Title

A Metaheuristic Adaptive Cubature Based Algorithm to Find Bayesian Optimal Designs for Nonlinear Models

Journal

Journal of Compuational and Graphical Statistics, 00(0)

Authors

Masoudi, Ehsan
Holling, Heinz
Duarte, Belmiro P.M.
et al.

Publication Date

2019

DOI

10.1080/10618600.2019.1601097

Peer reviewed

UCGS1601097, VOL 00, ISS 0

A Metaheuristic Adaptive Cubature Based Algorithm to Find Bayesian Optimal Designs for Nonlinear Models

Ehsan Masoudi, Heinz Holling, Belmiro P.M. Duarte, and Weng Kee Wong

QUERY SHEET

This page lists questions we have about your paper. The numbers displayed at left are hyperlinked to the location of the query in your paper.

The title and author names are listed on this sheet as they will be published, both on your paper and on the Table of Contents. Please review and ensure the information is correct and advise us if any changes need to be made. In addition, please review your paper as a whole for typographical and essential corrections.

Your PDF proof has been enabled so that you can comment on the proof directly using Adobe Acrobat. For further information on marking corrections using Acrobat, please visit http://journalauthors.tandf.co.uk/production/acrobat.asp; https://authorservices.taylorandfrancis.com/how-to-correct-proofs-with-adobe/

The CrossRef database (www.crossref.org/) has been used to validate the references.

AUTHOR QUERIES

Q1 Please note that there is a mismatch in article title between CATS and manuscript, we have retained the title present in the manuscript. Kindly correct if this is inaccurate.

Q2 Please check whether the author names (first name followed by last name) and affiliations are correct as presented in the proofs.

Q3 Please check whether the designated corresponding author is correct as presented in the proofs.

Q4 As per the journal specification, only 3 to 6 keywords are allowed. Please check and delete the extra keyword.

Q5 The citation "Table (18)" has been changed to "Table 4." Please check and correct if this is inaccurate.

Q6 Please provide an update for references "Lukemire et al. (2018) and Xu et al. (2018)."

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2019, VOL. 00, NO. 0, 1–15
https://doi.org/10.1080/10618600.2019.1601097

A Metaheuristic Adaptive Cubature Based Algorithm to Find Bayesian Optimal Designs for Nonlinear Models

Ehsan Masoudi[a], Heinz Holling[a], Belmiro P. M. Duarte[b,c], and Weng Kee Wong[d]

[a]Department of Psychology, University of Münster, Münster, Germany; [b]Department of Chemical and Biological Engineering, ISEC, Polytechnic Institute of Coimbra, Coimbra, Portugal; [c]CIEPQPF, Department of Chemical Engineering, University of Coimbra, Coimbra, Portugal; [d]Department of Biostatistics, UCLA Fielding School of Public Health, Los Angeles, CA

ABSTRACT

Finding Bayesian optimal designs for nonlinear models is a difficult task because the optimality criterion typically requires us to evaluate complex integrals before we perform a constrained optimization. We propose a hybridized method where we combine an adaptive multidimensional integration algorithm and a metaheuristic algorithm called imperialist competitive algorithm to find Bayesian optimal designs. We apply our numerical method to a few challenging design problems to demonstrate its efficiency. They include finding *D*-optimal designs for an item response model commonly used in education, Bayesian optimal designs for survival models, and Bayesian optimal designs for a four-parameter sigmoid Emax dose response model. Supplementary materials for this article are available online and they contain an R package for implementing the proposed algorithm and codes for reproducing all the results in this paper.

ARTICLE HISTORY
Received January 2018
Revised January 2019

KEYWORDS
Adaptive subregion algorithm; Compound criterion; *D*-efficiency; Equivalence theorem, Fisher information matrix; Optimization; P-optimality

1. Introduction

The cost of running an experiment is rising and optimal design ideas are increasingly used in various disciplines to rein in experimental expenses. Given the objective or objectives of the study, an optimal design determines how to take observations from the experimental region to realize maximum statistical efficiency. Given a statistical model defined on an interval and a design criterion, the model-based optimal design problem is to determine the optimal number of design points to observe the responses, their optimal locations and the optimal number of replications at the design points.

A popular design criterion for estimating model parameters accurately is *D*-optimality. When errors are normally distributed, *D*-optimal designs minimize the volume of the confidence ellipsoid of the model parameters by minimizing the generalized variance. For nonlinear models, the generalized variance depends on the unknown parameters which we want to estimate. The simplest approach is to replace the unknown parameters by some initial estimates from a similar study or a pilot study so that the criterion is free of unknown parameters and can be directly optimized. Such an approach results in optimal designs that depend on the initial estimates or nominal values and are called *locally* optimal designs (Chernoff 1953).

Locally optimal designs are not generally robust to misspecification in the nominal parameters. This means that, if the nominal values are wrongly specified, the resulting locally *D*-optimal design can be inefficient under the true values of the parameters. One method to overcome this drawback is to adopt a minimax approach. A *minimax* optimal design problem specifies a known region for the parameter values and finds a design under the worst-case scenario. For example, Sitter (1992) proposed minimax optimal designs assuming a known parameter space for the unknown parameters and minimized the maximum inefficiency that arises from using the worst set of values for the model parameters. Minimax types of design criteria are complicated and usually require advanced algorithms to solve nested multimodal continuous optimization problems (see, e.g., Masoudi, Holling, and Wong 2017; Chen et al. 2015).

Another approach to quantify the uncertainty in the unknown parameters is to use a prior distribution for the unknown parameters and optimize the criterion averaged over the prior distribution. The resulting optimal design is termed *Bayesian* optimal design (Chaloner and Larntz 1989; Chaloner and Verdinelli 1995; Atkinson 1996). Bayesian optimal designs are difficult to determine because they involve evaluating multiple integrals before a global optimization algorithm can be applied to optimize the averaged criterion, which typically has many variables and constraints to account for.

The purpose of our work is to propose a novel method for finding Bayesian optimal designs by combining a metaheuristic algorithm with a hcubature algorithm to efficiently tackle the optimization problem. The hcubature algorithm is a self-learning algorithm that uses a multivariate subdivision strategy to learn the behavior of the integrand and provide a reliable estimate of the integral. In contrast, a metaheuristic algorithm is a heuristic designed to find or generate a sufficiently good solution to an optimization problem with incomplete or imperfect information. Evolutionary algorithms are metaheuristic algorithms that are increasingly used in artificial intelligence and typically inspired by biological evolution or occurrences in nature. There

CONTACT Ehsan Masoudi ehsan.masoudi@wwu.de Department of Psychology, University of Münster, Fliednerstr. 21, 48149 Münster, Germany.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.
Supplementary materials for this article are available online. Please go to www.tanfonline.com/r/JCGS.

are many such algorithms widely used to solve optimization problems in computer science and engineering (Yang 2010).

Our interest is in one such algorithm called imperialist competitive algorithm (ICA), which is inspired from the sociopolitical process of humans and proposed in Atashpaz-Gargari and Lucas (2007). Our choice of ICA is in part due to our earlier positive experience, where we were successful in our search for hard-to-find minimax and standardized maximin optimal designs for different nonlinear models and where we experienced fewer premature convergence issues (Masoudi, Holling, and Wong 2017). Furthermore, as a state-of-the-art optimization algorithm, ICA: (a) does not require the design space to be discretized, (b) is a multi-start or population-based derivative-free algorithm and usually is not sensitive to the starting designs, and (c) can be easily modified to find other types of optimal designs.

To improve performance of a metaheuristic algorithm, it is a common practice to hybridize it with another metaheuristic algorithm or a numerical procedure so that it captures the advantages of both algorithms. Our main challenge here is to hybridize ICA with another numerical method to find Bayesian optimal designs efficiently. Specifically, we search for one that is capable of handling numerical integration well and which is able to compromise on the speed required to find the sought-after design and optimality of the generated design.

A common method to approximate the integrals in a Bayesian approach is the Monte Carlo (MC) integration method. The MC method uses a sequence of random draws from the prior distribution to approximate the integral and it usually requires a large number of function evaluations to provide accurate estimates of the integrals. Quadrature formulas are alternative deterministic methods that approximate the integrals by a weighted sum of the integrand values at specific points, called *nodes*; for a review, see Goos and Mylona (2018). Although the quadrature techniques are fast, they usually do not provide estimation errors and their accuracy depends on the prior distribution and properties of the integrand. Further, quadrature rules usually need to be extended to solve high-dimensional integrals and sometimes the extension rules can make them inefficient and slow. Quadrature rules in higher dimensions are called *cubature rules*.

To achieve a better trade-off between the speed of the algorithm and the accuracy of the generated designs, we implement an adaptive multidimensional integration method over hypercubes named *hcubature* in ICA. This algorithm studies properties of the integrand using a subdivision strategy and concentrates on the subregions where the shape of the integrand is most irregular. Unlike the traditional quadrature methods, the hcubature algorithm provides an estimation error that is used to build stopping rules to determine if and when to increase the reliability of the approximations. To emphasize the difference between the previous minimax algorithm in Masoudi, Holling, and Wong (2017) and the current version, we call the algorithm proposed here Bayesian imperialist competitive algorithm (BICA). Our work is novel because this is the first study that hybridizes a subregion adaptive algorithm with a metaheuristic algorithm to find Bayesian optimal designs.

In the next section, we review the statistical setup and theory for finding optimal designs. Section 3 provides a general description of the hcubature algorithm and Section 4 presents details and implementation information for BICA. In Section 4.1, we apply our algorithm to find Bayesian D-optimal designs for test-item calibration using the two-parameter logistic (2PL) model frequently used in item response theory (IRT). We use different prior distributions for the item parameters and show that BICA can find the optimal designs efficiently. In Section 4.2, we apply BICA to find Bayesian D-optimal designs for a four-parameter sigmoid Emax model where we have to evaluate a four-dimensional integral and the design space is large. Section 4.3 presents Bayesian optimal designs for the Cox proportional hazards model in survival analysis. We also investigate the performance of BICA for a compound criterion called DP-criterion in Section 4.4. Section 5 provides a sensitivity analysis of BICA with respect to the tuning parameters of the hcubature algorithm. In Section 6, we compare BICA with recent algorithms for finding Bayesian optimal designs and provide details in the supplementary materials, which also contains R codes for implementing BICA and generating all the results in our paper. Section 7 concludes with a discussion.

2. Bayesian Optimal Design

Following Kiefer (1985), an approximate design is a probability measure defined on a user-selected design space χ. Let Ξ be the space of all such designs on χ and let ξ be an approximate design with k support points at $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$ from χ with corresponding weights $w_1, \ldots, w_k$, $w_i > 0$, $\sum_{i=1}^{k} w_i = 1$. Given a predetermined total number of observations for the study, say N, we implement ξ by taking Nw_i number of observations at $\boldsymbol{x}_i$ subject to $Nw_1 + \cdots + Nw_k = N$ and each Nw_i is an integer. Pukelsheim and Rieder (1992) provide details for various rounding procedures.

Let Y be a univariate response variable and $\boldsymbol{x}$ be the set of covariates in the model. Let $E(Y) = f(\boldsymbol{x}, \boldsymbol{\theta})$, where f is a known function and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_q)^T$ is the vector of unknown parameters. Assuming errors are independent and normally distributed with mean zero, we first calculate its Fisher information matrix (FIM), which measures the worth of the design. If ξ has k support points, its FIM is proportional to

$$M(\xi, \boldsymbol{\theta}) = \sum_{i=1}^{k} w_i I(\boldsymbol{x}_i, \boldsymbol{\theta}), \tag{1}$$

where

$$I(\boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{1}{\text{var}(Y_i)} \nabla f(\boldsymbol{x}_i, \boldsymbol{\theta}) \nabla f(\boldsymbol{x}_i, \boldsymbol{\theta})^T,$$

and $\nabla f(\boldsymbol{x}_i, \boldsymbol{\theta})^T = \left(\frac{\partial f(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \theta_1}, \frac{\partial f(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \theta_2}, \ldots, \frac{\partial f(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \theta_q} \right)$. The FIM depends on $\boldsymbol{\theta}$ for nonlinear models and it does not for linear models; in the latter case, we denote it by $M(\xi)$. We note that the FIM is singular if $k < q$ and for this reason, our work here assumes $k \geq q$.

