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial b_3} = x_i\Big[- \left(\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)y_{i3} + \left(1-\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)y_{i4}\Big].$$

Taking the second derivatives,

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_1^2} = -p_5(x_i)(1-p_5(x_i)),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial b_1^2} = -x_i^2 p_5(x_i)(1-p_5(x_i)),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_2^2} = -(y_{i1}+y_{i2})\left(\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)\left(1-\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial b_2^2} = -x_i^2(y_{i1}+y_{i2})\left(\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)\left(1-\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_3^2} = -(y_{i3}+y_{i4})\left(\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)\left(1-\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial b_3^2} = -x_i^2(y_{i3}+y_{i4})\left(\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)\left(1-\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_1 b_1} = -x_i p_5(x_i)(1-p_5(x_i)),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_2 b_2} = -x_i(y_{i1}+y_{i2})\left(\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right)\left(1-\frac{p_2(x_i)}{p_1(x_i)+p_2(x_i)}\right),$$

$$\frac{\partial^2 l_i(\mathbf{y_i};\boldsymbol{\theta})}{\partial a_3 b_3} = -x_i(y_{i3}+y_{i4})\left(\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right)\left(1-\frac{p_4(x_i)}{p_3(x_i)+p_4(x_i)}\right).$$

The other partial derivatives are equal to 0.

Let

$$\mathbf{H_{1i}}(\boldsymbol{\theta_1}, x_i) = \begin{pmatrix} \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_1^2} & \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_1 b_1} \\ \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_1 b_1} & \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial b_1^2} \end{pmatrix},$$

$$\mathbf{H_{2i}}(\boldsymbol{\theta_2}, x_i) = \begin{pmatrix} \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_2^2} & \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_2 b_2} \\ \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_2 b_2} & \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial b_2^2} \end{pmatrix},$$

and

$$\mathbf{H_{3i}}(\boldsymbol{\theta_3}, x_i) = \begin{pmatrix} \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_3^2} & \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_3 b_3} \\ \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial a_3 b_3} & \frac{\partial^2 l_i(\boldsymbol{\theta})}{\partial b_3^2} \end{pmatrix}.$$

Then for the i^{th} subject, the Hessian matrix at dose x_i is:

$$\mathbf{H_i}(\boldsymbol{\theta}, x_i) = \begin{pmatrix} \mathbf{H_{1i}}(\boldsymbol{\theta_1}, x_i) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H_{2i}}(\boldsymbol{\theta_2}, x_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H_{3i}}(\boldsymbol{\theta_3}, x_i) \end{pmatrix}.$$

The contribution to the Fisher information matrix from the i^{th} subject is

$$\boldsymbol{\mathcal{I}_i}(\boldsymbol{\theta}, x_i) = -E\big[\mathbf{H_i}(\boldsymbol{\theta}, x_i)\big] = \begin{pmatrix} \boldsymbol{\mathcal{I}_{1i}}(\boldsymbol{\theta_1}, x_i) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{I}_{2i}}(\boldsymbol{\theta_2}, x_i) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{I}_{3i}}(\boldsymbol{\theta_3}, x_i) \end{pmatrix},$$

where

$$\begin{aligned}
\boldsymbol{\mathcal{I}_{1i}}(\boldsymbol{\theta_1}, x_i) &= -E\big[\mathbf{H_{1i}}(\boldsymbol{\theta_1}, x_i)\big] \\
&= p_5(x_i)(1 - p_5(x_i)) \begin{pmatrix} 1 & x_i \\ x_i & x_i^2 \end{pmatrix}, \\
\boldsymbol{\mathcal{I}_{2i}}(\boldsymbol{\theta_2}, x_i) &= -E\big[\mathbf{H_{2i}}(\boldsymbol{\theta_2}, x_i)\big] \\
&= (1 - p_5(x_i)) \left(\frac{p_2(x_i)}{p_1(x_i) + p_2(x_i)}\right) \left(1 - \frac{p_2(x_i)}{p_1(x_i) + p_2(x_i)}\right) \begin{pmatrix} 1 & x_i \\ x_i & x_i^2 \end{pmatrix}, \\
\boldsymbol{\mathcal{I}_{3i}}(\boldsymbol{\theta_3}, x_i) &= -E\big[\mathbf{H_{3i}}(\boldsymbol{\theta_3}, x_i)\big] \\
&= p_5(x_i) \left(\frac{p_4(x_i)}{p_3(x_i) + p_4(x_i)}\right) \left(1 - \frac{p_4(x_i)}{p_3(x_i) + p_4(x_i)}\right) \begin{pmatrix} 1 & x_i \\ x_i & x_i^2 \end{pmatrix}.
\end{aligned}$$

Denote n be the number of observations, and $\eta = (x_1, ..., x_n)$ be a design. Then $\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}, \eta) = \sum_i \boldsymbol{\mathcal{I}_i}(\boldsymbol{\theta}, x_i)$, and under regulatory conditions, as $n \to \infty$, $(n\boldsymbol{\mathcal{I}}(\boldsymbol{\theta}, \eta))^{-1/2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{D} N(\mathbf{0}, \mathbf{I_6})$. Note that although there are six unknown parameters, two distinct design points are sufficient to give a non-singular Hessian matrix.

Reparametrize $\boldsymbol{\theta}$ as $\boldsymbol{\xi}$, $\boldsymbol{\xi} = (\mu_1, \mu_1 + \delta_1, \mu_2, \mu_2 + \delta_2, \mu_3, \mu_3 + \delta_3)^T$, and define

$\mathbf{J}$ as below,

$$\mathbf{J} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} = \begin{pmatrix} -\frac{\log 9(\delta_1+\mu_1)}{\delta_1^2} & \frac{\mu_1 \log 9}{\delta_1^2} & 0 & 0 & 0 & 0 \\ \frac{\log 9}{\delta_1^2} & -\frac{\log 9}{\delta_1^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\log 9(\delta_2+\mu_2)}{\delta_2^2} & \frac{\mu_2 \log 9}{\delta_2^2} & 0 & 0 \\ 0 & 0 & \frac{\log 9}{\delta_2^2} & -\frac{\log 9}{\delta_2^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{\log 9(\delta_3+\mu_3)}{\delta_3^2} & \frac{\mu_3 \log 9}{\delta_3^2} \\ 0 & 0 & 0 & 0 & \frac{\log 9}{\delta_3^2} & -\frac{\log 9}{\delta_3^2} \end{pmatrix}.$$

Then using the Delta-method, the posterior variance-covariance matrix of $\boldsymbol{\xi}$ can be approximated by $\mathbf{J}^T \boldsymbol{\mathcal{I}}(\boldsymbol{\theta}, \eta)^{-1} \mathbf{J}$ [33].

8.3 Approximate and Exact Optimal Designs

Denote n_j $(j = 1, ..., m)$ be the number of subjects at dose x_j, $\sum_j n_j = n$. Let $w_j = \frac{n_j}{n}$, $\sum_j w_j = 1$, and for a design η, the normalized Fisher information for all subjects is $M(\boldsymbol{\theta}, \eta) = \sum_j w_j \boldsymbol{\mathcal{I}}_{\boldsymbol{j}}(\boldsymbol{\theta}, x_j)$. Following the usual practice in optimal designs, define η to be a probability measure on a compact design space $\mathcal{X}$ and $\mathcal{H}$ to be the set of all η. This relaxes the integer programming problem, where n_j are constrained to be integers, to a continuous problem which finds an "approximate" design, where w_j, $j = 1, ..., m$, are an non-negative weights that sum to one [50, 4]. In any implementation, "exact" designs where n is the sample size and nw_j are integers for all $j = 1, ..., m$, are needed. Rounding of approximated designs to exact designs is required (page 308, [44]).

8.4 Bayesian Optimality

For Bayesian optimality, it is assumed that the posterior distribution of the parameter can be approximated by normal distribution [7, 17]. Let $\hat{\boldsymbol{\theta}}$ be the MLE of $\boldsymbol{\theta}$, then approximately

$$\boldsymbol{\theta}|\mathbf{y}, \eta \sim N\big(\hat{\boldsymbol{\theta}}, (\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}}, \eta))^{-1}\big).$$

8.4.1 Bayesian D-optimality

Following the notation from Chaloner and Verdinelli [17], let $\phi_1(\eta)$ be the criterion value for Bayesian D-optimality, and the D-optimal design is to maximize the log of the determinant of $M(\boldsymbol{\theta}, \eta)$, $\phi_1(\eta)$:

$$\phi_1(\eta) = E_{\boldsymbol{\theta}}\big\{ \log\big[det\big(M(\boldsymbol{\theta}, \eta)\big)\big]\big\} = \int \log\big[det\big(M(\boldsymbol{\theta}, \eta)\big)\big]\ p(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

where $p(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$. Define for $l = 1, 2, 3$,

$$\begin{aligned}
r_{j1} &= p_5(x_j)(1 - p_5(x_j)), \\
r_{j2} &= \left(\frac{p_2(x_j)}{p_1(x_j) + p_2(x_j)}\right)\left(1 - \frac{p_2(x_j)}{p_1(x_j) + p_2(x_j)}\right), \\
r_{j3} &= \left(\frac{p_4(x_j)}{p_3(x_j) + p_4(x_j)}\right)\left(1 - \frac{p_4(x_j)}{p_3(x_j) + p_4(x_j)}\right), \\
t_l &= \sum_j w_j r_{jl}, \\
\bar{x}_l &= \frac{\sum_j w_j r_{jl} x_j}{t_l}, \\
s_l &= \sum_j w_j r_{jl} (x_j - \bar{x}_l)^2.
\end{aligned}$$

Adapting from Chaloner and Larntz [16], and the property of block diagonal matrix, the determinant of $M(\boldsymbol{\theta}, \eta)$ in the model is:

$$\begin{aligned} det\big(M(\boldsymbol{\theta}, \eta)\big) &= \Big(det\big(M(\boldsymbol{\theta_1}, \eta)\big)\Big)\Big(det\big(M(\boldsymbol{\theta_2}, \eta)\big)\Big)\Big(det\big(M(\boldsymbol{\theta_3}, \eta)\big)\Big) \\ &= (b_1^2 t_1 s_1)(b_2^2 t_2 s_2)(b_3^2 t_3 s_3). \end{aligned} \tag{8.1}$$

8.4.2 Bayesian c- and Bayesian A-optimality

Suppose a function of the coefficients, $g(\boldsymbol{\theta})$, is of interest, and denote the vector, $c(\boldsymbol{\theta})$, be the gradient vector of $g(\boldsymbol{\theta})$, the Bayesian c-optimal design is to maximize $\phi_2(\eta)$:

$$\phi_2(\eta) = E_{\boldsymbol{\theta}}\big\{c(\boldsymbol{\theta})^T M(\boldsymbol{\theta}, \eta)^{-1} c(\boldsymbol{\theta})\big\} = -\int c(\boldsymbol{\theta})^T M(\boldsymbol{\theta}, \eta)^{-1} c(\boldsymbol{\theta})\ p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

If the goal is to estimate $g_i(\boldsymbol{\theta})$, functions of the coefficients, Bayesian A-optimality corresponding to maximizing

$$\phi_{2A}(\eta) = \sum_q \Big\{ -\int c_q(\boldsymbol{\theta})^T M(\boldsymbol{\theta}, \eta)^{-1} c_q(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \Big\},$$

and this is equivalent to minimize the sum of square error. In the H-Flu example, both the $HCD50$ and $HCD90$ are of interest, and it can be shown that the corresponding Bayesian A optimal criterion value is

$$\begin{aligned} \phi_{2A}(\eta) &= -E_{\boldsymbol{\theta_1}}\big\{b_1^{-2}[t_1^{-1} + (\bar{x}_1 - \mu_1)^2 s_1^{-1}]\big\} \\ &\quad -E_{\boldsymbol{\theta_1}}\big\{b_1^{-2}[t_1^{-1} + (\log 9 - b_1(\bar{x}_1 - \mu_1))^2 b_1^{-2} s_1^{-1}]\big\}. \end{aligned} \tag{8.2}$$

8.5 Local Optimality

Alternatively, using a best guess for $\boldsymbol{\theta}$ to approximate the expected utility is referred to as locally optimal design. Suppose $\boldsymbol{\theta_0}$ represents the best guess, the

locally D-optimal design is the design η that maximizes Equation 8.1 evaluated at $\boldsymbol{\theta_0}$. That is, it does not require taking the expectation.

$$\phi_{1\boldsymbol{\theta_0}}(\eta) = det\Big\{ M(\boldsymbol{\theta_0}, \eta) \Big\}.$$

The locally c-optimal design maximizes $\phi_{2\boldsymbol{\theta_0}}(\eta)$:

$$\begin{aligned}\phi_{2\boldsymbol{\theta_0}}(\eta) &= -c(\boldsymbol{\theta_0})^T M(\boldsymbol{\theta_0}, \eta)^{-1} c(\boldsymbol{\theta_0}) \\ &= -\mathrm{tr}\Big(c(\boldsymbol{\theta_0}) c(\boldsymbol{\theta_0})^T M(\boldsymbol{\theta_0}, \eta)^{-1} \Big).\end{aligned}$$

The locally A-optimal design maximizes $\phi_{2A\boldsymbol{\theta_0}}(\eta)$:

$$\begin{aligned}\phi_{2A\boldsymbol{\theta_0}}(\eta) &= -\sum_q c_q(\boldsymbol{\theta_0})^T M(\boldsymbol{\theta_0}, \eta)^{-1} c_q(\boldsymbol{\theta_0}) \\ &= -\sum_q \mathrm{tr}\Big(c_q(\boldsymbol{\theta_0}) c_q(\boldsymbol{\theta_0})^T M(\boldsymbol{\theta_0}, \eta)^{-1} \Big).\end{aligned}$$

For the locally A-optimal criterion that minimizes the sum of the asymptotic variances of the $HCD50$ and the $HCD90$, as Equation 8.3 below, evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta_0}$:

$$\begin{aligned}\phi_{2A\boldsymbol{\theta_0}}(\eta) \;=\; & -\big\{ b_1^{-2}[t_1^{-1} + (\bar{x}_1 - \mu_1)^2 s_1^{-1}] \\ & + b_1^{-2}[t_1^{-1} + (\log 9 - b_1(\bar{x}_1 - \mu_1))^2 b_1^{-2} s_1^{-1}] \big\}. \qquad (8.3)\end{aligned}$$

8.6 Efficiency of One Design Measure Compared to Another Design Measure

To compare approximate design measures, the efficiency defined by Clyde and Chaloner [20] is used. The D-efficiency of the design η compared to η^* is

$$\mathrm{Eff}_D(\eta, \eta^*) = \exp\{ \frac{(\phi_1(\eta) - \phi_1(\eta^*))}{p} \},$$

where p is the number of unknown parameters, and in this case, $p = 6$. For A-optimality, the A-efficiency of η compared to η^* is

$$\text{Eff}_A(\eta, \eta^*) = \frac{\phi_{2A}(\eta^*)}{\phi_{2A}(\eta)}.$$

The value of efficiency is the factor that the sample size of η must be multiplied by to give the same criterion value as η^*. For example, if η^* is the D-optimal design, and $\text{Eff}_D(\eta, \eta^*) = 0.90$, then the optimal design η^* with a sample size of $0.90n$ gives the same D-criterion value as η with a sample size n, for any integer n, assuming the approximate is being considered where designs are probability measures on a set $\mathcal{X}$ as in Section 8.3.

8.7 Algorithm for Bayesian and Local Optimality

For the H-Flu example, different optimality criteria are applied to find the best designs. There are six parameters in the model, and each Bayesian criterion involves multiple integration which adds to the complexity in computing. The method of generating prior samples is implemented to compute the D and A criteria. The algorithm for Bayesian optimal designs is given below.

8.7.1 Algorithm for Bayesian Optimality

1. Generate the prior samples, $(\mu_{1k}, \delta_{1k}, \mu_{2k}, \delta_{2k}, \mu_{3k}, \delta_{3k})$, $k = 1, ..., N$. In the simulation, the primary prior distribution is used to generate the prior samples.

2. For $j = 1, 2, 3$, compute

$$a_{jk} = -\mu_{jk} b_j \text{ and } b_{jk} = \frac{\log 9}{\delta_{jk}}.$$

3. Define

$$
\begin{aligned}
p_{5k}(x_j) &= \frac{\exp(a_{1k} + b_{1k}x_j)}{1 + \exp(a_{1k} + b_{1k}x_j)} \\
p_{1k}(x_j) &= (1 - p_{5k}(x_j))\left(\frac{1}{1 + \exp(a_{2k} + b_{2k}x_j)}\right) \\
p_{2k}(x_j) &= (1 - p_{5k}(x_j))\left(\frac{\exp(a_{2k} + b_{2k}x_j)}{1 + \exp(a_{2k} + b_{2k}x_j)}\right) \\
p_{3k}(x_j) &= p_{5k}(x_j)\left(\frac{1}{1 + \exp(a_{3k} + b_{3k}x_j)}\right) \\
p_{4k}(x_j) &= p_{5k}(x_j)\left(\frac{\exp(a_{3k} + b_{3k}x_j)}{1 + \exp(a_{3k} + b_{3k}x_j)}\right).
\end{aligned}
$$

Then for $l = 1, 2, 3$ and any design represented by m non-negative weights that sum to one, w_j, $j = 1, ..., m$, $w_j \geq 0$, $\sum_j w_j = 1$. The value of m is fixed at a particular integer, $m = 2, 3, ...$, and then best design for each m is found.

$$
\begin{aligned}
r_{j1k} &= p_{5k}(x_j)(1 - p_{5k}(x_j)) \\
r_{j2k} &= \left(\frac{p_{2k}(x_j)}{p_{1k}(x_j) + p_{2k}(x_j)}\right)\left(1 - \frac{p_{2k}(x_j)}{p_{1k}(x_j) + p_{2k}(x_j)}\right) \\
r_{j3k} &= \left(\frac{p_{4k}(x_j)}{p_{3k}(x_j) + p_{4k}(x_j)}\right)\left(1 - \frac{p_{4k}(x_j)}{p_{3k}(x_j) + p_{4k}(x_j)}\right) \\
t_{lk} &= \sum_j w_j r_{jlk} \\
\bar{x}_{lk} &= \frac{\sum_j w_j r_{jlk} x_j}{t_{lk}} \\
s_{lk} &= \sum_j w_j r_{jlk}(x_j - \bar{x}_{lk})^2.
\end{aligned}
$$

4. For each m, starting with $m = 2$ compute the criteria, $\phi_1(\eta)$ and $\phi_{2A}(\eta)$:

(a) For Bayesian D-optimality,

$$\begin{aligned}\phi_1(\eta) &= E_\theta\big\{\log((b_1^2 t_1 s_1)(b_2^2 t_2 s_2)(b_3^2 t_3 s_3))\big\} \\ &= \frac{1}{N}\sum_k (\log(b_{1k}^2 t_{1k} s_{1k})(b_{2k}^2 t_{2k} s_{2k})(b_{3k}^2 t_{3k} s_{3k})).\end{aligned}$$

(b) For Bayesian A-optimality,

$$\begin{aligned}\phi_{2A}(\eta) &= \sum_q \Big\{ -\int c_q(\boldsymbol{\theta})^T M(\boldsymbol{\theta},\eta)^{-1} c_q(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}\Big\} \\ &= -\frac{1}{N}\sum_{q=1}^{2}\sum_k \Big\{ c_q(\boldsymbol{\theta})^T M(\boldsymbol{\theta},\eta)^{-1} c_q(\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}\Big\} \\ &= -\frac{1}{N}\Big\{ \sum_k \big\{ b_{1k}^{-2}[t_{1k}^{-1} + (\bar{x}_{1k} - \mu_{1k})^2 s_{1k}^{-1}]\big\} \\ &\quad + \sum_k \big\{ b_{1k}^{-2}[t_{1k}^{-1} + (\log 9 - b_{1k}(\bar{x}_{1k} - \mu_{1k}))^2 b_{1k}^{-2} s_{1k}^{-1}]\big\}\Big\}.\end{aligned}$$

Then find the corresponding $(x_1, ..., x_m)$ and $(w_1, .., w_m)$ that maximize $\phi_1(\eta)$ or $\phi_{2A}(\eta)$. If $m = 2$, track the criterion value, then repeat step 4 for $m = 3$. For $m \geq 3$, if the criterion value increases from m to $(m + 1)$, increases to $(m+2)$. When criterion value cease to increase significantly, stop and proceed to step 5.

5. Choose the design with the highest value of the criterion and examine the directional derivatives. For a candidate design η, to verify if it is optimal, the directional derivatives is computed [16], $d(\eta, x)$. Denote z_g $(g = 1, ..., G)$ be

any dose and $1.5 \leq z_g \leq 5$, and define

$$
\begin{aligned}
r_{g1k} &= p_{5k}(z_g)(1 - p_{5k}), \\
r_{g2k} &= \left(\frac{p_{2k}(z_g)}{p_{1k}(z_g) + p_{2k}(z_g)}\right)\left(1 - \frac{p_{2k}(z_g)}{p_{1k}(z_g) + p_{2k}(z_g)}\right), \\
r_{g3k} &= \left(\frac{p_{4k}(z_g)}{p_{3k}(z_g) + p_{4k}(z_g)}\right)\left(1 - \frac{p_{4k}(z_g)}{p_{3k}(z_g) + p_{4k}(z_g)}\right),
\end{aligned}
$$

the directional derivatives are:

(a) For Bayesian D-optimality,

$$
d_1(\eta, z_g) = \frac{1}{N}\left(\sum_{l=1}^{3}\sum_{k}\left\{r_{glk}[t_{lk}^{-1} + (\bar{x}_{lk} - z_g)^2 s_{lk}^{-1}]\right\}\right) - q,
$$

where q is the number of parameters.

(b) For Bayesian A-optimality,

$$
\begin{aligned}
d_{2A}(\eta, z_g) = &\left\{\frac{1}{N}\sum_{k}\left(r_{g1k}(b_{1k}s_{1k}t_{1k})^{-2}[t_{1k}(\bar{x}_{1k} - z_g)(\bar{x}_{1k} - \mu_{1k}) + s_{1k}]^2\right)\right. \\
&\left. - \frac{1}{N}\sum_{k}\left(b_{1k}^{-2}[t_{1k}^{-1} + (\bar{x}_{1k} - \mu_{1k})^2 s_{1k}^{-1}]\right)\right\} \\
&+ \left\{\frac{1}{N}\sum_{k}\left(r_{g1k}(b_{1k}^2 s_{1k}t_{1k})^{-2}[t_{1k}(\bar{x}_{1k} - z_g)(b_{1k}(\bar{x}_{1k} - \mu_{1k})\right.\right. \\
&\qquad \left. - \log 9) + b_{1k}s_{1k}]^2\right) \\
&\qquad \left. - \frac{1}{N}\sum_{k}\left(b_{1k}^{-2}[t_{1k}^{-1} + (\log 9 - b_{1k}(\bar{x}_{1k} - \mu_{1k}))^2 b_{1k}^{-2} s_{1k}^{-1}]\right)\right\}.
\end{aligned}
$$

If a design is optimal, the General Equivalence Theorem from optimal design theory shows that the value of the directional derivatives at those doses will be zero, and negative elsewhere [16]. In practice, the derivatives will be close to zero at a design close to the optimal design. Examining the directional

derivatives over all possible doses is a method to validate the optimality of a design. In some cases, the weight, w_j, is not close to zero, and the derivative at that point is negative, in which case setting that w_j to zero is examined.

8.7.2 Algorithm for Local Optimality

The locally optimal designs can be found using a similar algorithm. The only difference is to substitute the best guess, $\theta_{\mathbf{0}}$, in $\phi_1(\eta)$ and $\phi_{2A}(\eta)$ and let $k = 1$. The directional derivative is evaluated at $\theta_{\mathbf{0}}$.

8.8 Designs with Unrestricted Doses

In R, there are several packages for optimization. The general-purpose optimization based on quasi-Newton algorithm with box constraints [14] is applied here, and the corresponding R functions is `optim` [45]. The box constraints allow an upper and/or lower bound for each variable. In the H-Flu example, the doses are constrained to lie within the region between 1.5 and 5.0, and the weights are any value between 0 and 1 that sum to 1.0 so that η is a probability measure.

8.8.1 Bayesian D-optimality

Similar to previous simulations for the bivariate model (see Chapter 7) , the prior samples are generated using the primary prior distribution, BPP. That is for $l = 1, 2, 3$ and $k = 1, ..., 1000$,

$$v_{lk} \sim \text{Beta}(5, 5), \quad z_{lk} \sim \text{Beta}(1, 6)$$

$$\mu_{lk} = 3.5v_l + 1.5, \quad \delta_{lk} = 5z_{lk} + 0.1, \quad \gamma_{lk} = \mu_{lk} + \delta_{lk}.$$

The results from Bayesian D-optimality are shown in Figure 8.1 and Table 8.1. Denote η_m^D to be the best design for m points, $m = 2, ..., 12$. As m increases, the range of doses increases. The dose and weight assignment for the Bayesian D-optimal designs are to put approximate equal weight on each dose. For example, in η_4^D roughly 25% of the weight put on each of the doses, 2.4662, 2.9911, 3.5344 and 4.0739. Based on the value of $\phi_1(\eta_m^D)$, either η_{11}^D or η_{12}^D may be the Bayesian D-optimal design for this example. The value of directional derivatives are evaluated and plotted. The plots of directional derivatives for η_{11}^D and η_{12}^D are presented in Figure 8.2. As seen in the graphs, the values of the directional derivatives at the support points are close to 0 for η_{12}^D, although neither are "the optimal design" because the derivative is clearly positive in some regions. The efficiency for η_m, $m = 2, ..., 11$, with respect to η_{12}^D is computed and presented in Figure 8.3. The graph shows that the performance of the design is very similar when $m \geq 9$, and the improvement by increasing m further is likely small.

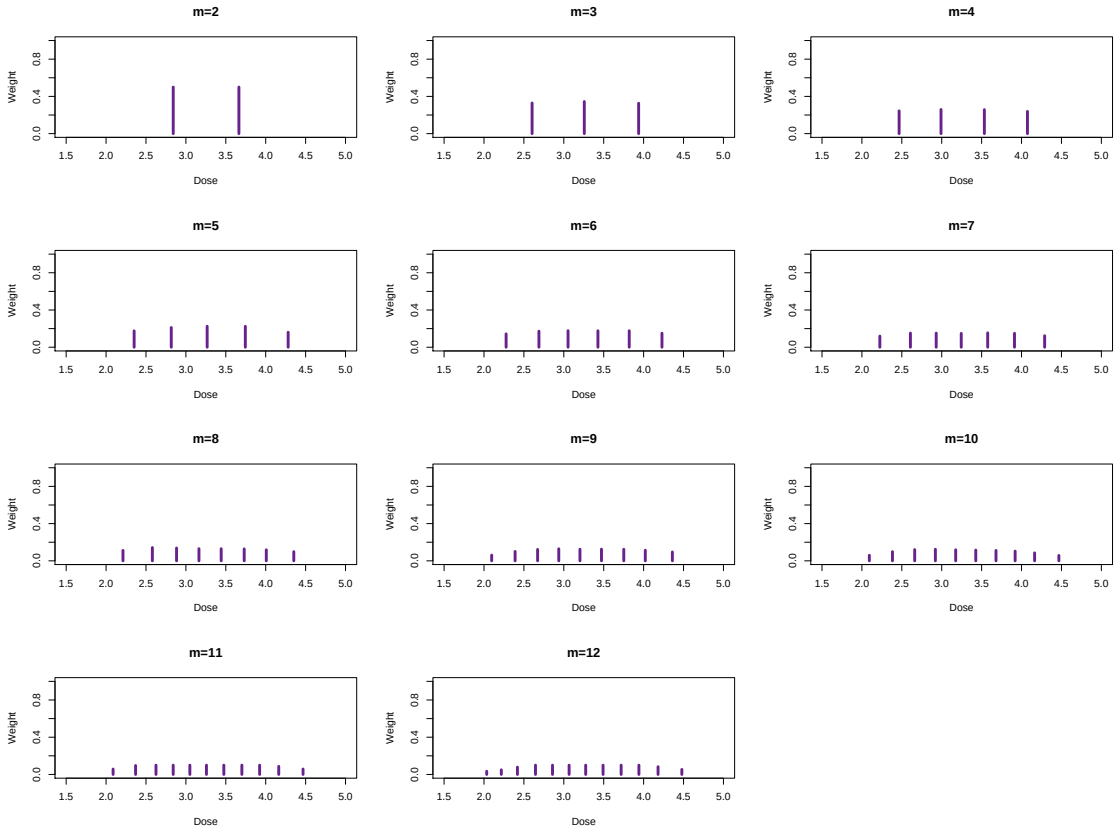

Figure 8.1: The Best m-point Bayesian D-optimal Designs, $m = 2$ to 12

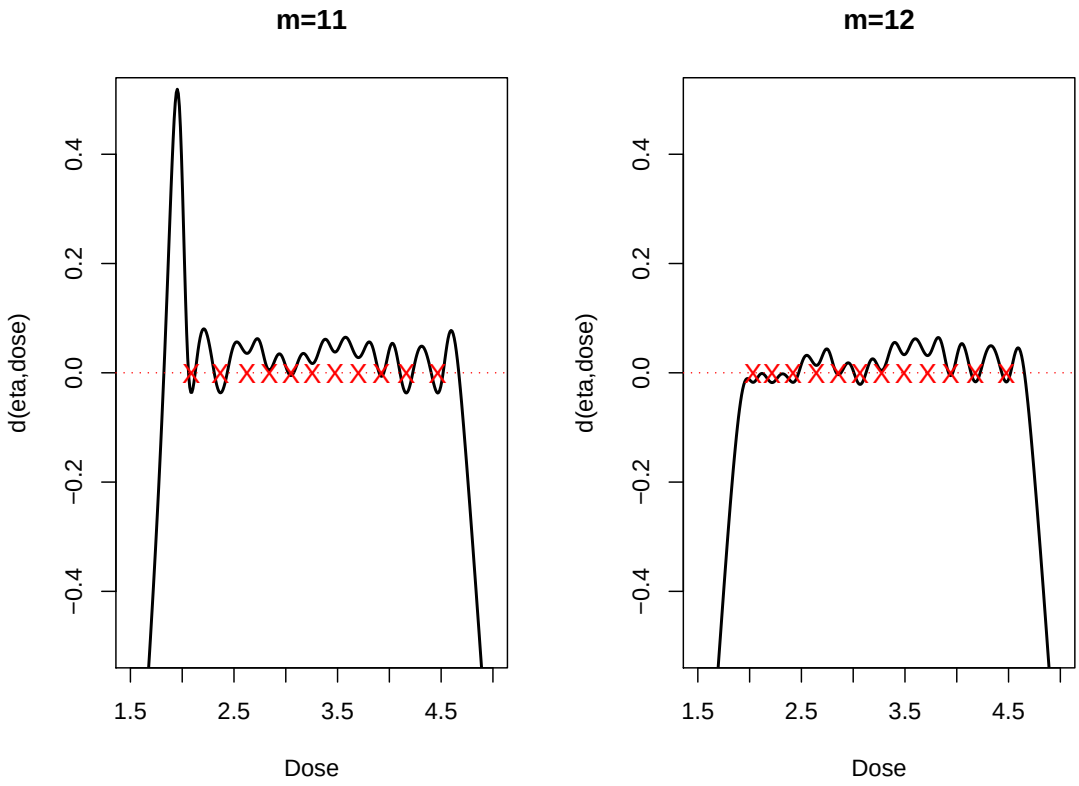

Figure 8.2: The Directional Derivatives at the Bayesian D-optimal Designs with 11 and 12 Support Points, η_{11}^D and η_{12}^D. The Red Marks (X) Are Points of Support.

