Published in final edited form as:

Chemometr Intell Lab Syst. 2016 February 15; 151: 153–163. doi:10.1016/j.chemolab.2015.12.014.

Model-based optimal design of experiments - semidefinite and nonlinear programming formulations

Belmiro P.M. Duarte^{a,b,*}, Weng Kee Wong^c, and Nuno M.C. Oliveira^a

Belmiro P.M. Duarte: bduarte@isec.pt; Weng Kee Wong: wkwong@ucla.edu; Nuno M.C. Oliveira: nuno@eq.uc.pt

^aCentro de Investigação em Processos Químicos e Produtos da Floresta, Department of Chemical Engineering, University of Coimbra, Pólo II, R. Sílvio Lima, 3030-790 Coimbra, Portugal. Tel. +351 239 798700

^bDepartment of Chemical and Biological Engineering, ISEC, Polytechnic Institute of Coimbra, R. Pedro Nunes, 3030-199 Coimbra, Portugal. Tel. +351 239 790200

^cDepartment of Biostatistics, Fielding School of Public Health, UCLA, 10833 Le Conte Ave., Los Angeles, California 90095-1772, U.S.A

Abstract

We use mathematical programming tools, such as Semidefinite Programming (SDP) and Nonlinear Programming (NLP)-based formulations to find optimal designs for models used in chemistry and chemical engineering. In particular, we employ local design-based setups in linear models and a Bayesian setup in nonlinear models to find optimal designs. In the latter case, Gaussian Quadrature Formulas (GQFs) are used to evaluate the optimality criterion averaged over the prior distribution for the model parameters. Mathematical programming techniques are then applied to solve the optimization problems. Because such methods require the design space be discretized, we also evaluate the impact of the discretization scheme on the generated design. We demonstrate the techniques for finding D -, A - and E -optimal designs using design problems in biochemical engineering and show the method can also be directly applied to tackle additional issues, such as heteroscedasticity in the model. Our results show that the NLP formulation produces highly efficient D -optimal designs but is computationally less efficient than that required for the SDP formulation. The efficiencies of the generated designs from the two methods are generally very close and so we recommend the SDP formulation in practice.

Keywords

Approximate Design; Bayesian Optimal Design; Global Optimization; Gaussian Quadrature Formula; Information Matrix

*Department of Chemical Engineering, University of Coimbra, Pólo II, R. Sílvio Lima, 3030-790 Coimbra, Portugal. Tel. +351 239 798700.

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

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

1. Introduction

We consider finding model-based optimal designs of experiments (M-bODE) for models that describe constitutive relations, commonly used to represent physical properties or kinetic data. For M-bODE problems, we have a given parametric model defined on a given design space and a given design criterion; our task is to find the number of design points required, where these design points are and the number of replicates at these design points that optimally meet the criterion. These design issues can be difficult to answer even for some relatively simple model. A general observation is that while there has been important advances made in solving estimation problems, innovation in techniques for finding efficient designs have not kept pace. In particular, it is helpful to explore the applicability of the increasing array of optimization numerical techniques used in other disciplines to solve statistical design problems where analytical approaches are no longer feasible. Continuing advances in algorithmic development is crucial to tackling more complex and high dimensional design problems.

In the subfield of optimal design of experiments in Statistics, various algorithms have been developed and continually improved for generating different types of optimal designs for algebraic models. Some examples are those proposed by Fedorov (1972) [1], Wynn (1972) [2], Mitchell (1974) [3] and, Gail and Kiefer (1980) [4]. Recently multiplicative algorithms seem to be gaining in popularity [5, 6]. Some of these algorithms are reviewed, compared and discussed in Cook and Nachtsheim (1982) [7] and Pronzato (2008) [8], among others. A common issue is how to confirm the global optimality of the design found from an algorithm. In selected situations, verification can be accomplished using an equivalence theorem [9]. These algorithms typically require a starting design and a stopping criterion to terminate the search for the optimal design. A common stopping rule comes from the general equivalence theorem, which we will use in this paper. Some algorithms also require that the space be discretized and so the generated optimal design depends on the size of the grid used in the search.

Mathematical programming algorithms and solvers have been used and continue to be widely used outside the field of statistics. These tools have improved substantially over the last two decades and they can solve complex high-dimensional optimization problems accurately and efficiently. In particular, mathematical programming approaches have been successfully employed to solve M-bODE problems. Some examples of such tools are Semidefinite Programming (SDP) [10, 11, 12], Semi Infinite Programming (SIP) [13], Nonlinear Programming (NLP) [14, 15], NLP combined with stochastic procedures such as Genetic Algorithms [16, 17], and Global Optimization [18]. This paper describes and compares a few mathematical programming tools for finding a variety of optimal designs used in chemistry and chemical engineering problems.