We formulate a design optimality criterion, ψ, as a convex function of the FIM, and a design that minimizes ψ is a ψ-optimal design. This means that we have to determine the optimal number of support points, k, the optimal support

points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ and their corresponding $w_1, \ldots, w_k$. The minimization is among all designs defined on $\Xi = \chi^k \times [0,1]^k$, where the symbol $\times$ is the Cartesian product. An example is D-optimality for linear models defined by $\psi(\xi) = -\log|M(\xi)|$ and $|.|$ denotes determinant. It is probably the most widely used criterion for estimating the model parameters and it can be shown to be a convex function of FIM (Kiefer 1985). When the model is nonlinear, the dependence of the FIM on $\boldsymbol{\theta}$ may be averaged out by a prior distribution. The Bayesian D-optimality criterion is

$$\psi_D(\xi;\pi) = \int_{\boldsymbol{\theta}\in\Theta} -\log|M(\xi,\boldsymbol{\theta})|\pi(\boldsymbol{\theta})d\boldsymbol{\theta}, \qquad (2)$$

where $\pi(\boldsymbol{\theta})$ is a user-selected prior distribution for the unknown parameters $\boldsymbol{\theta}$ and Θ is the set of all possible values of θ. It can be shown the criterion is convex as a function of the FIM (Firth and Hinde 1997) and a Bayesian D-optimal design is a design that minimizes (2) over Ξ.

One advantage of working with approximate designs is that we can use an equivalence theorem to verify the optimality of a given design if the criterion is a convex function of the FIM. For example, for Bayesian D-optimality, the equivalence theorem states that ξ_D^* is D-optimal if and only if

$$c_D(\boldsymbol{x},\xi_D^*;\pi) = \int_\Theta \mathrm{tr}\{M^{-1}(\xi_D^*,\boldsymbol{\theta})I(\boldsymbol{x},\boldsymbol{\theta})\}\pi(\boldsymbol{\theta})d\theta - q \le 0, \quad \forall \boldsymbol{x}\in\chi \qquad (3)$$

with equality in (3) at all support points of ξ_D^* (Kiefer and Wolfowitz 1959; Chaloner and Larntz 1989). The equivalence theorem is obtained by considering the directional derivative of the convex functional at the optimum in the direction of $\xi_{\boldsymbol{x}}$, the degenerate design at $\boldsymbol{x}$, that is, the design that puts all its mass at $\boldsymbol{x}$. For this reason, the function $c_D(\boldsymbol{x},\xi_D^*;\pi)$ is sometimes called the derivative function. When χ is one- or two-dimensional, one may plot $c_D(\boldsymbol{x},\xi_D^*;\pi)$ versus $\boldsymbol{x}\in\chi$ and visually inspect whether the graph meets the conditions in the equivalence theorem. If it does, the design ξ_D^* is Bayesian optimal; otherwise, it is not. All of the BICA-generated designs in this paper have been validated by the equivalence theorem.

We measure the closeness of a design ξ to the Bayesian D-optimal design ξ_D^* by its Bayesian D-efficiency defined by

$$\mathrm{eff}_D(\xi,\xi_D^*;\pi) = \exp\left(\frac{\psi_D(\xi_D^*;\pi) - \psi_D(\xi;\pi)}{q}\right). \qquad (4)$$

Using argument similar to Atwood (1969), it is straightforward to obtain a Bayesian D-efficiency lower bound (ELB) for any design ξ without knowing ξ_D^*. This ELB is $q/(q + \max_{\boldsymbol{x}\in\chi} c_D(\boldsymbol{x},\xi;\pi))$ and its value may be used to measure proximity of the generated design ξ to the optimum without knowing the latter.

3. A Short Description of the hcubature Algorithm

The basic version of the hcubature algorithm was introduced in Van Dooren and de Ridder (1976) and named HALF. Genz and Malik (1980) replaced the basic rule of HALF with a symmetric multidimensional polynomial degree 7 rule. Berntsen, Espelid, and Genz (1991a) proposed another version of the algorithm called DCUHRE (Berntsen, Espelid, and Genz 1991b) that allows parallelism at the lowest level during the integrand evaluations and at the subregion level. A free/open-source version of the algorithm is available in the C package cubature under the name hcubature (Johnson 2013). In what is to follow, we present in Algorithm 1, a pseudo-code of the hcubature algorithm and explain the general structure of the hcubature algorithm for scalar integrands (nonparallel version), assuming that the integration region is $\Theta \in \mathbb{R}^q$, and q is the number of unknown model parameters.

The algorithm begins with the original integration region Θ and dynamically subdivides it into finer and finer subregions, with smaller subregions concentrated at where the integrand is most irregular. At stage i, the subdivision S_i of Θ is defined as a set of disjoint subregions $S_i = \{R_{1_i}, R_{2_i}, \ldots, R_{M_i}\}$ with $\bigcup R_j = \Theta$ and $S_0 = \Theta$, where M_i is the total number of subregions at stage i. The hcubature algorithm applies a degree 7 Genz-Malik's rule to each subregion to estimate the local integrals. The local error at each subregion is estimated by the absolute difference between the degree 7 Genz-Malik's rule and an embedded degree 5 rule (Genz and Malik 1980). The nodes of the degree 5 Genz-Malik's rule belong to a subset of the nodes produced by the 7 degree rule. Therefore, no additional integrand evaluations are necessary to estimate the errors. The global estimate of the integral $\hat{I}_g$ and error $\hat{E}_g$ are equal to the sum of the estimated local integrals and errors over all the subregions. Let `tol` denote the user-specified tolerance and let `maxEval` denote the maximum number of function evaluations. The algorithm stops if $\mathrm{abs}(\hat{E}_g) < \mathrm{abs}(\hat{I}_g)$.`tol` or the total number of function evaluations is (approximately) larger than `maxEval`. Here, "abs" is the absolute function and "." is the scalar multiplication. If neither of the stopping conditions is met, to create subdivision S_{i+1}, the algorithm chooses the subregion with the largest error estimate and subdivides it in half along a coordinate axis l, where the integrand has the largest local absolute fourth difference along it. Similar to the local error estimates, calculating the local absolute fourth difference does not require extra integrand evaluations to be performed and is already embedded in the degree 7 Genz-Malik's rule.

The hcubature algorithm can produce a very accurate estimate of the integral and is best suited for a moderate number of dimensions, say, less than 7, which is the case for many problems. For more details, see Genz and Bretz (2009) and Evans and Swartz (2000). In this paper, we incorporate the hcubature algorithm into BICA using the R (R Core Team 2018) package `cubature` (Narasimhan and Johnson 2017) to approximate all the integrals. To produce accurate results in our examples, we set the values of `maxEval` and `tol` equal to 50,000 and 10^{-5}, respectively. Since the equivalence theorem is only used to verify the optimality of the generated design, the associated values of the hcubature tuning parameters for approximating the derivative function do not influence the convergence speed of the algorithm. Our reported CPU times do not include time for verifying optimality of a design by plotting the derivative function or calculating the ELB.

Algorithm 1 Pseudo-code for the hcubature algorithm (nonparallel version)

1: **Input** `tol`, `maxEval`, Θ
2: **Initialization** Set $S_0 = \Theta$ and apply the Genz-Malik's rule, returning $\hat{I}_g$, $\hat{E}_g$.
3: **while** abs($\hat{E}_g$) $\geq$ abs($\hat{I}_g$)`.tol` or total number of function evaluations <= `maxEval` **do**
4: Choose the subregion with largest estimated error from the collection of subregions.
5: Select the coordinate axis l where the integrand has the largest local absolute fourth difference along it.
6: Divide the chosen subregion in half along axis l and create two new subregions.
7: Apply the Genz-Malik's rule to each new resulted subregion.
8: Return local integral estimates and local error estimates from each subregion.
9: Update $\hat{I}_g$ and $\hat{E}_g$ by aggregating the local integral estimates and error estimates.
10: **end while**
11: **Output** $\hat{I}_g$ and $\hat{E}_g$.

4. ICA for Bayesian Optimal Design Problems

The structure of BICA is similar to the minimax version introduced in Masoudi, Holling, and Wong (2017). The major difference lies in the evaluation of the cost function or, equivalently, the optimality criterion. The minimax ICA is a nested optimization algorithm and calls another optimization problem over the parameter space to evaluate the cost function, while the Bayesian version runs the hcubature algorithm to approximate the integrals over the parameter prior distributions. Therefore, the cost value of each solution or design in BICA is equivalent to the approximated value of the Bayesian criterion.

Similar to the ICA algorithm, BICA requires the user to select the number of support points, k, in advance. In practice, one can start with $k = q$ and incrementally increase this value until the equivalence theorem verifies the generated k-point design as the global optimal design. After choosing k, the algorithm starts with a random population of solutions, named *countries*, and N_{count} is the number of countries. In optimal design problems, the position of a country is identical to the location of the support points and the associated weights of a design. The population of countries is divided into some subpopulations called *empires*. Each empire consists of one *imperialist* and the remaining countries form the *colonies*. The number of colonies of an empire is proportional to the *power* of the associated imperialist, which in turn is a function of its cost value. In this scheme, the more powerful imperialists receive a larger number of colonies.

After initialization step, BICA enters the evolution stage. The most influential steps that govern the evolution of the algorithm are the *assimilation* within each empire and the *competition* among the empires. In the assimilation step, the colonies of each empire improve their power by moving toward their relevant imperialist. During the competition, powerful empires take possession of the colonies of the weaker empires. For more details, see Atashpaz-Gargari and Lucas (2007) and Masoudi, Holling, and Wong (2017).

BICA inherits the tuning parameters of ICA and the hcubature algorithm. According to Masoudi, Holling, and Wong (2017), the most influential ICA tuning parameter is the number of countries N_{count}. Its value should be set according to the number of support points k and number of model covariates n. Larger k and n mean a larger number of decision variables in the optimization problem and, as a result, BICA requires more number of countries to be able to find the optimal design. Our numerical results suggest that a value between 40 and 300 for N_{count} usually results in a (nearly) optimal design. The first strategy when encountering a premature convergence is to increase the value of N_{count}. For more discussion about the ICA tuning parameters, see Masoudi, Holling, and Wong (2017). We delay the sensitivity analysis of BICA with respect to the hcubature tuning parameters until Section 5.

In what is to follow, we apply BICA to find Bayesian optimal designs for different types of problems. Except for the results in Section 6, all the BICA-generated designs in this paper were constructed using a Macintosh machine with a Core i5 processor running macOS version 10.12.6. BICA was written in R version 3.5.2.

4.1. Bayesian D-Optimal Designs for Test-Item Calibration

The 2PL model is frequently used in IRT to describe the probability of an examinee answering a particular item correctly, given the examinee's ability level and the item parameters. The 2PL model is given by

$$E(Y) = P(Y = 1) = \frac{1}{1 + \exp(-b(x - a))}, \tag{5}$$

where a and b are, respectively, the difficulty and discrimination item parameters, and the variable x is the examinee's ability known as the person parameter. In practice, $\boldsymbol{\theta} = (a, b)^T$ belongs to $\Theta = [a^L, a^U] \times [b^L, b^U]$, where a^L, a^U, b^L, b^U are the prespecified lower and upper bounds for a and b. The estimation of the item parameters a and b is known as *item calibration*. In *optimal calibration designs* the purpose is to find the best ability levels over a prespecified space of abilities χ (design interval) to obtain efficient item parameter estimates (Berger and Wong 2009, chap. 5). A common design interval in IRT is $\chi = [-3, 3]$, where small values denote examinees with lower ability levels to solve the item and large values denote examinees with higher ability levels. Similar to Lu (2014), Passos and Berger (2004), and Berger, King, and Wong (2000), we assume that the experimenter is able to find examinees whose abilities match the ability levels of the optimal design.