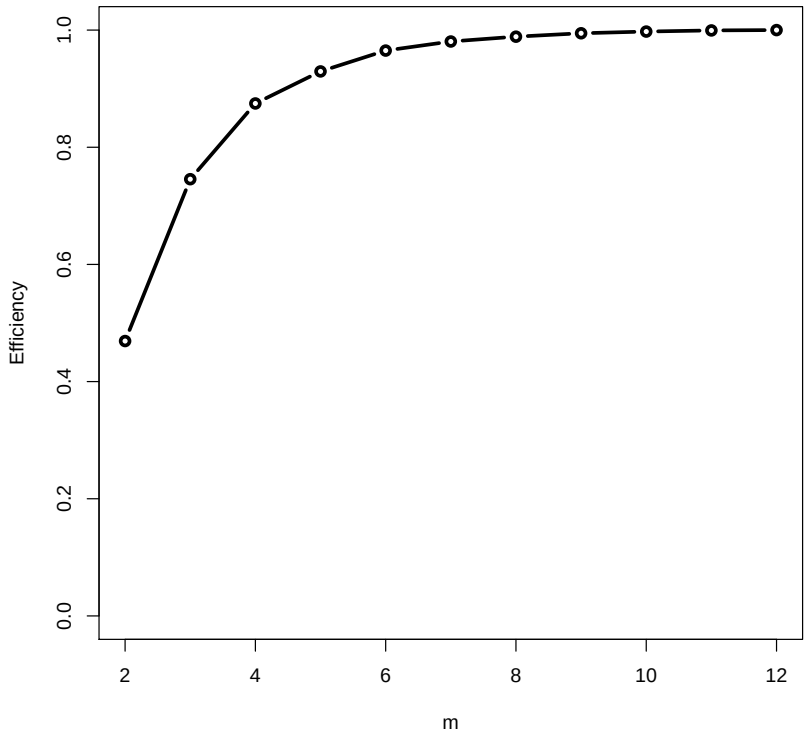

Figure 8.3: Efficiencies of the m-point D-optimal Designs Compared to the 12-point D-optimal Design, η_{12}^{D}

Table 8.1: Bayesian D-optimal Designs, $m = 2$ to 12

Dose	Weight	$\phi_1(\eta)$	Dose	Weight	$\phi_1(\eta)$	Dose	Weight	$\phi_1(\eta)$
	m=2			m=8			m=11	
2.8413	0.5000		2.2117	0.1122		2.0880	0.0581	
3.6643	0.5000	-17.1105	2.5800	0.1429		2.3697	0.0960	
	m=3		2.8831	0.1380		2.6253	0.1000	
2.6027	0.3290		3.1637	0.1307		2.8410	0.1000	
3.2566	0.3444		3.4421	0.1304		3.0497	0.1000	
3.9375	0.3266	-14.3329	3.7305	0.1284		3.2575	0.1000	
	m=4		4.0073	0.1186		3.4754	0.1000	
2.4662	0.2447		4.3520	0.0988	-12.6384	3.7014	0.1000	
2.9911	0.2591			m=9		3.9238	0.1000	
3.5344	0.2575		2.0963	0.0616		4.1632	0.0873	
4.0739	0.2387	-13.3734	2.3899	0.1010		4.4676	0.0585	-12.5740
	m=5		2.6718	0.1230			m=12	
2.3516	0.1763		2.9370	0.1297		2.0340	0.0343	
2.8167	0.2118		3.2022	0.1248		2.2170	0.0497	
3.2663	0.2256		3.4719	0.1260		2.4189	0.0786	
3.7447	0.2249		3.7514	0.1245		2.6452	0.1000	
4.2812	0.1614	-13.0095	4.0208	0.1141		2.8584	0.1000	
	m=6		4.3603	0.0953	-12.6032	3.0664	0.1000	
2.2772	0.1439			m=10		3.2742	0.1000	
2.6884	0.1729		2.0932	0.0600		3.4918	0.1000	
3.0532	0.1780		2.3828	0.0986		3.7183	0.1000	
3.4275	0.1770		2.6615	0.1210		3.9409	0.1000	
3.8197	0.1773		2.9205	0.1250		4.1808	0.0832	
4.2298	0.1509	-12.7841	3.1751	0.1190		4.4789	0.0543	-12.5706
	m=7		3.4266	0.1160				
2.2248	0.1198		3.6798	0.1121				
2.6078	0.1518		3.9197	0.1046				
2.9309	0.1522		4.1637	0.0858				
3.2445	0.1487		4.4678	0.0579	-12.5857			
3.5760	0.1545							
3.9116	0.1495							
4.2894	0.1236	-12.6886						

8.8.2 Bayesian A-optimality

A similar graphical presentation for Bayesian A-optimality is in Figure 8.4, and more detailed results are in Table 8.2. Denote η_m^A as the best design found for $m = 2, ..., 15$. Similar to the Bayesian D-optimal designs, the range of doses increases with m. From the values of $\phi_{2A}(\eta_m^A)$, η_2^A and η_3^A are clearly not optimal. Increasing the number of support points improves the performance of the design. The designs η_{14}^A and η_{15}^A are considerably better than η_2^A and η_3^A. The fifteen-point design is the best design found. The plots of directional derivatives for η_{14}^A and η_{15}^A are shown in Figures 8.5. Figure 8.6 shows the efficiencies compared to η_{15}^D. When $m \geq 9$, the performance of the designs does not differ that much; as for the D-optimal example, not a lot may be gained by increasing m further. As for D-optimality, the exact A-optimal design is difficult to identify numerically.

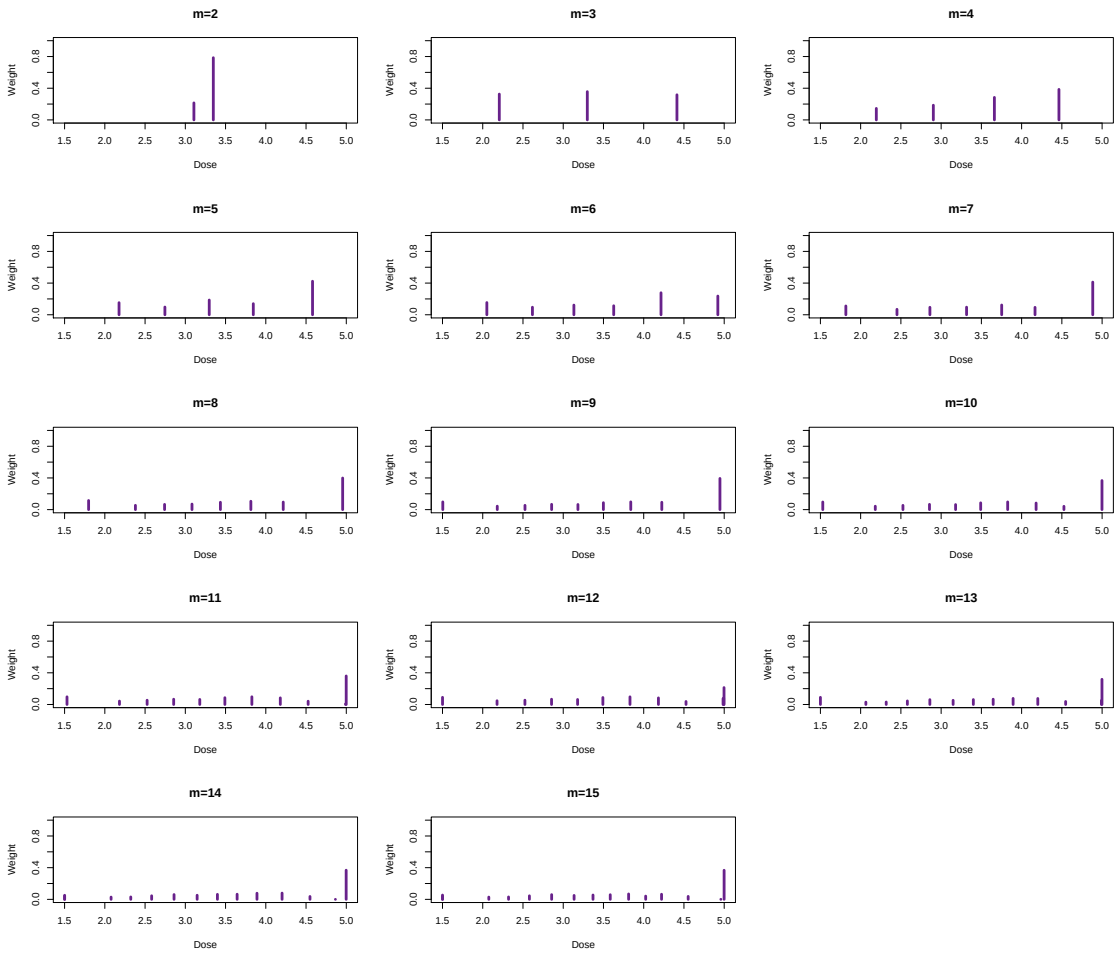

Figure 8.4: The Best m-point Bayesian A-optimal Designs, $m = 2$ to 15

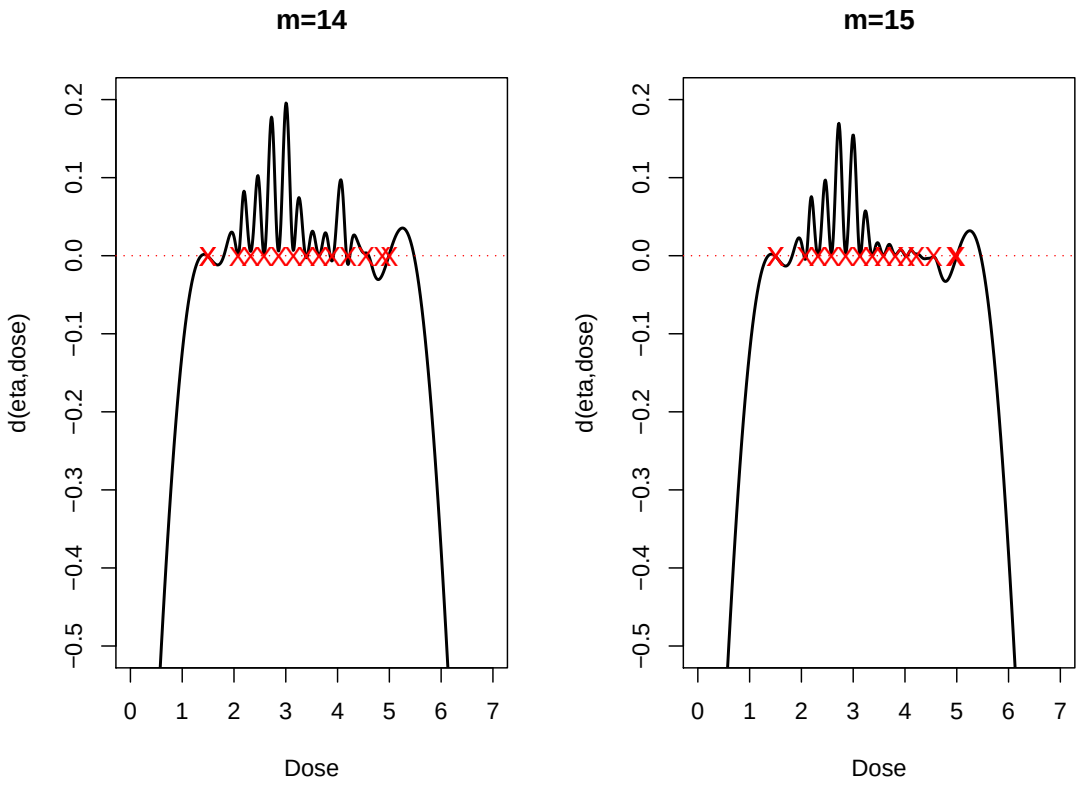

Figure 8.5: The Directional Derivatives at Bayesian A-optimal Designs with 14 and 15 Support Points, η_{14}^{A} and η_{15}^{A}. The Red Marks (X) Are Points of Support.

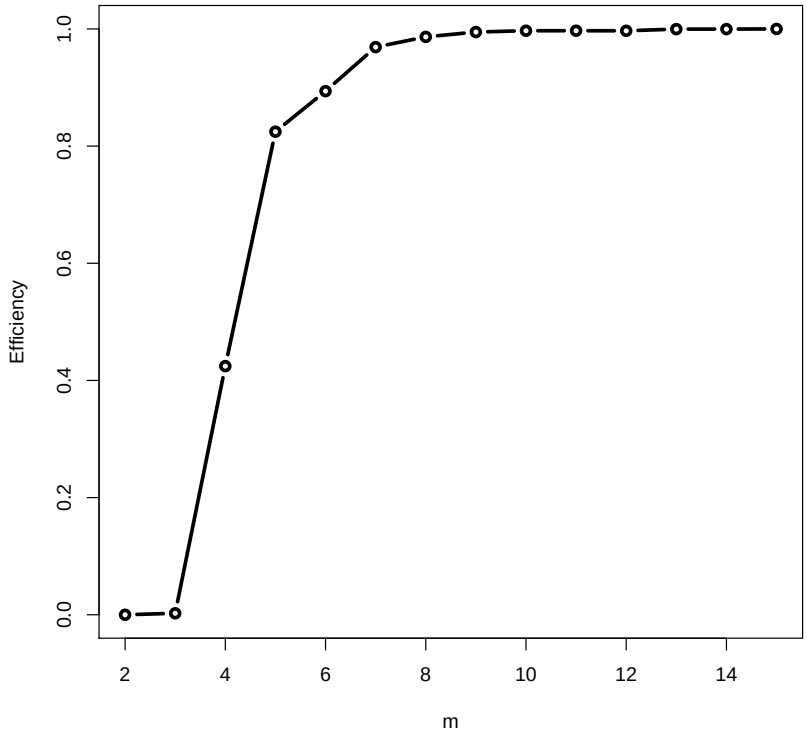

Figure 8.6: Efficiencies of the m-point A-optimal Designs Compared to the Best 15-point Design, $\eta 15^A$

Table 8.2: Bayesian A-optimal Designs, $m = 2$ to 15

Dose	Weight	$\phi_{2A}(\eta)$	Dose	Weight	$\phi_{2A}(\eta)$	Dose	Weight	$\phi_{2A}(\eta)$
	m=2			m=10			m=14	
3.1068	0.2148		1.5295	0.0966		1.5004	0.0417	
3.3479	0.7852	-76822653	2.1818	0.0432		1.5006	0.0516	
	m=3		2.5254	0.0540		2.0776	0.0289	
2.2058	0.3261		2.8553	0.0667		2.3223	0.0305	
3.2992	0.3568		3.1796	0.0639		2.5817	0.0453	
4.4130	0.3171	-3013.2700	3.4914	0.0858		2.8600	0.0606	
	m=4		3.8268	0.0975		3.1469	0.0521	
2.1948	0.1451		4.1800	0.0833		3.3988	0.0633	
2.9031	0.1856		4.5257	0.0418		3.6437	0.0649	
3.6617	0.2841		4.9985	0.3672	-7.6758	3.8904	0.0775	
4.4637	0.3851	-18.0290		m=11		4.2021	0.0792	
	m=5		1.5300	0.0968		4.5488	0.0362	
2.1765	0.1527		2.1819	0.0430		4.8649	0.0001	
2.7464	0.0977		2.5254	0.0541		4.9984	0.3681	-7.6545
3.2973	0.1862		2.8554	0.0667			m=15	
3.8447	0.1400		3.1798	0.0639		1.5005	0.0386	
4.5808	0.4234	-9.2796	3.4916	0.0858		1.5003	0.0544	
	m=6		3.8269	0.0974		2.0755	0.0289	
2.0511	0.1542		4.1803	0.0837		2.3208	0.0306	
2.6174	0.0954		4.5260	0.0415		2.5798	0.0449	
3.1326	0.1224		4.9903	0.0070		2.8562	0.0598	
3.6273	0.1138		4.9989	0.3602	-7.6758	3.1351	0.0496	
4.2149	0.2778			m=12		3.3700	0.0551	
4.9211	0.2364	-8.5613	1.5005	0.0906		3.5841	0.0587	
	m=7		2.1777	0.0473		3.8125	0.0684	
1.8162	0.1112		2.5241	0.0546		4.0245	0.0412	
2.4513	0.0687		2.8550	0.0669		4.2217	0.0647	
2.8605	0.0944		3.1785	0.0632		4.5524	0.0372	
3.3161	0.0961		3.4927	0.0882		4.9583	0.0005	
3.7524	0.1234		3.8294	0.0967		4.9989	0.3673	-7.6522
4.1666	0.0937		4.1829	0.0834				
4.8838	0.4125	-7.8970	4.5256	0.0379				
	m=8		4.9859	0.0741				
1.7984	0.1154		4.9941	0.0834				
2.3800	0.0544		4.9979	0.2136	-7.6767			
2.7423	0.0662			m=13				
3.0830	0.0703		1.5000	0.0904				
3.4360	0.0931		2.0647	0.0299				
3.8144	0.1057		2.3174	0.0314				
4.2166	0.0958		2.5804	0.0453				
4.9550	0.3991	-7.7576	2.8600	0.0612				
	m=9		3.1489	0.0528				
1.5038	0.0977		3.4001	0.0622				
2.1805	0.0438		3.6466	0.0667				
2.5248	0.0543		3.8943	0.0768				
2.8560	0.0670		4.2010	0.0761				
3.1827	0.0649		4.5470	0.0379				
3.4991	0.0875		4.9916	0.0513				
3.8392	0.0979		4.9983	0.3179	-7.6547			
4.2248	0.0931							
4.9472	0.3938	-7.6931						

8.8.3 Local Optimality

The best guess for $\boldsymbol{\xi}$ is $\boldsymbol{\xi}_0 = (3.3, 4.8, 3.7, 5.7, 3.8, 5.7)$. For the two-point locally D-optimal design, equal weight is put at $x = 2.3252, 4.8249$, and $\phi_1(\eta_2^D) = -9.1282$. The corresponding probability of colonization and toxicity can be found in Table 8.3 and is shown in Figure 8.7. The locally three- and four-point D-optimal designs give the same results as two-point design. In other words, the best locally D-optimal design for this example is a two-point design. Similar results are found for A-optimal designs. The two-point design putting weights $(0.2929, 0.7071)$ on doses $(2.2432, 4.3568)$ is the A-optimal design, and $\phi_{2A}(\eta_2^A) = -11.0431$ with corresponding response probabilities also shown in Figure 8.7. The results are also illustrated in Figure 8.7. The plot of directional derivatives for both designs are in Figure 8.8, and the graphs confirm that for D- and A-optimality, the two-point designs are optimal.

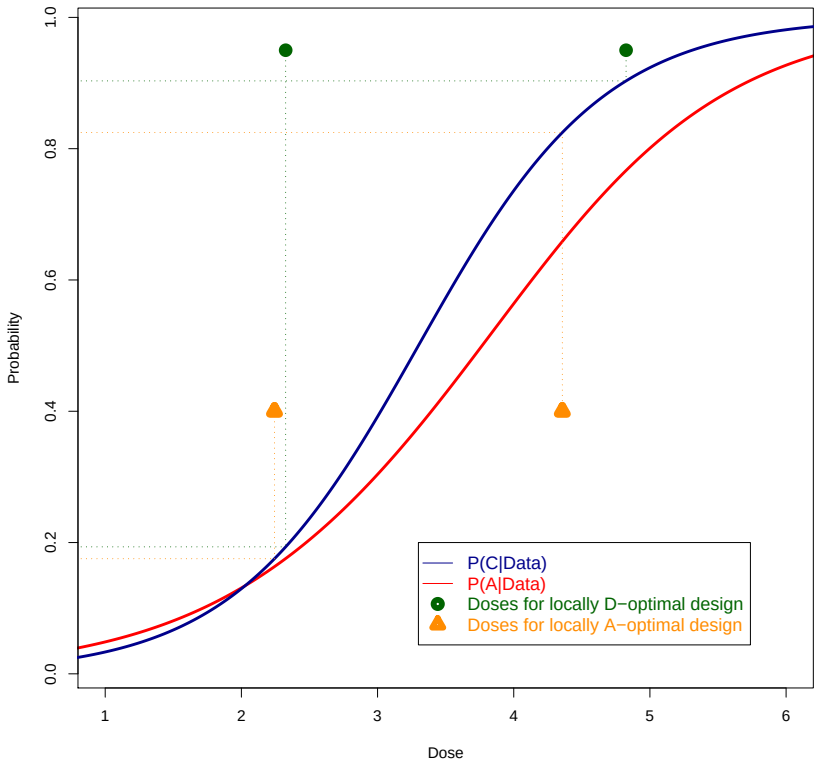

Figure 8.7: The Designs for Local D- and A-Optimality with Unrestricted Doses and $\boldsymbol{\xi}_0 = (3.3, 4.8, 3.7, 5.7, 3.8, 5.7)$

Table 8.3: Local Optimality

	Dose	Weight	Pr(C|Dose)	Pr(A|Dose)	Directional Derivatives
D-optimal	2.3252	0.5	0.1937	0.1756	1.1488×10^{-5}
	4.8249	0.5	0.9032	0.7668	1.1488×10^{-5}
	$\phi_1(\eta_2^D) =$	-9.1282			
A-optimal	2.2432	0.2929	0.1754	0.1633	7.5523×10^{-6}
	4.3568	0.7071	0.8246	0.6587	-3.12830×10^{-6}
	$\phi_{2A}(\eta_2^A) =$	-11.0431			

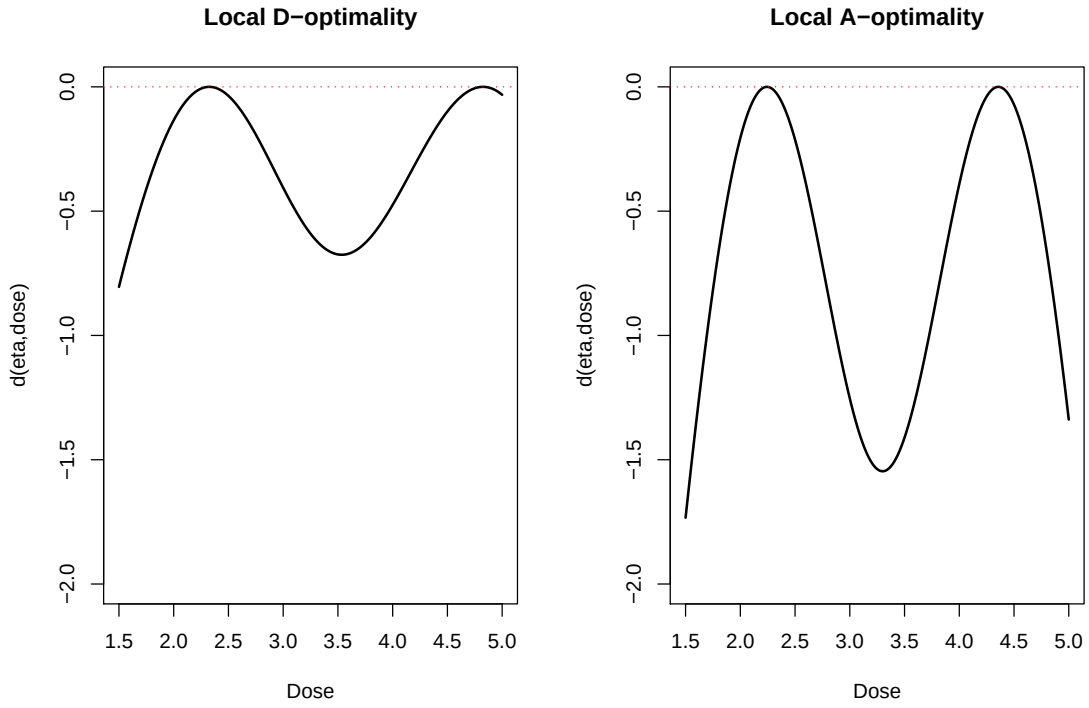

Figure 8.8: Directional Derivatives for Local Optimality with Unrestricted Doses and $\boldsymbol{\xi}_0 = (3.3, 4.8, 3.7, 5.7, 3.8, 5.7)$

8.9 Designs with Fixed Doses

In the previous section, the optimal designs with unrestricted doses are described where the design space $\mathcal{X}$ is the interval, $\mathcal{X} = [1.5, 5.0]$. However, it may be more practical to use a fixed set of doses for $\mathcal{X}$, $\mathcal{X} = \{x_1, ..., x_j\}$. Define $\mathcal{X}_F$ to be the set of eight pre-specified dose: $\mathcal{X} = \{1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$. The optimization is then only on the weights and the complexity of the problem is reduced. In the H-Flu example, this is the case. In this section, the Bayesian and locally optimal designs with dose restricted to $\mathcal{X}_F$ are evaluated. The directional derivatives are examined to check if a design is close to optimal. The values of directional derivatives have to be zero at the the points of support, and less than zero at the points that have zero weights. If the value of the derivative is less than zero but not close to zero, the weight may be zero. The algorithm in the examples starts with $j = 8$ design points, and then the number of design points is reduced when the best design found has a design point with weight close to zero and the directional derivative is clearly negative. That design point is then assumed a weight of zero. The optimization is performed again with the reduced number of design points until all design points have positive weight and directional derivatives are close to zero.

8.9.1 Bayesian Optimality

The results of finding the D- and A-optimal designs with restricted doses can be found in Table 8.4. Denote $\eta_m^{D_f}$ be the m-point D-optimal design with fixed doses. For D-optimality, the best design is a four-point design with doses (2.5, 3.0, 3.5, 4.0) and weights (0.2549, 0.2448, 0.2435, 0.2568).

Denote $\eta_m^{A_f}$ be the m-point A-optimal design with fixed doses. The best A-optimal design with fixed doses is $\eta_8^{A_f}$ (Table 8.4). Similar to the A-optimal designs with unrestricted doses, the optimal design puts more weights on higher doses. The directional derivatives for the best D- and A-optimal designs are presented in Figure 8.9. It is not surprisingly that there are positive values in some regions, more specifically, $\mathcal{X}_F^c$, where intersects with the interval $[1.5, 5.0]$. These designs are not as good as the designs with unrestricted doses in terms of the criterion, although they may be the easiest to implement in a particular practical example. A comparison is given in section 8.9.3.

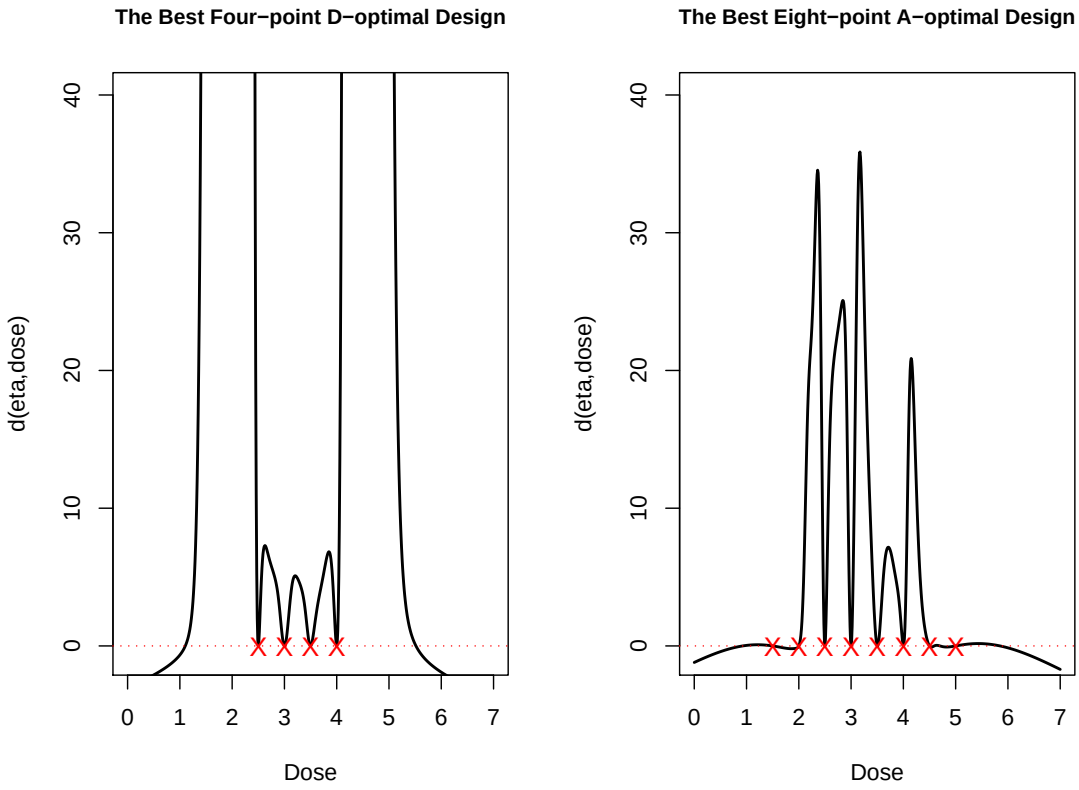

Figure 8.9: Directional Derivatives for Bayesian Optimality with Fixed Doses. The Red Marks (X) Are Points of Support.

Table 8.4: Bayesian Optimality with Fixed Doses

	Dose	Weight	Directional Derivatives
D-optimality	1.5	0.0000	2.6107×10^{2}
	2.0	0.0000	1.0744×10^{6}
	2.5	0.2549	-8.6958×10^{-6}
	3.0	0.2448	6.5980×10^{-5}
	3.5	0.2435	3.1473×10^{-6}
	4.0	0.2568	-6.4107×10^{-7}
	4.5	0.0000	5.2730×10^{4}
	5.0	0.0000	1.7350×10^{2}
	$\phi_1(\eta_5^{D_f}) =$	-13.4248	
A-optimality	1.5	0.0654	5.9868×10^{-4}
	2.0	0.0627	9.5997×10^{-4}
	2.5	0.0765	7.2254×10^{-4}
	3.0	0.1071	-4.3595×10^{-4}
	3.5	0.1330	-1.6509×10^{-3}
	4.0	0.1198	-7.6807×10^{-4}
	4.5	0.1239	2.0552×10^{-3}
	5.0	0.3115	-1.6387×10^{-4}
	$\phi_{2A}(\eta_8^{A_f}) =$	-8.0371	

8.9.2 Local Optimality

The designs for local optimality with $\mathcal{X}_F$ are also evaluated, and the results are shown in Figure 8.10 and Table 8.5. The graphs of directional derivatives are shown in Figure 8.11, and it is confirmed that the best design for D- and A-optimality are the two-point designs.