Section 2 presents background for SDP and NLP formulations for solving selected design problems, including Bayesian optimal design problems. Section 3 describes SDP formulations for linear and nonlinear models with applications to chemical engineering problems. Section 4 introduces the NLP formulations for finding D -optimal designs and

compares results with those from the SDP formulations in Section 3. A conclusion is offered in Section 5.

2. Background

2.1. Preliminaries

Throughout we assume that we have a regression model with a given mean function $f(\mathbf{x}, \boldsymbol{\theta})$ with differentiable components. The vector of regressors is $\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^{n_x}$ and $\mathbf{X}$ is a user-selected compact design space. The continuous response is y and its mean response at $\mathbf{x}$ is modeled by

$$\mathbb{E}[y|\mathbf{x}, \boldsymbol{\theta}] = f(\mathbf{x}, \boldsymbol{\theta}), \quad (1)$$

where the notation $\mathbb{E}[\bullet]$ is the expectation of the argument in $[\bullet]$. The $n_p \times 1$ vector of unknown model parameters $\boldsymbol{\theta}$ is assumed to belong to a known n_p -dimensional cartesian box $\Theta = \times_{j=1}^{n_p} [l_j, u_j] \in \mathbb{R}^{n_p}$ with each interval $[l_j, u_j]$ representing the known plausible range of values for the j^{th} parameter. We assume errors are homoscedastic but when the responses have different variances depending on where the $\mathbf{x}$'s are selected to observe the responses, methods discussed here can also apply and some brief results for such situations are also presented. Given a design criterion and a predetermined sample size, N , the research question is how to select the N sets of values for the covariates to observe the responses that maximize information in some optimal way.

A common goal of the M-bODE problem is to find an optimal *design* to maximize the information of the design of experiments carried out. Optimality depends on the objective of the study. For example, if predicting the responses at a few user-selected points in the design space is the primary goal, then one chooses a set of values of covariates in the design that will minimize the variances of the predicted responses at those points.

We focus on *approximate design problems*, which requires determination of a probability measure over the given design space $\mathbf{X}$. Such a design ξ is characterized by the number of support points, their locations in the design space and the proportions of observations to be taken at these points. If the sample size for the experiment is fixed at N , the approximate design ξ is implemented by taking roughly $N \times w_i$ observations at the design point $\mathbf{x}_i$, $i = 1, \dots, k$, subject to each $N \times w_i$ is a positive integer and $N \times w_1 + \dots + N \times w_k = N$. In what is to follow, we represent such a design by rows where each row shows one of the design points and the last component in the row is the weight at the support point. If there are n_x covariates in the model, the i^{th} design point is $\mathbf{x}_i^T = (x_{i,1}, \dots, x_{i,n_x})$ and if there are k of them, the design can be represented by k rows: $(\mathbf{x}_i^T, w_i), i \in \{1, \dots, k\}$ with $\sum_{i=1}^k w_i = 1$. In what is to follow, we let $[k] = \{1, \dots, k\}$.

An optimal approximate design optimizes a given criterion over Ξ , the space of all approximate designs on $\mathbf{X}$. The key advantages of working with *approximate designs* are that there is a unified framework for finding optimal continuous designs for M-bODE

problems and when the design criterion is a convex or concave functional of the information matrix, equivalence theorems are available to provide a practical way to check the optimality of any design among all *continuous designs*. In case the design is not optimal, the equivalence theorem also provides a lower bound of the design efficiency of the current design relative to the optimum (without the need to find the optimum). In addition, there are algorithms for finding several types of optimal approximate designs.

To fix ideas, we assume that all N responses have constant variances, are identically, independently and normally distributed and there are r_i replicates at each of the k points $\mathbf{x}_i$, $i \in [k]$ with $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,n_x})^T$. If $y_{i,j}$ is the j^{th} observation from $\mathbf{x}_i$, the total log-likelihood function is

$$\mathcal{L}(\xi, \boldsymbol{\theta}) = -\frac{1}{2} \sum_{i=1}^k r_i \log[2\pi] - \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^{r_i} [y_{i,j} - f(\mathbf{x}_i, \boldsymbol{\theta})]^2. \quad (2)$$

The maximum likelihood estimator (MLE) for $\boldsymbol{\theta}$ is:

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg \min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^k \sum_{j=1}^{r_i} [y_{i,j} - f(\mathbf{x}_i, \boldsymbol{\theta})]^2.$$