One application of the 2PL model is in computerized adaptive testing (CAT), which is a computer-based test that tailors the items to the examinee's ability. CAT requires a large collection of calibrated items known as *item pool* or *item bank* and it is heavily dependent on the efficient prior estimation of the item parameters. Thus, using optimal calibration designs is necessary to achieve the best efficiency. Test administrations typically calibrate multiple items at the same time. A typical assumption in IRT is that the dependency among responses for each pair of items of a test for a fixed ability level is zero.

This well-known assumption in IRT is called *stochastic local independence* (Fischer 2004; Rost 2001). Therefore, the total information for a whole test with m items is a block-diagonal matrix $\mathcal{M}(\xi, \tilde{\boldsymbol{\theta}})$, with main diagonal design matrices $M_i(\xi, \boldsymbol{\theta}_i)$, where $\boldsymbol{\theta}_i = (a_i, b_i)^T, i = 1, \ldots, m$ and $\tilde{\boldsymbol{\theta}} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_m)^T$. The Bayesian D-criterion based on the FIM for the m items is

$$\int_{\tilde{\Theta}} -\log|\mathcal{M}(\xi, \tilde{\boldsymbol{\theta}})|\pi(\tilde{\boldsymbol{\theta}})d\tilde{\boldsymbol{\theta}} = \int_{\tilde{\Theta}} \left(-\sum_{i=1}^{m} \log|M_i(\xi, \boldsymbol{\theta}_i)|\right)\pi(\tilde{\boldsymbol{\theta}})d\tilde{\boldsymbol{\theta}}, \quad (6)$$

where $\tilde{\Theta} = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_m, \pi(\tilde{\boldsymbol{\theta}}) = \pi_1(\boldsymbol{\theta}_1) \times \pi_2(\boldsymbol{\theta}_2) \times \cdots \times \pi_m(\boldsymbol{\theta}_m)$ and $\pi_i(\boldsymbol{\theta}_i)$ is the prior distribution for $\boldsymbol{\theta}_i$ on Θ_i. Clearly, even a moderate number of items requires a large number of integrals over the items prior distributions. In practice, $\pi_i(\boldsymbol{\theta}_i)$ and Θ_i are chosen to be the same for all m items, and (6) simplifies to

$$-m\int_{\Theta_i} \log|M_i(\xi, \boldsymbol{\theta}_i)|\pi_i(\boldsymbol{\theta}_i)\mathrm{d}\boldsymbol{\theta}_i, \quad (7)$$

where only a two-dimensional integral must be approximated. This means that the optimal design obtained for one item is also optimal for the rest of the items under a common prior distribution for the item parameters. The shape of the common prior distribution may be specified in a way that reflects the variation among the item parameters for different items of a test. For example, an uncertain prior distribution for the item difficulty is appropriate when a test includes both easy and difficult items.

We now apply BICA to find Bayesian D-optimal designs for the 2PL model assuming one item and different prior distributions on $\Theta = [-3, 3] \times [0.1, 2]$. The prior distributions of interest come from Blum et al. (2016) and they are:

1. Independent uniform priors with $a \sim U(-3, 3)$ and $b \sim U(0.1, 2)$. We denote the prior distributions and the BICA-generated design by π_{uni} and ξ_{uni}, respectively.
2. Two normal prior distributions $\pi_{\text{norm1}} = \mathcal{N}_\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1)$ and $\pi_{\text{norm2}} = \mathcal{N}_\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma}_2)$, where
$$\boldsymbol{\mu} = (0, 1)^T, \ \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1.00 & -0.17 \\ -0.17 & 0.50 \end{bmatrix}, \text{ and } \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1.00 & 0 \\ 0 & 0.50 \end{bmatrix}. \quad (8)$$
We denote the BICA-generated designs by ξ_{norm1} and ξ_{norm2}, respectively, and also use the same mean $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_1$ in the specification of the next two sets of prior distributions.
3. A bivariate t prior distribution $\pi_t = t_\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1, df)$ where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_1$ are, respectively, the mean vector and the covariance matrix for $\boldsymbol{\theta} = (a, b)^T$ and df is the number of degrees of freedom. Following Dunnett and Sobel (1954), we set $df = 3$ and denote the BICA-generated design by ξ_t.
4. Two bivariate skewed normal prior distributions denoted by $\pi_{\text{skew1}} = \mathcal{SN}_\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1, \boldsymbol{\alpha}_1)$ and $\pi_{\text{skew2}} = \mathcal{SN}_\Theta(\boldsymbol{\mu}, \boldsymbol{\Sigma}_1, \boldsymbol{\alpha}_2)$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_1$ are the location parameters and the scale matrix for $\boldsymbol{\theta} = (a, b)^T$, respectively. The vectors of skewness factors are $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ and they are set to $\boldsymbol{\alpha}_1 = (1, 0)^T$ and $\boldsymbol{\alpha}_2 = (-1, 0)^T$, respectively. If the experimenter believes the item is difficult prior to data collection, choosing $\boldsymbol{\alpha}_1$ is appropriate. On the other hand, if the experimenter feels that the item is more likely to be easy, $\boldsymbol{\alpha}_2$ is a more appropriate choice. We denote the BICA-generated designs by ξ_{skew1} and ξ_{skew2}, respectively, and note that such prior distributions were considered in Azzalini and Capitanio (1999).

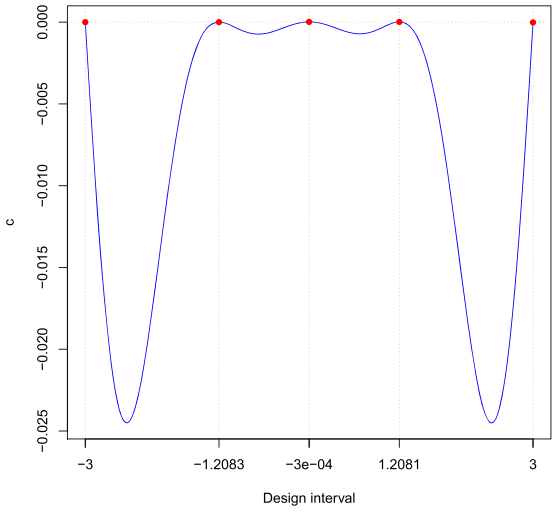
Table 1. BICA-generated designs ξ for the 2PL model for different prior distributions for $\boldsymbol{\theta} = (a, b)^T$ when $\Theta = \in [-3, 3] \times [0.1, 2]$ and $x \in [-3, 3]$.

ξ	ξ_{uni}	ξ_t	ξ_{norm1}	ξ_{norm2}
	−3.000(0.247)	−2.417(0.263)	−2.255(0.318)	−2.230(0.314)
	−1.208(0.183)	−1.167(0.182)	−0.763(0.182)	−0.670(0.186)
	0.000(0.140)	0.044(0.142)	0.546(0.182)	0.672(0.186)
	1.208(0.183)	1.185(0.168)	2.160(0.319)	2.231(0.314)
	3.000(0.247)	2.406(0.244)		
π	π_{uni}	π_t	π_{norm1}	π_{norm2}
$\psi_D(\xi; \pi)$	3.931	2.567	2.968	2.972
$\text{eff}_D(\xi_{\text{uni}}, \xi; \pi)$	1.000	0.985	0.979	0.981
CPU(s)	354	4788	861	719
ξ	ξ_{skew1}	ξ_{skew2}		
	−1.512(0.377)	−2.509(0.358)		
	0.120(0.141)	−1.168(0.110)		
	1.053(0.127)	−0.369(0.156)		
	2.597(0.356)	1.292(0.376)		
π	π_{skew1}	π_{skew2}		
$\psi_D(\xi; \pi)$	2.925	2.935		
$\text{eff}_D(\xi_{\text{uni}}, \xi; \pi)$	0.970	0.952		
CPU(s)	835	1273		

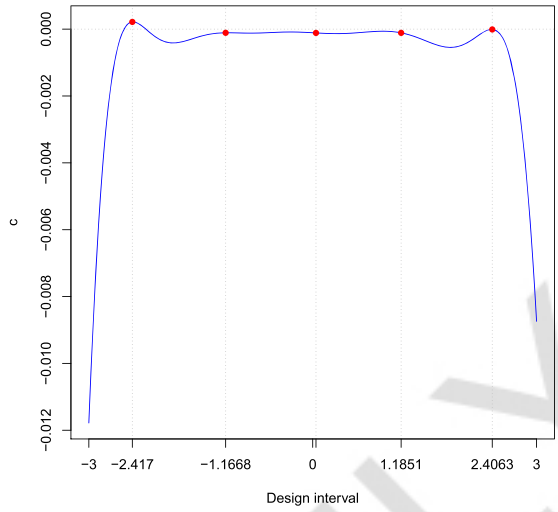
Table 1 displays BICA-generated designs after running our algorithm 1000 iterations. In this and subsequent tables, a design is presented in one column where each of the support points is listed one on top of the other with its corresponding weight in parentheses. The value of N_{count} for the independent uniform and both the bivariate normal prior distributions was set to 40, for the bivariate skewed normal priors it was set to 80 and for the bivariate t distribution, it was increased to 200.

A direct calculation shows that the value of the ELB of each of the BICA-generated design is at least 0.999 and so all of them are optimal for all practical purposes. Figure 1 displays the plots of the derivative functions $c_D(x, \xi; \pi)$ on the design space $[-3, 3]$ for some BICA-generated designs ξ from Table 1. The values of all derivative functions are equal or less than zero on the design space and equal to zero at the support points of the BICA-generated designs. Based on the equivalence theorem, this confirms the Bayesian D-optimality of the obtained designs.

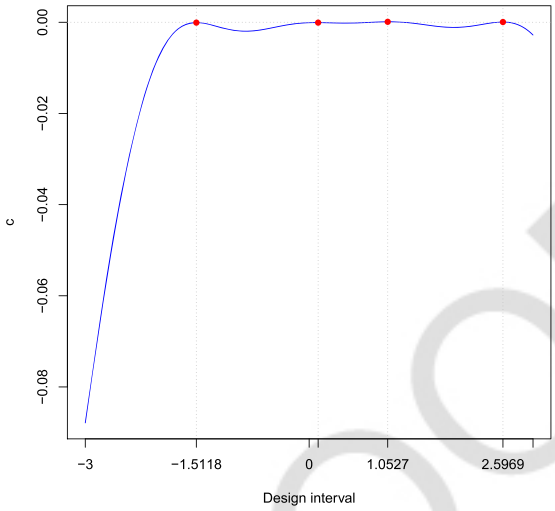
We observe from Table 1 that ξ_{uni} has 5 support points with nearly 50% of its weight at the endpoints of the design interval. In practice, it is unlikely that we have this distribution of test-takers in an examinee sample, having extreme abilities at the endpoints of χ with very competent or incompetent examinees. The designs ξ_{norm1} and ξ_{norm2} have one less support point than ξ_{uni} because the assumed bivariate normal prior distributions provide more information about the locations of the item parameters and reduce the uncertainty about the parameters. Compared to ξ_{uni}, the support points of ξ_{norm1} and ξ_{norm2} are closer to each other, which implies that there is no need to have examinees with extremely high and low ability levels. The design ξ_t has more support points than the design ξ_{norm1}, partly because the assumed bivariate t distribution increases



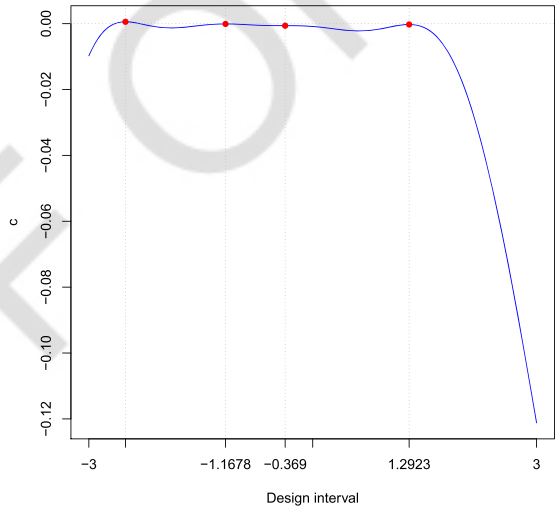
(a) $\xi = \xi_{uni}, \pi = \pi_{uni}$



(b) $\xi = \xi_t, \pi = \pi_t$



(c) $\xi = \xi_{skew1}, \pi = \pi_{skew1}$



(d) $\xi = \xi_{skew2}, \pi = \pi_{skew2}$

Figure 1. Plots of the nonpositive derivative functions $c_D(x, \xi; \pi)$ of the BICA-generated designs ξ for selected prior distributions π from Table 1 on the design space $[-3, 3]$.

the uncertainty about the locations of the item parameters by having more mass on the tails. The designs ξ_{skew1} and ξ_{skew2} have the same number of support points as ξ_{norm1}, but they differ in locations and the weight distributions. The design ξ_{skew1} requires examinees with higher ability levels when the item was assumed to be difficult and the design ξ_{skew2} requires examinees with lower ability levels when the item was assumed to be easy.