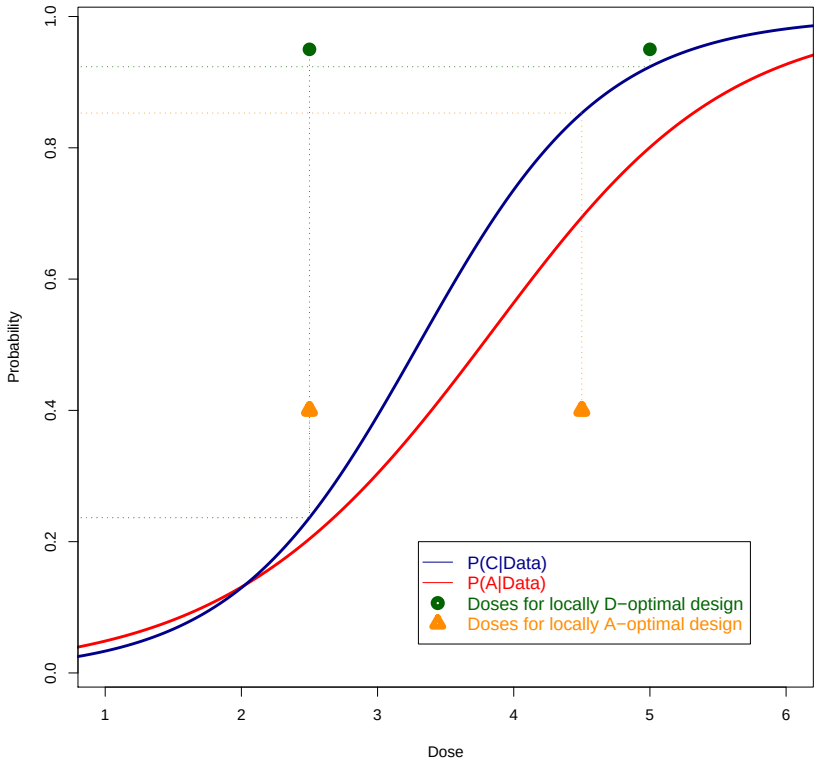

Figure 8.10: The Designs for Local D- and A-Optimality with Fixed Doses and $\boldsymbol{\xi}_0 = (3.3, 4.8, 3.7, 5.7, 3.8, 5.7)$

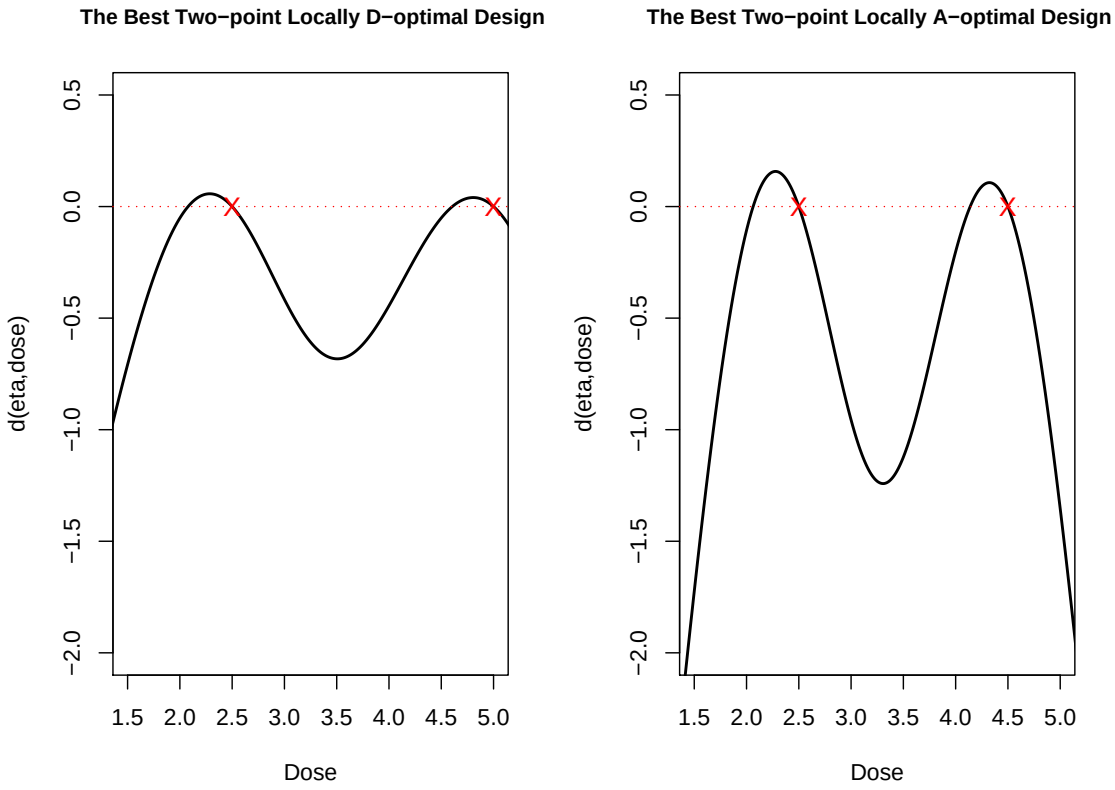

Figure 8.11: Directional Derivatives for Local Optimality with Fixed Doses and $\boldsymbol{\xi}_0 = (3.3, 4.8, 3.7, 5.7, 3.8, 5.7)$. The Red Marks (X) Are Points of Support.

Table 8.5: Local Optimality with Fixed Doses

	Dose	Weight	Pr(C|Dose)	Pr(A|Dose)	Directional Derivatives
D-optimality	1.5	0.0000			-7.0828×10^{-1}
	2.0	0.0000			-5.1583×10^{-2}
	2.5	0.5000	0.2365	0.2042	-1.1895×10^{-12}
	3.0	0.0000			-4.1568×10^{-1}
	3.5	0.0000			-6.8242×10^{-1}
	4.0	0.0000			-4.4450×10^{-1}
	4.5	0.0000			-5.1086×10^{-2}
	5.0	0.5000	0.9234	0.8007	1.1893×10^{-12}
	$\phi_1(\eta_2^D) =$	-9.1689			
A-optimality	1.5	0.0000			-1.7245
	2.0	0.0000			-1.0653×10^{-1}
	2.5	0.2975	0.2365	0.2042	8.1287×10^{-7}
	3.0	0.0000			-9.5642×10^{-1}
	3.5	0.0000			-1.1229
	4.0	0.0000			-2.0187×10^{-1}
	4.5	0.7025	0.8529	0.6941	-3.4413×10^{-7}
	5.0	0.0000			-1.3539
	$\phi_{2A}(\eta_2^A) =$	-11.1589			

8.9.3 Comparison of Optimality Criteria on $\mathcal{X}$ and $\mathcal{X}_F$

To compare the designs with fixed doses to unrestricted doses, the efficiencies are computed. The best Bayesian D-optimal fixed-dose design is a four-point design, and the efficiencies compared to the unrestricted-dose designs, η_4^D and η_{12}^D, are 0.9331 and 0.8673. The best Bayesian A-optimal fixed-dose designs is a eight-point design, and efficiencies compared to unrestricted designs, η_8^A and η_{15}^A, are 0.9652 and 0.9521. For both D- and A-optimality, the efficiencies of the designs with fixed doses compared to unrestricted doses are reduced by less than 15% and 5%, respectively. The results can be found at Table 8.6.

For local optimality with fixed doses, two-point D-optimal and two-point A-optimal designs are the best designs. As in Table 8.6, compared to the designs without dose restriction, the efficiencies are 0.9932 for D-optimality, and 0.9896 for A-optimality.

Table 8.6: Comparison between Designs with Unrestricted and Fixed Doses

	Bayesian Optimal Designs	Local Optimal Designs
D-optimality	$\text{Eff}_D(\eta_4^{D_f}, \eta_4^D) = 0.9915$	$\text{Eff}_D(\eta_2^{D_f}, \eta_2^D) = 0.9932$
	$\text{Eff}_D(\eta_4^{D_f}, \eta_{12}^D) = 0.8673$	
A-optimality	$\text{Eff}_A(\eta_8^{A_f}, \eta_8^A) = 0.9652$	$\text{Eff}_A(\eta_2^{A_f}, \eta_2^A) = 0.9896$
	$\text{Eff}_A(\eta_8^{A_f}, \eta_{15}^A) = 0.9521$	

8.10 Comparison between Sequential and Non-sequential Designs

As in Chaloner and Verdinelli [17], when dealing with nonlinear problem, it is more beneficial to implement sequential designs compared to non-sequential designs because the posterior variance or asymptotic variance depends on the outcome data, $\mathbf{y}$. In this section, the results from myopic procedures with no restriction on the posterior probability of toxicity are compared with the best Bayesian and Locally A-optimal designs with fixed doses. The primary prior distribution, BPP, is used in the Bayesian approaches as described in Chapter 7. For both sequential and non-sequential designs, the ratio of expected losses are computed, and the myopic one-step-look-ahead procedure is the benchmark.

Table 8.7 and 8.8 list the best m-point designs for Bayesian and local A-optimality with fixed doses. For local optimality, when $m = 3$ and 4 the weight for at least one dose is extremely small, and there is little or no improvement over the two-point design. When $m = 5$, 6 or 7, a similar trend is found, and most of the weight is put at doses 2.0, 2.5, 4.0 and 4.5. For $m = 8$, there is no weight at doses 1.5 and 3.0. Therefore, only two-, four- and six- point designs are used for local A-optimality. The optimal designs found in Section 8.9: eight-point Bayesian A-optimal ($\eta_8^{A_f}$) and two-point local A-optimal ($\eta_2^{A_f}$) designs are also used. When comparing to the myopic strategies for a total of n observations, the best m-point designs are used where m is the best design for $m \leq n$. For example, when $n > 8$, the best eight-point design is used. Since some of the best m-point designs have less than m support points, the following rules are applied for computing the expected

loss for local A-optimality:

- When $n = 3$, the best two-point design is used.
- When $n = 5$, the best four-point design is used.
- When $n \geq 6$, the best six-point design is used.

Table 8.7: The Best m-point Bayesian A-optimal Designs with Fixed Doses

m	Dose	Weight	$\phi_{2A}(\eta_m^{A_f})$	m	Dose	Weight	$\phi_{2A}(\eta_m^{A_f})$
2	3.0	0.0629		7	2.0	0.1229	
	4.0	0.9371	-1.84×10^8		2.5	0.0770	
3	2.5	0.3004			3.0	0.1036	
	3.0	0.5000			3.5	0.1326	
	4.5	0.1996	-2.64×10^4		4.0	0.1180	
4	2.0	0.1583			4.5	0.1192	
	3.0	0.3333			5.0	0.3268	-8.0758
	3.5	0.2726		8	1.5	0.0654	
	4.5	0.2358	-84.6037		2.0	0.0627	
5	2.0	0.0741			2.5	0.0766	
	2.5	0.0347			3.0	0.1071	
	3.0	0.1063			3.5	0.1330	
	3.5	0.2500			4.0	0.1198	
	4.5	0.5350	-31.1928		4.5	0.1242	
6	2.0	0.1649			5.0	0.3113	-8.0371
	2.5	0.0746					
	3.0	0.0996					
	3.5	0.1137					
	4.0	0.0903					
	4.5	0.4569	-8.5694				

Table 8.8: The Best m-point Local A-optimal Designs with Fixed Doses

m	Dose	Weight	$\phi_{2A}(\eta_m^{A_f})$	m	Dose	Weight	$\phi_{2A}(\eta_m^{A_f})$
2	2.5	0.2974		7	1.5	1.40×10^{-123}	
	4.5	0.7026	-11.1588		2.0	0.1246	
3	2.0	1.93×10^{-7}			2.5	0.1667	
	2.5	0.2974			3.0	4.80×10^{-111}	
	4.5	0.7026	-11.1589		3.5	1.31×10^{-69}	
4	2.0	1.21×10^{-7}			4.0	0.0926	
	2.5	0.2974			4.5	0.6161	-11.1938
	3.5	2.11×10^{-118}		8	1.5	0.0000	
	4.5	0.7026	-11.1589		2.0	0.1002	
5	2.0	0.0452			2.5	0.1429	
	2.5	0.2500			3.0	0.0000	
	3.0	0.0000			3.5	0.0976	
	4.0	0.0156			4.0	0.1429	
	4.5	0.6891	-11.1697		4.5	0.1429	
6	1.5	0.0000			5.0	0.3736	-11.8649
	2.0	0.0929					
	2.5	0.2000					
	3.0	2.14×10^{-9}					
	4.0	0.0618					
	4.5	0.6453	-11.1841				

For an optimal design measure, and for a particular sample size n, nw_j may not be an integer (see Section 8.3), and must be rounded to an integer. The rounded A-optimal designs for the simulations are listed in Table 8.9 and 8.10. The number of subjects is computed by multiplying n by w_j, and rounding to integers so that the sum of the number of observations is n (page 308, [44]).

Table 8.9: Rounded Bayesian A-optimal Designs for Simulations

n	Dose	Rounded number of subjects per dose	n	Dose	Rounded number of subjects per dose
2	3.0	0	7	2.0	1
	4.0	2		2.5	1
3	2.5	1		3.0	1
	3.0	1		3.5	1
	4.5	1		4.0	1
4	2.0	1		4.5	1
	3.0	1		5.0	1
	3.5	1	8	1.5	1
	4.5	1		2.0	1
5	2.0	0		2.5	1
	2.5	0		3.0	1
	3.0	1		3.5	1
	3.5	1		4.0	1
	4.5	3		4.5	1
6	2.0	1		5.0	1
	2.5	0	9	1.5	1
	3.0	1		2.0	1
	3.5	1		2.5	1
	4.0	1		3.0	1
	4.5	2		3.5	1
				4.0	1
				4.5	1
				5.0	2

Table 8.10: Rounded Local A-optimal Designs for Simulations

n	Dose	Rounded number of subjects per dose	n	Dose	Rounded number of subjects per dose
2	2.5	1	7	2.0	1
	4.5	1		2.5	1
3	2.5	1		3.5	1
	4.5	2		4.0	1
4	2.0	0		4.5	1
	2.5	1		5.0	2
	4.0	0	8	2.0	1
	4.5	3		2.5	1
5	2.0	0		3.5	1
	2.5	1		4.0	1
	4.0	0		4.5	1
	4.5	4		5.0	3
6	2.0	1	9	2.0	1
	2.5	1		2.5	1
	3.5	1		3.5	1
	4.0	1		4.0	1
	4.5	1		4.5	1
	5.0	1		5.0	4

The comparisons are summarized in Figure 8.12. As in myopic procedures, the expected losses for both Bayesian and locally A-optimal designs decreases as number of subjects increases. Neither A-optimal designs perform as well as the designs from myopic strategies, indicating that sequential strategy in the design of experiment can lead to substantial improvements.

The results of a similar comparison using an alternative approach to round to exact designs is given in Appendix A.

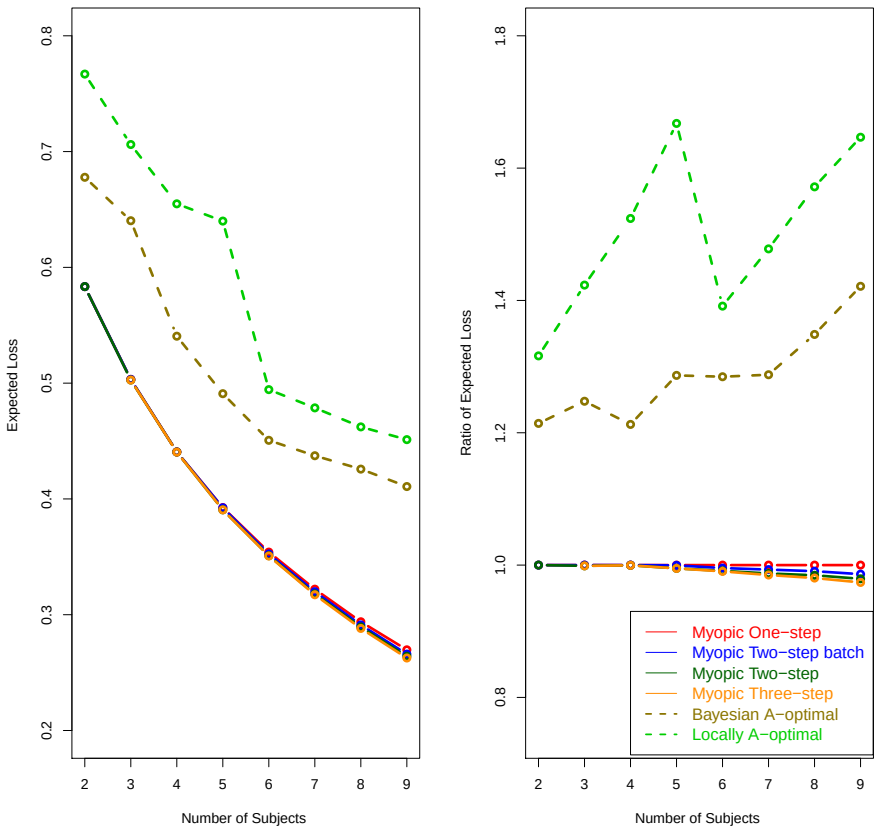

Figure 8.12: Comparison between Sequential and Non-sequential Designs. The Left Plot Is the Expected Loss. The Right Plot Is the Ratio of Expected Loss, and Myopic One-step-look-ahead Procedure Is the Benchmark.

8.11 Conclusions

From the results, it is found that optimal design may be very inefficient when n is small since the criteria involve asymptotic approximation. In addition, as in Section 8.8, the Bayesian optimal designs are hard to identify because the computation involves taking expectation, and the posterior distribution of the parameter is approximated by normal distribution. In the example, different sizes for prior samples, k, are also examined and they give similar results: sequential design is clearly better than non-sequential design in the cases examined.

CHAPTER 9
FUTURE WORK

9.1 Developing the R Package

A number of R packages have been developed for clinical trial designs, such as `gsDesign` [3], `GroupSeq` [42], `AGSDest` [31], `ldbounds` [15], `seqmon` [47], `DoseFinding` [11], and `MCPMod` [10]. The first five packages are for group sequential designs or sequential monitoring, and the last two are used in dose-finding desings. It appears that there is no R package incorporating Bayesian myopic procedures in dose-finding experiments, and one of the possible extensions is to develop an R package. The following are some of the components should be included in the package:

- Models and parameters: A variety of models can be included, for example, models for univariate responses (logistic regression model, probit model), and for bivariate responses (proportional odds model, continuation-ratio model, and the new model).

- Loss function: Different loss functions can be selected by user.

- Prior distribution: The prior distribution should be specified by user.

- Options for non-sequential design: This allows the user to apply non-sequential strategies, and to compare them.

- Options for real-time sequential strategy: The feature is practical for clinicians or principle investigators who conduct the clinical trials.

9.2 Incorporating Toxicity into Loss

As discussed in Chapter 7, toxicity could be incorporated into the loss. Since toxicity and variance of estimation are in different units, this may be challenging [20]. In this thesis, we introduced toxicity as a constraint rather than a loss: $\Pr(A = 1|\mathbf{h^{(i-1)}}, x_i) < \epsilon,\ 0 < \epsilon \leq 1$. Further investigation in quantifying the trade-off between toxicity and dose finding is another possible extension of this thesis.

9.3 Consideration of Time-to-event Data

In the H-Flu study, the colonization status was evaluated on day three, six, ten, fourteen and eighteen after inoculation. Adverse events were also monitored during the follow-up period. The time to event is available, and this information could be incorporated in determining the dose allocation using methods for censored data in real time. Several methods have been proposed for such design: Thall et al. [54], Cheung and Chappell [18], Husing et al. [34], and Braun et al. [13] applied time-to-toxicity data on dose-finding trials, and the method of time-to-event data considering both efficacy and toxicity is proposed by Yuan and Yin [66]. Myopic strategies could be implemented modeling efficacy and toxicity jointly as time-to-event outcomes.

9.4 The H-Flu Study

The H-Flu study had several novel aspects. First, the subjects were healthy volunteers rather than then subjects with advanced disease in phase I cancer trials. The sponsors from the National Institute of Health, were understandably concerned

about the potential for making volunteers sick. However, understanding colonization and estimating a dose at which colonization reliably occurs is essential to developing a vaccine.

Second, the bacteria, *Haemophilus influenzae*, used was a stain that is resistant to an antibiotic, streptomycin, although susceptible to other drugs. The sponsors were also concerned with the potential release of a drug resistant bacteria into the community. In the second study, currently underway, intimate partners of the volunteers getting inoculated are also being tested for colonization.

These kinds of novel aspects of studies with infectious biological agents present new problems in designing trials. This is a rich area for methodological development.

9.5 Other Applications

This thesis was motivated by the H-Flu study but led to the results that are more widely applicable, for example, the algorithms can be implemented in other applications such as animal studies. Grounding methodological developments for design in a real application is a good way to develop new methods that can be helpful in other situations.

APPENDIX A
RESULTS FOR COMPARING SEQUENTIAL AND NON-SEQUENTIAL DESIGNS (ALTERNATIVE APPROACH)

As discussed in Chapter 8, an alternative approach is considered. That is to use $\eta_8^{A_f}$ for Bayesian A-optimality, and $\eta_2^{A_f}$ for local A-optimality. For Bayesian A-optimality, when $n < 8$, the doses with n largest weights are selected. For example, when $n = 2$, the two doses with the largest weights from $\eta_8^{A_f}$ are chosen, and the number of subjects at each is rounded to one.

As discussed in Chapter 8, an alternative approach could be considered for rounding an approximate design to integer design for different integers. An alternative approach was used for Bayesian A-optimality where, for $n < 8$, the doses with the n largest weights in $\eta_2^{A_f}$ were selected. For example, when $n = 2$, the two doses with the largest weights in $\eta_2^{A_f}$ could be rounded to one. The results of simulations using these alternative designs are in Figure A.1.

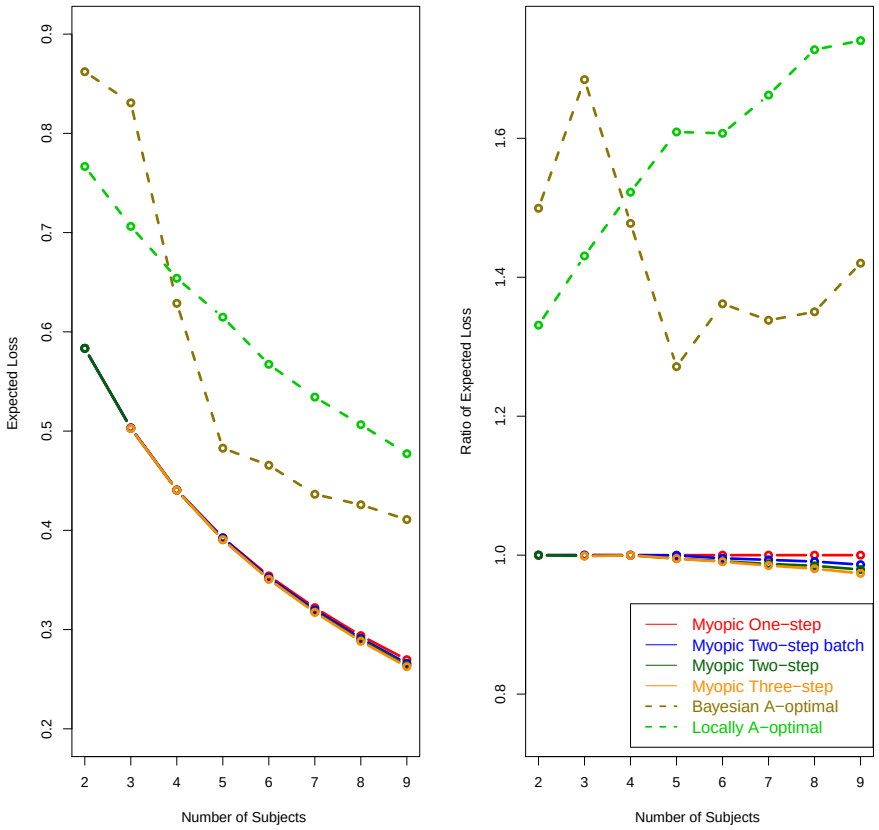

Figure A.1: Comparison between Sequential and Non-sequential Designs. The Left Plot Is the Expected Loss. The Right Plot Is the Ratio of Expected Loss, and Myopic One-step-look-ahead Procedure Is the Benchmark.

APPENDIX B
R FUNCTIONS FOR MYOPIC PROCEDURES

B.1 R Functions for Myopic Procedures: Bivariate Model

In this section, R functions for myopic procedures used in Chapter 4 are listed.

```
#************************************************************#
#    Functions for myopic procedure:
#    (1) sim.prior: simulate the random sample from
#                   prior distribution
#    (2) prior.prob: compute the prior probability
#    (3) post.wgt: compute the posterior weight
#    (4) Eloss: compute the expected loss
#************************************************************ #
# ------- (1) Function : Simulate the random sample ------- #
# -------                from prior distribution    ------- #
# sim.prior(n, constz, a, b, e, f)
# n = number of sample
# constz = constant
# a, b = parameters for beta distribution
#       (for v: related to the lower HCD)
# e, f = parameters for beta distribution
#       (for z: related to the higher HCD - lower HCD)
# VALUE
#   sim.prior gives n by 2 matrix, the 1st column is mu and
#    2nd column is gamma
#-----------------------------------------------------------#
sim.prior <- function(nsample, constz, a, b, e, f){
set.seed(39)
  v <- rbeta(nsample, a, b)
  z <- rbeta(nsample, e, f)
  mu.k <- 1.5 + 3.5*v
  ga.k <- constz*z + 0.1 + mu.k
  smp <- cbind(mu.k, ga.k)
  return(smp)
}
# ------ (2) Function: calculate prior probability -------- #
# prior.prob(p.low, p.high, dose, sim.mu, sim.ga)
# p.low / p.high = prob of subject get infected  (HCDa, HCDb)
```

```
# dose = dose
# sim.mu / sim.ga = mu / gamm simulated from the
#                   function sim.prior
# VALUE
#  prior.prob gives the inidividual probabiliy of success
#    (1st column) for a given dose
# ------------------------------------------------------- #
prior.prob <- function(p.low, p.high, dose, sim.mu, sim.ga) {
  odds.low <- p.low/(1-p.low)
  odds.high <- p.high/(1-p.high)
  OR <- odds.high / odds.low
  M <- (dose - sim.mu) / (sim.ga - sim.mu)
  numer <- odds.low * (OR^M)
  p.i <- numer / (1+numer)
  return(p.i)
}
# ------ (3) Function: calculate posterior weights -------- #
# post.wgt(prior.weight, ind.prob)
# prior.weight = prior weights for success/failure
# curr.prob = individual prob for success / failure
              (from prior.prob)
# VALUE
#   post.wgt calcuate the updated weight given prior weight
# ------------------------------------------------------- #
post.wgt <- function(prior.weight, ind.prob) {
  n <- length(prior.weight)
  numer <- prior.weight*ind.prob
  denom <- sum(numer)
  post.w <- (numer/denom) * n
  avg.p <- denom / sum(prior.weight)
  return(cbind(post.w, avg.p))
}
# ----- (4) Function: calculate the expected loss --------- #
# Eloss(w1, w0, m, g, p.success, p.failure)
# w1 / w0 = posterior weights (vector)
# m = prior mu (vector)
# g = prior gamma (vector)
# p.success = average prob of success
# p.failure = average prob of failure
# VALUE
#   Eloss calculate the expected loss
#-------------------------------------------------------- #
Eloss <- function(w1, w0, m, g, p.success, p.failure){
    # calculate posterior mean and variance based
```

```
    # on the value of y
    # y=1
    #w1 <- postweights[,1]
    post.Emu1 <- sum(w1 * m)/sum(w1)
    post.Ega1 <- sum(w1 * g)/sum(w1)
    post.Vmu1 <- sum(w1 * ((m - post.Emu1)^2))/sum(w1)
    post.Vga1 <- sum(w1 * ((g - post.Ega1)^2))/sum(w1)
    Loss1 <- post.Vmu1 + post.Vga1
    # y =0
    #w0 <- postweights[,2]
    post.Emu0 <- sum(w0 * m)/sum(w0)
    post.Ega0 <- sum(w0 * g)/sum(w0)
    post.Vmu0 <- sum(w0 * ((m - post.Emu0)^2))/sum(w0)
    post.Vga0 <- sum(w0 * ((g - post.Ega0)^2))/sum(w0)
    Loss0 <- post.Vmu0 + post.Vga0
    exp.loss <- p.success*Loss1 + p.failure*Loss0
    return(c(exp.loss, Loss1, Loss0,
            post.Vmu1, post.Vmu0, post.Vga1, post.Vga0))
}
```

B.1.1 One-step-look-ahead Procedure

```
# ************************************************************#
# One-step-look-ahead Procedure
#
# onestep.univariate(nsubj,doses,HCDlow,HCDhigh,
#                      n.mc,v.a,v.b,z.a,z.b)
# nsubj = number of subjects
# doses = a vector of possible doses
# HCDlow / HCDhigh = lower or higher values for HCD
# n.mc = number of samples from prior distribution
# v.a /v.b = parameters for beta distribution
#           (for v: related to the lower HCD)
# z.a /z.b = parameters for beta distribution
#            (for z: related to the higher HCD - lower HCD)
# VALUE
#   Returns nsubj and expected loss
# ************************************************************#
onestep.univariate <-
   function(nsubj,doses,HCDlow,HCDhigh,n.mc,v.a,v.b,z.a,z.b) {
  ndose <- length(doses)
  prior.p <- matrix(NA, nrow=n.mc, ncol=ndose)
  w.init <- rep(1, n.mc)
  p.k <- q.k <- rep(NA, n.mc)
```

```
# simulate MC sample for mu and gamma
prior.sample <- sim.prior(n.mc, cz, v.a, v.b, z.a, z.b)
# calculate the prior probability of success for each dose,
# and prob of failure is 1-prior.p
for (i in 1:ndose) {
  prior.p[,i] <- prior.prob(HCDlow, HCDhigh, doses[i],
            prior.sample[,1], prior.sample[,2])
}
# ------------------------ the first dose ---------------- #
p1 <- post.weights1 <- OPDOSE <- list(NULL)
WGT.OPT <- P.OPT <- list(NULL)
Exp.Loss1 <- matrix(NA, nrow=2^(1-1), ncol=ndose)
# the upper bound of dose index
up.dd <- ndose
for (dd1 in 1:up.dd){
  # calculate the posterior weights
  w1 <- w0 <- matrix(NA, nrow=n.mc, ncol=2)
  w1 <- post.wgt(w.init, prior.p[,dd1])
  w0 <- post.wgt(w.init, 1-prior.p[,dd1])
  post.weights1[[dd1]] <- cbind(w1[,1], w0[,1])
  # calculate the average prob of success and failure
  p1[[dd1]] <- c(w1[1,2], w0[1,2])
  Exp.Loss1[,dd1] <- Eloss(w1[,1], w0[,1], prior.sample[,1],
                     prior.sample[,2], w1[1,2], w0[1,2])[1]
}
OPDOSE1.idx <- apply(Exp.Loss1, 1, which.min)
# list of optimal dose
OPDOSE[[1]] <- doses[OPDOSE1.idx]
# list of weights and prob for stage 1
WGT.OPT[[1]] <- post.weights1[[OPDOSE1.idx]]
P.OPT[[1]] <- p1[[OPDOSE1.idx]]
# -------------------- the i^{th} dose (i>1) ------------- #
for (i in 2:nsubj) {
  priwgt.i0 <- WGT.OPT[[i-1]]
  up.ddi0 <- rep(NA, 2^(i-2))
  opdoseidx <- 2*OPDOSE[[i-1]]-2
  optimal.dose <- rep(NA, 2^(i-1))
  doseidx.i1 <- list(NULL)
  # n.bb: number of optimal dose for the (i-1)th stage
  # for stage i, the number of maximum dose is 2^(i-2)
  # n.bb <- 2^(i-2)
  for (bb in 1:n.bb) up.ddi0[bb] <- min(opdoseidx[bb]+1, 8)
  for (bb in 1:n.bb) {
    eloss.i0 <- matrix(NA, nrow=2, ncol=up.ddi0[bb])
```

```
for (ireps0 in 1:2) {
  for (dd.i0 in 1:up.ddi0[bb]) {
    w1.i0 <- post.wgt(priwgt.i0[,(2*(bb-1)+ireps0)],
                      prior.p[,dd.i0])
    w0.i0 <- post.wgt(priwgt.i0[,(2*(bb-1)+ireps0)],
                      1-prior.p[,dd.i0])
    # updated weight for i and prior weight for (i+1)
    priwgt.i1 <- cbind(w1.i0[,1], w0.i0[,1])
    up.ddi1 <- min(dd.i0+1, 8)
    eloss.i1 <- matrix(NA, nrow=2, ncol=up.ddi1)
      for (ireps1 in 1:2) {
        for (dd.i1 in 1:up.ddi1) {
          w1.i1 <- post.wgt(priwgt.i1[,ireps1],
                            prior.p[,dd.i1])
          w0.i1 <- post.wgt(priwgt.i1[,ireps1],
                            1-prior.p[,dd.i1])
          eloss.i1[ireps1,dd.i1] <-
               Eloss(w1.i1[,1], w0.i1[,1],
                     prior.sample[,1], prior.sample[,2],
                     w1.i1[1,2], w0.i1[1,2])[1]
        }
      }
      minELid.i1 <- apply(eloss.i1, 1, which.min)
      minEL.i1 <- c(eloss.i1[1, minELid.i1[1]],
                    eloss.i1[2, minELid.i1[2]])
      eloss.i0[ireps0, dd.i0] <-
        w1.i0[1,2]*minEL.i1[1] + w0.i0[1,2]*minEL.i1[2]
    }
  }
  minELid.i0 <- apply(eloss.i0, 1, which.min)
  minEL.i0 <- c(eloss.i0[1, minELid.i0[1]],
                eloss.i0[2, minELid.i0[2]])
  # list of optimal doses
  optimal.dose[(2*(bb-1)+1):(2*(bb-1)+2)] <-
            doses[minELid.i0]
}
OPDOSE[[i]] <- optimal.dose
wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^nsubj)
p.temp <- matrix(NA, nrow=1, ncol=2^i)
doseidx <- 2*OPDOSE[[i]] - 2
loss.last <- rep(NA, 2^i)
# given the optimal dose for stage i, obtain the updated
# weight and ave probability
for (mm in 1:(2^(i-1))) {
```

```
        # list of weight and prob
        w1.curr <-
           post.wgt(priwgt.i0[,mm], prior.p[,doseidx[mm]])
        w0.curr <-
           post.wgt(priwgt.i0[,mm], 1-prior.p[,doseidx[mm]])
        wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)]  <-
           cbind(w1.curr[,1], w0.curr[,1])
        p.temp[,(2*(mm-1)+1):(2*(mm-1)+2)] <-
           c(w1.curr[1,2], w0.curr[1,2])
        loss.last[(2*(mm-1)+1):(2*(mm-1)+2)] <-
           Eloss(w1.curr[,1], w0.curr[,1], prior.sample[,1],
           prior.sample[,2], w1.curr[1,2], w0.curr[1,2])[2:3]
      }
      WGT.OPT[[i]] <- wgt.temp
      P.OPT[[i]] <- p.temp
  }
OEL <- sum(P.OPT[[nsubj]]*loss.last)
return{c(nsubj, OEL)}
}
```