Our work suggests that BICA requires the shortest CPU time to find ξ_{uni} because, unlike other prior distributions studied here, no function evaluation is required to calculate the probability density of the uniform distribution. Uniform prior distribution generally allows more uncertainty on the unknown parameters and results in a larger number of support points, therefore, a larger number of decision variables in the optimization problem. We also calculated Bayesian D-efficiency of ξ_{uni} relative to each ξ from Table 1, and $\text{eff}_D(\xi_{\text{uni}}, \xi; \pi)$ in the same table shows the results. Its values suggest that in our design setting, ξ_{uni} is very efficient under other prior distributions.

The above examples show that BICA is a flexible algorithm and can be applied for different prior distributions. To fix ideas, we assume independent uniform prior distributions for the unknown parameters in the models for the rest of the examples in this paper. In what follows, we investigate situations where Bayesian optimal designs have more design points and how the uncertainty of the prior information for each parameter affects the number of support points of the optimal design.

4.2. Sigmoid Emax Model

The sigmoid Emax model belongs to the class of nonlinear sigmoid models commonly used in pharmacokinetics to describe

Table 2. BICA-generated designs ξ for the sigmoid Emax model when $\boldsymbol{\theta}$ has independent uniform prior distributions on $\pi_{\Theta_i}, i = 1, 2, 3, 4$ and $\chi = [0.001, 500]$.

ξ	ξ_{Θ_1}	ξ_{Θ_2}	ξ_{Θ_3}	ξ_{Θ_4}
				0.018(0.205)
	0.003(0.250)	0.089(0.250)	0.362(0.243)	96.433(0.112)
	84.751(0.250)	90.194(0.250)	94.593(0.194)	117.319(0.102)
	123.833(0.250)	127.843(0.250)	113.665(0.116)	133.068(0.100)
	500.000(0.250)	500.000(0.250)	138.300(0.203)	152.844(0.130)
			500.000(0.244)	188.081(0.139)
				500.000(0.212)
π	π_{Θ_1}	π_{Θ_2}	π_{Θ_3}	π_{Θ_4}
$\psi_D(\xi, \pi_\Theta)$	12.159	12.325	12.724	13.332
$\text{eff}_D(\xi, \xi_{\Theta_4}, \pi_{\Theta_4})$	0.364	0.451	0.680	1.000
N_{count}	40	40	40	300
CPU(s)	18	18	243	21654

the dose-response S-shape curves (Macdougall 2006). The mean of the response variable Y at a given dose x is given by

$$E(Y) = f(x, \boldsymbol{\theta}) = \theta_1 + (\theta_2 - \theta_1)\frac{x^{\theta_4}}{x^{\theta_4} + \theta_3^{\theta_4}}, \tag{9}$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)^T$, $\theta_2 > \theta_1$, $\theta_3 > 0$, $x \in \chi = (0, x_0]$ and x_0 (in mg) is user-selected. All errors are assumed to be independent and normally distributed with mean zero and constant variance. Here, θ_1 is the minimum mean response, θ_2 is the maximum mean response, θ_3 is the ED50, that is, the dose at which 50% of the maximum mean effect is achieved, and θ_4 is the slope parameter.

Dragalin, Hsuan, and Padmanabhan (2007) and Wang and Yang (2014) introduced adaptive D-optimal designs for the sigmoid Emax model. Roughly, such procedures find an adaptive design by first estimating the unknown parameters from an initial design in the first stage. The parameter estimates will then be used in the next stage to find the locally optimal design. The process is repeated until some user-specified stopping conditions are met. Obviously, replicating the experiment at each stage is likely going to take a considerable amount of time and effort. Further, the efficiency of adaptive designs usually depends on the sample size and the quality of the initial design.

Bayesian optimal designs avoid the necessity of having just a single set of best initial estimates for the parameters. However, finding them is a challenge here because a four-dimensional integral must first be solved to evaluate the D-optimality criterion. Let $\theta_i \sim U(\theta_i^L, \theta_i^U), i = 1, 2, 3, 4$ and all the uniform prior distributions be independent. For simplicity, we denote the independent uniform distributions for $\theta_i, i = 1, 2, 3, 4$ by π_Θ, where $\Theta = [\theta_1^L, \theta_1^U] \times [\theta_2^L, \theta_3^U] \times [\theta_3^L, \theta_3^U] \times [\theta_4^L, \theta_4^U]$ is the parameter region. To gain insights into the properties of the Bayesian D-optimal design when the spread in π_Θ varies, we assume different parameter regions as follows: $\Theta_1 = [4, 5] \times [11, 12] \times [100, 105] \times [5, 6]$, $\Theta_2 = [4, 6] \times [11, 13] \times [100, 115] \times [5, 7]$, $\Theta_3 = [4, 8] \times [11, 15] \times [100, 130] \times [5, 9]$, and $\Theta_4 = [4, 10] \times [11, 18] \times [100, 180] \times [5, 11]$, where the symbol $\times$ is the Cartesian product. We applied BICA to find Bayesian D-optimal designs ξ_{Θ_i} when the prior distribution is $\pi_{\Theta_i}, i = 1, 2, 3, 4$. Table 2 reports the results when the design interval is $\chi = [0.001, 500]$.

A direct calculation shows that the value of the ELB of each of the BICA-generated designs is approximately 1. Figure 2 displays the plots of the derivative functions on the design space $[0.001, 500]$ for the BICA-generated designs from Table 2 and they confirm the Bayesian D-optimality of the obtained designs.

We observe that as the uniform prior distributions become more diffuse, the number of support points of BICA-generated designs increases and so do the CPU times, especially when $\pi = \pi_{\Theta_4}$. This is because the number of countries was increased to 300 to find the sought-after design accurately under this prior distribution. Furthermore, the hcubature algorithm requires subdividing a larger number of subregions to estimate the integrals when the integrand domain is larger.

To investigate the efficiency loss due to misspecification of the prior distribution in this scenario, we assume the true prior distribution is π_{Θ_4}, but the Bayesian D-optimal design is found under a more informative prior distribution, say π_{Θ_1}, π_{Θ_2}, or π_{Θ_3}. We calculated the Bayesian D-efficiencies (4) of designs ξ_{Θ_1}, ξ_{Θ_2}, and ξ_{Θ_3} relative to ξ_{Θ_4}, when the prior distribution is π_{Θ_4}, and denote them in Table 2 by $\text{eff}_D(\xi, \xi_{\Theta_4}, \pi_{\Theta_4})$ when $\xi = \xi_{\Theta_i}, i = 1, 2, 3$. Their values suggest that, at least in our scenario, the Bayesian D-optimal designs for the sigmoid Emax model are not robust with respect to the misspecification of the prior. For example, ξ_{Θ_1} has only around 36% of the efficiency of ξ_{Θ_4} when the prior distribution is π_{Θ_4}.

4.3. Cox Proportional-Hazards Model

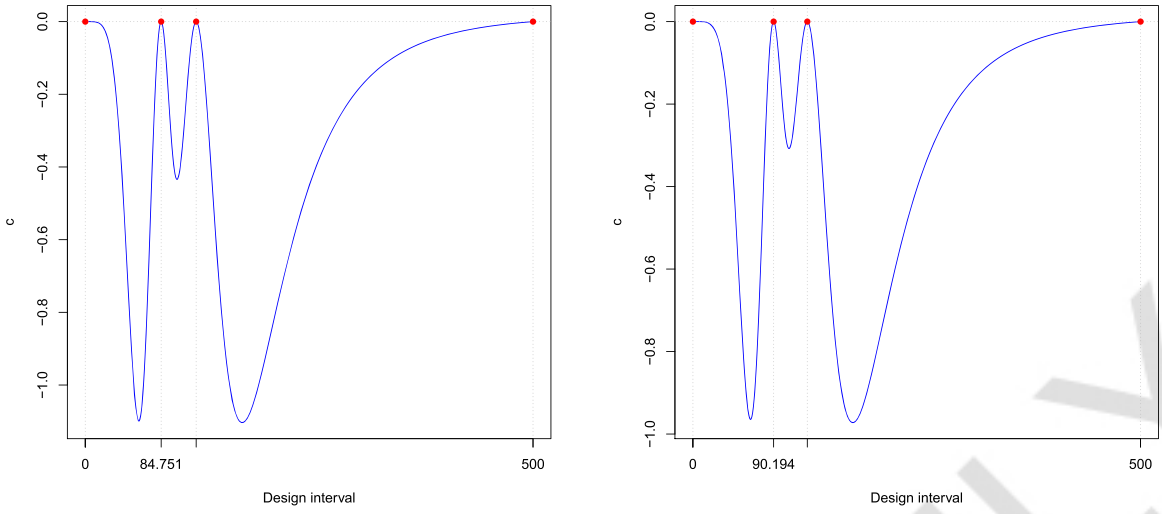
A popular model for analyzing survival data is the Cox proportional hazards model. The model relates the constant hazard function $\lambda(t; \boldsymbol{x}_i)$ for the ith subject at a given time t to the vector of covariates $\boldsymbol{x}_i$. We investigate the proportional hazards model with a constant baseline hazard function that is $\lambda_0(t) = \lambda = \exp(\beta_0)$ when all covariates are zero. Then, we obtain the model

$$\lambda(t, \boldsymbol{x}_i) = \exp(\boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\beta}), \tag{10}$$

where $\lambda(t, \boldsymbol{x}_i)$ is the constant hazard function for the ith individual given $\boldsymbol{x}_i \in \chi$, $\boldsymbol{g} = (1, g_1, g_2, \ldots, g_{q-1})$ is a q-dimensional vector of some known regression functions of the covariates and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{q-1})^T \in \mathbb{R}^q$. It follows that the survival time Y_i is exponentially distributed

$$Y_i \sim \text{Exp}\{\exp[\boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\beta}]\}. \tag{11}$$

Suppose the duration of the study is the interval $[0, c]$ where c is known. In type one censoring, all the individuals enter the experiment at the same time and stay until either the censoring time c or failure, whichever is earlier (right censored data). In



(a) $\xi = \xi_{\Theta_1}, \pi = \pi_{\Theta_1}$ (b) $\xi = \xi_{\Theta_2}, \pi = \pi_{\Theta_2}$

(c) $\xi = \xi_{\Theta_3}, \pi = \pi_{\Theta_3}$ (d) $\xi = \xi_{\Theta_4}, \pi = \pi_{\Theta_4}$

Figure 2. Plots of the nonpositive derivative functions $c_D(x, \xi; \pi)$ of the BICA-generated designs ξ from Table 2 on the design space [0.001, 500] using different prior distributions.

random censoring, the individuals join the experiment at random times before c. Because the procedure of finding optimal designs for the random censoring model is very similar to the type one censoring, we only focus on the latter.