B.1.2 Two-step-look-ahead Procedure: Batch and Moving Window

```
# ***********************************************************#
# Two-step-look-ahead Procedure (Batch)
#  -- selecting two doses at a time
# twostepBatch.univariate(nsubj,doses,HCDlow,HCDhigh,
#                         n.mc,v.a,v.b,z.a,z.b)
# nsubj = number of subjects
# doses = a vector of possible doses
# HCDlow / HCDhigh = lower or higher values for HCD
# n.mc = number of samples from prior distribution
# v.a /v.b = parameters for beta distribution
#           (for v: related to the lower HCD)
# z.a /z.b = parameters for beta distribution
#            (for z: related to the higher HCD - lower HCD)
# VALUE
#   Returns nsubj and expected loss
# ***********************************************************#
twostepBatch.univariate <-
   function(nsubj,doses,HCDlow,HCDhigh,n.mc,v.a,v.b,z.a,z.b) {
  ndose <- length(doses)
  prior.p <- matrix(NA, nrow=n.mc, ncol=ndose)
  w.init <- rep(1, n.mc)
  # simulate MC sample for mu and gamma
```

```
prior.sample <- sim.prior(n.mc, cz, v.a, v.b, z.a, z.b)
# calculate the prior probability of success for each dose,
# and prob of failure is 1-prior.p
for (i in 1:ndose) {
  prior.p[,i] <- prior.prob(HCDlow, HCDhigh, doses[i],
                      prior.sample[,1], prior.sample[,2])
}
# check if nsubj is odd or even number
if ((nsubj %% 2) == 0) {
  odd <- 0; nsubj.i <- seq(from=1, to=nsubj-1, by=2) }
if ((nsubj %% 2) == 1) {
  odd <- 1; nsubj.i <- seq(from=1, to=nsubj,   by=2) }

# ------------ the first and second stage ------------ #
OPDOSE <- P.OPT <- list(NULL)
eloss1 <- matrix(NA, nrow=2^(1-1), ncol=ndose)
up.dd1 <- ndose
# dose12[y_1, dd1]
dose12 <- matrix(NA, nrow=2, ncol=up.dd1)
# avg.p12[y_2, dd2, y_1(ireps2), dd1]
avg.p12 <- array(NA, c(2,8,2,up.dd1))
# avg.p1[y_1, dd1]
avg.p1 <- matrix(NA, nrow=2, ncol=up.dd1)
for (dd1 in 1:up.dd1){
  # first stage #
  # calculate the posterior weights
  w1.1 <- w0.1 <- matrix(NA, nrow=n.mc, ncol=2)
  w1.1 <- post.wgt(w.init, prior.p[,dd1])
  w0.1 <- post.wgt(w.init, 1-prior.p[,dd1])
  priwgt1 <- cbind(w1.1[[1]], w0.1[[1]])
  # second stage #
  up.dd2 <- min(dd1+1, 8)
  eloss2 <- matrix(NA, nrow=2^(2-1), ncol=up.dd2)
  for (dd2 in 1:up.dd2) {
    for (ireps2 in 1:2){
      # update the posterior weights for
      # (y1,Y2)=(1,1),(1,0),(0,1),(0,0)
      # and calculate the expected loss
      w1.2<-post.wgt(priwgt1[,ireps2], prior.p[,dd2])
      w0.2<-post.wgt(priwgt1[,ireps2], 1-prior.p[,dd2])
      eloss.pre <- Eloss(w1.2[[1]], w0.2[[1]],
                   prior.sample[,1], prior.sample[,2],
                   w1.2[[2]], w0.2[[2]])[1:3]
      eloss2[ireps2,dd2] <- eloss.pre[1]
```

```
      avg.p12[,dd2,ireps2,dd1] <- c(w1.2[[2]], w0.2[[2]])
    }
  }
  minELid.i2 <- apply(eloss2, 1, which.min)
  dose12[, dd1] <- minELid.i2
  minEL.i2 <- c(eloss2[1, minELid.i2[1]],
                eloss2[2, minELid.i2[2]])
  eloss1[dd1] <- w1.1[[2]]*minEL.i2[1] +
                 w0.1[[2]]*minEL.i2[2]
  avg.p1[,dd1] <- c(w1.1[[2]], w0.1[[2]])
}
# optimal dose for stage 1 and 2
# stage 1
doseid.1 <- which.min(eloss1)
OPDOSE[[1]] <- doses[doseid.1]
# stage 2
doseid.2 <- dose12[,doseid.1]
OPDOSE[[2]] <- doses[doseid.2]
# average prob for stage 1 and 2
# stage 1
P.OPT[[1]] <- avg.p1[,doseid.1]
# stage 2
p2.pre <- rep(NA, 2^2)
for (mm in 1:(2^(2-1))) {
  idx <- doseid.2[mm]
  temp <- avg.p12[,idx, mm, doseid.1]
  p2.pre[(2*(mm-1)+1):(2*(mm-1)+2)] <- temp
}
P.OPT[[2]] <- p2.pre
# obtain posterior weigths given the first and second dose
# first dose
w1.1 <- post.wgt(w.init, prior.p[,doseid.1])
w0.1 <- post.wgt(w.init, 1-prior.p[,doseid.1])
# second dose
priwgt1 <- cbind(w1.1[[1]], w0.1[[1]])
wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^2)
for (mm in 1:(2^(2-1))) {
  idx <- doseid.2[mm]
  w1.2 <- post.wgt(priwgt1[,mm], prior.p[,idx])
  w0.2 <- post.wgt(priwgt1[,mm], 1-prior.p[,idx])
  wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)] <-
                             cbind(w1.2[[1]], w0.2[[1]])
}
rm(priwgt1, w1.1, w0.1, w1.2, w0.2)
```

```
gc(reset=TRUE)
# -------------------- the i^{th} dose (i>1) ------------- #
# -- if number of subject is even number, perform batch
# -- at the last two subjects
if (odd == 0) {
  for (ii in 2:length(nsubj.i)) {
    i <- nsubj.i[ii]
      priwgt0 <- wgt.temp
      up.dd0 <- rep(NA, 2^(i-2))
      opdoseidx <- 2*OPDOSE[[i-1]]-2
      # dose0
      dose0 <- matrix(NA, nrow=2, ncol=2^(i-2))
      # dose01[y_i, dd0, ireps0, bb]
      dose01 <- array(NA, c(2,8,2,2^(i-2)))
      # avg.p0[y_i, dd0, ireps0, bb]
      avg.p0 <- array(NA, c(2,8,2,2^(i-2)))
      # avg.p01[y_{i+1}, dd1, ireps1, dd0, ireps0, bb]
      avg.p01 <- array(NA, c(2,8,2,8,2,2^(i-2)))
      # loss01[y_{i+1}, dd1, ireps1, dd0, ireps0, bb]
      loss01 <- array(NA, c(2,8,2,8,2,2^(i-2)))
      # n.bb: number of optimal dose for the (i-1)th stage
      n.bb <- 2^(i-2)
      for (bb in 1:n.bb) up.dd0[bb]<-min(opdoseidx[bb]+1, 8)
      for (bb in 1:n.bb) {
        eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
        for (ireps0 in 1:2) {
          for (dd0 in 1:up.dd0[bb]) {
            w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                             prior.p[,dd0])
            w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                             1-prior.p[,dd0])
            # updated weight for  stage i and it is the
            # prior weight for stage (i+1)
            priwgt1 <- cbind(w1.0[[1]], w0.0[[1]])
            up.dd1 <- min(dd0+1, 8)
            eloss1 <- matrix(NA, nrow=2, ncol=up.dd1)
            for (ireps1 in 1:2) {
              for (dd1 in 1:up.dd1) {
                w1.1 <- post.wgt(priwgt1[,ireps1],
                                 prior.p[,dd1])
                w0.1 <- post.wgt(priwgt1[,ireps1],
                                 1-prior.p[,dd1])
                eloss.pre <- Eloss(w1.1[[1]], w0.1[[1]],
                     prior.sample[,1], prior.sample[,2],
```

```
                w1.1[[2]], w0.1[[2]])[1:3]
            eloss1[ireps1, dd1] <- eloss.pre[1]
            avg.p01[, dd1, ireps1, dd0, ireps0, bb] <-
                                c(w1.1[[2]], w0.1[[2]])
            loss01[ , dd1, ireps1, dd0, ireps0, bb] <-
                                eloss.pre[2:3]
          }
        }
        minELid.i1 <- apply(eloss1, 1, which.min)
        dose01[ , dd0, ireps0, bb] <- minELid.i1
        minEL.i1 <- c(eloss1[1, minELid.i1[1]],
                      eloss1[2, minELid.i1[2]])
        eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.i1[1] +
                               w0.0[[2]]*minEL.i1[2]
        avg.p0[ ,dd0,ireps0,bb]<-c(w1.0[[2]], w0.0[[2]])
      }
    }
    minELid.i0 <- apply(eloss0, 1, which.min)
    dose0[ , bb] <- minELid.i0
    minEL.i0 <- c(eloss0[1, minELid.i0[1]],
                  eloss0[2, minELid.i0[2]])
}
# optimal dose for stage i and i+1
# stage i
doseid.0 <- rep(NA, 2^(i-1))
for (bb in 1:n.bb) {
  doseid.0[(2*(bb-1)+1):(2*(bb-1)+2)] <- dose0[ , bb]
}
OPDOSE[[i]] <- doses[doseid.0]
# stage (i+1)
doseid.1 <- rep(NA, 2^i)
d1.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0 <- doseid.0[2*(bb-1)+m0]
    temp1 <- dose01[, idx, m0, bb]
    d1.pre[2*(bb-1)+m0,] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
# average prob for stage i, i+1 and i+2
# stage i
p0.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
```

```
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0 <- doseid.0[2*(bb-1)+m0]
    temp1 <- avg.p0[,idx0,m0,bb]
    p0.pre[2*(bb-1)+m0,] <- temp1
  }
}
P.OPT[[i]] <- as.vector(t(p0.pre))
# stage (i+1)
p1.pre <- loss.pre <- matrix(NA, nrow=2^i, ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0.i <- 2*(bb-1)+m0
    idx0 <- doseid.0[idx0.i]
    for (m1 in 1:2) {
      idx1 <- doseid.1[2*(idx0.i-1)+m1]
      temp1 <- avg.p01[ , idx1, m1, idx0, m0, bb]
      loss.temp1 <- loss01[ ,idx1, m1, idx0, m0, bb]
      p1.pre[2*(idx0.i-1)+m1, ] <- temp1
      loss.pre[2*(idx0.i-1)+m1, ] <- loss.temp1
    }
  }
}
P.OPT[[i+1]] <- as.vector(t(p1.pre))
loss.last <- as.vector(t(loss.pre))
# obtain the posterior weights for (i+1)th stage
wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^i)
# stage i
for (mm in 1:(2^(i-1))) {
  idx <- doseid.0[mm]
  w1.curr <- post.wgt(priwgt0[,mm], prior.p[,idx])
  w0.curr <- post.wgt(priwgt0[,mm], 1-prior.p[,idx])
  wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)] <-
                    cbind(w1.curr[[1]], w0.curr[[1]])
}
# stage i+1
priwgt1 <- wgt.temp
wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^(i+1))
for (mm in 1:(2^i)) {
  idx <- doseid.1[mm]
  w1.curr <- post.wgt(priwgt1[,mm], prior.p[,idx])
  w0.curr <- post.wgt(priwgt1[,mm], 1-prior.p[,idx])
  wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)] <-
                    cbind(w1.curr[[1]], w0.curr[[1]])
```

```
      }
  }
rm(priwgt0, w1.0, w0.0, priwgt1, w1.1, w0.1, w1.curr,
   w0.curr, wgt.temp, dose0, dose01, avg.p0, avg.p01, loss01)
gc(reset=TRUE)
}
# -- If number of subject is odd number, at stages 3 to
# -- (i-1), perform two-step batch, At the last stage,
# -- perform one-step-look-ahead preocedure
if (odd == 1) {
  for (ii in 2:length(nsubj.i)) {
    i <- nsubj.i[ii]
    if (i < nsubj) {
      priwgt0 <- wgt.temp
      up.dd0 <- rep(NA, 2^(i-2))
      opdoseidx <- 2*OPDOSE[[i-1]]-2
      dose0 <- matrix(NA, nrow=2, ncol=2^(i-2))
      dose01 <- array(NA, c(2,8,2,2^(i-2)))
      avg.p0 <- array(NA, c(2,8,2,2^(i-2)))
      avg.p01 <- array(NA, c(2,8,2,8,2,2^(i-2)))
      loss01 <- array(NA, c(2,8,2,8,2,2^(i-2)))
      n.bb <- 2^(i-2)
      for (bb in 1:n.bb) p.dd0[bb]<-min(opdoseidx[bb]+1, 8)
      for (bb in 1:n.bb) {
        eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
        for (ireps0 in 1:2) {
          for (dd0 in 1:up.dd0[bb]) {
            w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                             prior.p[,dd0])
            w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                             1-prior.p[,dd0])
            priwgt1 <- cbind(w1.0[[1]], w0.0[[1]])
            up.dd1 <- min(dd0+1, 8)
            eloss1 <- matrix(NA, nrow=2, ncol=up.dd1)
            for (ireps1 in 1:2) {
              for (dd1 in 1:up.dd1) {
                w1.1 <- post.wgt(priwgt1[,ireps1],
                                 prior.p[,dd1])
                w0.1 <- post.wgt(priwgt1[,ireps1],
                                 1-prior.p[,dd1])
                eloss.pre <- Eloss(w1.1[[1]], w0.1[[1]],
                        prior.sample[,1], prior.sample[,2],
                        w1.1[[2]], w0.1[[2]])[1:3]
                eloss1[ireps1, dd1] <- eloss.pre[1]
```

```
          avg.p01[,dd1,ireps1,dd0,ireps0,bb] <-
                                  c(w1.1[[2]], w0.1[[2]])
          loss01[,dd1,ireps1,dd0,ireps0,bb] <-
                                  eloss.pre[2:3]
        }
      }
      minELid.i1 <- apply(eloss1, 1, which.min)
      dose01[, dd0, ireps0, bb] <- minELid.i1
      minEL.i1 <- c(eloss1[1, minELid.i1[1]],
                    eloss1[2, minELid.i1[2]])
      eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.i1[1] +
                             w0.0[[2]]*minEL.i1[2]
      avg.p0[,dd0,ireps0,bb] <- c(w1.0[[2]],w0.0[[2]])
    }
  }
  minELid.i0 <- apply(eloss0, 1, which.min)
  dose0[,bb] <- minELid.i0
  minEL.i0 <- c(eloss0[1, minELid.i0[1]],
                eloss0[2, minELid.i0[2]])
}
# optimal dose for stage i and i+1
# stage i
doseid.0 <- rep(NA, 2^(i-1))
for (bb in 1:n.bb) {
  doseid.0[(2*(bb-1)+1):(2*(bb-1)+2)] <- dose0[,bb]
}
OPDOSE[[i]] <- doses[doseid.0]
# stage (i+1)
doseid.1 <- rep(NA, 2^i)
d1.pre <- matrix(NA, nrow=n.bb*2, ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx <- doseid.0[2*(bb-1)+m0]
    temp1 <- dose01[, idx, m0,bb]
    d1.pre[2*(bb-1)+m0,] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
# average prob for stage i, i+1 and i+2
# stage i
p0.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
```

```
    idx0 <- doseid.0[2*(bb-1)+m0]
    temp1 <- avg.p0[ , idx0, m0, bb]
    p0.pre[2*(bb-1)+m0,] <- temp1
  }
}
P.OPT[[i]] <- as.vector(t(p0.pre))
# stage (i+1)
p1.pre <- loss.pre <- matrix(NA, nrow=2^i, ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0.i <- 2*(bb-1)+m0
    idx0 <- doseid.0[idx0.i]
    for (m1 in 1:2) {
      idx1 <- doseid.1[2*(idx0.i-1)+m1]
      temp1 <- avg.p01[ , idx1, m1, idx0, m0, bb]
      loss.temp1 <- loss01[ , idx1, m1, idx0, m0, bb]
      p1.pre[2*(idx0.i-1)+m1, ] <- temp1
      loss.pre[2*(idx0.i-1)+m1, ] <- loss.temp1
    }
  }
}
P.OPT[[i+1]] <- as.vector(t(p1.pre))
loss.last <- as.vector(t(loss.pre))
# obtain the posterior weights
wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^i)
# stage i
for (mm in 1:(2^(i-1))) {
  idx <- doseid.0[mm]
  w1.curr <- post.wgt(priwgt0[,mm], prior.p[,idx])
  w0.curr <- post.wgt(priwgt0[,mm], 1-prior.p[,idx])
  wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)]  <-
                    cbind(w1.curr[[1]], w0.curr[[1]])
}
# stage i+1
priwgt1 <- wgt.temp
wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^(i+1))
# given the optimal dose for stage i and i+1,
# obtain the updated weight and ave probability
for (mm in 1:(2^i)) {
  idx <- doseid.1[mm]
  w1.curr <- post.wgt(priwgt1[,mm], prior.p[,idx])
  w0.curr <- post.wgt(priwgt1[,mm], 1-prior.p[,idx])
  wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)] <-
                      cbind(w1.curr[[1]], w0.curr[[1]])
```

```
  }
  rm(priwgt0, w1.0, w0.0, priwgt1, w1.1, w0.1, w1.curr,
     w0.curr, wgt.temp, dose0, dose01, avg.p0, avg.p01, loss01)
  gc(reset=TRUE)
}
# -- at the last stage: one-step-look-ahead
if (i == nsubj) {
  priwgt0 <- wgt.temp
  up.dd0 <- rep(NA, 2^(i-2))
  opdoseidx <- 2*OPDOSE[[i-1]]-2
  dose0 <- matrix(NA, nrow=2, ncol=2^(i-2))
  avg.p0 <- array(NA, c(2,8,2,2^(i-2)))
  loss0 <- array(NA, c(2,8,2,2^(i-2)))
  n.bb <- 2^(i-2)
  for (bb in 1:n.bb) up.dd0[bb]<-min(opdoseidx[bb]+1, 8)
  for (bb in 1:n.bb) {
    eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
    temp <- matrix(NA, nrow=2, ncol=up.dd0[bb])
    for (ireps0 in 1:2) {
      for (dd0 in 1:up.dd0[bb]) {
        w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                         prior.p[,dd0])
        w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                         1-prior.p[,dd0])
        eloss.pre <- Eloss(w1.0[[1]], w0.0[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.0[[2]], w0.0[[2]])[1:3]
        eloss0[ireps0, dd0] <- eloss.pre[1]
        avg.p0[,dd0,ireps0,bb] <- c(w1.0[[2]],w0.0[[2]])
        loss0[, dd0, ireps0, bb] <- eloss.pre[2:3]
      }
    }
    minELid.i0 <- apply(eloss0, 1, which.min)
    dose0[,bb] <- minELid.i0
    minEL.i0 <- c(eloss0[1, minELid.i0[1]],
                  eloss0[2, minELid.i0[2]])
  }
  # optimal dose for stage i
  doseid.0 <- rep(NA, 2^(i-1))
  for (bb in 1:n.bb) {
    doseid.0[(2*(bb-1)+1):(2*(bb-1)+2)] <- dose0[,bb]
  }
  OPDOSE[[i]] <- doses[doseid.0]
  # avg prob for stage i
```

```
        p0.pre <- loss.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
        for (bb in 1:n.bb) {
          for (m0 in 1:2) {
            idx0 <- doseid.0[2*(bb-1)+m0]
            temp1 <- avg.p0[ , idx0, m0, bb]
            p0.pre[2*(bb-1)+m0,] <- temp1
            loss.temp1 <- loss0[ , idx0, m0, bb]
            loss.pre[2*(bb-1)+m0,] <- loss.temp1
          }
        }
        P.OPT[[i]] <- as.vector(t(p0.pre))
        loss.last <- as.vector(t(loss.pre))
      }
    }
    rm(priwgt0, w1.0, w0.0, wgt.temp, dose0, avg.p0, loss0)
    gc(reset=TRUE)
  }
  prob.matrix <- matrix(NA, nrow=2^nsubj, ncol=nsubj)
  for (ncolumn in 1:nsubj) {
    p <- rep(P.OPT[[ncolumn]], each=2^(nsubj-ncolumn))
    prob.matrix[,ncolumn] <- p
  }
  prob.expt <- apply(prob.matrix, 1, prod)
  OEL.2batch <- sum(prob.expt*loss.last)
  rstt <- matrix(c(nsubj, OEL.2batch), ncol=2, nrow=1)
  return(rstt)
}

# ***********************************************************#
# Two-step-look-ahead Procedure (Moving Window)
#  -- selecting one doses at a time
# twostepMoving.univariate(nsubj,doses,HCDlow,HCDhigh,
#                          n.mc,v.a,v.b,z.a,z.b)
# nsubj = number of subjects
# doses = a vector of possible doses
# HCDlow / HCDhigh = lower or higher values for HCD
# n.mc = number of samples from prior distribution
# v.a /v.b = parameters for beta distribution
#           (for v: related to the lower HCD)
# z.a /z.b = parameters for beta distribution
#            (for z: related to the higher HCD - lower HCD)
# VALUE
#   Returns nsubj and expected loss
# ***********************************************************#
twostepMoving.univariate <-
```

```
 function(nsubj,doses,HCDlow,HCDhigh,n.mc,v.a,v.b,z.a,z.b) {
ndose <- length(doses)
prior.p <- matrix(NA, nrow=n.mc, ncol=ndose)
w.init <- rep(1, n.mc)
prior.sample <- sim.prior(n.mc, cz, v.a, v.b, z.a, z.b)
for (i in 1:ndose) {
  prior.p[,i] <- prior.prob(HCDlow, HCDhigh, doses[i],
                      prior.sample[,1], prior.sample[,2])
}
# ------------------------ the first dose ---------------- #
OPDOSE <- P.OPT <- list(NULL)
eloss1 <- matrix(NA, nrow=2^(1-1), ncol=ndose)
up.dd1 <- ndose
avg.p1 <- matrix(NA, nrow=2, ncol=up.dd1)
for (dd1 in 1:up.dd1){
  # first stage #
  w1.1 <- w0.1 <- matrix(NA, nrow=n.mc, ncol=2)
  w1.1 <- post.wgt(w.init, prior.p[,dd1])
  w0.1 <- post.wgt(w.init, 1-prior.p[,dd1])
  priwgt2 <- cbind(w1.1[[1]], w0.1[[1]])
  # second stage #
  up.dd2 <- min(dd1+1, 8)
  eloss2 <- matrix(NA, nrow=2^(2-1), ncol=up.dd2)
  for (dd2 in 1:up.dd2) {
    for (ireps2 in 1:2){
      w1.2 <- post.wgt(priwgt2[,ireps2], prior.p[,dd2])
      w0.2 <- post.wgt(priwgt2[,ireps2], 1-prior.p[,dd2])
      eloss.pre <- Eloss(w1.2[[1]], w0.2[[1]],
           prior.sample[,1], prior.sample[,2],
           w1.2[[2]], w0.2[[2]])[1:3]
      eloss2[ireps2,dd2] <- eloss.pre[1]
    }
  }
  minELid.i2 <- apply(eloss2, 1, which.min)
  minEL.i2 <- c(eloss2[1, minELid.i2[1]],
                eloss2[2, minELid.i2[2]])
  eloss1[dd1] <- w1.1[[2]]*minEL.i2[1] +
                 w0.1[[2]]*minEL.i2[2]
  avg.p1[,dd1] <- c(w1.1[[2]], w0.1[[2]])
}
# optimal dose for stage 1 and 2
# stage 1
doseid.1 <- which.min(eloss1)
OPDOSE[[1]] <- doses[doseid.1]
```

```
# average prob for stage 1 and 2
# stage 1
P.OPT[[1]] <- avg.p1[,doseid.1]
w1 <- post.wgt(w.init, prior.p[,doseid.1])
w0 <- post.wgt(w.init, 1-prior.p[,doseid.1])
wgt.temp <- cbind(w1[[1]], w0[[1]])
rm(w1, w0, w1.1, w0.1, w1.2, w0.2, priwgt2)
gc(reset=TRUE)
# -------------------- the i^{th} dose (i>1) ------------- #
# - for i<(nsubj -1), select the optimal dose for each stage
# - for i=(nsubj-1), select the optimal doses for i and
# -   (i+1) stage
for (i in 2:nsubj) {
  if (i < (nsubj-1)) {
    priwgt0 <- wgt.temp
    up.dd0 <- rep(NA, 2^(i-2))
    opdoseidx <- 2*OPDOSE[[i-1]]-2
    dose0 <- matrix(NA, nrow=2, ncol=2^(i-2))
    avg.p0 <- array(NA, c(2,8,2,2^(i-2)))
    n.bb <- 2^(i-2)
    for (bb in 1:n.bb) up.dd0[bb] <- min(opdoseidx[bb]+1, 8)
    for (bb in 1:n.bb) {
      eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
      for (ireps0 in 1:2) {
        for (dd0 in 1:up.dd0[bb]) {
          w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                           prior.p[,dd0])
          w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                           1-prior.p[,dd0])
          priwgt1 <- cbind(w1.0[[1]], w0.0[[1]])
          up.dd1 <- min(dd0+1, 8)
          eloss1 <- matrix(NA, nrow=2, ncol=up.dd1)
          for (ireps1 in 1:2) {
            for (dd1 in 1:up.dd1) {
              w1.1 <- post.wgt(priwgt1[,ireps1],
                               prior.p[,dd1])
              w0.1 <- post.wgt(priwgt1[,ireps1],
                               1-prior.p[,dd1])
              eloss1[ireps1,dd1]< Eloss(w1.1[[1]], w0.1[[1]],
                          prior.sample[,1], prior.sample[,2],
                          w1.1[[2]], w0.1[[2]])[1]
            }
          }
          minELid.i1 <- apply(eloss1, 1, which.min)
```

```
        minEL.i1 <- c(eloss1[1, minELid.i1[1]],
                      eloss1[2, minELid.i1[2]])
        eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.i1[1] +
                               w0.0[[2]]*minEL.i1[2]
        avg.p0[,dd0, ireps0, bb] <- c(w1.0[[2]],w0.0[[2]])
      }
    }
    minELid.i0 <- apply(eloss0, 1, which.min)
    minEL.i0 <- c(eloss0[1, minELid.i0[1]],
                  eloss0[2, minELid.i0[2]])
    dose0[,bb] <- minELid.i0
  }
  # optimal dose for stage i
  # stage i
  doseid.0 <- rep(NA, 2^(i-1))
  for (bb in 1:n.bb) {
      doseid.0[(2*(bb-1)+1):(2*(bb-1)+2)] <- dose0[ ,bb]
  }
  OPDOSE[[i]] <- doses[doseid.0]
  # average prob for stage i, i+1 and i+2
  # stage i
  p0.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
  for (bb in 1:n.bb) {
    for (m0 in 1:2) {
      idx0 <- doseid.0[2*(bb-1)+m0]
      temp1 <- avg.p0[ , idx0, m0, bb]
      p0.pre[2*(bb-1)+m0, ] <- temp1
    }
  }
  P.OPT[[i]] <- as.vector(t(p0.pre))
  # obtain the posterior weights
  wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^i)
  # stage i
  for (mm in 1:(2^(i-1))) {
    idx <- doseid.0[mm]
    w1.curr <- post.wgt(priwgt0[,mm], prior.p[,idx])
    w0.curr <- post.wgt(priwgt0[,mm], 1-prior.p[,idx])
    wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)]  <-
                        cbind(w1.curr[[1]], w0.curr[[1]])
  }
}
if (i == (nsubj-1)) {
  priwgt0 <- wgt.temp
  up.dd0 <- rep(NA, 2^(i-2))
```

```
opdoseidx <- 2*OPDOSE[[i-1]]-2
dose0 <- matrix(NA, nrow=2, ncol=2^(i-2))
dose01 <- array(NA, c(2,8,2,2^(i-2)))
avg.p0 <- array(NA, c(2,8,2,2^(i-2)))
avg.p01 <- array(NA, c(2,8,2,8,2,2^(i-2)))
loss01 <- array(NA, c(2,8,2,8,2,2^(i-2)))
n.bb <- 2^(i-2)
for (bb in 1:n.bb) up.dd0[bb] <- min(opdoseidx[bb]+1, 8)
for (bb in 1:n.bb) {
  eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
  for (ireps0 in 1:2) {
    for (dd0 in 1:up.dd0[bb]) {
      w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                       prior.p[,dd0])
      w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                       1-prior.p[,dd0])
      priwgt1 <- cbind(w1.0[[1]], w0.0[[1]])
      up.dd1 <- min(dd0+1, 8)
      eloss1 <- matrix(NA, nrow=2, ncol=up.dd1)
      for (ireps1 in 1:2) {
        for (dd1 in 1:up.dd1) {
          w1.1 <- post.wgt(priwgt1[,ireps1],
                           prior.p[,dd1])
          w0.1 <- post.wgt(priwgt1[,ireps1],
                           1-prior.p[,dd1])
          eloss.pre <- Eloss(w1.1[[1]], w0.1[[1]],
               prior.sample[,1], prior.sample[,2],
               w1.1[[2]], w0.1[[2]])[1:3]
          eloss1[ireps1, dd1] <- eloss.pre[1]
          avg.p01[, dd1, ireps1, dd0, ireps0, bb] <-
                                c(w1.1[[2]], w0.1[[2]])
          loss01[,dd1,ireps1,dd0,ireps0,bb] <-
                                          eloss.pre[2:3]
        }
      }
      minELid.i1 <- apply(eloss1, 1, which.min)
      dose01[, dd0, ireps0, bb] <- minELid.i1
      minEL.i1 <- c(eloss1[1, minELid.i1[1]],
                    eloss1[2, minELid.i1[2]])
      eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.i1[1] +
                             w0.0[[2]]*minEL.i1[2]
      avg.p0[,dd0, ireps0, bb] <- c(w1.0[[2]],w0.0[[2]])
    }
  }
```

```
  minELid.i0 <- apply(eloss0, 1, which.min)
  dose0[,bb] <- minELid.i0
}
# optimal dose for stage i and i+1
# stage i
doseid.0 <- rep(NA, 2^(i-1))
for (bb in 1:n.bb) {
    doseid.0[(2*(bb-1)+1):(2*(bb-1)+2)] <- dose0[,bb]
}
OPDOSE[[i]] <- doses[doseid.0]
# stage (i+1)
doseid.1 <- rep(NA, 2^i)
d1.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx <- doseid.0[2*(bb-1)+m0]
    temp1 <- dose01[, idx, m0, bb]
    d1.pre[(2*(bb-1)+m0),] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
# average prob for stage i, i+1 and i+2
# stage i
p0.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0 <- doseid.0[2*(bb-1)+m0]
    temp1 <- avg.p0[ , idx0, m0, bb]
    p0.pre[2*(bb-1)+m0, ] <- temp1
  }
}
P.OPT[[i]] <- as.vector(t(p0.pre))
# stage (i+1)
p1.pre <- loss.pre <- matrix(NA, nrow=2^i, ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0.i <- 2*(bb-1)+m0
    idx0 <- doseid.0[idx0.i]
    for (m1 in 1:2) {
      idx1 <- doseid.1[2*(idx0.i-1)+m1]
      temp1 <- avg.p01[ , idx1, m1, idx0, m0, bb]
      loss.temp1 <- loss01[ , idx1, m1, idx0, m0, bb]
      p1.pre[2*(idx0.i-1)+m1,] <- temp1
```

```
            loss.pre[2*(idx0.i-1)+m1,] <- loss.temp1
          }
        }
      }
      P.OPT[[i+1]] <- as.vector(t(p1.pre))
      loss.last <- as.vector(t(loss.pre))
    }
  }
  rm(priwgt0, w1.0, w0.0, priwgt1, w1.1, w0.1, wgt.temp,
     dose0, dose01, avg.p0, avg.p01, loss01)
  gc(reset=TRUE)
  prob.matrix <- matrix(NA, nrow=2^nsubj, ncol=nsubj)
  for (ncolumn in 1:nsubj) {
    p <- rep(P.OPT[[ncolumn]], each=2^(nsubj-ncolumn))
    prob.matrix[,ncolumn] <- p
  }
  prob.expt <- apply(prob.matrix, 1, prod)
  OEL.2moving <- sum(prob.expt*loss.last)
  rstt <- matrix(c(nsubj, OEL.2moving), ncol=2, nrow=1)
  return(rstt)
}
```

B.1.3 Three-step-look-ahead Procedure: Moving Window

```
# ************************************************************#
# Three-step-look-ahead Procedure (Moving Window)
#  -- selecting one doses at a time
# threestepMoving.univariate(nsubj,doses,HCDlow,HCDhigh,
#                            n.mc,v.a,v.b,z.a,z.b)
# nsubj = number of subjects
# doses = a vector of possible doses
# HCDlow / HCDhigh = lower or higher values for HCD
# n.mc = number of samples from prior distribution
# v.a /v.b = parameters for beta distribution
#           (for v: related to the lower HCD)
# z.a /z.b = parameters for beta distribution
#            (for z: related to the higher HCD - lower HCD)
# VALUE
#   Returns nsubj and expected loss
# ************************************************************#
threestepMoving.univariate <-
   function(nsubj,doses,HCDlow,HCDhigh,n.mc,v.a,v.b,z.a,z.b) {
  ndose <- length(doses)
  prior.p <- matrix(NA, nrow=n.mc, ncol=ndose)
```

```
w.init <- rep(1, n.mc)
prior.sample <- sim.prior(n.mc, cz, v.a, v.b, z.a, z.b)
for (i in 1:ndose) {
  prior.p[,i] <- prior.prob(HCDlow, HCDhigh, doses[i],
                      prior.sample[,1], prior.sample[,2])
}
# ----------------------- the first dose --------------- #
OPDOSE <- P.OPT <- list(NULL)
up.dd1 <- ndose
eloss1 <- rep(NA, up.dd1)
avg.p1 <- matrix(NA, nrow=2, ncol=up.dd)
for (dd1 in 1:up.dd1){
  # first stage #
  # calculate the posterior weights
  w1.1 <- post.wgt(w.init, prior.p[,dd1])
  w0.1 <- post.wgt(w.init, 1-prior.p[,dd1])
  priwgt2 <- cbind(w1.1[[1]], w0.1[[1]])
  # second stage #
  up.dd2 <- min(dd1+1, 8)
  eloss2 <- matrix(NA, nrow=2, ncol=up.dd2)
  for (ireps2 in 1:2){
    for (dd2 in 1:up.dd2) {
      w1.2 <- post.wgt(priwgt2[,ireps2], prior.p[,dd2])
      w0.2 <- post.wgt(priwgt2[,ireps2], 1-prior.p[,dd2])
      priwgt3 <- cbind(w1.2[[1]], w0.2[[1]])
      up.dd3 <- min(dd2+1, 8)
      eloss3 <- matrix(NA, nrow=2, ncol=up.dd3)
      # third stage #
      # ireps3=1: y2=1; ireps3=2: y2=0
      for (ireps3 in 1:2) {
        for (dd3 in 1:up.dd3) {
          w1.3 <- post.wgt(priwgt3[,ireps3],
                           prior.p[,dd3])
          w0.3 <- post.wgt(priwgt3[,ireps3],
                           1-prior.p[,dd3])
          eloss.pre <- Eloss(w1.3[[1]], w0.3[[1]],
                      prior.sample[,1], prior.sample[,2],
                      w1.3[[2]], w0.3[[2]])[1:3]
          eloss3[ireps3, dd3] <- eloss.pre[1]
        }
      }
      minELid.i3 <- apply(eloss3, 1, which.min)
      minEL.i3 <- c(eloss3[1, minELid.i3[1]],
                    eloss3[2, minELid.i3[2]])
```

```
      eloss2[ireps2, dd2] <- w1.2[[2]]*minEL.i3[1] +
                             w0.2[[2]]*minEL.i3[2]
    }
  }
  minELid.i2 <- apply(eloss2, 1, which.min)
  minEL.i2 <- c(eloss2[1, minELid.i2[1]],
                eloss2[2, minELid.i2[2]])
  eloss1[dd1] <- w1.1[[2]]*minEL.i2[1] +
                 w0.1[[2]]*minEL.i2[2]
  avg.p1[ ,dd1] <- c(w1.1[[2]], w0.1[[2]])
}
# optimal doses for stage 1
doseid.1 <- which.min(eloss1)
# stage 1
OPDOSE[[1]] <- doses[doseid.1]
P.OPT[[1]] <- avg.p1[ ,doseid.1]
w1.1 <- post.wgt(w.init, prior.p[,doseid.1])
w0.1 <- post.wgt(w.init, 1-prior.p[,doseid.1])
wgt.temp <- cbind(w1.1[[1]], w0.1[[1]])
rm(w1.1, w0.1, priwgt2, w1.2, w0.2, priwgt3, w1.3, w0.3)
gc(reset=TRUE)
# -------------------- the i^{th} dose (i>1) ------------- #
# - for i<(nsubj -2), select the optimal dose for each stage
# - for i=(nsubj-2), select the optimal doses for i, (i+1),
# - and (i+2) stage
for (i in 2:nsubj) {
  if (i < (nsubj-2)) {
    priwgt0 <- wgt.temp
    up.dd0 <- rep(NA, 2^(i-2))
    opdoseidx <- 2*OPDOSE[[i-1]]-2
    dose0 <- matrix(NA, nrow=2, ncol=2^(i-2))
    avg.p0 <- array(NA, c(2,8,2,2^(i-2)))
    n.bb <- 2^(i-2)
    for (bb in 1:n.bb) up.dd0[bb] <- min(opdoseidx[bb]+1, 8)
    for (bb in 1:n.bb) {
      eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
      for (ireps0 in 1:2) {
        for (dd0 in 1:up.dd0[bb]) {
          w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                           prior.p[,dd0])
          w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                           1-prior.p[,dd0])
          priwgt1 <- cbind(w1.0[[1]], w0.0[[1]])
          up.dd1 <- min(dd0+1, 8)
```

```
      eloss1 <- matrix(NA, nrow=2, ncol=up.dd1)
      for (ireps1 in 1:2) {
        for (dd1 in 1:up.dd1) {
          w1.1 <- post.wgt(priwgt1[,ireps1],
                           prior.p[,dd1])
          w0.1 <- post.wgt(priwgt1[,ireps1],
                           1-prior.p[,dd1])
          priwgt2 <- cbind(w1.1[[1]], w0.1[[1]])
          up.dd2 <- min(dd1+1, 8)
          eloss2 <- matrix(NA, nrow=2, ncol=up.dd2)
          for (ireps2 in 1:2) {
            for (dd2 in 1:up.dd2) {
               w1.2 <- post.wgt(priwgt2[,ireps2],
                                prior.p[,dd2])
               w0.2 <- post.wgt(priwgt2[,ireps2],
                                1-prior.p[,dd2])
               eloss2[ireps2, dd2] <-
                      Eloss(w1.2[[1]], w0.2[[1]],
                      prior.sample[,1], prior.sample[,2],
                      w1.2[[2]], w0.2[[2]])[1]
            }
          }
          minELid.i2 <- apply(eloss2, 1, which.min)
          minEL.i2 <- c(eloss2[1, minELid.i2[1]],
                        eloss2[2, minELid.i2[2]])
          eloss1[ireps1, dd1] <- w1.1[[2]]*minEL.i2[1] +
                                 w0.1[[2]]*minEL.i2[2]
        }
      }
      minELid.i1 <- apply(eloss1, 1, which.min)
      minEL.i1 <- c(eloss1[1, minELid.i1[1]],
                    eloss1[2, minELid.i1[2]])
      eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.i1[1] +
                             w0.0[[2]]*minEL.i1[2]
      avg.p0[,dd0,ireps0,bb] <- c(w1.0[[2]], w0.0[[2]])
    }
  }
  minELid.i0 <- apply(eloss0, 1, which.min)
  minEL.i0 <- c(eloss0[1, minELid.i0[1]],
                eloss0[2, minELid.i0[2]])
  dose0[,bb] <- minELid.i0
}
# optimal dose for stage i, i+1 and i+2
# stage i
```

```
  doseid.0 <- rep(NA, 2^(i-1))
  for (bb in 1:n.bb) {
      doseid.0[(2*(bb-1)+1):(2*(bb-1)+2)] <- dose0[,bb]
  }
  OPDOSE[[i]] <- doses[doseid.0]
  # average prob for stage i, i+1 and i+2
  # stage i
  p0.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
  for (bb in 1:n.bb) {
    for (m0 in 1:2) {
      idx0 <- doseid.0[2*(bb-1)+m0]
      temp1 <- avg.p0[, idx0, m0, bb]
      p0.pre[(2*(bb-1)+m0),] <- temp1
    }
  }
  P.OPT[[i]] <- as.vector(t(p0.pre))
  # obtain posterior weight at stage i
  wgt.temp <- matrix(NA, nrow=n.mc, ncol=2^i)
  for (mm in 1:(2^(i-1))) {
    idx <- doseid.0[mm]
    w1.curr <- post.wgt(priwgt0[,mm], prior.p[,idx])
    w0.curr <- post.wgt(priwgt0[,mm], 1-prior.p[,idx])
    wgt.temp[,(2*(mm-1)+1):(2*(mm-1)+2)] <-
                    cbind(w1.curr[[1]], w0.curr[[1]])
  }
}
if (i == (nsubj-2)) {
  priwgt0 <- wgt.temp
  up.dd0 <- rep(NA, 2^(i-2))
  opdoseidx <- 2*OPDOSE[[i-1]]-2
  n.bb <- 2^(i-2)
  dose0 <- matrix(NA, nrow=2, ncol=n.bb)
  # -- dose01, a 4-way array. For each block and each
  # -- outcome y_i, there is a 2x8 matrix.
  # -- row- ireps1=1: y_i=1; ireps1=2: y_i=0.
  # -- for stage i, there are 2^(i-2) blocks
  # -- dose01[ireps1, dd0, ireps0, bb]
  dose01 <- array(NA, c(2,8,2,n.bb))
  # -- dose01, a 6-way array. For each block and each
  # -- outcome y_i, there is a 2x8 matrix.
  # -- row- ireps1=1: y_i=1; ireps1=2: y_i=0
  # -- for stage i, there are 2^(i-2) blocks
  # -- dose012[ireps2, dd1, ireps1, dd0, ireps0, bb]
  dose012 <- array(NA, c(2,8,2,8,2,n.bb))
```

```
# -- avg.p012[y_{i+2},dd2,ireps2,dd1,ireps1,dd0,ireps0,bb]
avg.p012 <- array(NA, c(2,8,2,8,2,8,2,n.bb))
# -- avg.01[y_{i+1}, dd1, ireps1, dd0, ireps0, bb]
avg.p01 <- array(NA, c(2,8,2,8,2,n.bb))
# --  avg.p0[y_i, dd0, ireps0, bb]
avg.p0 <- array(NA, c(2,8,2,n.bb))
# -- loss012[y_{i+2},dd2,ireps2,dd1,ireps1,dd0,ireps0,bb]
loss012 <- array(NA, c(2,8,2,8,2,8,2,n.bb))
for (bb in 1:n.bb) up.dd0[bb] <- min(opdoseidx[bb]+1, 8)
for (bb in 1:n.bb) {
  eloss0 <- matrix(NA, nrow=2, ncol=up.dd0[bb])
  for (ireps0 in 1:2) {
    for (dd0 in 1:up.dd0[bb]) {
      w1.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                       prior.p[,dd0])
      w0.0 <- post.wgt(priwgt0[,(2*(bb-1)+ireps0)],
                       1-prior.p[,dd0])
      priwgt1 <- cbind(w1.0[[1]], w0.0[[1]])
      up.dd1 <- min(dd0+1, 8)
      eloss1 <- matrix(NA, nrow=2, ncol=up.dd1)
      for (ireps1 in 1:2) {
        for (dd1 in 1:up.dd1) {
          w1.1 <- post.wgt(priwgt1[,ireps1],
                           prior.p[,dd1])
          w0.1 <- post.wgt(priwgt1[,ireps1],
                           1-prior.p[,dd1])
          priwgt2 <- cbind(w1.1[[1]], w0.1[[1]])
          up.dd2 <- min(dd1+1, 8)
          eloss2 <- matrix(NA, nrow=2, ncol=up.dd2)
          for (ireps2 in 1:2) {
            for (dd2 in 1:up.dd2) {
              w1.2 <- post.wgt(priwgt2[,ireps2],
                               prior.p[,dd2])
              w0.2 <- post.wgt(priwgt2[,ireps2],
                               1-prior.p[,dd2])
              eloss.pre <- Eloss(w1.2[[1]], w0.2[[1]],
                   prior.sample[,1], prior.sample[,2],
                   w1.2[[2]], w0.2[[2]])[1:3]
              eloss2[ireps2, dd2] <- eloss.pre[1]
              avg.p012[,dd2,ireps2,dd1,
                        ireps1,dd0,ireps0,bb] <-
                            c(w1.2[[2]], w0.2[[2]])
              loss012[,dd2,ireps2,dd1,
                       ireps1,dd0,ireps0,bb] <-
```

```
                            eloss.pre[2:3]
            }
          }
          minELid.i2 <- apply(eloss2, 1, which.min)
          dose012[,dd1,ireps1,dd0,ireps0,bb]<-minELid.i2
          minEL.i2 <- c(eloss2[1, minELid.i2[1]],
                        eloss2[2, minELid.i2[2]])
          eloss1[ireps1, dd1] <- w1.1[[2]]*minEL.i2[1] +
                                 w0.1[[2]]*minEL.i2[2]
          avg.p01[,dd1,ireps1,dd0,ireps0,bb] <-
                                 c(w1.1[[2]], w0.1[[2]])
        }
      }
      minELid.i1 <- apply(eloss1, 1, which.min)
      dose01[,dd0, ireps0, bb] <- minELid.i1
      minEL.i1 <- c(eloss1[1, minELid.i1[1]],
                    eloss1[2, minELid.i1[2]])
      eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.i1[1] +
                             w0.0[[2]]*minEL.i1[2]
      avg.p0[,dd0,ireps0,bb] <- c(w1.0[[2]], w0.0[[2]])
    }
  }
  minELid.i0 <- apply(eloss0, 1, which.min)
  dose0[,bb] <- minELid.i0
  minEL.i0 <- c(eloss0[1, minELid.i0[1]],
                eloss0[2, minELid.i0[2]])
}
# optimal dose for stage i, i+1 and i+2
# stage i
# doseid.0 <- rep(NA, 2^(i-1))
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
# stage (i+1)
doseid.1 <- rep(NA, 2^i)
d1.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx <- doseid.0[2*(bb-1)+m0]
    temp1 <- dose01[, idx, m0, bb]
    d1.pre[(2*(bb-1)+m0),] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
```

```
# stage (i+2)
doseid.2 <- rep(NA, 2^(i+1))
d2.pre <- matrix(NA, nrow=2^i, ncol=2)
for (bb in 1:n.bb){
  for (m0 in 1:2) {
    idx0.i <- 2*(bb-1)+m0
    idx0 <- doseid.0[idx0.i]
    for (m1 in 1:2) {
      idx1 <- doseid.1[2*(idx0.i-1)+m1]
      temp1 <- dose012[,idx1,m1,idx0,m0,bb]
      d2.pre[(2*(idx0.i-1)+m1),] <- temp1
    }
  }
}
doseid.2 <- as.vector(t(d2.pre))
OPDOSE[[i+2]] <- doses[doseid.2]
# average prob for stage i, i+1 and i+2
# stage i
p0.pre <- matrix(NA, nrow=2^(i-1), ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0 <- doseid.0[2*(bb-1)+m0]
    temp1 <- avg.p0[,idx0,m0,bb]
    p0.pre[2*(bb-1)+m0,] <- temp1
  }
}
P.OPT[[i]] <- as.vector(t(p0.pre))
# stage (i+1)
p1.pre <- matrix(NA, nrow=2^i, ncol=2)
for (bb in 1:n.bb) {
  for (m0 in 1:2) {
    idx0.i <- 2*(bb-1)+m0
    idx0 <- doseid.0[idx0.i]
    for (m1 in 1:2) {
      idx1 <- doseid.1[2*(idx0.i-1)+m1]
      temp1 <- avg.p01[,idx1,m1,idx0,m0,bb]
      p1.pre[(2*(idx0.i-1)+m1),] <- temp1
    }
  }
}
P.OPT[[i+1]] <- as.vector(t(p1.pre))
# stage (i+2)
p2.pre <- matrix(NA, nrow=2^(i+1), ncol=2)
loss.pre <- matrix(NA, nrow=2^(i+1), ncol=2)
```

```
      for (bb in 1:n.bb) {
        for (m0 in 1:2) {
          idx0.i <- 2*(bb-1)+m0
          idx0 <- doseid.0[idx0.i]
          for (m1 in 1:2) {
            idx1.i <- 2*(idx0.i-1)+m1
            idx1 <- doseid.1[idx1.i]
            for (m2 in 1:2) {
              idx2 <- doseid.2[2*(idx1.i-1)+m2]
              temp1 <- avg.p012[,idx2,m2,idx1,m1,idx0,m0,bb]
              loss.temp1<-loss012[,idx2,m2,idx1,m1,idx0,m0,bb]
              p2.pre[  (2*(idx1.i-1)+m2), ] <- temp1
              loss.pre[(2*(idx1.i-1)+m2), ] <- loss.temp1
            }
          }
        }
      }
      P.OPT[[i+2]] <- as.vector(t(p2.pre))
      loss.last <- as.vector(t(loss.pre))
    }
  }
  prob.matrix <- matrix(NA, nrow=2^nsubj, ncol=nsubj)
  for (ncolumn in 1:nsubj) {
    p <- rep(P.OPT[[ncolumn]], each=2^(nsubj-ncolumn))
    prob.matrix[,ncolumn] <- p
  }
  prob.expt <- apply(prob.matrix, 1, prod)
  OEL.3moving <- sum(prob.expt*loss.last)
  rstt <- matrix(c(nsubj, OEL.3moving), ncol=2, nrow=1)
  return(rstt)
}
```

B.2 R Functions for Myopic Procedures: Bivariate Model

In this section, R functions for myopic procedures used in Chapter 7 are listed.

```
#************************************************************#
#    Functions for myopic procedure:
#    (1) sim.prior: simulate the random sample from
#                   prior distribution
#    (2) prior.prob: compute the prior probability
#    (3) post.wgt: compute the posterior
#    (4) Eloss: compute the expected loss
#************************************************************ #
# -- (1) Function :---------------------------------------- #
# --    Simulate the random sample from prior distribution -- #
# sim.prior(n, constz, a, b, e, f)
# n = number of sample
# constz = constant
# a, b = parameters for beta distribution
#        (for v1, v2, v3: related to the lower HCD, DANC, DAC)
# e, f = parameters for beta distribution
#        (for z1, z2, z3: related to the difference, delta)
# VALUE
#   sim.prior gives n by 2 matrix, the 1st column is mu
#   and 2nd column is gamma
#------------------------------------------------------------ #
sim.prior <-
 function(nsample,constz,va,vb,za,zb,seed1,seed2,seed3) {
  set.seed(seed1)
  v1 <- rbeta(nsample, va, vb)
  z1 <- rbeta(nsample, za, zb)
  set.seed(seed2)
  v2 <- rbeta(nsample, va, vb)
  z2 <- rbeta(nsample, za, zb)
  set.seed(seed3)
  v3 <- rbeta(nsample, va, vb)
  z3 <- rbeta(nsample, za, zb)
  mu1.k <- 1.5 + 3.5*v1
  ga1.k <- (constz*z1 + 0.1) + mu1.k
  mu2.k <- 1.5 + 3.5*v2
  ga2.k <- (constz*z2 + 0.1) + mu2.k
  mu3.k <- 1.5 + 3.5*v3
  ga3.k <- (constz*z3 + 0.1) + mu3.k
```

```
  return(cbind(mu1.k, ga1.k, mu2.k, ga2.k, mu3.k, ga3.k))
}
# -------- (2) Function: calculate prior probability -------- #
# prior.prob(p.low, p.high, dose, sim.mu1, sim.ga1,
#            sim.mu2, sim.ga2, sim.mu3, sim.ga3)
# p.low / p.high = prob of subject get infected
#                  (HCDa/DANCa/DACa, HCDb/DANCb/DACb)
# dose = dose
# sim.mu1 - sim.mu3 / sim.ga1 - sim.ga3 =
#    mu1 - mu2 / gamma1 - gamma3 simulated from the sim.prior
# VALUE
#   prior.prob1: gives the inidividual probabiliy of A=0, C=0
#   prior.prob2: gives the inidividual probabiliy of A=1, C=0
#   prior.prob3: gives the inidividual probabiliy of A=0, C=1
#------------------------------------------------------------ #
prior.prob <-
 function(p.low,p.high,dose,sim.mu1,sim.ga1,
          sim.mu2,sim.ga2, im.mu3,sim.ga3) {
  odds.low <- p.low/(1-p.low)
  odds.high <- p.high/(1-p.high)
  OR <- odds.high / odds.low
  # P(C=1)
  M5 <- (dose - sim.mu1) / (sim.ga1 - sim.mu1)
  numer5 <- odds.low * (OR^M5)
  p5 <- numer5/ (1+numer5)
  # P(A=0,C=0) and P(A=1, C=0)
  MC0 <- (dose - sim.mu2) / (sim.ga2 - sim.mu2)
  numerC0 <- odds.low * (OR^MC0)
  p1 <- (      1/ (1+numerC0)) * (1 - p5)
  p2 <- (numerC0/ (1+numerC0)) * (1 - p5)
  # P(A=0,C=1) and P(A=1, C=1)
  MC1 <- (dose - sim.mu3) / (sim.ga3 - sim.mu3)
  numerC1 <- odds.low * (OR^MC1)
  p3 <- (     1/ (1+numerC1)) * (p5)
  return(cbind(p1, p2, p3))
}
# -------- (3) Function: calculate posterior weights -------- #
# post.wgt(prior.weight, ind.prob)
# prior.weight = prior weights for (A=0,C=0), (A=1,C=0),
#                (A=0,C=1) or (A=1,C=1)
# ind.prob = individual prob for (A=0,C=0), (A=1,C=0),
#            (A=0,C=1) or (A=1,C=1) (from prior.prob)
# VALUE
#   post.wgt calcuate the updated weight given prior weight,
```

```
#   and average probability of success
#------------------------------------------------------------ #
post.wgt <- function(prior.weight, ind.prob) {
  n <- length(prior.weight)
  numer <- prior.weight*ind.prob
  denom <- sum(numer)
  post.w <- (numer/denom) * n
  avg.p <- denom / sum(prior.weight)
  return(list(post.w, avg.p))
}
# -------- (4) Function: calculate the expected loss -------- #
# Eloss(w1, w0, m, g, p.success, p.failure)
# w1 / w0 = posterior weights (vector)
# m = prior mu (vector)
# g = prior gamma (vector)
# p1 - p4 = average prob of outcome
# VALUE
#   Eloss calculate the expected loss (exp.loss)
#   loss (loss1 - loss4)
#------------------------------------------------------------ #
Eloss <- function(w1, w2, w3, w4, m1, g1, p1, p2, p3, p4){
  # y=1 (A=0, C=0)
  post1.Emu1 <- sum(w1 * m1)/sum(w1)
  post1.Ega1 <- sum(w1 * g1)/sum(w1)
  post1.Vmu1 <- sum(w1 * ((m1 - post1.Emu1)^2))/sum(w1)
  post1.Vga1 <- sum(w1 * ((g1 - post1.Ega1)^2))/sum(w1)
  Loss1 <- post1.Vmu1 + post1.Vga1
  # y =2 (A=1, C=0)
  post2.Emu1 <- sum(w2 * m1)/sum(w2)
  post2.Ega1 <- sum(w2 * g1)/sum(w2)
  post2.Vmu1 <- sum(w2 * ((m1 - post2.Emu1)^2))/sum(w2)
  post2.Vga1 <- sum(w2 * ((g1 - post2.Ega1)^2))/sum(w2)
  Loss2 <- post2.Vmu1 + post2.Vga1
  # y =3 (A=0, C=1)
  post3.Emu1 <- sum(w3 * m1)/sum(w3)
  post3.Ega1 <- sum(w3 * g1)/sum(w3)
  post3.Vmu1 <- sum(w3 * ((m1 - post3.Emu1)^2))/sum(w3)
  post3.Vga1 <- sum(w3 * ((g1 - post3.Ega1)^2))/sum(w3)
  Loss3 <- post3.Vmu1 + post3.Vga1
  # y =4 (A=1, C=1)
  post4.Emu1 <- sum(w4 * m1)/sum(w4)
  post4.Ega1 <- sum(w4 * g1)/sum(w4)
  post4.Vmu1 <- sum(w4 * ((m1 - post4.Emu1)^2))/sum(w4)
  post4.Vga1 <- sum(w4 * ((g1 - post4.Ega1)^2))/sum(w4)
```

```
  Loss4 <- post4.Vmu1 + post4.Vga1
  exp.loss <- p1*Loss1 + p2*Loss2 + p3*Loss3 + p4*Loss4
  return(c(exp.loss, Loss1, Loss2, Loss3, Loss4))
}
```

B.2.1 One-step-look-ahead Procedure

```
# ************************************************************#
#  One-step-look-ahead procedure
#  NEW MODEL (efficacy and toxicity)
#  onestep.AE1(nsubj, AE.threshold)
#      nsubj: number of subjects
#      AE.threshold: the upper bound of posterior probability
#                    of toxicity
# VALUE:
#    This function returns a list:
#      [[1]]: c(nsubj, expected loss, E(nsubj)
#      [[2]]: optimal doses for each stage
# ************************************************************#
onestep.AE1 <- function(nsubj, AE.threshold) {
  rstt <- matrix(NA, nrow=1, ncol=3)
  prob.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
  loss.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
  sub.matrix <- matrix(NA, nrow=4^(nsubj-1), ncol=nsubj)
  # ----------------------- the first dose ---------------- #
  OPDOSE <- P.OPT <- Nsub <- STOP.IDX <- LOSS <- list(NULL)
  eloss1 <- matrix(NA, nrow=4^(1-1), ncol=ndose)
  up.dd1 <- ndose
  # probabiliy of adverse event
  p.ae <- rep(NA, up.dd1)
  for (dd1 in 1:up.dd1){
    # first stage #
    w1.0 <- post.wgt(w.init, prior.p1[ ,dd1])
    w2.0 <- post.wgt(w.init, prior.p2[ ,dd1])
    w3.0 <- post.wgt(w.init, prior.p3[ ,dd1])
    w4.0 <- post.wgt(w.init,
         (1-prior.p1[ ,dd1]-prior.p2[ ,dd1]-prior.p3[ ,dd1]))
    eloss1[ ,dd1] <-
            Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
              prior.sample[,1], prior.sample[,2],
              w1.0[[2]], w2.0[[2]], w3.0[[2]], w4.0[[2]])[1]
      p.ae[dd1] <- w2.0[[2]] + w4.0[[2]]
  }
  # restriction on probability of AE
  doseid.1a <- which(p.ae <= AE.threshold)
```

```
# optimal dose for stage 1
doseid.1 <- which.min(eloss1[,doseid.1a])
OPDOSE[[1]] <- doses[doseid.1]
# compute posterior weigths given the first optimal dose
w1.0 <- post.wgt(w.init, prior.p1[ ,doseid.1])
w2.0 <- post.wgt(w.init, prior.p2[ ,doseid.1])
w3.0 <- post.wgt(w.init, prior.p3[ ,doseid.1])
w4.0 <- post.wgt(w.init, (1-prior.p1[ ,doseid.1]-
                prior.p2[ ,doseid.1]-prior.p3[ ,doseid.1]))
wgt.temp <- cbind(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]])
# average prob for stage 1
P.OPT[[1]] <- cbind(w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])
Nsub[[1]] <- 1
STOP.IDX[[1]] <- 0
LOSS[[1]] <- Eloss(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]],
              prior.sample[ ,1],prior.sample[ ,2],
              w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
# -------------------- the i^{th} dose (i>1) ------------- #
for (i in 2:nsubj) {
  priwgt0 <- wgt.temp
  # the largest dose can be used in the i stage. dimension:
  # number of dose at (i-1) stage
  up.dd0 <- rep(NA, 4^(i-2))
  # index for optimal dose(s) selected from stage (i-1)
  opdoseidx <- 2*OPDOSE[[i-1]] - 2
  # dose with min expected loss for each outcome for stage i
  dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  # record number of subject up to stage i
  sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  # record if the trial stops or not
  stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  minELid.0 <- rep(NA, 4)
  n.bb <- 4^(i-2)
  for (bb in 1:n.bb) {
    if (is.na(opdoseidx[bb]) == FALSE){
      up.dd0 <- min(opdoseidx[bb]+1, 8)
      eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
      p.ae <- rep(NA, up.dd0)
      # ireps0 is # of outcomes
      for (ireps0 in 1:4) {
        for (dd0 in 1:up.dd0) {
          w1.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                           prior.p1[ ,dd0])
          w2.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
```

```
                        prior.p2[ ,dd0])
          w3.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                        prior.p3[ ,dd0])
          w4.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                      (1-prior.p1[ ,dd0]-prior.p2[ ,dd0]-
                       prior.p3[ ,dd0]))
          eloss0[ireps0, dd0] <-
             Eloss(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]],
               prior.sample[ ,1], prior.sample[ ,2],
               w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[1]
          p.ae[dd0] <- w2.0[[2]] + w4.0[[2]]
        }
        # restriction on probability of AE
        doseid.0a <- which(p.ae <= AE.threshold)
        # if P(AE) <= AE.threshold)
        if (length(doseid.0a) != 0) {
          # choose the one with minimun eloss
          temp <- which.min(eloss0[ireps0, doseid.0a])
          if (length(temp) != 0) {
            minELid.0[ireps0] <- temp
            sub0[ireps0, bb] <- i
            stop0[ireps0, bb] <- 0
          }
          if (length(temp) == 0) {
            minELid.0[ireps0] <- NA
            sub0[ireps0, bb] <- i-1
            stop0[ireps0, bb] <- 1
          }
        }
        # if P(AE) > AE.threshold, the trial
        # stops at this stage
        if (length(doseid.0a) == 0) {
          minELid.0[ireps0] <- NA
          sub0[ireps0, bb] <- i-1
          stop0[ireps0, bb] <- 1
        }
      }
      dose0[ ,bb] <- minELid.0
    }
    if (is.na(opdoseidx[bb]) == TRUE) {
      dose0[ ,bb] <- rep(NA, 4)
    }
  }
  # optimal dose for stage i
```

```
  # stage i
  doseid.0 <- as.vector(dose0)
  OPDOSE[[i]] <- doses[doseid.0]
  Nsub[[i]] <- as.vector(sub0)
  STOP.IDX[[i]] <- as.vector(stop0)
  # obtain the posterior weights
  wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
  p.temp <- loss.all <- rep(NA, 4^i)
  # stage i
  for (mm in 1:(4^(i-1))) {
    idx <- doseid.0[mm]
    w1.0 <- post.wgt(priwgt0[,mm], prior.p1[,idx])
    w2.0 <- post.wgt(priwgt0[,mm], prior.p2[,idx])
    w3.0 <- post.wgt(priwgt0[,mm], prior.p3[,idx])
    w4.0 <- post.wgt(priwgt0[,mm], (1-prior.p1[,idx]-
                    prior.p2[,idx]-prior.p3[,idx]))
    p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <- c(w1.0[[2]],
                      w2.0[[2]], w3.0[[2]], w4.0[[2]])
    # loss at stage i (for all paths)
    loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
        Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
              prior.sample[ ,1], prior.sample[ ,2],
              w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
    if (i < nsubj)   {
      wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
           cbind(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]])
    }
    if (i == nsubj) wgt.temp <- NA
  }
  P.OPT[[i]] <- p.temp
  LOSS[[i]] <- loss.all
}
rm(w1.0, w2.0, w3.0, w4.0, wgt.temp, p.temp, priwgt0)
gc(reset=TRUE)
for (ncolumn in 1:nsubj) {
  p <- rep(P.OPT[[ncolumn]], each=4^(nsubj-ncolumn))
  prob.matrix[ ,ncolumn] <- p
  p <- rep(LOSS[[ncolumn]], each=4^(nsubj-ncolumn))
  loss.matrix[ ,ncolumn] <- p
  sub.matrix[ ,ncolumn] <-
               rep(Nsub[[ncolumn]], each=4^(nsubj-ncolumn))
}
# compute average number of subjects
# loss and probability for each outcome for each path
```

```
  sub.final <- sub.matrix[,nsubj]
  total.scenario <- length(sub.final)  # should be 4^(nsubj-1)
  loss.final <- loss.matrix[,nsubj]
  total.lossscenario <- length(loss.final) # hould be 4^nsubj
  for (jj in 2:nsubj) {
    s.temp <- which(STOP.IDX[[jj]] == 1)
    for (ss in s.temp) {
      s.idx <- (total.scenario / (4^(jj-1))) * (ss-1) + 1
      sub.final[s.idx] <- sub.matrix[s.idx, jj]
      l.idx <- (total.lossscenario / (4^(jj-1))) * (ss-1) + 1
      loss.final[l.idx] <- loss.matrix[l.idx, jj-1]
    }
  }
  Esubj <- mean(sub.final, na.rm=TRUE)
  rm(P.OPT, sub.matrix, STOP.IDX, sub.final)
  gc(reset=TRUE)
  # calculate overall expected loss
  idx <- 1-is.na(loss.final)
  prob.expt <- rep(NA, 4^nsubj)
  for (k in 1:4^nsubj) {
    if (idx[k] ==1) {
      prob.expt[k] <- prod(prob.matrix[k,], na.rm=T)
    }
  }
  OEL.onestep <- sum(prob.expt*loss.final, na.rm=T)
  rstt[1, ] <- c(nsubj, OEL.onestep, Esubj)
  return(list(OEL=rstt, optimal.dose=OPDOSE))
}
```