A direct calculation shows that the FIM of a k-point design ξ for the type one censoring has the form (Schmidt and Schwabe 2014)

$$M(\xi, \boldsymbol{\beta}) = \sum_{i=1}^{k} w_i \left\{1 - \exp\left[-c \exp(\boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\beta})\right]\right\} \boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{g}(\boldsymbol{x}_i). \tag{12}$$

The FIM depends on the unknown regression parameters and some locally and standardized maximin optimal designs have been found for this model. Qiu et al. (2014) applied particle swarm optimization (PSO) to find locally optimal designs for the two-parameter model. Schmidt and Schwabe (2014) obtained the locally D-optimal designs for the two- and three-parameter models. Konstantinou, Biedermann, and Kimber (2014) assumed a plausible region for the unknown parameters and found standardized maximin D-optimal designs within the class of two-point designs for the two-parameter model. We now apply BICA to generate Bayesian optimal designs for the above model for two cases: (1) a two-parameter model when $\boldsymbol{g} = (1, x)$ and $\boldsymbol{\beta} = (\beta_0, \beta_1)$ and (2) a three-parameter model when $\boldsymbol{g} = (1, x, x^2)$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)$. We suppose the prior distributions on the three parameters $\beta_0, \beta_1, \beta_2$ are independent uniform densities over the interval $[-d, d]$, $x \in \chi = [0, 1]$ and $c = 30$ in (12).

To investigate the influence of the diffuseness of the uniform prior distributions on the number of support points of the Bayesian optimal design, we set $d = 3, 11$, and denote the resulting independent uniform distributions by π_d and the

Table 3. BICA-generated designs ξ_d for the two- and three-parameter proportional hazards models on $\chi = [0, 1]$ with $c = 30$ in (12).

	Two-parameter model		Three-parameter model	
d	3	11	3	11
ξ_d	0.000(0.500) 1.000(0.500)	0.000(0.405) 0.307(0.173) 0.553(0.014) 1.000(0.407)	0.000(0.333) 0.490(0.333) 1.000(0.333)	0.000(0.303) 0.188(0.102) 0.453(0.197) 0.645(0.129) 1.000(0.269)
$\psi_D(\xi_d, \pi_d)$	1.529	4.705	6.372	11.366
eff_D	0.995	0.721	0.966	0.552
CPU(s)	29	272	407	9847

NOTE: The two sets of independent uniform prior distributions π_d for the parameters are given by $\beta_0 \sim U(-d, d)$, $\beta_1 \sim U(-d, d)$, and $\beta_2 \sim U(-d, d)$ with $d = 3$ and $d = 11$.

BICA-generated design by ξ_d. Table 3 reports the designs ξ_d for the two- and three-parameter Cox proportional hazard models found using 40 countries in BICA. The last three rows report the criterion values, the average simulated D-optimal efficiencies of the locally optimal designs relative to the corresponding ξ_d, and the CPU times for the four cases.

A direct calculation shows that all the generated designs have a minimum ELB of 0.998 and the derivative plots in Figure 3 confirm they are (nearly) optimal. Similar to the previous examples, the number of support points in the Bayesian optimal designs increases as the uniform distributions become more diffuse. We also note that the increase in the CPU time is substantial when d is increased from 3 to 11 and the two extreme ends of the design interval $\chi = [0, 1]$ are always the support points of the optimal designs.

It is interesting to assess the performance of the locally D-optimal designs when we use different prior distributions for the unknown parameters. We first consider the three-parameter Cox proportional hazards model. To investigate this, we first sampled 20 equidistant points from the interval $[-d, d]$ to construct $20^3 = 8000$ vectors of initial estimates for the vector of parameters $(\beta_0, \beta_1, \beta_2)$. Next, we found locally D-optimal design for each vector of initial estimates using the ICA algorithm (Masoudi, Holling, and Wong 2018) and, then, using (4), we calculated the Bayesian D-efficiency of the obtained locally D-optimal design relative to the BICA-generated design ξ_d when $\pi = \pi_d$. A similar procedure was done for the two-parameter Cox proportional hazards model. In Table 3, $\bar{\text{eff}}_D$ is the averaged value of the Bayesian D-efficiencies for each model and $d = 3, 11$. The results suggest that in our design settings, locally D-optimal designs have on average at least 96% Bayesian D-efficiencies when $d = 3$, but their performance deteriorates when the prior distributions become more diffuse, that is, when $d = 11$. This observation is similar to locally optimal designs becoming less efficient when the initial estimates are far from the true values of the unknown parameters, which is the case when the prior distributions become diffuse.

4.4. Compound DP-Criterion for the Logistic Model With Two Predictors

Sometimes, it is desirable to find a design that maximizes the probability of the occurrence of an event. For example, in medical trials we want to have a design that maximizes the probability that patients receive an effective treatment. McGree and Eccleston (2008) introduced probability-based optimal designs that maximize the weighted sum or average of the probabilities of success. The average P-optimal criterion is defined by

$$\phi_P(\xi, \boldsymbol{\theta}) = \sum_{i=1}^{k} w_i p_i(\boldsymbol{x}_i, \boldsymbol{\theta}), \tag{13}$$

where $\boldsymbol{x}_i$ and w_i are the support points and weights of the design ξ, and $p_i(\boldsymbol{x}_i, \boldsymbol{\theta})$ is the probability of success at $\boldsymbol{x}_i$ in a binary response model with parameter $\boldsymbol{\theta}$.

For the average P-optimal criterion, the P-efficiency of a design ξ is given by

$$P_{\text{eff}}(\xi) = \frac{\phi_P(\xi, \boldsymbol{\theta})}{\phi_P(\xi_P^*, \boldsymbol{\theta})}, \tag{14}$$

where ξ_P^* is the average P-optimal design. McGree and Eccleston (2008) presented the equivalence theorem for the P-optimality criterion: given $\boldsymbol{\theta}$, a design $\xi_{loc,P}^*$ is locally P-optimal design if and only if for all $\boldsymbol{x} \in \chi$

$$d_{loc,P}(\boldsymbol{x}, \xi_{loc,P}^*, \boldsymbol{\theta}) = \left\{ \frac{\phi_P(\xi_{\boldsymbol{x}}, \boldsymbol{\theta}) - \phi_P(\xi_{loc,P}^*, \boldsymbol{\theta})}{\phi_P(\xi_{loc,P}^*, \boldsymbol{\theta})} \right\} \leq 0, \tag{15}$$

with equality at the support points of $\xi_{loc,P}^*$.

A study may have two or more objectives at the onset. For example, D-optimality may be of interest in addition to P-optimality. To find a design that meets the dual goals of parameter estimation and that maximizes the probability of an event, a compromise is necessary. McGree and Eccleston (2008) proposed a compound or combined DP-criterion that is a weighted product of the D- and average P-efficiencies of a design. The weight is a predefined mixing constant α between 0 and 1 and represents the relative importance of the two criteria. This criterion is similar to that of A. C. Atkinson (2008) for DT-optimality for both model discrimination and parameter estimation. Specifically, for fixed α and $\boldsymbol{\theta}$, the compound criterion that we want to maximize by choice of a design is

$$\phi_{DP}(\xi, \boldsymbol{\theta}) = \frac{\alpha}{q} \log |M(\xi, \boldsymbol{\theta})| + (1 - \alpha) \log \left(\sum_{i=1}^{k} w_i p_i(\boldsymbol{x}_i, \boldsymbol{\theta}) \right), \tag{16}$$

and a locally DP-optimal design maximizes (16) over all possible designs.

When a prior distribution $\pi(\boldsymbol{\theta})$ is available for $\boldsymbol{\theta}$, similar to (2), a Bayesian P-optimal design ξ_P^* minimizes

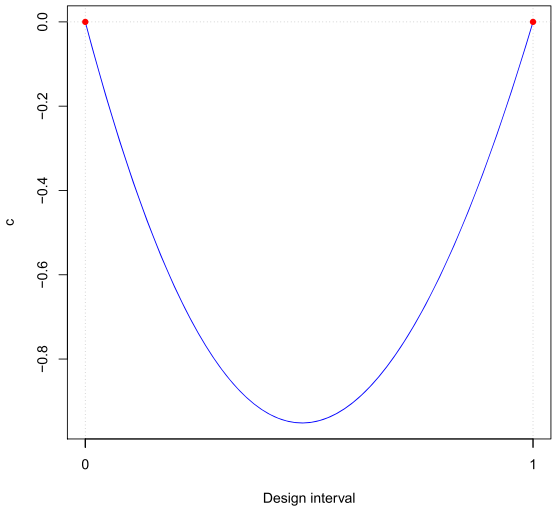
$$\psi_P(\xi, \pi) = \int_{\Theta} -\log \phi_P(\xi, \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{17}$$

The closeness of a design ξ to ξ_P^* can be measured by its Bayesian P-efficiency given by

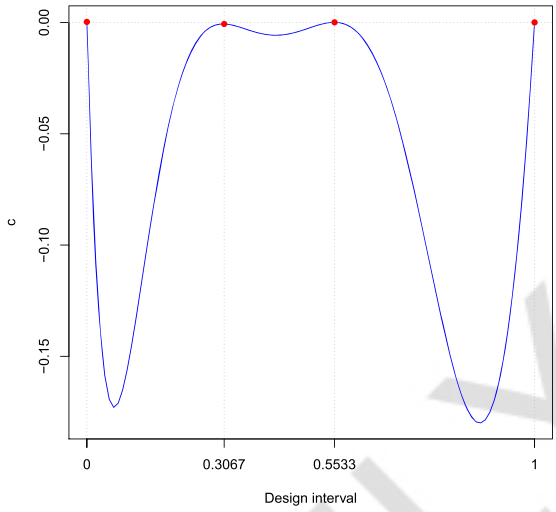
$$\text{eff}_P(\xi, \xi_P^*; \pi) = \exp(\psi_P(\xi_P^*; \pi) - \psi_P(\xi; \pi)). \tag{18}$$

The equivalence theorem for (17) comes from Theorem 2 of McGree and Eccleston (2008): given $\pi(\boldsymbol{\theta})$, a design ξ_P^* is Bayesian P-optimal design if and only if

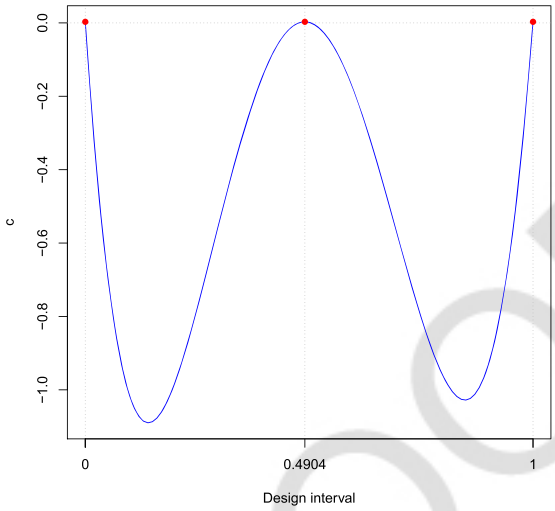
$$c_P(\boldsymbol{x}; \xi_P^*; \pi) = \int_{\Theta} d_{loc,P}(\boldsymbol{x}; \xi_P^*; \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} \leq 0 \tag{19}$$



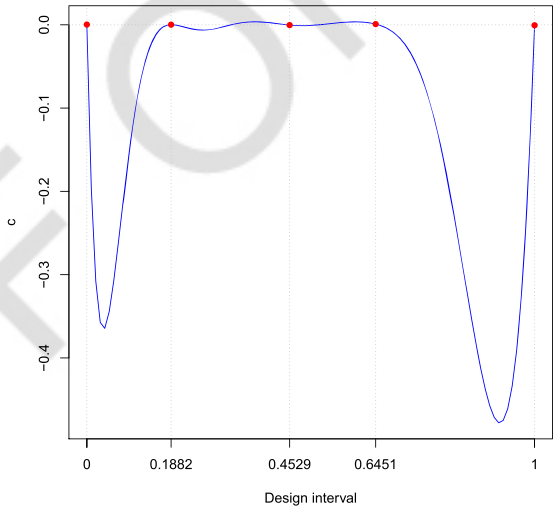
(a) Two-parameter model, $d = 3$



(b) Two-parameter model, $d = 11$



(c) Three-parameter model, $d = 3$



(d) Three-parameter model, $d = 11$

Figure 3. Plots of the nonpositive derivative functions $c_D(x, \xi_d; \pi)$ of the BICA-generated designs ξ_d on the design space [0, 1] from Table 3 when $d = 3$ and $d = 11$.