B.2.2 Two-step-look-ahead Procedure: Batch and Moving Window

```
# ***********************************************************#
#  Two-step-look-ahead procedure  (BATCH)
#  NEW MODEL (efficacy and toxicity)
#  twostep.batchAE1(nsubj, AE.threshold)
#      nsubj: number of subjects
#      AE.threshold: the upper bound of posterior probability
#                    of toxicity
# VALUE:
#    This function returns a list:
#      [[1]]: c(nsubj, expected loss, E(nsubj)
#      [[2]]: optimal doses for each stage
# ***********************************************************#
twostep.batchAE1 <- function(nsubj, AE.threshold) {
```

```
# check if nsubj is odd or even number
if ((nsubj %% 2) == 0) {
  odd <- 0; nsubj.i <- seq(from=1, to=nsubj-1, by=2)
}
if ((nsubj %% 2) == 1) {
  odd <- 1; nsubj.i <- seq(from=1, to=nsubj,   by=2)
}
rstt <- matrix(NA, nrow=1, ncol=3)
prob.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
loss.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
sub.matrix <- matrix(NA, nrow=4^(nsubj-1), ncol=nsubj)
# ------------------------ the first dose ---------------- #
OPDOSE <- P.OPT <- Nsub <- STOP.IDX <- LOSS <- list(NULL)
eloss1 <- matrix(NA, nrow=4^(1-1), ncol=ndose)
up.dd1 <- ndose
# indicator for stop at stage 1
stop1 <- rep(NA, up.dd1)
# number of subject after stage 1
sub1 <- rep(NA, up.dd1)
# dose12[y_1, dd1]
dose12 <- matrix(NA, nrow=4, ncol=up.dd1)
p.ae1 <- rep(NA, up.dd1)
minELid.2 <- minEL.2 <- rep(NA, 4)
sub2 <- matrix(NA, nrow=4, ncol=up.dd1)
stop2 <- matrix(NA, nrow=4, ncol=up.dd1)
for (dd1 in 1:up.dd1){
  # first stage #
  # calculate the posterior weights
  w1.1 <- post.wgt(w.init, prior.p1[,dd1])
  w2.1 <- post.wgt(w.init, prior.p2[,dd1])
  w3.1 <- post.wgt(w.init, prior.p3[,dd1])
  w4.1 <- post.wgt(w.init, (1-prior.p1[,dd1]-
                            prior.p2[,dd1]-prior.p3[,dd1]))
  priwgt2 <- cbind(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]])
  p.ae1[dd1] <- w2.1[[2]] + w4.1[[2]]
  if (p.ae1[dd1] <= AE.threshold) {
    # second stage #
    up.dd2 <- min(dd1+1, 8)
    eloss2 <- matrix(NA, nrow=4^(2-1), ncol=up.dd2)
    # probability of adverse event at second stage
    p.ae12 <- rep(NA, up.dd2)
    for (ireps2 in 1:4){
      for (dd2 in 1:up.dd2) {
        # update the posterior weights for
```

```
    # y=(1,1), (1,0), (0,1), (0,0)
    # and calculate the expected loss for y1=1 and y1=0
    w1.2 <- post.wgt(priwgt2[,ireps2], prior.p1[,dd2])
    w2.2 <- post.wgt(priwgt2[,ireps2], prior.p2[,dd2])
    w3.2 <- post.wgt(priwgt2[,ireps2], prior.p3[,dd2])
    w4.2 <- post.wgt(priwgt2[,ireps2],(1-prior.p1[,dd2]-
                    prior.p2[,dd2]-prior.p3[,dd2]))
    p.ae12[dd2] <- w2.2[[2]] + w4.2[[2]]
    if (p.ae12[dd2] <= AE.threshold) {
      eloss2[ireps2,dd2] <- Eloss(w1.2[[1]], w2.2[[1]],
                  w3.2[[1]],w4.2[[1]],prior.sample[,1],
                  prior.sample[,2],w1.2[[2]],w2.2[[2]],
                  w3.2[[2]], w4.2[[2]])[1]
    }
    if (p.ae12[dd2] > AE.threshold) {
      eloss2[ireps2, dd2] <- NA
    }
  }
  temp <- which.min(eloss2[ireps2, ])
  if (length(temp) != 0) {
    minELid.2[ireps2] <- temp
    dose12[ireps2, dd1] <- minELid.2[ireps2]
    sub2[ireps2, dd1] <- 2
    stop2[ireps2, dd1] <- 0
    minEL.2[ireps2] <- eloss2[ireps2, minELid.2[ireps2]]
  }
  if (length(temp) == 0) {
    minELid.2[ireps2] <- NA
    dose12[ireps2, dd1] <- NA
    sub2[ireps2, dd1] <- 1
    stop2[ireps2, dd1] <- 1
    minEL.2[ireps2] <- NA
  }
}
eloss1[dd1]<-w1.1[[2]]*minEL.2[1]+w2.1[[2]]*minEL.2[2]+
            w3.1[[2]]*minEL.2[3] + w4.1[[2]]*minEL.2[4]
# when p.ae2 > AE.threshold, p.ae1 < AE.threshold,
# perform only one step dose selection for the 1st stage
if (is.na(eloss1[dd1]) == TRUE) {
  eloss1[1, dd1] <-
       Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[1]
}
```

```
    stop1[dd1] <- 0
    sub1[dd1] <- 1
  }
   if (p.ae1[dd1] > AE.threshold) {
     eloss1[dd1] <- NA
     stop1[dd1] <- 1
     sub1[dd1] <- 0
   }
}
# optimal dose for stage 1 and 2
# stage 1
temp1 <- which.min(eloss1)
if (length(temp1) != 0) {
  doseid.1 <- temp1
  Nsub[[1]] <- 1
}
if (length(temp1) == 0) {
  doseid.1 <- NA
  Nsub[[1]] <- 0
}
OPDOSE[[1]] <- doses[doseid.1]
STOP.IDX[[1]] <- stop1[doseid.1]
# stage 2
doseid.2 <- dose12[ ,doseid.1]
OPDOSE[[2]] <- doses[doseid.2]
Nsub[[2]] <- sub2[ ,doseid.1]
STOP.IDX[[2]] <- stop2[, doseid.1]
# obtain posterior weigths given the first and second dose
# first dose
w1.1 <- post.wgt(w.init, prior.p1[ ,doseid.1])
w2.1 <- post.wgt(w.init, prior.p2[ ,doseid.1])
w3.1 <- post.wgt(w.init, prior.p3[ ,doseid.1])
w4.1 <- post.wgt(w.init, (1-prior.p1[ ,doseid.1]-
                prior.p2[ ,doseid.1]-prior.p3[ ,doseid.1]))
priwgt1 <- cbind(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]])
P.OPT[[1]] <- c(w1.1[[2]], w2.1[[2]], w3.1[[2]], w4.1[[2]])
LOSS[[1]] <- Eloss(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]],
            prior.sample[,1], prior.sample[,2],
            w1.1[[2]], w2.1[[2]], w3.1[[2]], w4.1[[2]])[2:5]
# second dose
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^2)
p.temp <- loss.all <- rep(NA, 4^2)
for (mm in 1:(4^(2-1))) {
  idx <- doseid.2[mm]
```

```
  w1.2 <- post.wgt(priwgt1[ ,mm], prior.p1[ ,idx])
  w2.2 <- post.wgt(priwgt1[ ,mm], prior.p2[ ,idx])
  w3.2 <- post.wgt(priwgt1[ ,mm], prior.p3[ ,idx])
  w4.2 <- post.wgt(priwgt1[ ,mm], (1-prior.p1[ ,idx]-
                            prior.p2[ ,idx]-prior.p3[ ,idx]))
  wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
           cbind(w1.2[[1]], w2.2[[1]], w3.2[[1]], w4.2[[1]])
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)]<-c(w1.2[[2]],w2.2[[2]],
                                       w3.2[[2]], w4.2[[2]])
  loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
      Eloss(w1.2[[1]], w2.2[[1]], w3.2[[1]], w4.2[[1]],
            prior.sample[,1], prior.sample[,2],
            w1.2[[2]], w2.2[[2]], w3.2[[2]], w4.2[[2]])[2:5]
}
P.OPT[[2]] <- p.temp
LOSS[[2]] <- loss.all
rm(priwgt1, priwgt2, w1.1, w2.1, w3.1, w4.1, w1.2, w2.2,
   w3.2, w4.2, p.temp, stop1, sub1, stop2, sub2, loss.all)
gc(reset=TRUE)
# -------------------- the i^{th} dose (i>1) ------------- #
if (odd == 0) {
  for (i in nsubj.i[-1]) {
    priwgt0 <- wgt.temp
    opdoseidx <- 2*OPDOSE[[i-1]]-2
    # dose0[ireps0,bb]
    dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    # record number of subject up to stage i
    sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    # record if the trail stops or not
    stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    minELid.0 <- rep(NA, 4)
    # dose01[y_i, dd0, ireps0, bb]
    dose01 <- array(NA, c(4,8,4,4^(i-2)))
    # sub01[y_i, dd0, ireps0, bb]
    sub01 <- array(NA, c(4,8,4,4^(i-2)))
    # stop01[y_i, dd0, ireps0, bb]
    stop01 <- array(NA, c(4,8,4,4^(i-2)))
    minELid.1 <- minEL.1 <- rep(NA, 4)
    n.bb <- 4^(i-2)
    # stage i
    for (bb in 1:n.bb) {
      if (is.na(opdoseidx[bb]) == FALSE) {
        up.dd0 <- min(opdoseidx[bb]+1, 8)
        eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
```

```
for (ireps0 in 1:4) {
  for (dd0 in 1:up.dd0) {
    w1.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                     prior.p1[ ,dd0])
    w2.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                     prior.p2[ ,dd0])
    w3.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                     prior.p3[ ,dd0])
    w4.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                     (1-prior.p1[ ,dd0]-
                prior.p2[ ,dd0]-prior.p3[ ,dd0]))
    priwgt1 <- cbind(w1.0[[1]], w2.0[[1]],
                     w3.0[[1]], w4.0[[1]])
    p.ae0<- w2.0[[2]] + w4.0[[2]]
    up.dd1 <- min(dd0+1, 8)
    eloss1 <- matrix(NA, nrow=4, ncol=up.dd1)
    p.ae01 <- rep(NA, up.dd1)
    if (p.ae0 <= AE.threshold) {
      # stage (i+1)
      for (ireps1 in 1:4) {
        for (dd1 in 1:up.dd1) {
          w1.1 <- post.wgt(priwgt1[ ,ireps1],
                           prior.p1[,dd1])
          w2.1 <- post.wgt(priwgt1[ ,ireps1],
                           prior.p2[,dd1])
          w3.1 <- post.wgt(priwgt1[ ,ireps1],
                           prior.p3[,dd1])
          w4.1 <- post.wgt(priwgt1[ ,ireps1],
                    (1-prior.p1[,dd1]-
                   prior.p2[,dd1]-prior.p3[,dd1]))
          p.ae01[dd1] <- w2.1[[2]] + w4.1[[2]]
          if (p.ae01[dd1] <= AE.threshold) {
            eloss1[ireps1, dd1] <-
              Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]],
               w4.1[[1]], prior.sample[ ,1],
               prior.sample[ ,2], w1.1[[2]],
               w2.1[[2]], w3.1[[2]], w4.1[[2]])[1]
          }
          if (p.ae01[dd1] > AE.threshold) {
            eloss1[ireps1, dd1] <- NA
          }
        }
        # restriction on P(AE) at stage (i+1)
        temp <- which.min(eloss1[ireps1, ])
```

```
            if (length(temp) != 0) {
              minELid.1[ireps1] <- temp
              dose01[ireps1,dd0,ireps0,bb] <-
                                  minELid.1[ireps1]
              sub01[ireps1,dd0,ireps0,bb] <- i+1
              stop01[ireps1,dd0,ireps0,bb] <- 0
            }
            if (length(temp) == 0) {
              minELid.1[ireps1] <- NA
              dose01[ireps1,dd0,ireps0,bb] <- NA
              sub01[ireps1,dd0,ireps0,bb] <- i
              stop01[ireps1,dd0,ireps0,bb] <- 1
            }
            minEL.1[ireps1] <-
                eloss1[ireps1, minELid.1[ireps1]]
        }
        eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.1[1] +
          w2.0[[2]]*minEL.1[2] + w3.0[[2]]*minEL.1[3]+
          w4.0[[2]]*minEL.1[4]
        if (is.na(eloss0[ireps0,dd0]) == TRUE) {
          eloss0[ireps0, dd0] <-
            Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]],
            w4.0[[1]],prior.sample[ ,1],
            prior.sample[ ,2], w1.0[[2]], w2.0[[2]],
            w3.0[[2]], w4.0[[2]])[1]
        }
      }
      if (p.ae0 > AE.threshold) {
        eloss0[ireps0, dd0] <-  NA
      }
    }
    # P(AE) restriction at stage i
    temp <- which.min(eloss0[ireps0, ])
    if (length(temp) != 0) {
      minELid.0[ireps0] <- temp
      sub0[ireps0, bb] <- i
      stop0[ireps0, bb] <- 0
    }
    if (length(temp) == 0) {
      minELid.0[ireps0] <- NA
      sub0[ireps0, bb] <- i-1
      stop0[ireps0, bb] <- 1
    }
  }
```

```
    dose0[ ,bb] <- minELid.0
  }
  if (is.na(opdoseidx[bb]) == TRUE) {
      dose0[ ,bb] <- rep(NA, 4) }
} ## bb
# optimal dose for stage i and i+1
# stage i
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
STOP.IDX[[i]] <- as.vector(stop0)
Nsub[[i]] <- as.vector(sub0)
# stage (i+1)
doseid.1 <- rep(NA, 4^i)
d1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
stop1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
sub1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
for (bb in 1:n.bb) {
  for (m0 in 1:4) {
    idx0 <- doseid.0[4*(bb-1)+m0]
    temp1 <- dose01[ ,idx0,m0,bb]
    d1.pre[(4*(bb-1)+m0), ] <- temp1
    temp1 <- stop01[ ,idx0,m0,bb]
    stop1.pre[(4*(bb-1)+m0), ] <- temp1
    temp1 <- sub01[ ,idx0,m0,bb]
    sub1.pre[(4*(bb-1)+m0), ] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
STOP.IDX[[i+1]] <- as.vector(t(stop1.pre))
Nsub[[i+1]] <- as.vector(t(sub1.pre))
# obtain the posterior weights for (i+1)th stage
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
p.temp <- loss.all <- rep(NA, 4^i)
# stage i
for (mm in 1:(4^(i-1))) {
  idx0 <- doseid.0[mm]
  w1.0 <- post.wgt(priwgt0[ ,mm], prior.p1[ ,idx0])
  w2.0 <- post.wgt(priwgt0[ ,mm], prior.p2[ ,idx0])
  w3.0 <- post.wgt(priwgt0[ ,mm], prior.p3[ ,idx0])
  w4.0 <- post.wgt(priwgt0[ ,mm], (1-prior.p1[ ,idx0]-
                  prior.p2[ ,idx0]-prior.p3[ ,idx0]))
  wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
       cbind(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]])
```

```
    p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <- c(w1.0[[2]],
                          w2.0[[2]], w3.0[[2]], w4.0[[2]])
    loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
        Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
  }
  P.OPT[[i]] <- p.temp
  LOSS[[i]] <- loss.all
  # stage i+1
  priwgt1 <- wgt.temp
  wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^(i+1))
  p.temp <- loss.all <- rep(NA, 4^(i+1))
  for (mm in 1:(4^i)) {
    idx1 <- doseid.1[mm]
    w1.1 <- post.wgt(priwgt1[ ,mm], prior.p1[ ,idx1])
    w2.1 <- post.wgt(priwgt1[ ,mm], prior.p2[ ,idx1])
    w3.1 <- post.wgt(priwgt1[ ,mm], prior.p3[ ,idx1])
    w4.1 <- post.wgt(priwgt1[ ,mm], (1-prior.p1[ ,idx1]-
                     prior.p2[ ,idx1]-prior.p3[ ,idx1]))
    p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <- c(w1.1[[2]],
                        w2.1[[2]], w3.1[[2]], w4.1[[2]])
    loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
        Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]],
              prior.sample[ ,1], prior.sample[ ,2],
            w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[2:5]
    if (i < (nsubj-1)) {
      wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
        cbind(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]])
    }
    if (i == (nsubj-1)) wgt.temp <- NA
  }
  P.OPT[[i+1]] <- p.temp
  LOSS[[i+1]] <- loss.all
}
rm(priwgt0, priwgt1, w1.0, w2.0, w3.0, w4.0, w1.1, w2.1,
   w3.1, w4.1, dose0, dose01, wgt.temp, p.temp, loss.all,
   sub0, sub01, stop0, stop01)
gc(reset=TRUE)
}
if (odd == 1) {
  for (i in nsubj.i[-1]) {
    if (i < nsubj) {
      priwgt0 <- wgt.temp
```

```
opdoseidx <- 2*OPDOSE[[i-1]]-2
dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
minELid.0 <- rep(NA, 4)
# dose01[y_i, dd0, ireps0, bb]
dose01 <- array(NA, c(4,8,4,4^(i-2)))
# sub01[y_i, dd0, ireps0, bb]
sub01 <- array(NA, c(4,8,4,4^(i-2)))
# stop01[y_i, dd0, ireps0, bb]
stop01 <- array(NA, c(4,8,4,4^(i-2)))
minELid.1 <- minEL.1 <- rep(NA, 4)
n.bb <- 4^(i-2)
# stage i
for (bb in 1:n.bb) {
  if (is.na(opdoseidx[bb]) == FALSE) {
    up.dd0 <- min(opdoseidx[bb]+1, 8)
    eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
    for (ireps0 in 1:4) {
      for (dd0 in 1:up.dd0) {
        w1.0 <-post.wgt(priwgt0[,(4*(bb-1)+ireps0)],
                        prior.p1[ ,dd0])
        w2.0 <-post.wgt(priwgt0[,(4*(bb-1)+ireps0)],
                        prior.p2[ ,dd0])
        w3.0 <-post.wgt(priwgt0[,(4*(bb-1)+ireps0)],
                        prior.p3[ ,dd0])
        w4.0 <-post.wgt(priwgt0[,(4*(bb-1)+ireps0)],
                  (1-prior.p1[ ,dd0]-prior.p2[ ,dd0]-
                   prior.p3[ ,dd0]))
        priwgt1 <- cbind(w1.0[[1]], w2.0[[1]],
                                 w3.0[[1]], w4.0[[1]])
        p.ae0 <- w2.0[[2]] + w4.0[[2]]
        up.dd1 <- min(dd0+1, 8)
        eloss1 <- matrix(NA, nrow=4, ncol=up.dd1)
        p.ae01 <- rep(NA, up.dd1)
        if (p.ae0 <= AE.threshold) {
          # stage (i+1)
          for (ireps1 in 1:4) {
            for (dd1 in 1:up.dd1) {
              w1.1 <- post.wgt(priwgt1[ ,ireps1],
                               prior.p1[,dd1])
              w2.1 <- post.wgt(priwgt1[ ,ireps1],
                               prior.p2[,dd1])
              w3.1 <- post.wgt(priwgt1[ ,ireps1],
```

```
                    prior.p3[,dd1])
    w4.1 <- post.wgt(priwgt1[ ,ireps1],
         (1-prior.p1[,dd1]-
          prior.p2[,dd1]-prior.p3[,dd1]))
    p.ae01[dd1] <- w2.1[[2]] + w4.1[[2]]
    if (p.ae01[dd1] <= AE.threshold) {
      eloss1[ireps1,dd1] <-
        Eloss(w1.1[[1]],w2.1[[1]],
        w3.1[[1]], w4.1[[1]],
        prior.sample[,1],prior.sample[,2],
        w1.1[[2]], w2.1[[2]],
        w3.1[[2]], w4.1[[2]])[1]
    }
    if (p.ae01[dd1] > AE.threshold) {
            eloss1[ireps1, dd1] <- NA
    }
  }
  temp <- which.min(eloss1[ireps1, ])
  if (length(temp) != 0) {
    minELid.1[ireps1] <- temp
    dose01[ireps1,dd0,ireps0,bb] <-
                        minELid.1[ireps1]
    sub01[ireps1,dd0,ireps0,bb] <- i+1
    stop01[ireps1,dd0,ireps0,bb] <- 0
  }
  if (length(temp) == 0) {
    minELid.1[ireps1] <- NA
    dose01[ireps1,dd0,ireps0,bb] <- NA
    sub01[ireps1,dd0,ireps0,bb] <- i
    stop01[ireps1,dd0,ireps0,bb] <- 1
  }
  minEL.1[ireps1] <-
         eloss1[ireps1, minELid.1[ireps1]]
} ## ireps1
eloss0[ireps0, dd0]<-w1.0[[2]]*minEL.1[1]+
                     w2.0[[2]]*minEL.1[2]+
                     w3.0[[2]]*minEL.1[3]+
                     w4.0[[2]]*minEL.1[4]
if (is.na(eloss0[ireps0, dd0]) == TRUE) {
  eloss0[ireps0, dd0] <-
    Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]],
      w4.0[[1]],prior.sample[ ,1],
      prior.sample[ ,2],w1.0[[2]],
      w2.0[[2]], w3.0[[2]], w4.0[[2]])[1]
```

```
          }
        }
        if (p.ae0 > AE.threshold) {
          eloss0[ireps0, dd0] <- NA
        }
      }
      temp <- which.min(eloss0[ireps0, ])
      if (length(temp) != 0) {
        minELid.0[ireps0] <- temp
        sub0[ireps0, bb] <- i
        stop0[ireps0, bb] <- 0
      }
      if (length(temp) == 0) {
        minELid.0[ireps0] <- NA
        sub0[ireps0, bb] <- i-1
        stop0[ireps0, bb] <- 1
      }
    }
    dose0[ ,bb] <- minELid.0
  }
  if (is.na(opdoseidx[bb]) == TRUE) {
      dose0[ ,bb] <- rep(NA, 4)
  }
}
# optimal dose for stage i and i+1
# stage i
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
STOP.IDX[[i]] <- as.vector(stop0)
Nsub[[i]] <- as.vector(sub0)
# stage (i+1)
doseid.1 <- rep(NA, 4^i)
d1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
stop1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
sub1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
for (bb in 1:n.bb) {
  for (m0 in 1:4) {
    idx0 <- doseid.0[4*(bb-1)+m0]
    temp1 <- dose01[ ,idx0,m0,bb]
    d1.pre[(4*(bb-1)+m0), ] <- temp1
    temp1 <- stop01[ ,idx0,m0,bb]
    stop1.pre[(4*(bb-1)+m0), ] <- temp1
    temp1 <- sub01[ ,idx0,m0,bb]
    sub1.pre[(4*(bb-1)+m0), ] <- temp1
```

```
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
STOP.IDX[[i+1]] <- as.vector(t(stop1.pre))
Nsub[[i+1]] <- as.vector(t(sub1.pre))
# obtain the posterior weights for (i+1)th stage
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
p.temp <- loss.all <- rep(NA, 4^i)
# stage i
for (mm in 1:(4^(i-1))) {
  idx0 <- doseid.0[mm]
  w1.0 <- post.wgt(priwgt0[ ,mm], prior.p1[ ,idx0])
  w2.0 <- post.wgt(priwgt0[ ,mm], prior.p2[ ,idx0])
  w3.0 <- post.wgt(priwgt0[ ,mm], prior.p3[ ,idx0])
  w4.0 <- post.wgt(priwgt0[ ,mm],
    (1-prior.p1[ ,idx0]-prior.p2[ ,idx0]-
      prior.p3[ ,idx0]))
  wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
     cbind(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]])
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
     c(w1.0[[2]], w2.0[[2]], w3.0[[2]], w4.0[[2]])
  loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
     Eloss(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]],
     prior.sample[,1], prior.sample[,2],
     w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
}
P.OPT[[i]] <- p.temp
LOSS[[i]] <- loss.all
# stage i+1
priwgt1 <- wgt.temp
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^(i+1))
p.temp <- loss.all <- rep(NA, 4^(i+1))
for (mm in 1:(4^i)) {
  idx1 <- doseid.1[mm]
  w1.1 <- post.wgt(priwgt1[ ,mm], prior.p1[ ,idx1])
  w2.1 <- post.wgt(priwgt1[ ,mm], prior.p2[ ,idx1])
  w3.1 <- post.wgt(priwgt1[ ,mm], prior.p3[ ,idx1])
  w4.1 <- post.wgt(priwgt1[ ,mm],
   (1-prior.p1[ ,idx1]-prior.p2[ ,idx1]-
    prior.p3[ ,idx1]))
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
      c(w1.1[[2]], w2.1[[2]], w3.1[[2]], w4.1[[2]])
  loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
```

```
      Eloss(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]],
         prior.sample[ ,1], prior.sample[ ,2],
         w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[2:5]
    if (i < (nsubj-1)) {
      wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
        cbind(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]])
    }
    if (i == (nsubj-1))  wgt.temp <- NA
  }
  P.OPT[[i+1]] <- p.temp
  LOSS[[i+1]] <- loss.all
}
if (i == nsubj) {
  priwgt0 <- wgt.temp
  up.dd0 <- rep(NA, 4^(i-2))
  opdoseidx <- 2*OPDOSE[[i-1]]-2
  dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  minELid.0 <- rep(NA, 4)
  n.bb <- 4^(i-2)
  for (bb in 1:n.bb) {
    if (is.na(opdoseidx[bb])== FALSE) {
      up.dd0 <- min(opdoseidx[bb]+1, 8)
      eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
      p.ae0 <- rep(NA, up.dd0)
      for (ireps0 in 1:4) {
        for (dd0 in 1:up.dd0) {
          w1.0<-post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                         prior.p1[ ,dd0])
          w2.0<-post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                         prior.p2[ ,dd0])
          w3.0<-post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                         prior.p3[ ,dd0])
          w4.0<-post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                   (1-prior.p1[ ,dd0]-prior.p2[ ,dd0]-
                    prior.p3[ ,dd0]))
          eloss0[ireps0, dd0] <-
            Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]],
            w4.0[[1]],prior.sample[ ,1],
            prior.sample[ ,2],w1.0[[2]], w2.0[[2]],
            w3.0[[2]], w4.0[[2]])[1]
          p.ae0[dd0] <- w2.0[[2]] + w4.0[[2]]
        }
```

```
      # restriction on P(AE)
      doseid.0a <- which(p.ae0 <= AE.threshold)
      # if P(AE) <= AE.threshold
      if (length(doseid.0a) != 0) {
        # choose the one with minimum eloss
        temp <- which.min(eloss0[ireps0, doseid.0a])
        if (length(temp) != 0) {
          minELid.0[ireps0] <- temp
          sub0[ireps0, bb] <- i
          stop0[ireps0, bb] <- 0
        }
        if (length(temp) == 0) {
          minELid.0[ireps0] <- NA
          sub0[ireps0, bb] <- i-1
          stop0[ireps0, bb] <- 1
        }
      }
      # if P(AE) > AE.threshold
      if (length(doseid.0a) == 0) {
        minELid.0[ireps0] <- NA
        sub0[ireps0, bb] <- i-1
        stop0[ireps0, bb] <- 1
      }
    }
    dose0[ ,bb] <- minELid.0
  } ## if (is.na(opdoseidx[bb]) == FALSE)
  if (is.na(opdoseidx[bb]) == TRUE) {
       dose0[ ,bb] <- rep(NA, 4) }
}
# optimal dose for stage i
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
Nsub[[i]] <- as.vector(sub0)
STOP.IDX[[i]] <- as.vector(stop0)
# obtain the posterior weights
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
p.temp <- loss.all <- rep(NA, 4^i)
# stage i
for (mm in 1:(4^(i-1))) {
  idx0 <- doseid.0[mm]
   w1.0 <- post.wgt(priwgt0[ ,mm],prior.p1[ ,idx0])
   w2.0 <- post.wgt(priwgt0[ ,mm],prior.p2[ ,idx0])
   w3.0 <- post.wgt(priwgt0[ ,mm],prior.p3[ ,idx0])
   w4.0 <- post.wgt(priwgt0[ ,mm],
```

```
            (1-prior.p1[ ,idx0]-prior.p2[ ,idx0]-
              prior.p3[ ,idx0]))
         p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
            c(w1.0[[2]], w2.0[[2]], w3.0[[2]], w4.0[[2]])
         loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
           Eloss(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]],
                 prior.sample[ ,1], prior.sample[ ,2],
                 w1.0[[2]], w2.0[[2]],
                 w3.0[[2]], w4.0[[2]])[2:5]
      }
      P.OPT[[i]] <- p.temp
      LOSS[[i]] <- loss.all
    }
  }
  rm(priwgt0, priwgt1, w1.0, w2.0, w3.0, w4.0, w1.1,
     w2.1, w3.1, w4.1, dose0, wgt.temp, p.temp,
     sub0, stop0, loss.all)
  gc(reset=TRUE)
}
for (ncolumn in 1:nsubj) {
  p <- rep(P.OPT[[ncolumn]], each=4^(nsubj-ncolumn))
  prob.matrix[ ,ncolumn] <- p
  p <- rep(LOSS[[ncolumn]], each=4^(nsubj-ncolumn))
  loss.matrix[ ,ncolumn] <- p
  sub.matrix[ ,ncolumn] <-
     rep(Nsub[[ncolumn]], each=4^(nsubj-ncolumn))
}
# compute average number of subjects
# loss and probability for each outcome for each path
sub.final <- sub.matrix[,nsubj]
total.scenario <- length(sub.final)
loss.final <- loss.matrix[,nsubj]
total.lossscenario <- length(loss.final)
for (jj in 2:nsubj) {
  s.temp <- which(STOP.IDX[[jj]] == 1)
  if (length(s.temp) != 0) {
    for (ss in s.temp) {
      s.idx <- (total.scenario / (4^(jj-1))) * (ss-1) + 1
      sub.final[s.idx] <- sub.matrix[s.idx, jj]
      l.idx <- (total.lossscenario/(4^(jj-1)))*(ss-1) + 1
      loss.final[l.idx] <- loss.matrix[l.idx, jj-1]
    }
  }
}
```

```
    Esubj <- mean(sub.final, na.rm=TRUE)
    rm(P.OPT, p, sub.matrix, STOP.IDX, sub.final)
    gc(reset=TRUE)
    # calculate overall expected loss
    idx <- 1 - is.na(loss.final)
    prob.expt <- rep(NA, 4^nsubj)
    for (k in 1:4^nsubj) {
      if (idx[k] ==1) {
        prob.expt[k] <- prod(prob.matrix[k,], na.rm=T)
      }
    }
    OEL.2batch <- sum(prob.expt*loss.final, na.rm=T)
    rstt[1, ] <- c(nsubj, OEL.2batch, Esubj)
  return(list(OEL=rstt, optimal.dose=OPDOSE))
}

# ************************************************************#
#  Two-step-look-ahead procedure  (Moving Window)
#  NEW MODEL (efficacy and toxicity)
#  twostep.movingAE1(nsubj, AE.threshold)
#      nsubj: number of subjects
#      AE.threshold: the upper bound of posterior probability
#                    of toxicity
# VALUE:
#    This function returns a list:
#      [[1]]: c(nsubj, expected loss, E(nsubj)
#      [[2]]: optimal doses for each stage
# ************************************************************#
twostep.movingAE1 <- function(nsubj, AE.threshold) {
  rstt <- matrix(NA, nrow=1, ncol=3)
  prob.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
  loss.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
  sub.matrix <- matrix(NA, nrow=4^(nsubj-1), ncol=nsubj)
  # ------------------------ the first dose ---------------- #
  OPDOSE <- P.OPT <- Nsub <- STOP.IDX <- LOSS <- list(NULL)
  eloss1 <- matrix(NA, nrow=4^(1-1), ncol=ndose)
  up.dd1 <- ndose
  stop1 <- rep(NA, up.dd1)
  sub1 <- rep(NA, up.dd1)
  dose12 <- matrix(NA, nrow=4, ncol=up.dd1)
  p.ae1 <- rep(NA, up.dd1)
  minELid.2 <- rep(NA, 4)
  sub2 <- matrix(NA, nrow=4, ncol=up.dd1)
  stop2 <- matrix(NA, nrow=4, ncol=up.dd1)
  for (dd1 in 1:up.dd1){
```

```
# first stage #
# calculate the posterior weights
w1.1 <- post.wgt(w.init, prior.p1[,dd1])
w2.1 <- post.wgt(w.init, prior.p2[,dd1])
w3.1 <- post.wgt(w.init, prior.p3[,dd1])
w4.1 <- post.wgt(w.init, (1-prior.p1[,dd1]-
                          prior.p2[,dd1]-prior.p3[,dd1]))
priwgt2 <- cbind(w1.1[[1]], w2.1[[1]],
                 w3.1[[1]], w4.1[[1]])
p.ae1[dd1] <- w2.1[[2]] + w4.1[[2]]
if (p.ae1[dd1] <= AE.threshold) {
  # second stage ###
  up.dd2 <- min(dd1+1, 8)
  eloss2 <- matrix(NA, nrow=4^(2-1), ncol=up.dd2)
  p.ae12 <- rep(NA, up.dd2)
  for (ireps2 in 1:4){
    for (dd2 in 1:up.dd2) {
      w1.2 <- post.wgt(priwgt2[,ireps2], prior.p1[,dd2])
      w2.2 <- post.wgt(priwgt2[,ireps2], prior.p2[,dd2])
      w3.2 <- post.wgt(priwgt2[,ireps2], prior.p3[,dd2])
      w4.2 <- post.wgt(priwgt2[,ireps2],
       (1-prior.p1[,dd2]-prior.p2[,dd2]-prior.p3[,dd2]))
      p.ae12[dd2] <- w2.2[[2]] + w4.2[[2]]
      if (p.ae12[dd2] <= AE.threshold) {
        eloss2[ireps2,dd2] <-
           Eloss(w1.2[[1]],w2.2[[1]],w3.2[[1]],w4.2[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.2[[2]],w2.2[[2]],w3.2[[2]],w4.2[[2]])[1]
      }
      if (p.ae12[dd2] > AE.threshold) {
        eloss2[ireps2, dd2] <- NA
      }
    }
    temp <- which.min(eloss2[ireps2, ])
    if (length(temp) != 0) {
      minELid.2[ireps2] <- temp
      sub2[ireps2, dd1] <- 2
      stop2[ireps2, dd1] <- 0
    }
    if (length(temp) == 0) {
      minELid.2[ireps2] <- NA
      sub2[ireps2, dd1] <- 1
      stop2[ireps2, dd1] <- 1
    }
```

```
    }
    dose12[, dd1] <- minELid.2
    minEL.2 <- c(eloss2[1, minELid.2[1]],
        eloss2[2, minELid.2[2]], eloss2[3, minELid.2[3]],
        eloss2[4, minELid.2[4]])
    eloss1[dd1] <- w1.1[[2]]*minEL.2[1] +
         w2.1[[2]]*minEL.2[2] + w3.1[[2]]*minEL.2[3] +
         w4.1[[2]]*minEL.2[4]
    if (is.na(eloss1[dd1]) == TRUE) {
      eloss1[1, dd1] <-
         Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]],
               prior.sample[,1], prior.sample[,2],
               w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[1]
      }
    stop1[dd1] <- 0; sub1[dd1] <- 1
  }
    if (p.ae1[dd1] > AE.threshold) {
      eloss1[dd1] <- NA
      stop1[dd1] <- 1; sub1[dd1] <- 1
    }
}
# optimal dose for stage 1 and 2
# stage 1
temp1 <- which.min(eloss1)
if (length(temp1) != 0) {
  doseid.1 <- temp1; Nsub[[1]] <- 1
}
if (length(temp1) == 0) {
  doseid.1 <- NA; Nsub[[1]] <- 0
}
OPDOSE[[1]] <- doses[doseid.1]
STOP.IDX[[1]] <- stop1[doseid.1]
# obtain posterior weigths given the first and second dose
# first dose
w1.1 <- post.wgt(w.init, prior.p1[ ,doseid.1])
w2.1 <- post.wgt(w.init, prior.p2[ ,doseid.1])
w3.1 <- post.wgt(w.init, prior.p3[ ,doseid.1])
w4.1 <- post.wgt(w.init, (1-prior.p1[ ,doseid.1]-
           prior.p2[ ,doseid.1]-prior.p3[ ,doseid.1]))
wgt.temp <- cbind(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]])
P.OPT[[1]] <- c(w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])
LOSS[[1]] <- Eloss(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]],
               prior.sample[,1], prior.sample[,2],
               w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[2:5]
```

```
rm(priwgt2,w1.1,w2.1,w3.1,w4.1,w1.2,w2.2,w3.2,w4.2,stop1,sub1)
gc(reset=TRUE)
# -------------------- the i^{th} dose (i>1) ------------- #
for (i in 2:nsubj) {
  # for i < (nsubj -1), moving window
  if (i < (nsubj-1)) {
    priwgt0 <- wgt.temp
    opdoseidx <- 2*OPDOSE[[i-1]]-2
    dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    minELid.0 <- minEL.0 <- rep(NA, 4)
    minELid.1 <- minEL.1 <- rep(NA, 4)
    n.bb <- 4^(i-2)
    for (bb in 1:n.bb) {
      if (is.na(opdoseidx[bb]) == FALSE) {
        up.dd0 <- min(opdoseidx[bb]+1, 8)
        eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
        for (ireps0 in 1:4) {
          for (dd0 in 1:up.dd0) {
            w1.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
            prior.p1[ ,dd0])
            w2.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                             prior.p2[ ,dd0])
            w3.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                             prior.p3[ ,dd0])
            w4.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                        (1-prior.p1[,dd0]-prior.p2[,dd0]-
                         prior.p3[,dd0]))
            priwgt1 <- cbind(w1.0[[1]], w2.0[[1]],
                             w3.0[[1]], w4.0[[1]])
            p.ae0 <- w2.0[[2]] + w4.0[[2]]
            if (p.ae0 <= AE.threshold) {
              up.dd1 <- min(dd0+1, 8)
              eloss1 <- matrix(NA, nrow=4, ncol=up.dd1)
              # stage (i+1)
              for (ireps1 in 1:4) {
                for (dd1 in 1:up.dd1) {
                  w1.1 <- post.wgt(priwgt1[ ,ireps1],
                                   prior.p1[ ,dd1])
                  w2.1 <- post.wgt(priwgt1[ ,ireps1],
                                   prior.p2[ ,dd1])
                  w3.1 <- post.wgt(priwgt1[ ,ireps1],
                                   prior.p3[ ,dd1])
```

```
          w4.1 <- post.wgt(priwgt1[ ,ireps1],
           (1-prior.p1[ ,dd1]-prior.p2[ ,dd1]-
            prior.p3[ ,dd1]))
          p.ae01 <- w2.1[[2]] + w4.1[[2]]
          if (p.ae01 <= AE.threshold) {
            eloss1[ireps1, dd1] <-
             Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]],
              w4.1[[1]],prior.sample[ ,1],
              prior.sample[ ,2],w1.1[[2]],w2.1[[2]],
              w3.1[[2]], w4.1[[2]])[1]
          }
          if (p.ae01 > AE.threshold) {
            eloss1[ireps1, dd1] <- NA
          }
         }
         temp <- which.min(eloss1[ireps1, ])
         if (length(temp) != 0) {
           minELid.1[ireps1] <- temp}
         if (length(temp) == 0) {
           minELid.1[ireps1] <- NA }
         minEL.1[ireps1] <-
                 eloss1[ireps1, minELid.1[ireps1]]
      }
      eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.1[1] +
        w2.0[[2]]*minEL.1[2] +w3.0[[2]]*minEL.1[3]+
        w4.0[[2]]*minEL.1[4]
      if (is.na(eloss0[ireps0,dd0]) == TRUE) {
        eloss0[ireps0, dd0] <-
           Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]],
             w4.0[[1]],prior.sample[ ,1],
             prior.sample[ ,2],w1.0[[2]],w2.0[[2]],
             w3.0[[2]], w4.0[[2]])[1]
      }
    }
    if (p.ae0 > AE.threshold) {
      eloss0[ireps0, dd0] <-  NA }
  }
  temp <- which.min(eloss0[ireps0, ])
  if (length(temp) != 0) {
    minELid.0[ireps0] <- temp
    sub0[ireps0, bb] <- i
    stop0[ireps0, bb] <- 0
  }
  if (length(temp) == 0) {
```

```
          minELid.0[ireps0] <- NA
          sub0[ireps0, bb] <- i-1
          stop0[ireps0, bb] <- 1
        }
      }
      dose0[ ,bb] <- minELid.0
    }
    if (is.na(opdoseidx[bb])==TRUE) dose0[ ,bb]<-rep(NA,4)
  }
  # optimal dose for stage i
  # stage i
  doseid.0 <- as.vector(dose0)
  OPDOSE[[i]] <- doses[doseid.0]
  STOP.IDX[[i]] <- as.vector(stop0)
  Nsub[[i]] <- as.vector(sub0)
  # obtain the posterior weights
  wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
  p.temp <- loss.all <- rep(NA, 4^i)
  # stage i
  for (mm in 1:(4^(i-1))) {
    idx0 <- doseid.0[mm]
    w1.0 <- post.wgt(priwgt0[ ,mm], prior.p1[,idx0])
    w2.0 <- post.wgt(priwgt0[ ,mm], prior.p2[,idx0])
    w3.0 <- post.wgt(priwgt0[ ,mm], prior.p3[,idx0])
    w4.0 <- post.wgt(priwgt0[ ,mm], (1-prior.p1[,idx0]-
                    prior.p2[,idx0]-prior.p3[,idx0]))
    wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
         cbind(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]])
    p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
         c(w1.0[[2]], w2.0[[2]], w3.0[[2]], w4.0[[2]])
    loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
        Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
           prior.sample[,1], prior.sample[,2],
           w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
  }
  P.OPT[[i]] <- p.temp; LOSS[[i]] <- loss.all
  rm(priwgt0, priwgt1, w1.0, w2.0, w3.0, w4.0, w1.1,
     w2.1, w3.1, w4.1, p.temp, loss.all, stop0, sub0)
  gc(reset=TRUE)
}
if (i == (nsubj-1)) {
  priwgt0 <- wgt.temp
  opdoseidx <- 2*OPDOSE[[i-1]]-2
  dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
```

```
sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
minELid.0 <- rep(NA, 4)
dose01 <- array(NA, c(4,8,4,4^(i-2)))
sub01 <- array(NA, c(4,8,4,4^(i-2)))
stop01 <- array(NA, c(4,8,4,4^(i-2)))
minELid.1 <- minEL.1 <- rep(NA, 4)
n.bb <- 4^(i-2)
# stage i
for (bb in 1:n.bb) {
  if (is.na(opdoseidx[bb]) == FALSE) {
    up.dd0 <- min(opdoseidx[bb]+1, 8)
    eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
    for (ireps0 in 1:4) {
      for (dd0 in 1:up.dd0) {
        w1.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                         prior.p1[ ,dd0])
        w2.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                         prior.p2[ ,dd0])
        w3.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                         prior.p3[ ,dd0])
        w4.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                        (1-prior.p1[ ,dd0]-
                        prior.p2[,dd0]-prior.p3[,dd0]))
        priwgt1 <- cbind(w1.0[[1]], w2.0[[1]],
                         w3.0[[1]], w4.0[[1]])
        p.ae0 <- w2.0[[2]] + w4.0[[2]]
        up.dd1 <- min(dd0+1, 8)
        eloss1 <- matrix(NA, nrow=4, ncol=up.dd1)
        if (p.ae0 <= AE.threshold) {
          # stage (i+1)
          for (ireps1 in 1:4) {
            for (dd1 in 1:up.dd1) {
              w1.1 <- post.wgt(priwgt1[ ,ireps1],
                prior.p1[,dd1])
              w2.1 <- post.wgt(priwgt1[ ,ireps1],
                  prior.p2[,dd1])
              w3.1 <- post.wgt(priwgt1[ ,ireps1],
                  prior.p3[,dd1])
              w4.1 <- post.wgt(priwgt1[ ,ireps1],
                    (1-prior.p1[,dd1]-prior.p2[,dd1]-
                     prior.p3[,dd1]))
              p.ae01 <- w2.1[[2]] + w4.1[[2]]
              if (p.ae01 <= AE.threshold) {
```

```
            eloss1[ireps1, dd1] <-Eloss(w1.1[[1]],
                w2.1[[1]], w3.1[[1]], w4.1[[1]],
                prior.sample[ ,1],
                 prior.sample[,2], w1.1[[2]],
                 w2.1[[2]],w3.1[[2]] w4.1[[2]])[1]
          }
          if (p.ae01 > AE.threshold) {
            eloss1[ireps1, dd1] <- NA}
        }
        temp <- which.min(eloss1[ireps1, ])
        if (length(temp) != 0) {
          minELid.1[ireps1] <- temp
          dose01[ireps1,dd0,ireps0,bb] <-
                                  minELid.1[ireps1]
          sub01[ireps1,dd0,ireps0,bb] <- i+1
          stop01[ireps1,dd0,ireps0,bb] <- 0
        }
        if (length(temp) == 0) {
          minELid.1[ireps1] <- NA
          dose01[ireps1,dd0,ireps0,bb] <- NA
          sub01[ireps1,dd0,ireps0,bb] <- i
           stop01[ireps1,dd0,ireps0,bb] <- 1
        }
        minEL.1[ireps1] <-
            eloss1[ireps1, minELid.1[ireps1]]
      }
      eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.1[1]+
          w2.0[[2]]*minEL.1[2]+w3.0[[2]]*minEL.1[3]+
          w4.0[[2]]*minEL.1[4]
      if (is.na(eloss0[ireps0,dd0]) == TRUE) {
          eloss0[ireps0, dd0] <- Eloss(w1.0[[1]],
                 w2.0[[1]], w3.0[[1]], w4.0[[1]],
                 prior.sample[,1],prior.sample[,2],
                 w1.0[[2]], w2.0[[2]], w3.0[[2]],
                 w4.0[[2]])[1]
      }
    }
    if (p.ae0 > AE.threshold) {
      eloss0[ireps0, dd0] <-  NA }
  }
  temp <- which.min(eloss0[ireps0, ])
  if (length(temp) != 0) {
    minELid.0[ireps0] <- temp
    sub0[ireps0, bb] <- i
```

```
      stop0[ireps0, bb] <- 0
    }
    if (length(temp) == 0) {
      minELid.0[ireps0] <- NA
      sub0[ireps0, bb] <- i-1
      stop0[ireps0, bb] <- 1
    }
  }
  dose0[ ,bb] <- minELid.0
}
if (is.na(opdoseidx[bb]) == TRUE) {
    dose0[ ,bb] <- rep(NA, 4)}
}
# optimal dose for stage i and i+1
# stage i
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
STOP.IDX[[i]] <- as.vector(stop0)
Nsub[[i]] <- as.vector(sub0)
# stage (i+1)
doseid.1 <- rep(NA, 4^i)
d1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
stop1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
sub1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
for (bb in 1:n.bb) {
  for (m0 in 1:4) {
    idx0 <- doseid.0[4*(bb-1)+m0]
    temp1 <- dose01[ ,idx0,m0,bb]
    d1.pre[(4*(bb-1)+m0), ] <- temp1
    temp1 <- stop01[ ,idx0,m0,bb]
    stop1.pre[(4*(bb-1)+m0), ] <- temp1
    temp1 <- sub01[ ,idx0,m0,bb]
    sub1.pre[(4*(bb-1)+m0), ] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
STOP.IDX[[i+1]] <- as.vector(t(stop1.pre))
Nsub[[i+1]] <- as.vector(t(sub1.pre))
# obtain the posterior weights for (i+1)th stage
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
p.temp <- loss.all <- rep(NA, 4^i)
# stage i
for (mm in 1:(4^(i-1))) {
```

```
    idx0 <- doseid.0[mm]
    w1.0 <- post.wgt(priwgt0[ ,mm], prior.p1[ ,idx0])
    w2.0 <- post.wgt(priwgt0[ ,mm], prior.p2[ ,idx0])
    w3.0 <- post.wgt(priwgt0[ ,mm], prior.p3[ ,idx0])
    w4.0 <- post.wgt(priwgt0[ ,mm], (1-prior.p1[ ,idx0]-
                    prior.p2[ ,idx0]-prior.p3[ ,idx0]))
    wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
         cbind(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]])
    p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
         c(w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])
    loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
      Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
         prior.sample[,1], prior.sample[,2],
         w1.0[[2]] w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
  }
  P.OPT[[i]] <- p.temp
  LOSS[[i]] <- loss.all
  # stage i+1
  priwgt1 <- wgt.temp
  wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^(i+1))
  p.temp <- loss.all <- rep(NA, 4^(i+1))
  for (mm in 1:(4^i)) {
    idx1 <- doseid.1[mm]
    w1.1 <- post.wgt(priwgt1[ ,mm], prior.p1[ ,idx1])
    w2.1 <- post.wgt(priwgt1[ ,mm], prior.p2[ ,idx1])
    w3.1 <- post.wgt(priwgt1[ ,mm], prior.p3[ ,idx1])
    w4.1 <- post.wgt(priwgt1[ ,mm], (1-prior.p1[ ,idx1]-
                    prior.p2[ ,idx1]-prior.p3[ ,idx1]))
    p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
           c(w1.1[[2]], w2.1[[2]], w3.1[[2]], w4.1[[2]])
    loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
       Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]],
         prior.sample[ ,1], prior.sample[ ,2],
         w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[2:5]
    if (i < (nsubj-1)) {
      wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
        cbind(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]])
    }
    if (i == (nsubj-1)) wgt.temp <- NA
  }
  P.OPT[[i+1]] <- p.temp
  LOSS[[i+1]] <- loss.all
}
rm(priwgt1, w1.0, w2.0, w3.0, w4.0, w1.1, w2.1,
```

```
          w3.1, w4.1, p.temp, loss.all)
      gc(reset=TRUE)
    }
    for (ncolumn in 1:nsubj) {
        p <- rep(P.OPT[[ncolumn]], each=4^(nsubj-ncolumn))
        prob.matrix[ ,ncolumn] <- p
        p <- rep(LOSS[[ncolumn]], each=4^(nsubj-ncolumn))
        loss.matrix[ ,ncolumn] <- p
        sub.matrix[ ,ncolumn] <-
             rep(Nsub[[ncolumn]], each=4^(nsubj-ncolumn))
    }
    # compute average number of subjects
    # loss and probability for each outcome for each path
    sub.final <- sub.matrix[,nsubj]
    total.scenario <- length(sub.final)
    loss.final <- loss.matrix[,nsubj]
    total.lossscenario <- length(loss.final)
    for (jj in 2:nsubj) {
      s.temp <- which(STOP.IDX[[jj]] == 1)
      if (length(s.temp) != 0) {
        for (ss in s.temp) {
          s.idx <- (total.scenario / (4^(jj-1))) * (ss-1) + 1
          sub.final[s.idx] <- sub.matrix[s.idx, jj]
          l.idx <- (total.lossscenario/(4^(jj-1)))*(ss-1) + 1
          loss.final[l.idx] <- loss.matrix[l.idx, jj-1]
        }
      }
    }
    Esubj <- mean(sub.final, na.rm=TRUE)
    rm(P.OPT, p, sub.matrix, STOP.IDX, sub.final)
    gc(reset=TRUE)
    # calculate overall expected loss
    idx <- 1 - is.na(loss.final)
    prob.expt <- rep(NA, 4^nsubj)
    for (k in 1:4^nsubj) {
      if (idx[k] ==1) {
        prob.expt[k] <- prod(prob.matrix[k,], na.rm=T)
      }
    }
    OEL.2moving <- sum(prob.expt*loss.final, na.rm=T)
    rstt[1, ] <- c(nsubj, OEL.2moving, Esubj)
  return(list(OEL=rstt, optimal.dose=OPDOSE))
}
```