for all $\boldsymbol{x} \in \chi$ with equality at the support points of ξ_P^*. Accordingly, given α, a Bayesian *DP*-optimal design ξ_{DP}^* is the design that maximizes a weighted product of the Bayesian *P*- and *D*-efficiencies

$$\{\text{eff}_P(\xi, \xi_P^*; \pi)\}^{1-\alpha}\{\text{eff}_D(\xi, \xi_D^*; \pi)\}^{\alpha}, \tag{20}$$

or equivalently, minimizes the negative of the log function of (20). After ignoring constant terms involving ξ_P^* and ξ_D^*, we have

$$\begin{aligned}\psi_{DP}(\xi; \pi) &= \frac{\alpha}{q}\int_{\Theta} -\log|M(\xi, \boldsymbol{\theta})|\pi(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &\quad +(1-\alpha)\int_{\Theta} -\log\left(\sum_{i=1}^{k} w_i p_i(\boldsymbol{x}_i, \boldsymbol{\theta})\right)\pi(\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= \frac{\alpha}{q}\psi_D(\xi; \pi) + (1-\alpha)\psi_P(\xi; \pi). \end{aligned} \tag{21}$$

The derivative function for (21) is a weighted sum of (3) and (19) and one can directly show that ξ_{DP}^* is a Bayesian *DP*-optimal design if and only if

$$c_{DP}(\boldsymbol{x}, \xi_{DP}^*; \pi) = \frac{\alpha}{q}c_D(\boldsymbol{x}, \xi_{DP}^*; \pi) + (1-\alpha)c_P(\boldsymbol{x}, \xi_{DP}^*; \pi) \leq 0, \tag{22}$$

for all $\boldsymbol{x} \in \chi$ with equality at the support points of ξ_{DP}^*.

As an illustrative application, consider a logistic model with two factors of the form

$$\begin{aligned} P(Y_i = 1) &= p_i(\boldsymbol{x}_i, \boldsymbol{\theta}) \\ &= \frac{\exp(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_3 x_{i1} x_{i2})}{1 + \exp(\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_3 x_{i1} x_{i2})}, \end{aligned} \tag{23}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2})$, $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)^T$ and $\boldsymbol{x} \in \chi = [x_1^L x_1^U] \times [x_2^L, x_2^U]$. McGree and Eccleston (2008) used simulated annealing to find locally *DP*-optimal designs for different α where $\chi = [-1, 1] \times [-1, 1]$.

Table 4. BICA-generated Bayesian *DP*-optimal designs ξ_{DP} on $\chi = [-1,1] \times [-1,1]$ for different values of α for model (23).

α	0.25	0.5	0.75	1
ξ_{DP}	1.000, −1.000(0.085) −0.625, 1.000(0.431) 0.114, −1.000(0.012) −1.000, −1.000(0.145) 1.000, 1.000(0.059) 0.282, 1.000(0.090) −1.000, 0.094(0.177)	−1.000, −0.083(0.092) 0.302, 1.000(0.103) 1.000, −1.000(0.156) 1.000, 1.000(0.131) 0.074, −1.000(0.020) −1.000, −1.000(0.224) −0.320, 1.000(0.275)	1.000, 1.000(0.190) −1.000, −1.000(0.240) 0.283, 1.000(0.100) 1.000, −1.000(0.205) −1.000, −0.168(0.039) −0.197, 1.000(0.201) 0.033, −1.000(0.025)	1.000, 1.000(0.230) −1.000, −0.205(0.016) 0.266, 1.000(0.095) −0.134, 1.000(0.162) 1.000, −1.000(0.237) −0.009, −1.000(0.027) −1.000, −1.000(0.232)
$\psi_{DP}(\xi_{DP};\pi_U)$	0.929	1.539	2.064	2.554
$\text{eff}_D(\xi_{DP},\xi_D^*;\pi_U)$	0.644	0.911	0.987	1.000
$\text{eff}_P(\xi_{DP},\xi_P^*;\pi_U)$	0.805	0.667	0.588	0.545

NOTE: The prior distribution π_U for $\boldsymbol{\theta} = (\theta_0,\theta_1,\theta_2,\theta_3)^T$ is a product of independent uniform priors for each of its components with $\theta_0 \sim U(-0.5, 2.5)$, $\theta_1 \sim U(-3.5, -0.5)$, $\theta_2 \sim U(-0.5, 2.5)$, and $\theta_3 \sim U(-2.5, 0.5)$. The last three rows show the criterion values, the Bayesian *D*-efficiencies, and the *P*-efficiencies of the designs ξ_{DP} as the α values for the weighted criterion vary from 0.25, 0.50, 0.75 to 1.

We extend their application by finding Bayesian optimal designs. Let π_U be the prior distribution for the model parameters $\boldsymbol{\theta} = (\theta_0,\theta_1,\theta_2,\theta_3)^T$ and suppose it comprises the following independent uniform prior distributions for the four parameters: $\theta_0 \sim U(-0.5, 2.5)$, $\theta_1 \sim U(-3.5, -0.5)$, $\theta_2 \sim U(-0.5, 2.5)$, and $\theta_3 \sim U(-2.5, 0.5)$. Given π_U and different values for α, we used BICA to minimize (21) over the space of all possible designs to find Bayesian *DP*-optimal designs. In the algorithm, we used 40 countries and it was run for 1000 iterations. Table 4 displays the BICA-generated Bayesian *DP*-optimal designs for four weight values of α: 0.25, 0.50, 0.75, and 1.00.

A direct calculation shows that the values of the ELB of the BICA-generated designs are all nearly equal to 1. Figure 4 displays the derivative plot for the BICA-generated design when $\alpha = 0.5$ and confirms its optimality. The CPU time to find each design was about 50 min. This is not a surprise given that a four-dimensional integral has to be approximated with high accuracy and the optimization problem has 21 decision variables. Table 4 displays the Bayesian *D*- and *P*-efficiencies of the obtained designs using (4) and (18), respectively. The Bayesian *P*-optimal design ξ_P^* in our example is obtained when $\alpha = 0$ and the Bayesian *D*-optimal design is obtained when $\alpha = 1$. The Bayesian *P*-optimal design is not shown and is the degenerate design at $(x_1, x_2) = (-1, 1)$. As expected, the design efficiencies under one criterion increase and the efficiencies under the other criterion decrease as α varies. The varying values of the Bayesian *D*- and *P*-efficiencies suggest that a compromise between the two opposing criteria may be achieved by choosing an appropriate value for α. For example, by choosing $\alpha = 0.5$, the generated design has a Bayesian *P*-efficiency of 0.667 and a Bayesian *D*-efficiency of 0.911.

5. Sensitivity Analysis

This section briefly evaluates the sensitivity of the BICA-generated designs to the tuning parameters in the hcubature algorithm (`tol` and `maxEval`). To fix ideas and for space consideration, we chose a benchmark problem that required the longest CPU time for BICA to find the optimal design among all the examples in this paper. The benchmark problem is the Bayesian *D*-optimal design problem for the sigmoid Emax model in Section 4.2 when $\pi = \pi_{\Theta_4}$. To investigate the influence of different combinations of tuning parameters on the final result, we assigned one conservative and one less conservative value to each of the tuning parameters below. We note that the last pair of the tuning parameters is the most conservative and is the one used to produce all the previous examples in this paper. The complete set of scenarios is:

1. `tol` $= 10^{-4}$, `maxEval` $= 1000$ and the BICA-generated design is denoted by ξ_1.
2. `tol` $= 10^{-4}$, `maxEval` $= 50000$ and the BICA-generated design is denoted by ξ_2.
3. `tol` $= 10^{-5}$, `maxEval` $= 1000$ and the BICA-generated design is denoted by ξ_3.
4. `tol` $= 10^{-5}$, `maxEval` $= 50000$ and the BICA-generated design is denoted by ξ_4.

We ran BICA for each pair of the tuning parameters using the same design setting and initialization and Table 5 reports the results.

In all cases, the values of the criterion $\psi_D(\xi;\pi_{\Theta_4})$ at termination are approximately the same and equal to 13.33. The optimal designs are close to one another even though their CPU times differ. Figure 5 displays the derivative plots for the four generated designs and shows that they are closer to optimality when we set `tol` $= 10^{-5}$. However, based on the equivalence theorem, design ξ_4 is only optimal because the corresponding derivative plot (d) is the only one that is nonpositive for every $x \in [0.001, 500]$ and equal to zero at the support points of ξ_4 (see red points). This suggests that using less conservative values for `maxEval` and `tol` is unlikely going to impact the generated designs in practice for this example.

6. Comparison With Other Algorithms

There are recent algorithms proposed for finding Bayesian optimal designs. For example, Duarte and Wong (2014) proposed a semidefinite programming (SDP) based algorithm to find approximate designs on a discretized design space. Some disadvantages of such an algorithm and others that require the search space to be discretized are that the generated designs depend on the grid set and the algorithm can become ineffective for models with many covariates because the optimization problem becomes high dimensional. Duarte, Wong, and Oliveira (2016) introduced a nonlinear programming (NLP) based algorithm that does not need any discretization on the design space. This algorithm uses a multistart heuristic algorithm named OQNLP

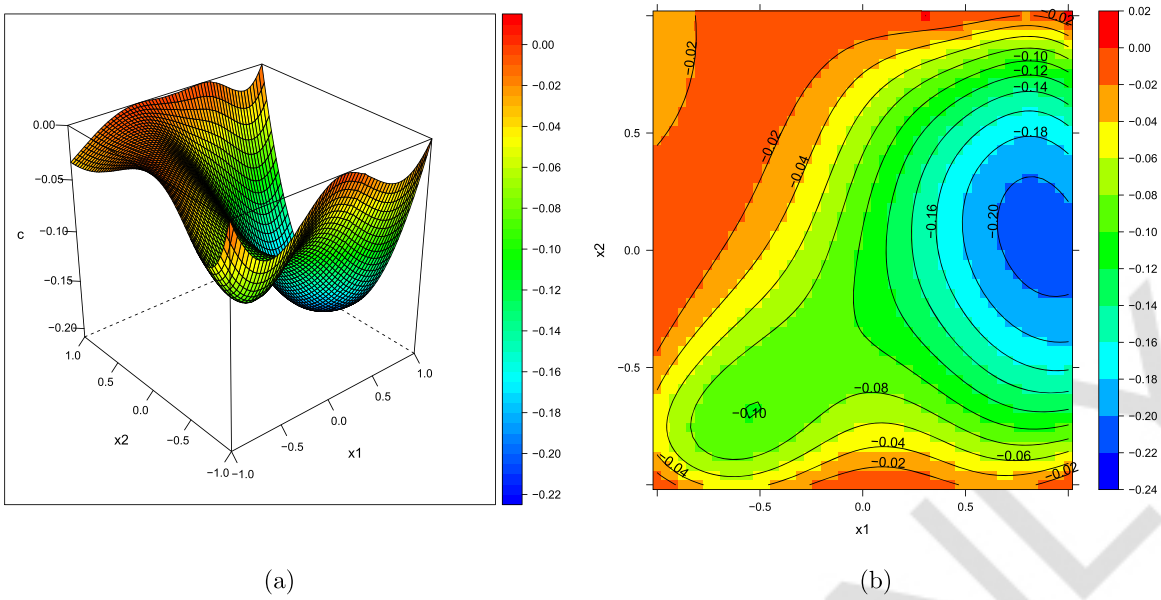


Figure 4. The left panel displays the derivative function $c_{DP}(\boldsymbol{x}, \xi_{DP}; \pi_U)$ for model (23) for $(x_1, x_2) \in \chi = [-1, 1] \times [-1, 1]$ when $\alpha = 0.5$ (a). The right panel displays the corresponding contour plot on χ (b).

Table 5. BICA-generated designs ξ for the sigmoid Emax model when $\boldsymbol{\theta} \sim \pi_{\Theta_4}$ and $\chi = [0.001, 500]$.

ξ	ξ_1	ξ_2	ξ_3	ξ_4
	12.726(0.206)	6.851(0.204)	0.699(0.205)	0.018(0.205)
	97.294(0.111)	96.599(0.113)	96.922(0.117)	96.433(0.112)
	115.720(0.090)	117.937(0.110)	120.154(0.131)	117.319(0.103)
	133.616(0.119)	136.465(0.118)	139.606(0.095)	133.068(0.100)
	152.882(0.123)	155.945(0.110)	154.525(0.102)	152.844(0.130)
	188.346(0.141)	188.795(0.133)	187.974(0.139)	188.081(0.139)
	500.000(0.211)	499.998(0.212)	500.000(0.212)	500.000(0.212)
$\psi_D(\xi, \pi_{\Theta_4})$	13.331	13.333	13.333	13.332
`maxEval`	1000	50000	1000	50000
`tol`	10^{-4}	10^{-4}	10^{-5}	10^{-5}
CPU(s)	3338	4600	3185	21,654

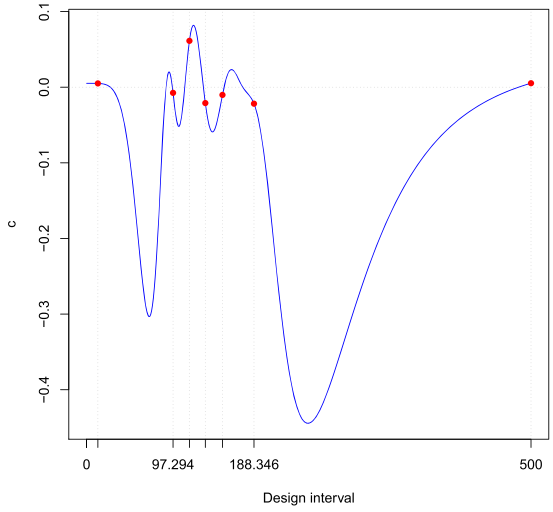
NOTE: The designs were found using different pairs of values for the tuning parameters `maxEval` and `tol`.

(Ugray et al. 2005) to determine an optimal design and it is codified in GAMS (GAMS Development Corporation 2013). Both of the SDP and the NLP-based algorithms are equipped with Gauss quadrature formulas to approximate the integrals. Duarte, Wong, and Oliveira (2016) showed that the NLP-based algorithm tend to generate more efficient designs with fewer number of support points than those designs found by the SDP-based algorithm.

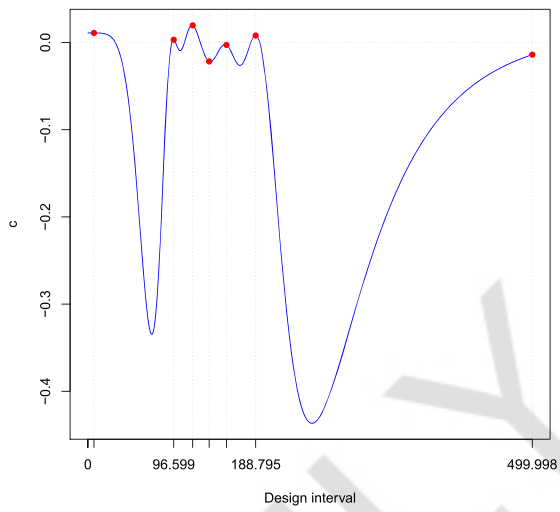
This section compares our algorithm with the NLP-based algorithm for finding D-optimal designs for a three-parameter alcohol kinetics model (Box and Hunter 1965) and the four-parameter sigmoid Emax model presented in Section 4.2. Due to space consideration, we offer details in Supplementary Section S1. In our comparison, the two algorithms produced very similar results, except that for the sigmoid Emax model when $\pi = \pi_{\Theta_4}$, the NLP-based algorithm was only able to obtain a local optimum. Our comparison using our examples suggests that BICA tends to be faster than the NLP-based algorithm when the prior distributions are uniform even when we use conservative default values for the hcubature tuning parameters. On the other hand, when the prior is normal, BICA tends to be slower than the NLP-based algorithm in finding the optimal design for the alcohol kinetics model. However, if we reduce the value of `maxEval` to 2000, BICA becomes considerably faster than the NLP-based algorithm again. A drawback of the NLP-based algorithm is that it relies on the expensive commercial software GAMS which we believe many statisticians may be unfamiliar with. Accordingly, we have devoted considerable effort to make our algorithm available in R and allow it to be directly amendable to find a wide variety of optimal designs for various nonlinear models. Our algorithm is free, open-source and available in the R package ICAOD (Masoudi, Holling, and Wong 2018).

7. Discussion

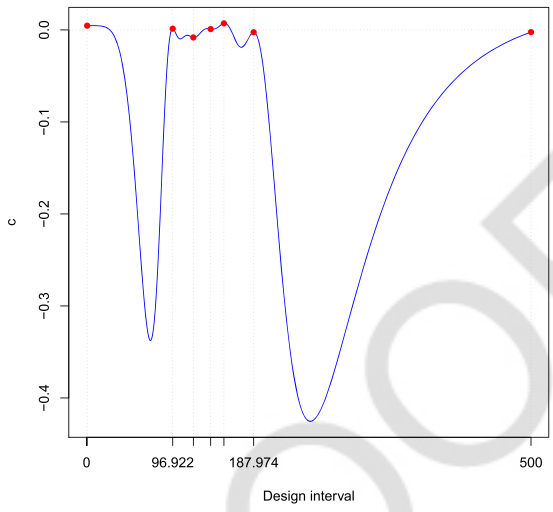
Metaheuristic algorithms are important and modern tools for tackling complicated optimization problems. They have been shown to be flexible, powerful and frequently give high quality solutions in computer science, engineering, and artificial intelligence (Yang 2010, 2011). Most metaheuristic algorithms including BICA are assumptions-free and this property makes them especially useful. They can even help to find a formula for the optimal design in a difficult problem. For instance, Chen, Chen, and Wong (2017) first applied PSO to find standardized



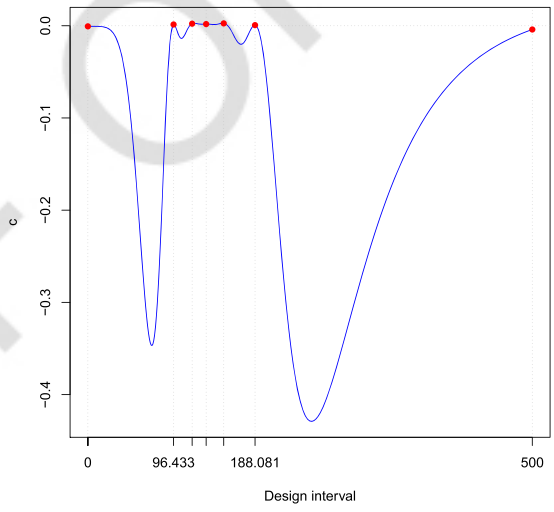
(a) $\xi = \xi_1$, $\texttt{maxeval} = 1000, \texttt{tol} = 10^{-4}$



(b) $\xi = \xi_2$, $\texttt{maxeval} = 50000, \texttt{tol} = 10^{-4}$



(c) $\xi = \xi_3$, $\texttt{maxeval} = 1000, \texttt{tol} = 10^{-5}$



(d) $\xi = \xi_4$, $\texttt{maxeval} = 50000, \texttt{tol} = 10^{-5}$

Figure 5. Plots of the derivative function $c_D(x, \xi; \pi_{\Theta_4})$ for the four designs in Table 5 found by BICA using different pairs of values for `maxEval` and `tol`. The derivative plot (d) proves the optimality of ξ_4.

maximin optimal designs for a three-parameter enzyme kinetic nonlinear model and then used information from the design structure and the equivalence theorem to obtain a formula for the optimal design. Two recent applications of using a nature-inspired metaheuristic algorithm to find optimal designs are Lukemire, Mandal, and Wong (2018) and Xu et al. (2018).

Convergence properties of metaheuristic algorithms are hard to study and we are not aware that there are rigorous proofs for many of them, even for the popular nature-inspired metaheuristic algorithms like PSO and ICA. The difficulty is largely due to the interactions of the various components in the algorithm that are highly nonlinear, complex, and stochastic (Yang 2011). Our view is that the lack of proof of convergence should not limit their use in practice, especially when there are theoretical methods to assess the proximity of a solution to the optimum without knowing the latter. In our case, a proof of convergence of ICA is not needed because with a convex functional to be optimized, an equivalence theorem is available to confirm optimality, or alternatively, an ELB can be theoretically constructed to assess proximity of the generated design to the optimum.

It is quite common to hybridize a metaheuristic algorithm with a properly selected procedure to enhance its performance. For example, if an algorithm is especially good at exploring where the optimum is, one may want to hybridize it with another algorithm that is very good for exploiting the neighborhood of the so optimum to locate it quickly. Generally, the two algorithms should have complementary advantages and be able to perform specific tasks that the other does not do as well. In this paper, we hybridized ICA with the hcubature algorithm which we found to be quite efficient for approximating the

Bayesian optimality criteria for models with a moderate number of parameters. The hcubature algorithm is distribution-free and may be used for evaluating integrals with different types of prior distributions. However, its performance depends on the spread and the shape of the prior distribution. In the current version (cubature-1.0.2), when the prior distribution is normal and located in the middle of the integral domain, the error estimation in the hcubature algorithm can become very conservative. For such a prior distribution, we recommend using a smaller value for `maxEval`, say 2000, to have a shorter CPU time.

Gaussian quadrature formulas are alternative methods to approximate the Bayesian optimality criteria. In Gaussian quadrature, the way in which the systematic nodes are determined depends on the underlying prior distributions. For example, when a uniform distribution is used, the systematic nodes and weights are usually obtained from a special type of orthogonal polynomials known as Legendre polynomials and the corresponding formula is called Gauss–Legendre quadrature. Similarly, the Gauss–Hermite formula, which is based on the Gauss-Hermite polynomials, can be applied when we have a normal prior distribution. For more details, see K. E. Atkinson (2008) and Goos and Mylona (2018). The ICA algorithm is very flexible and can also be combined with the Gaussian quadrature formulas. To avoid confusion, we call this algorithm BICA-Quad.

To compare the performance of the hcubature algorithm relative to results from the Gaussian quadrature formulas, we executed BICA and BICA-Quad in Section S2 of the supplementary materials to find Bayesian D-optimal designs for two benchmark models: the four-parameter sigmoid Emax model presented in Section 4.2, and a six-parameter generalized linear model with a Gamma distributed response. The results show that when we have a large enough number of nodes for the appropriately selected Gaussian quadrature formula, both algorithms can generate very similar designs. However, we experienced difficulty in finding the appropriate number of nodes for the sigmoid Emax model when $\pi = \pi_{\Theta_4}$ and we were only able to find nearly optimal designs. When the prior distribution is uniform, BICA was marginally slower than BICA-Quad for easy problems, that is, smaller parameter regions. This comes as no surprise because we used the conservative values for the hcubature tuning parameters and the number of nodes in the Gaussian quadrature was set to be the minimum sufficient value found by trial-and-error for each problem. If we were to use such a strategy for, say, `maxEval`, we could also reduce the CPU time for BICA. Further, when we used the uniform prior for the six-parameter generalized linear model, BICA became faster than BICA-Quad even when we used only a 3-point Gauss–Legendre quadrature formula. This was also the case when we required a larger number of nodes (more than 8) to find accurate designs for the four-parameter sigmoid Emax model. These observations suggest that the curse of dimensionality can have more negative effects on BICA-Quad than BICA, particularly when the prior is uniform. For the assumed normal prior distribution for the six-parameter generalized linear model, as explained above, the hcubature algorithm became very conservative and, therefore, we had to reduce the value of `maxEval`. For this example, the Gauss–Hermite formula resulted in a faster algorithm, although the CPU time for BICA was still reasonable.