B.2.3 Three-step-look-ahead Procedure: Moving Window

```
# ************************************************************#
#  Three-step-look-ahead procedure  (Moving Window)
#  NEW MODEL (efficacy and toxicity)
#  threestep.movingAE1(nsubj, AE.threshold)
#      nsubj: number of subjects
#      AE.threshold: the upper bound of posterior probability
#                    of toxicity
# VALUE:
#    This function returns a list:
#      [[1]]: c(nsubj, expected loss, E(nsubj)
#      [[2]]: optimal doses for each stage
# ************************************************************#
threestep.movingAE <- function(nsubj, AE.threshold) {
  rstt <- matrix(NA, nrow=1, ncol=3)
  prob.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
  loss.matrix <- matrix(NA, nrow=4^nsubj, ncol=nsubj)
  sub.matrix <- matrix(NA, nrow=4^(nsubj-1), ncol=nsubj)
  OPDOSE <- P.OPT <- Nsub <- STOP.IDX <- LOSS <- list(NULL)
  # ------------------------ the first dose ---------------- #
  up.dd1 <- ndose
  eloss1 <- rep(NA, up.dd1)
  minELid.1 <- NA
  sub1 <- NA
  stop1 <- NA
  p.ae1 <- rep(NA, up.dd1)
  minELid.2 <- minEL.2 <- rep(NA, 4)
  minELid.3 <- minEL.3 <- rep(NA, 4)
  dose1 <- NA
  for (dd1 in 1:up.dd1) {
    w1.1 <- post.wgt(w.init, prior.p1[ ,dd1])
    w2.1 <- post.wgt(w.init, prior.p2[ ,dd1])
    w3.1 <- post.wgt(w.init, prior.p3[ ,dd1])
    w4.1 <- post.wgt(w.init, (1-prior.p1[ ,dd1]-
                              prior.p2[ ,dd1]-prior.p3[ ,dd1]))
    p.ae1[dd1] <- w2.1[[2]] + w4.1[[2]]
    priwgt2 <- cbind(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]])
    up.dd2 <- min(dd1+1, 8)
    eloss2 <- matrix(NA, nrow=4, ncol=up.dd2)
    p.ae12 <- rep(NA, up.dd2)
    for (ireps2 in 1:4) {
      for (dd2 in 1:up.dd2) {
        w1.2 <- post.wgt(priwgt2[ ,ireps2], prior.p1[ ,dd2])
```

```
  w2.2 <- post.wgt(priwgt2[ ,ireps2], prior.p2[ ,dd2])
  w3.2 <- post.wgt(priwgt2[ ,ireps2], prior.p3[ ,dd2])
  w4.2 <- post.wgt(priwgt2[ ,ireps2],(1-prior.p1[ ,dd2]-
                  prior.p2[ ,dd2]-prior.p3[ ,dd2]))
  p.ae12[dd2] <- w2.2[[2]] + w4.2[[2]]
  priwgt3 <- cbind(w1.2[[1]], w2.2[[1]],
                  w3.2[[1]], w4.2[[1]])
  up.dd3 <- min(dd2+1, 8)
  eloss3 <- matrix(NA, nrow=4, ncol=up.dd3)
  p.ae123 <- matrix(NA, nrow=4, ncol=up.dd3)
  for (ireps3 in 1:4) {
    for (dd3 in 1:up.dd3) {
      w1.3 <- post.wgt(priwgt3[,ireps3],prior.p1[,dd3])
      w2.3 <- post.wgt(priwgt3[,ireps3],prior.p2[,dd3])
      w3.3 <- post.wgt(priwgt3[,ireps3],prior.p3[,dd3])
      w4.3 <- post.wgt(priwgt3[,ireps3],(1-
              prior.p1[ ,dd3]-prior.p2[ ,dd3]-
              prior.p3[ ,dd3]))
      p.ae123[dd3] <- w2.3[[2]] + w4.3[[2]]
      eloss3[ireps3, dd3] <- Eloss(w1.3[[1]], w2.3[[1]],
             w3.3[[1]], w4.3[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.3[[2]],w2.3[[2]],w3.3[[2]] w4.3[[2]])[1]
    }
    # select doses satisfying the restriction criteria
    # at stage 3
    doseid.3a <- which(p.ae123 <= AE.threshold)
    if (length(doseid.3a) != 0) {
      temp <- which.min(eloss3[ireps3, doseid.3a])
      if (length(temp) != 0) minELid.3[ireps3] <- temp
      if (length(temp) == 0) minELid.3[ireps3] <- NA
    }
    if (length(doseid.3a) == 0) minELid.3[ireps3] <- NA
      minEL.3[ireps3] <- eloss3[ireps3, minELid.3[ireps3]]
  }
  eloss2[ireps2, dd2] <- w1.2[[2]]*minEL.3[1] +
           w2.2[[2]]*minEL.3[2] + w3.2[[2]]*minEL.3[3] +
           w4.2[[2]]*minEL.3[4]
}
# restriction criteria for stage 2
doseid.2a <- which(p.ae12 <= AE.threshold)
if (length(doseid.2a) != 0) {
  temp <- which.min(eloss2[ireps2, doseid.2a])
  if (length(temp) != 0) { minELid.2[ireps2] <- temp }
```

```
      if (length(temp) == 0) { minELid.2[ireps2] <- NA }
    }
    if (length(doseid.2a) == 0) { minELid.2[ireps2] <- NA }
      minEL.2[ireps2] <- eloss2[ireps2, minELid.2[ireps2]]
  }
  eloss1[dd1] <- w1.1[[2]]*minEL.2[1]+w2.1[[2]]*minEL.2[2]+
               w3.1[[2]]*minEL.2[3] + w4.1[[2]]*minEL.2[4]
}
# restriction criteria for stage 1
doseid.1a <- which(p.ae1 <= AE.threshold)
if (length(doseid.1a) != 0) {
  temp <- which.min(eloss1[doseid.1a])
  if (length(temp) !=0 ) {
    minELid.1 <- temp
    sub1 <- 1; stop1 <- 0
  }
  if (length(temp) ==0 ) {
    minELid.1<- NA
    sub1 <- 0; stop1 <- 1
  }
}
if (length(doseid.1a) == 0) {
  minELid.1 <- NA
  sub1 <- 0; stop1 <- 1
}
dose1 <- minELid.1
# optimal dose for stage 1, 2 and 3
# stage 1
doseid.1 <- as.vector(dose1)
OPDOSE[[1]] <- doses[doseid.1]
Nsub[[1]] <- sub1
STOP.IDX[[1]] <- stop1
# obtain the posterior weights for (i+1)th stage
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^1)
p.temp <- rep(NA, 4^1)
# stage 1
for (mm in 1:(4^(1-1))) {
  idx1 <- doseid.1[mm]
  w1.1 <- post.wgt(w.init, prior.p1[ ,idx1])
  w2.1 <- post.wgt(w.init, prior.p2[ ,idx1])
  w3.1 <- post.wgt(w.init, prior.p3[ ,idx1])
  w4.1 <- post.wgt(w.init, (1-prior.p1[ ,idx1]-
                  prior.p2[ ,idx1]-prior.p3[ ,idx1]))
  wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
```

```
         cbind(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]])
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
          c(w1.1[[2]], w2.1[[2]], w3.1[[2]], w4.1[[2]])
}
P.OPT[[1]] <- p.temp
LOSS[[1]] <- Eloss(w1.1[[1]],w2.1[[1]],w3.1[[1]],w4.1[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.1[[2]],w2.1[[2]],w3.1[[2]],w4.1[[2]])[2:5]
rm(w1.1, w2.1, w3.1, w4.1, w1.2, w2.2, w3.2, w4.2, w1.3,
   w2.3, w3.3, w4.3, p.temp)
gc(reset=TRUE)
# -------------------- the i^{th} dose (i>1) ------------ #
for (i in 2:nsubj) {
  # for i < (nsubj -2), moving window
  if (i < (nsubj-2)) {
    priwgt0 <- wgt.temp
    opdoseidx <- 2*OPDOSE[[i-1]]-2
    # dose0[ireps0, bb]
    dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    # number of subject at stage i
    sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    # indicator whether the trials stops or not at stage i
    stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
    minELid.0 <- minEL.0 <- rep(NA, 4)
    minELid.1 <- minEL.1 <- rep(NA, 4)
    minELid.2 <- minEL.2 <- rep(NA, 4)
    n.bb <- 4^(i-2)
    for (bb in 1:n.bb) {
      if (is.na(opdoseidx[bb]) == FALSE) {
        up.dd0 <- min(opdoseidx[bb]+1, 8)
        eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
        p.ae0 <- rep(NA, up.dd0)
        for (ireps0 in 1:4) {
          for (dd0 in 1:up.dd0) {
            w1.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                             prior.p1[ ,dd0])
            w2.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                             prior.p2[ ,dd0])
            w3.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                             prior.p3[ ,dd0])
            w4.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                    (1-prior.p1[ ,dd0]-prior.p2[ ,dd0]-
                     prior.p3[ ,dd0]))
            priwgt1 <- cbind(w1.0[[1]], w2.0[[1]],
```

```
                w3.0[[1]], w4.0[[1]])
p.ae0[dd0] <- w2.0[[2]] + w4.0[[2]]
up.dd1 <- min(dd0+1, 8)
eloss1 <- matrix(NA, nrow=4, ncol=up.dd1)
p.ae01 <- rep(NA, up.dd1)
# stage (i+1)
for (ireps1 in 1:4) {
  for (dd1 in 1:up.dd1) {
    w1.1 <- post.wgt(priwgt1[ ,ireps1],
                     prior.p1[ ,dd1])
    w2.1 <- post.wgt(priwgt1[ ,ireps1],
                     prior.p2[ ,dd1])
    w3.1 <- post.wgt(priwgt1[ ,ireps1],
                     prior.p3[ ,dd1])
    w4.1 <- post.wgt(priwgt1[ ,ireps1],
           (1-prior.p1[ ,dd1]-
            prior.p2[ ,dd1]-prior.p3[ ,dd1]))
    priwgt2 <- cbind(w1.1[[1]], w2.1[[1]],
                     w3.1[[1]], w4.1[[1]])
    eloss1[ireps1, dd1] <-
      Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]],
       w4.1[[1]],prior.sample[ ,1],
       prior.sample[ ,2], w1.1[[2]], w2.1[[2]],
       w3.1[[2]], w4.1[[2]])[1]
    p.ae01[dd1] <- w2.1[[2]] + w4.1[[2]]
    up.dd2 <- min(dd1+1, 8)
    eloss2 <- matrix(NA, nrow=4, ncol=up.dd2)
    p.ae012 <- rep(NA, up.dd2)
    for (ireps2 in 1:4) {
      for (dd2 in 1:up.dd2) {
        w1.2 <- post.wgt(priwgt2[ ,ireps2],
                         prior.p1[ ,dd2])
        w2.2 <- post.wgt(priwgt2[ ,ireps2],
                         prior.p2[ ,dd2])
        w3.2 <- post.wgt(priwgt2[ ,ireps2],
                         prior.p3[ ,dd2])
        w4.2 <- post.wgt(priwgt2[ ,ireps2],
              (1-prior.p1[,dd2]-prior.p2[,dd2]-
               prior.p3[,dd2]))
        eloss2[ireps2, dd2] <-
           Eloss(w1.2[[1]], w2.2[[1]], w3.2[[1]],
             w4.2[[1]],prior.sample[ ,1],
             prior.sample[ ,2], w1.2[[2]],
             w2.2[[2]], w3.2[[2]], w4.2[[2]])[1]
```

```
            p.ae012[dd2] <- w2.2[[2]] + w4.2[[2]]
          }
          # restriction on P(AE|data)
          doseid.012a <- which(p.ae012<=AE.threshold)
          if (length(doseid.2a) != 0) {
            temp <- which.min(eloss2[ireps2,
                                     doseid.012a])
            if (length(temp) != 0) {
                      minELid.2[ireps2] <- temp
            if (length(temp) == 0)
                      minELid.2[ireps2] <- NA }
          }
          if (length(doseid.2a) == 0) {
                      minELid.2[ireps2] <- NA }
          minEL.2[ireps2] <- eloss2[ireps2,
                                    minELid.2[ireps2]]
        }
        eloss1[ireps1, dd1] <- w1.1[[2]]*minEL.2[1]+
          w2.1[[2]]*minEL.2[2]+w3.1[[2]]*minEL.2[3]+
          w4.1[[2]]*minEL.2[4]
      }
      # restriction on P(AE) at stage (i+1)
      doseid.01a <- which(p.ae01 <= AE.threshold)
      # if P(AE) <= AE.threshold: there's an optimal dose
      if (length(doseid.01a) != 0) {
        temp <- which.min(eloss1[ireps1, doseid.01a])
        if (length(temp) != 0) {
                 minELid.1 [ireps1] <- temp }
        if (length(temp) == 0) {
                 minELid.1 [ireps1] <- NA }
      }
      if (length(doseid.01a) == 0) {
                   minELid.1 [ireps1] <- NA }
      minEL.1[ireps1] <-
               eloss1[ireps1, minELid.1[ireps1]]
    }
    eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.1[1] +
                           w2.0[[2]]*minEL.1[2] +
                           w3.0[[2]]*minEL.1[3] +
                           w4.0[[2]]*minEL.1[4]
}
doseid.0a <- which(p.ae0 <= AE.threshold)
if (length(doseid.0a) != 0) {
   temp <- which.min(eloss0[ireps0, doseid.0a])
```

```
          if (length(temp) != 0) {
            minELid.0[ireps0] <- temp
            sub0[ireps0, bb] <- i
            stop0[ireps0, bb] <- 0
          }
          if (length(temp) == 0) {
            minELid.0[ireps0] <- NA
            sub0[ireps0, bb] <- i-1
            stop0[ireps0, bb] <- 1
          }
        }
        if (length(doseid.0a) == 0) {
          minELid.0[ireps0] <- NA
          sub0[ireps0, bb] <- i-1
          stop0[ireps0, bb] <- 1
        }
      }
      dose0[ ,bb] <- minELid.0
    }
    if (is.na(opdoseidx[bb]) == TRUE) {
               dose0[ ,bb] <- rep(NA, 4) }
}
# optimal dose for stage i
# stage i
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
STOP.IDX[[i]] <- as.vector(stop0)
Nsub[[i]] <- as.vector(sub0)
# obtain the posterior weights
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
p.temp <- loss.all <- rep(NA, 4^i)
# stage i
for (mm in 1:(4^(i-1))) {
  idx0 <- doseid.0[mm]
  w1.0 <- post.wgt(priwgt0[ ,mm], prior.p1[,idx0])
  w2.0 <- post.wgt(priwgt0[ ,mm], prior.p2[,idx0])
   w3.0 <- post.wgt(priwgt0[ ,mm], prior.p3[,idx0])
   w4.0 <- post.wgt(priwgt0[ ,mm], (1-prior.p1[,idx0]-
                    prior.p2[,idx0]-prior.p3[,idx0]))
   wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
          cbind(w1.0[[1]],w2.0[[1]],w3.0[[1]],w4.0[[1]])
   p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
          c(w1.0[[2]], w2.0[[2]], w3.0[[2]], w4.0[[2]])
   loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
```

```
        Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
             prior.sample[,1], prior.sample[,2],
             w1.0[[2]],w2.0[[2]],w3.0[[2]],w4.0[[2]])[2:5]
  }
  P.OPT[[i]] <- p.temp
  LOSS[[i]] <- loss.all
  rm(priwgt0, priwgt1, priwgt2, w1.0, w2.0, w3.0, w4.0,
     w1.1, w2.1, w3.1, w4.1, p.temp, loss.all, stop0, sub0)
  gc(reset=TRUE)
}
# -- for i = (nsubj -2), batching - select the dose
# for i, (i+1) and (i+2)
if (i == (nsubj-2)) {
  n.bb <- 4^(i-2)
  priwgt0 <- wgt.temp
  opdoseidx <- 2*OPDOSE[[i-1]]-2
  # dose0[ireps0, bb]
  dose0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  # number of subject at stage i
  sub0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  # indicator whether the trials stops or not at stage i
  stop0 <- matrix(NA, nrow=4, ncol=4^(i-2))
  minELid.0 <- minEL.0 <- rep(NA, 4)
  # dose01[ireps1, dd0, ireps0, bb]
  dose01 <- array(NA, c(4,8,4,n.bb))
  sub01 <- array(NA, c(4,8,4,n.bb))
  stop01 <- array(NA, c(4,8,4,n.bb))
  minELid.1 <- minEL.1 <- rep(NA, 4)
  #dose012[ireps2, dd1, ireps1, dd0, ireps0, bb]
  dose012 <- array(NA, c(4,8,4,8,4,n.bb))
  sub012 <- array(NA, c(4,8,4,8,4,n.bb))
  stop012 <- array(NA, c(4,8,4,8,4,n.bb))
  minELid.2 <- minEL.2 <- rep(NA, 4)
  for (bb in 1:n.bb) {
    if (is.na(opdoseidx[bb]) == FALSE) {
      up.dd0 <- min(opdoseidx[bb]+1, 8)
      eloss0 <- matrix(NA, nrow=4, ncol=up.dd0)
      p.ae0 <- rep(NA, up.dd0)
      for (ireps0 in 1:4) {
        for (dd0 in 1:up.dd0) {
          w1.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                           prior.p1[ ,dd0])
          w2.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                           prior.p2[ ,dd0])
```

```
w3.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
                 prior.p3[ ,dd0])
w4.0 <- post.wgt(priwgt0[ ,(4*(bb-1)+ireps0)],
            (1-prior.p1[ ,dd0]-prior.p2[ ,dd0]-
             prior.p3[ ,dd0]))
priwgt1 <- cbind(w1.0[[1]], w2.0[[1]],
                 w3.0[[1]], w4.0[[1]])
p.ae0[dd0] <- w2.0[[2]] + w4.0[[2]]
up.dd1 <- min(dd0+1, 8)
eloss1 <- matrix(NA, nrow=4, ncol=up.dd1)
p.ae01 <- rep(NA, up.dd1)
# stage (i+1)
for (ireps1 in 1:4) {
  for (dd1 in 1:up.dd1) {
    w1.1 <- post.wgt(priwgt1[ ,ireps1],
                     prior.p1[ ,dd1])
    w2.1 <- post.wgt(priwgt1[ ,ireps1],
                     prior.p2[ ,dd1])
    w3.1 <- post.wgt(priwgt1[ ,ireps1],
                     prior.p3[ ,dd1])
    w4.1 <- post.wgt(priwgt1[ ,ireps1],
       (1-prior.p1[ ,dd1]-prior.p2[ ,dd1]-
        prior.p3[ ,dd1]))
    priwgt2 <- cbind(w1.1[[1]], w2.1[[1]],
                     w3.1[[1]], w4.1[[1]])
    eloss1[ireps1, dd1] <-
        Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]],
          w4.1[[1]], prior.sample[ ,1],
          prior.sample[ ,2],w1.1[[2]],w2.1[[2]],
          w3.1[[2]], w4.1[[2]])[1]
    p.ae01[dd1] <- w2.1[[2]] + w4.1[[2]]
    up.dd2 <- min(dd1+1, 8)
    eloss2 <- matrix(NA, nrow=4, ncol=up.dd2)
    p.ae012 <- rep(NA, up.dd2)
    for (ireps2 in 1:4) {
      for (dd2 in 1:up.dd2) {
        w1.2 <- post.wgt(priwgt2[ ,ireps2],
                         prior.p1[ ,dd2])
        w2.2 <- post.wgt(priwgt2[ ,ireps2],
                         prior.p2[ ,dd2])
        w3.2 <- post.wgt(priwgt2[ ,ireps2],
                         prior.p3[ ,dd2])
        w4.2 <- post.wgt(priwgt2[ ,ireps2],
           (1-prior.p1[ ,dd2]-prior.p2[ ,dd2]-
```

```
        prior.p3[ ,dd2]))
    eloss2[ireps2, dd2] <-
       Eloss(w1.2[[1]],w2.2[[1]],w3.2[[1]],
           w4.2[[1]],prior.sample[ ,1],
           prior.sample[ ,2], w1.2[[2]],
           w2.2[[2]],w3.2[[2]],w4.2[[2]])[1]
    p.ae012[dd2] <- w2.2[[2]] + w4.2[[2]]
  }
  # restriction on P(AE|data)
  doseid.012a <-which(p.ae012<=AE.threshold)
  if (length(doseid.012a) != 0) {
    temp <- which.min(eloss2[ireps2,
                      doseid.012a])
    if (length(temp) != 0) {
      minELid.2[ireps2] <- temp
      dose012[ireps2, dd1, ireps1, dd0,
           ireps0, bb] <- minELid.2[ireps2]
      sub012[ireps2, dd1, ireps1, dd0,
           ireps0, bb] <- i+2
      stop012[ireps2, dd1, ireps1, dd0,
           ireps0, bb] <- 0
    }
    if (length(temp) == 0) {
      minELid.2[ireps2] <- NA
      dose012[ireps2, dd1, ireps1, dd0,
           ireps0, bb] <- NA
      sub012[ireps2, dd1, ireps1, dd0,
          ireps0, bb] <- i+1
      stop012[ireps2, dd1, ireps1, dd0,
          ireps0, bb] <- 1
    }
  }
  if (length(doseid.012a) == 0) {
    minELid.2[ireps2] <- NA
    dose012[ireps2, dd1, ireps1, dd0,
         ireps0, bb] <- NA
    sub012[ireps2, dd1, ireps1, dd0,
        ireps0, bb] <- i+1
    stop012[ireps2, dd1, ireps1, dd0,
         ireps0, bb] <- 1
  }
  minEL.2[ireps2] <-
      eloss2[ireps2, minELid.2[ireps2]]
}
```

```
        eloss1[ireps1, dd1] <- w1.1[[2]]*minEL.2[1]+
                               w2.1[[2]]*minEL.2[2]+
                               w3.1[[2]]*minEL.2[3]+
                               w4.1[[2]]*minEL.2[4]
      }
      # restriction on P(AE) at stage (i+1)
      doseid.01a <- which(p.ae01 <= AE.threshold)
      # if P(AE) <= AE.threshold
      if (length(doseid.01a) != 0) {
      temp <- which.min(eloss1[ireps1, doseid.01a])
      if (length(temp) != 0) {
        minELid.1[ireps1] <- temp
                                 minELid.1[ireps1]
        sub01[ireps1, dd0, ireps0, bb] <- i+1
        stop01[ireps1, dd0, ireps0, bb] <- 0
      }
      if (length(temp) == 0) {
        minELid.1[ireps1] <- NA
        dose01[ireps1, dd0, ireps0, bb] <- NA
        sub01[ireps1, dd0, ireps0, bb] <- i
        stop01[ireps1, dd0, ireps0, bb] <- 1
         dose01[ireps1, dd0, ireps0, bb] <-
       }
    }
    if (length(doseid.01a) == 0) {
      minELid.1[ireps1] <- NA
      dose01[ireps1, dd0, ireps0, bb] <- NA
      sub01[ireps1, dd0, ireps0, bb] <- i
      stop01[ireps1, dd0, ireps0, bb] <- 1
    }
    minEL.1[ireps1] <-
            eloss1[ireps1, minELid.1[ireps1]]
  }
  eloss0[ireps0, dd0] <- w1.0[[2]]*minEL.1[1] +
                         w2.0[[2]]*minEL.1[2] +
                         w3.0[[2]]*minEL.1[3] +
                         w4.0[[2]]*minEL.1[4]
}
doseid.0a <- which(p.ae0 <= AE.threshold)
length(doseid.0a) != 0) {
  temp <- which.min(eloss0[ireps0, doseid.0a])
  if (length(temp) != 0) {
    minELid.0[ireps0] <- temp
    sub0[ireps0, bb] <- i
```

```
        stop0[ireps0, bb] <- 0
      }
      if (length(temp) == 0) {
        minELid.0[ireps0] <- NA
        sub0[ireps0, bb] <- i-1
        stop0[ireps0, bb] <- 1
      }
    }
    if (length(doseid.0a) == 0) {
      minELid.0[ireps0] <- NA
      sub0[ireps0, bb] <- i-1
      stop0[ireps0, bb] <- 1
    }
  }
  dose0[ ,bb] <- minELid.0
  }
  if (is.na(opdoseidx[bb]) == TRUE) {
     dose0[ ,bb] <- rep(NA, 4) }
}
# optimal dose for stage i
# stage i
doseid.0 <- as.vector(dose0)
OPDOSE[[i]] <- doses[doseid.0]
STOP.IDX[[i]] <- as.vector(stop0)
Nsub[[i]] <- as.vector(sub0)
# stage (i+1)
doseid.1 <- rep(NA, 4^i)
d1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
sub1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
stop1.pre <- matrix(NA, nrow=4^(i-1), ncol=4)
for (bb in 1:n.bb) {
  for (ireps0 in 1:4) {
    idx0 <- doseid.0[4*(bb-1)+ireps0]
    temp1 <- dose01[ , idx0, ireps0, bb]
    d1.pre[(4*(bb-1)+ireps0),] <- temp1
    temp1 <- sub01[ , idx0, ireps0, bb]
    sub1.pre[(4*(bb-1)+ireps0),] <- temp1
    temp1 <- stop01[ , idx0, ireps0, bb]
    stop1.pre[(4*(bb-1)+ireps0),] <- temp1
  }
}
doseid.1 <- as.vector(t(d1.pre))
OPDOSE[[i+1]] <- doses[doseid.1]
Nsub[[i+1]] <- as.vector(t(sub1.pre))
```

```
STOP.IDX[[i+1]] <- as.vector(t(stop1.pre))
# stage (i+2)
doseid.2 <- rep(NA, 4^(i+1))
d2.pre <- matrix(NA, nrow=4^i, ncol=4)
sub2.pre <- matrix(NA, nrow=4^i, ncol=4)
stop2.pre <- matrix(NA, nrow=4^i, ncol=4)
for (bb in 1:n.bb){
  for (ireps0 in 1:4) {
    idx0.i <- 4*(bb-1)+ireps0
    idx0 <- doseid.0[idx0.i]
    for (ireps1 in 1:4) {
      idx1 <- doseid.1[4*(idx0.i-1)+ireps1]
      temp1 <- dose012[ ,idx1,ireps1,idx0,ireps0,bb]
      d2.pre[(4*(idx0.i-1)+ireps1), ] <- temp1
      temp1 <- sub012[ ,idx1,ireps1,idx0,ireps0,bb]
      sub2.pre[(4*(idx0.i-1)+ireps1), ] <- temp1
      temp1 <- stop012[ ,idx1,ireps1,idx0,ireps0,bb]
      stop2.pre[(4*(idx0.i-1)+ireps1), ] <- temp1
    }
  }
}
doseid.2 <- as.vector(t(d2.pre))
OPDOSE[[i+2]] <- doses[doseid.2]
Nsub[[i+2]] <- as.vector(t(sub2.pre))
STOP.IDX[[i+2]] <- as.vector(t(stop2.pre))
# obtain the posterior weights
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^i)
p.temp <- loss.all <- rep(NA, 4^i)
# stage i
for (mm in 1:(4^(i-1))) {
  idx0 <- doseid.0[mm]
  w1.0 <- post.wgt(priwgt0[ ,mm], prior.p1[,idx0])
  w2.0 <- post.wgt(priwgt0[ ,mm], prior.p2[,idx0])
  w3.0 <- post.wgt(priwgt0[ ,mm], prior.p3[,idx0])
  w4.0 <- post.wgt(priwgt0[ ,mm], (1-prior.p1[,idx0]-
                  prior.p2[,idx0]-prior.p3[,idx0]))
  wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
       cbind(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]])
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
       c(w1.0[[2]], w2.0[[2]], w3.0[[2]], w4.0[[2]])
  loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
      Eloss(w1.0[[1]], w2.0[[1]], w3.0[[1]], w4.0[[1]],
       prior.sample[,1], prior.sample[,2], w1.0[[2]],
       w2.0[[2]], w3.0[[2]], w4.0[[2]])[2:5]
```

```
}
P.OPT[[i]] <- p.temp
LOSS[[i]] <- loss.all
# stage i+1
priwgt1 <- wgt.temp
wgt.temp <- matrix(NA, nrow=n.mc, ncol=4^(i+1))
p.temp <- loss.all <- rep(NA, 4^(i+1))
for (mm in 1:(4^i)) {
  idx1 <- doseid.1[mm]
  w1.1 <- post.wgt(priwgt1[,mm], prior.p1[,idx1])
  w2.1 <- post.wgt(priwgt1[,mm], prior.p2[,idx1])
  w3.1 <- post.wgt(priwgt1[,mm], prior.p3[,idx1])
  w4.1 <- post.wgt(priwgt1[,mm], (1-prior.p1[,idx1]-
                 prior.p2[,idx1]-prior.p3[,idx1]))
  wgt.temp[ ,(4*(mm-1)+1):(4*(mm-1)+4)] <-
      cbind(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]])
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <-
      c(w1.1[[2]], w2.1[[2]], w3.1[[2]], w4.1[[2]])
  loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
      Eloss(w1.1[[1]], w2.1[[1]], w3.1[[1]], w4.1[[1]],
       prior.sample[,1], prior.sample[,2], w1.1[[2]],
       w2.1[[2]], w3.1[[2]], w4.1[[2]])[2:5]
}
P.OPT[[i+1]] <- p.temp
LOSS[[i+1]] <- loss.all
# stage i+2
priwgt2 <- wgt.temp
wgt.temp <- NA
p.temp <- loss.all <- rep(NA, 4^(i+2))
for (mm in 1:(4^(i+1))) {
  idx2 <- doseid.2[mm]
  w1.2 <- post.wgt(priwgt2[,mm], prior.p1[,idx2])
  w2.2 <- post.wgt(priwgt2[,mm], prior.p2[,idx2])
  w3.2 <- post.wgt(priwgt2[,mm], prior.p3[,idx2])
  w4.2 <- post.wgt(priwgt2[,mm],(1-prior.p1[,idx2]-
                prior.p2[,idx2]-prior.p3[,idx2]))
  p.temp[(4*(mm-1)+1):(4*(mm-1)+4)] <- c(w1.2[[2]],
                w2.2[[2]], w3.2[[2]], w4.2[[2]])
  loss.all[(4*(mm-1)+1):(4*(mm-1)+4)] <-
        Eloss(w1.2[[1]],w2.2[[1]],w3.2[[1]],w4.2[[1]],
        prior.sample[,1],prior.sample[,2],w1.2[[2]],
        w2.2[[2]],w3.2[[2]],w4.2[[2]])[2:5]
}
P.OPT[[i+2]] <- p.temp
```

```
      LOSS[[i+2]] <- loss.all
    }
  }
  for (ncolumn in 1:nsubj) {
    p <- rep(P.OPT[[ncolumn]], each=4^(nsubj-ncolumn))
    prob.matrix[ ,ncolumn] <- p
    p <- rep(LOSS[[ncolumn]], each=4^(nsubj-ncolumn))
    loss.matrix[ ,ncolumn] <- p
    sub.matrix[ ,ncolumn] <-
          rep(Nsub[[ncolumn]], each=4^(nsubj-ncolumn))
  }
  # compute average number of subjects
  # loss and probability for each outcome for each path
  sub.final <- sub.matrix[,nsubj]
  total.scenario <- length(sub.final)
  loss.final <- loss.matrix[,nsubj]
  total.lossscenario <- length(loss.final)
  for (jj in 2:nsubj) {
    s.temp <- which(STOP.IDX[[jj]] == 1)
    if (length(s.temp) != 0) {
      for (ss in s.temp) {
        s.idx <- (total.scenario / (4^(jj-1))) * (ss-1) + 1
        sub.final[s.idx] <- sub.matrix[s.idx, jj]
        l.idx <- (total.lossscenario/(4^(jj-1)))*(ss-1) + 1
        loss.final[l.idx] <- loss.matrix[l.idx, jj-1]
      }
    }
  }
  Esubj <- mean(sub.final, na.rm=TRUE)
  rm(P.OPT, p, sub.matrix, STOP.IDX, sub.final); gc(reset=TRUE)
  # calculate overall expected loss
  idx <- 1 - is.na(loss.final)
  prob.expt <- rep(NA, 4^nsubj)
  for (k in 1:4^nsubj) {
    if (idx[k] ==1) {
      prob.expt[k] <- prod(prob.matrix[k,], na.rm=T)
    }
  }
  OEL.3moving <- sum(prob.expt*loss.final, na.rm=T)
  rstt[1, ] <- c(nsubj, OEL.3moving, Esubj)
  return(list(OEL=rstt, optimal.dose=OPDOSE))
}
```