To further emphasize that our approach is flexible, we note that our codes can be directly modified to find other types of optimal designs, including optimal exact designs. For example, to find the Bayesian D-optimal exact design in Table 3 of Gotwalt, Jones, and Steinberg (2009) with $N = 18$ observations, we can set $k = 18$ and modify our algorithm to find the optimal approximate design within the class of equally weighted designs. In this case, the support points of the generated approximate design are the same as the exact design with 18 observations. Upon execution, BICA generates a design similar to the one in Gotwalt's paper and the Bayesian D-efficiency of the Gotwalt's design relative to our design is approximately equal to 0.99. The main difference between optimal approximate designs and optimal exact designs is that the former can be constructed and confirmed using theory whereas there is no unified theory for finding and confirming optimality of an exact design. Since we are proposing a new algorithm, it is desirable to have mathematical tools to confirm the optimality of the generated design and this explains why we have focused on approximate designs.

We conclude with a reminder that nature-inspired metaheuristic algorithms like ICA, BICA, and PSO are general purpose optimization tools and can be used to solve nondesign optimization problems. We encourage the readers to explore and apply such algorithms for solving other optimization problems in statistics.

Supplementary Materials

Here, we provide two packages of computer codes: one for implementing the BICA and the other for replicating all examples in this article. In addition, we compare performances of BICA and other algorithms in the supplementary materials.

ICAOD R package contains codes of the BICA algorithm (ICAOD_0.9.8.tar.gz, zipped tar file).

Rcodes contains codes that can be used to replicate all the examples, simulations and plots in this study (Rcodes.zip, .zip archive). Please see the file README.txt contained in the zip file for more details.

Sections S1 and S2 "Supplementary materials: a comparison between BICA and other algorithms" (a PDF file).

Acknowledgments

We are grateful to the editor, associate editor, and two reviewers for their timely review and valuable comments. The contents in this paper are solely the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health.

Disclaimer

The contents in this article are solely the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health.

Funding

The research of Wong reported in this article was partially supported by a grant award R01GM107639 from the National Institute of General Medical Sciences of the National Institutes of Health.

References

Atashpaz-Gargari, E., and Lucas, C. (2007), "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition," in *IEEE Congress on Evolutionary Computation*, pp. 4661–4667. [2,4]

Atkinson, A. C. (1996), "The Usefulness of Optimum Experimental Designs," *Journal of the Royal Statistical Society*, Series B, 58, 59–76. [1]

——— (2008), "DT-Optimum Designs for Model Discrimination and Parameter Estimation," *Journal of Statistical Planning and Inference*, 138, 56–64. [9]

Atkinson, K. E. (2008), *An Introduction to Numerical Analysis*, Hoboken, NJ: Wiley. [14]

Atwood, C. L. (1969), "Optimal and Efficient Designs of Experiments," *The Annals of Mathematical Statistics*, 40, 1570–1602. [3]

Azzalini, A., and Capitanio, A. (1999), "Statistical Applications of the Multivariate Skew Normal Distribution," *Journal of the Royal Statistical Society*, Series B, 61, 579–602. [5]

Berger, M. P., King, C. J., and Wong, W. K. (2000), "Minimax D-Optimal Designs for Item Response Theory Models," *Psychometrika*, 65, 377–390. [4]

Berger, M. P. F., and Wong, W. K. (2009), *An Introduction to Optimal Designs for Social and Biomedical Research*, Chichester: Wiley. [4]

Berntsen, J., Espelid, T. O., and Genz, A. (1991a), "An Adaptive Algorithm for the Approximate Calculation of Multiple Integrals," *ACM Transactions on Mathematical Software (TOMS)*, 17, 437–451. [3]

——— (1991b), "Algorithm 698: Dcuhre: An Adaptive Multidimensional Integration Routine for a Vector of Integrals," *ACM Transactions on Mathematical Software (TOMS)*, 17, 452–456. [3]

Blum, D., Holling, H., Galibert, M. S., and Forthmann, B. (2016), "Task Difficulty Prediction of Figural Analogies," *Intelligence*, 56, 72–81. [5]

Box, G. E., and Hunter, W. G. (1965), "The Experimental Study of Physical Mechanisms," *Technometrics*, 7, 23–42. [12]

Chaloner, K., and Larntz, K. (1989), "Optimal Bayesian Design Applied to Logistic Regression Experiments," *Journal of Statistical Planning and Inference*, 21, 191–208. [1,3]

Chaloner, K., and Verdinelli, I. (1995), "Bayesian Experimental Design: A Review," *Statistical Science*, 10, 273–304. [1]

Chen, R.-B., Chang, S.-P., Wang, W., Tung, H.-C., and Wong, W. K. (2015), "Minimax Optimal Designs via Particle Swarm Optimization Methods," *Statistics and Computing*, 25, 975–988. [1]

Chen, R. B., Chen, P. Y., and Wong, W. K. (2017), "Standardized Maximin D-Optimal Designs for Enzyme Kinetic Models via Particle Swarm Optimization Techniques," *Chemometrics and Intelligent Laboratory Systems*, 169, 79–86. [12]

Chernoff, H. (1953), "Locally Optimal Designs for Estimating Parameters," *The Annals of Mathematical Statistics*, 24, 586–602. [1]

Dragalin, V., Hsuan, F., and Padmanabhan, S. K. (2007), "Adaptive Designs for Dose-Finding Studies Based on Sigmoid Emax Model," *Journal of Biopharmaceutical Statistics*, 17, 1051–1070. [7]

Duarte, B. P., and Wong, W. K. (2014), "A Semi-infinite Programming Based Algorithm for Finding Minimax Optimal Designs for Nonlinear Models," *Statistics and Computing*, 24, 1063–1080. [11]

Duarte, B. P., Wong, W. K., and Oliveira, N. M. (2016), "Model-Based Optimal Design of Experiments—Semidefinite and Nonlinear Programming Formulations," *Chemometrics and Intelligent Laboratory Systems*, 151, 153–163. [11,12]

Dunnett, C. W., and Sobel, M. (1954), "A Bivariate Generalization of Student's t-Distribution, With Tables for Certain Special Cases," *Biometrika*, 41, 153–169. [5]

Evans, M., and Swartz, T. (2000), *Approximating Integrals via Monte Carlo and Deterministic Methods*, Oxford: Oxford University Press. [3]

Firth, D., and Hinde, J. (1997), "On Bayesian D-Optimum Design Criteria and the Equivalence Theorem in Non-linear Models," *Journal of the Royal Statistical Society*, Series B, 59, 793–797. [3]

Fischer, G. H. (2004), "Remarks on Equivalent Linear Logistic Test Models by Bechger, Verstralen, and Verhelst (2002)," *Psychometrika*, 69, 305–315. [5]

GAMS Development Corporation (2013), "GAMS—A User's Guide," GAMS Release 24.2.1, Washington, DC, USA. [12]

Genz, A., and Bretz, F. (2009), *Computation of Multivariate Normal and t Probabilities*, Berlin: Springer. [3]

Genz, A., and Malik, A. (1980), "Remarks on Algorithm 006: An Adaptive Algorithm for Numerical Integration Over an n-Dimensional Rectangular Region," *Journal of Computational and Applied Mathematics*, 6, 295–302. [3]

Goos, P., and Mylona, K. (2018), "Quadrature Methods for Bayesian Optimal Design of Experiments With Nonnormal Prior Distributions," *Journal of Computational and Graphical Statistics*, 27, 179–194. [2,14]

Gotwalt, C. M., Jones, B. A., and Steinberg, D. M. (2009), "Fast Computation of Designs Robust to Parameter Uncertainty for Nonlinear Settings," *Technometrics*, 51, 88–95. [14]

Johnson, S. G. (2013), "Cubature Package," Version 1.0.3, available at *https://github.com/stevengj/cubature*. [3]

Kiefer, J. C. (1985), *Collected Papers III: Design of Experiments*, New York: Springer. [2,3]

Kiefer, J., and Wolfowitz, J. (1959), "Optimum Designs in Regression Problems," *The Annals of Mathematical Statistics*, 30, 271–294. [3]

Konstantinou, M., Biedermann, S., and Kimber, A. (2014), "Optimal Designs for Two-Parameter Nonlinear Models With Application to Survival Models," *Statistica Sinica*, 24, 415–428. [8]

Lu, H.-Y. (2014), "Application of Optimal Designs to Item Calibration," *PLoS One*, 9, 1–8. [4]

Lukemire, J., Mandal, A., and Wong, W. K. (2018), "d-QPSO: A Quantum-Behaved Particle Swarm Technique for Finding *D*-Optimal Designs With Discrete and Continuous Factors and a Binary Response," *Technometrics* (in press). [13]

Macdougall, J. (2006), "Analysis of Dose–Response Studies—Emax Model," in *Dose Finding in Drug Development*, New York: Springer, pp. 127–145. [7]

Masoudi, E., Holling, H., and Wong, W. K. (2017), "Application of Imperialist Competitive Algorithm to Find Minimax and Standardized Maximin Optimal Designs," *Computational Statistics & Data Analysis*, 113, 330–345. [1,2,4]

——— (2018), "ICAOD: Optimal Designs for Nonlinear Models," R Package Version 0.9.6. [9,12]

McGree, J., and Eccleston, J. (2008), "Probability-Based Optimal Design," *Australian & New Zealand Journal of Statistics*, 50, 13–28. [9,10]

Narasimhan, B., and Johnson, S. G. (2017), "cubature: Adaptive Multivariate Integration Over Hypercubes," R Package Version 1.3-11. [3]

Passos, V. L., and Berger, M. P. (2004), "Maximin Calibration Designs for the Nominal Response Model: An Empirical Evaluation," *Applied Psychological Measurement*, 28, 72–87. [4]

Pukelsheim, F., and Rieder, S. (1992), "Efficient Rounding of Approximate Designs," *Biometrika*, 79, 763–770. [2]

Qiu, J., Chen, R.-B., Wang, W., and Wong, W. K. (2014), "Using Animal Instincts to Design Efficient Biomedical Studies via Particle Swarm Optimization," *Swarm and Evolutionary Computation*, 18, 1–10. [8]

R Core Team (2018), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. [3]

Rost, J. (2001), "The Growing Family of Rasch Models," in *Essays on Item Response Theory*, New York: Springer, pp. 25–42. [5]

Schmidt, D., and Schwabe, R. (2014), "On Optimal Designs for Censored Data," *Metrika*, 78, 237–257. [8]

Sitter, R. R. (1992), "Robust Designs for Binary Data," *Biometrics*, 48, 1145–1155. [1]

Ugray, Z., Lasdon, L., Plummer, J. C., Glover, F., Kelly, J., and Marti, R. (2005), "A Multistart Scatter Search Heuristic for Smooth NLP and MINLP Problems," in *Metaheuristic Optimization via Memory and Evolution*, Boston, MA: Springer, pp. 25–57. [12]

Van Dooren, P., and de Ridder, L. (1976), "An Adaptive Algorithm for Numerical Integration Over an n-Dimensional Cube," *Journal of Computational and Applied Mathematics*, 2, 207–217. [3]

Wang, T., and Yang, M. (2014), "Adaptive Optimal Designs for Dose-Finding Studies Based on Sigmoid Emax Models," *Journal of Statistical Planning and Inference*, 144, 188–197. [7]

Xu, W., Wong, W. K., Tan, K. C., and Xu, J. (2018), "Finding High-Dimensional D-Optimal Designs for Logistic Models via Differential Evolution," *IEEE Access* (in press). [13]

Yang, X.-S. (2010), *Engineering Optimization: An Introduction With Metaheuristic Applications*, Chichester: Wiley. [2,12]

——— (2011), "Metaheuristic Optimization: Algorithm Analysis and Open Problems," in *Experimental Algorithms*, eds. P. M. Pardalos and S. Rebennack, Berlin: Springer, pp. 21–32. [12,13]

Q6