REFERENCES

[1] M.A. Agin. *Optimal Bayesian Design for Nonlinear Models*. PhD thesis, University of Minnesota, School of Statistics, 1997.

[2] A. Agresti. *Categorical Data Analysis*. Wiley, New York, 2002.

[3] K. Anderson. *gsDesign: Group Sequential Design*, 2010. R package version 2.2-11.

[4] A.C. Atkinson and A.N. Donov. *Optimum Experimental Designs*. Oxford Science Publications, Oxford, 1996.

[5] J. Babb, A. Rogatko, and S Zacks. Cancer phase I clinical trials: Efficient dose escalation with overdose control. *Statistics in Medicine*, 17:1103–1120, 1998.

[6] J.S. Babb and A. Rogatko. Bayesian methods for cancer phase I clinical trials. In N.L. Geller, editor, *Advances in Clinical Trial Biostatistics*, chapter 1. Marcel Dekker, New York, 2004.

[7] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*, chapter 4. Spring-Verlag, New York, 1985.

[8] D.A. Berry. Experimental design for drug development: A Bayesian approach. *Journal of Biopharmaceutical Statistics*, 1:81–101, 1991.

[9] G.P. Bodey and S.S. Legha. The phase I study: General objectives, methods, and evaluation. In F.M. Muggia and M. Rozencweig, editors, *Clinical Evaluation of Antitumor Therapy*. Nijhoff, The Netherlands: Dordrecht, 1987.

[10] B. Bornkamp, J. Pinheiro, and F Bretz. MCPMod: An R package for the design and analysis of dose-finding studies. *Journal of Statistical Software*, 29(7):1–23, 2009.

[11] B. Bornkamp, J. Pinheiro, and F Bretz. *DoseFinding: Planning and Analyzing Dose Finding experiments*, 2010. R package version 0.1-1.

[12] T.M. Braun. The bivariate continual reassessment method: Extending the CRM to phase I trials of two competing outcomes. *Controlled Clinical Trials*, 23:240–256, 2002.

[13] T.M. Braun, Z. Yuan, and P.F. Thall. Determining a maximum-tolerated schedule of a cytotoxic agent. *Biometrics*, 61:335 – 343, 2005.

[14] R.H. Byrd, P. Lu, J. Nocedal, and C Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, 16:1190–1208, 1995.

[15] C. Casper and O.A. Perez. *ldbounds: Lan-DeMets Method for Group Sequential Boundaries. Based on FORTRAN program ld98*, 2006. R package version 1.0-1.

[16] K. Chaloner and K. Larntz. Optimal Bayesian design applied to logistic regression experiments. *Journal of Statistical Planning and Inference*, 21:191–208, 1989.

[17] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Sicence*, 10:273–304, 1995.

[18] Y.K. Cheung and R. Chappell. Sequential designs for phase I clinical trials with late-onset toxicities. *Biometrics*, 56:1177–1182, 2000.

[19] S. Chevret. Basic concepts in dose-finding. In S. Chevret, editor, *Statistical Methods for Dose-Finding Experiments*, chapter 1. Wiley, England, 2006.

[20] M. Clyde and K. Chaloner. The equivalence of constrained and weighted designs in multiple objective design problems. *Journal of the American Statistical Association*, 91:1236–1244, 1996.

[21] M.H. DeGroot. *Optimal Statistical Decisions*. Wiley, New Jersy, 2005.

[22] W.J. Dixon and D.A. Mood. A method for obtaining and analyzing sensitivity data. *Journal of the American Statistical Association*, 43:109–126, 1948.

[23] L. Edler and I. Burkholder. Overview of phase I trial. In J. Crowley and D.P. Ankerst, editors, *Handbook of Statistics in Clinical Oncology*, chapter 1. CRC Press, Florida, 2006.

[24] P.R. Edwards and J.K. Aronson. Adverse drug reactions: Definitions, diagnosis, and management. *Lancet*, 356:1255–1259, 2000.

[25] B.H. Eichhorn and S. Zacks. Sequential search of an optimal dosage, I. *Journal of the American Statistical Assiciation*, 68:594–598, 1973.

[26] S.K. Fan and K. Chaloner. Optimal designs and limiting optimal designs for a trinomial response. *Journal of Statistical Planning and Inference*, 126:347–360, 2003.

[27] P.R. Freeman. Optimal Bayesian sequential estimation of the median effective dose. *Biometrika*, 57:79–89, 1970.

[28] C. Gatsonis and J.B. Greenhouse. Bayesian methods for phase I clinical trials. *Statistics in Medicine*, 11:1377–1389, 1992.

[29] B. Gaydos, M. Krams, I. Perevozskaya, F. Bretz, Q. Liu, P. Gallo, D. Berry, C. Chuang-Stein, J. Pinheiro, and A. Bedding. Adaptive dose-response studies. *Drug Information Journal*, 40:451–461, 2006.

[30] S.N. Goodman, M.L. Zahurak, and S. Piantadosi. Some practical improvements in the continual reassessment method for phase I studies. *Statistics in Medicine*, 11:1377–1389, 1992.

[31] N. Hack and W. Brannath. *AGSDest: Estimation in adaptive group sequential trials*, 2008. R package version 1.0.

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

[33] R.V. Hogg, J.W. McKean, and A.T. Craig. *Introduction to Mathematical Statistics*, page 230. Pearson Prentice Hall, New Jersy, 2004.

[34] J. Husing, W. Sauerwein, K. Hideghety, and K.-H. Jockel. A scheme for a dose-escalation study when the event is lagged. *Statistics in Medicine*, 20:3323 – 3334, 2001.

[35] E.L. Korn, D. Midthune, T.T. Chen, L.V. Rubinstein, M.C. Christian, and R.M. Simon. A comparison of two phase I trial designs. *Statistics in Medicine*, 13:1799–1806, 1994.

[36] National Cancer Institute (NCI). *Common Terminology Criteria for Adverse Events Version 4.0.* Cancer Therapy Evaluation Program, National Cancer Institute, National Institutes of Health, U.S. Department of Health and Human Services, May 2009.

[37] J.R. Nebeker, P. Barach, and M.H. Samore. Calrifying adverse drug events: A clinician's guide to terminology, documentation, and reporting. *Annuals of Internal Medicine*, 140:795 – 801, 2004.

[38] J. O'Quigley. Another look at two phase I clinical trials designs. *Statistics in Medicine*, 18:2683–2690, 1999.

[39] J. O'Quigley and S. Chevert. Methods for dose finding studies in cancer clinical trials: A review and results of a monte carlo study. *Statistics in Medicine*, 10:1647–1664, 1991.

[40] J. O'Quigley, M. Pepe, and L. Fisher. Continual reassessment method: A practical design for phase I clinical trials in cancer. *Biometrics*, 46:33–48, 1990.

[41] J. O'Quigley and L.Z. Shen. Continual reassessment method: A likelihood approach. *Biometrics*, 52:673–684, 1996.

[42] R. Pahl. *GroupSeq: Performing computations related to group sequential designs.*, 2009. R package version 1.3.

[43] S. Piantadosi and G. Liu. Improved designs for dose escalation studies using pharmacokinetic measurements. *Statistics in Medicine*, 15:1605–1618, 1996.

[44] F. Pukelsheim. *Optimal Design of Experiments*, page 308. Wiley, New York, 1993.

[45] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. ISBN 3-900051-07-0.

[46] W.F. Rosenberger and L.M. Haines. Competing designs for phase I clinical trials: A review. *Statistics in Medicine*, 21:2757–2770, 2003.

[47] D.A. Schoenfeld. *seqmon: Sequential Monitoring of Clinical Trials*, 2007. R package version 0.2.

[48] S. Sethi and T.F. Murphy. Bacterial infection in chronic obstructive pulmonary disease in 2000: A state-of-the-art review. *Clinical Mivrobiology Reviews*, 14:336–63, 2001.

[49] M.J. Silvapulle. On the existence of maximum likelihood estimators for the binomial response models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 43:310–313, 1981.

[50] S.D. Silvey. *Optimal Design*. Chapman and Hall, London-New York, 1980.

[51] S.E. Stool and M. J. Field. The impact of otitis media. *Pediatric Infectious Disease Journal*, 8:S11–14, 1989.

[52] B.E. Storer. Design and analysis of phase I clinical trials. *Biometrics*, 45:925–937, 1989.

[53] P.F. Thall and J.D. Cook. Dose-finding based on efficacy-tosicity trade-offs. *Biometrics*, 60:684–693, 2004.

[54] P.F. Thall, J.J. Lee, C.-H. Tseng, and E.H. Estey. Accrual strategies for phase I trials with delayed patient outcome. *Statistics in Medicine*, 18:1155 – 1169, 1999.

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

[56] M. Tighiouart, A. Rogatko, and J.S. Babb. Flexible bayesian methods for cancer phase I clinical trials. dose escalation with overdose control. *Statistics in Medicine*, 24:2183–2196, 2005.

[57] S. Todd. A 25-year review of sequential methodology in clinical studies. *Statistics in Medicine*, 26:237–252, 2007.

[58] J. Whitehead. Bayesian decision procedures with application to dose-finding studies. *International Journal of Pharmaceutical Medicine*, 11:201 – 208, 1997.

[59] J. Whitehead and H. Brunier. Bayesian decision procedures for dose determining experiments. *Statistics in Medicine*, 14:885 – 893, 1995.

[60] J. Whitehead and D. Williamson. Bayesian decision procedures based on logistic regression models for dose-finding studies. *Journal of Biopharmaceutical Statistics*, 8:445 – 467, 1998.

[61] J. Whitehead, Y. Zhou, S. Patterson, D. Webber, and S. Francis. Easy-to-implement Bayesian methods for dose-excalation studies in healthy volunteers. *Biostatistics*, 2:47 – 61, 2001.

[62] J. Whitehead, Y. Zhou, J. Stevens, and G. Blakey. An evaluation of a Bayesian method of dose escalation based in bivariate binary responses. *Journal of Biopharmaceutical Statistics*, 14:969 – 983, 2004.

[63] P.L. Winokur, C. Taylor, H. Hamilton, J. Keefe, and J. Russell. *Nontypeable Haemophilus influenzae Human Colonization in Healthy Young Adults.* The University of Iowa Hospitals and Clinics, 2005. DMID Protocal Number: 04-022.

[64] G.G. Woodworth. *Biostatistics: A Bayesian Introduction*, chapter 1. Wiley, New Jersy, 2004.

[65] C.F.J. Wu. Efficient sequential designs with binary data. *Journal of American Statistical Association*, 80:974–987, 1985.

[66] Y. Yuan and G. Yin. Bayesin dose finding by jointly modelling toxicity and efficacy as time-to-event outcomes. *Journal of the Royal Statistical Soceity: Series C (Applied Statistics)*, 58:719–736, 2009.

[67] S. Zacks, A. Rogatko, and J. Babb. Optimal Bayesian-feasible dose escalation for cancer phase I trials. *Statistics and Probability Letters*, 38:215–220, 1998.