For an approximate k -point design with at $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ and weights $w_1, w_2, \dots, w_k$, the elements of the normalized FIM are the negative expectation of the second order derivatives of the total log-likelihood with respect to the parameters given by

$$\mathcal{M}(\xi, \boldsymbol{\theta}) = -\mathbb{E} \left[\frac{\partial}{\partial \boldsymbol{\theta}} \left(\frac{\partial \mathcal{L}(\xi, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right) \right] = \sum_{i=1}^k w_i \mathcal{M}(\boldsymbol{\delta}_{\mathbf{x}_i}, \boldsymbol{\theta}), \quad (3)$$

where $\mathcal{M}(\boldsymbol{\delta}_{\mathbf{x}_i}, \boldsymbol{\theta})$ is the FIM from the design $\boldsymbol{\delta}_{\mathbf{x}_i}$ that puts all weight at $\mathbf{x}_i$. Let $\mathbb{X}$ denote the discretized space from $\mathbf{X}$ using q points equally spaced in each dimension. The above information matrix is now approximated by

$$\sum_{\mathbf{x} \in \mathbb{X}} \mathcal{M}(\boldsymbol{\delta}_{\mathbf{x}}, \boldsymbol{\theta}) \chi(\mathbf{x})$$

where χ is the selected probability measure on $\mathbb{X}$ so that the above sum is equal to the integral in (3) as close as possible. We denote the set of q points in $\mathbb{X}$ by $[q] = \{1, \dots, q\}$.

The volume of the asymptotic confidence region of $\boldsymbol{\theta}$ is proportional to $\det[\mathcal{M}^{-1/2}(\xi, \boldsymbol{\theta})]$, and so maximizing the determinant by choice of the design provides the smallest possible volume. Maximizing the information matrix in other ways lead to other criteria, the most common ones are represented by a concave function of the information matrix. For example, D -, A - and E -optimal designs maximize each one of the following criteria, respectively:

$$\xi_D = \arg \max_{\xi \in \Xi} \{\log(\det[\mathcal{M}(\xi, \boldsymbol{\theta})])\}, \quad (4)$$

$$\xi_A = \arg \max_{\xi \in \Xi} \{\text{tr}[\mathcal{M}(\xi, \boldsymbol{\theta})^{-1}]\}^{-1}, \quad (5)$$

$$\xi_E = \arg \max_{\xi \in \Xi} \{\lambda_{\min}[\mathcal{M}(\xi, \boldsymbol{\theta})]\}, \quad (6)$$

where $\lambda_{\min}$ is the minimum eigenvalue of the FIM. The efficiency of a design ξ is its worth relative to the optimum, and the D -, A - and E -efficiency are defined, respectively, by

$$\text{eff}_D(\xi) = \left(\frac{\det[\mathcal{M}(\xi, \boldsymbol{\theta})]}{\det[\mathcal{M}(\xi_D, \boldsymbol{\theta})]} \right)^{1/n_p} \quad (7)$$

$$\text{eff}_A(\xi) = \left\{ \frac{\text{tr}[\mathcal{M}^{-1}(\xi, \boldsymbol{\theta})]}{\text{tr}[\mathcal{M}^{-1}(\xi_A, \boldsymbol{\theta})]} \right\}^{-1} \quad (8)$$

$$\text{eff}_E(\xi) = \frac{\max_{\lambda_i}[\mathcal{M}(\xi, \boldsymbol{\theta})]}{\max_{\lambda_i}[\mathcal{M}(\xi_E, \boldsymbol{\theta})]}. \quad (9)$$

Since the optimality criteria are concave functions of the FIM, we can use convex analysis theory [9, 19] or systematic mathematical convex programming solvers to obtain globally optimal designs [20, 21]. To verify the optimality of a design, Kiefer-Wolfowitz *equivalence theorems* [9, 1, 22] can be used. Pukelsheim (1993) [23] provides details on optimality criteria, equivalence theorems and the interpretation of design efficiency.

2.2. Pseudo-Bayesian designs

Nonlinear models are common in chemistry, thermodynamics, and chemical engineering, with typical applications ranging from modeling kinetic reaction rates to physical properties [24, 25]. For such models, the information matrix depends on the parameters and so all design criteria formulated in terms of the information matrix depend on the unknown parameters which we want to estimate. When nominal values are assumed for these parameters, the resulting designs are termed locally optimal. Some design strategies to handle this dependence include use of: (i) a series of locally optimal designs, each computed using the most updated estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ [26, Chap. 17]; (ii) Bayesian designs that optimize the expectation of the optimality criterion value averaged over the prior distribution model parameters $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$ [14]; (iii) minmax designs that maximize the design efficiency assuming that the true values for the model parameters are from the worst combination of parameters in $\boldsymbol{\Theta}$ [27]. Here we focus on finding Bayesian optimal designs, or perhaps more correctly, pseudo-Bayesian optimal designs. However, for the purpose of this paper, we use the two terms interchangeably.

The Bayesian approach assumes that the uncertainty of the parameters can be adequately captured in the prior distribution. This prior density averages out the parameter values so that the design criterion is no longer dependent on the unknown parameters. The Bayesian optimal designs are then found by optimizing the expectation of the design criterion, see Chaloner and Verdinelli (1995) [28]. Specifically, given a prior density $\pi(\boldsymbol{\theta})$ for $\boldsymbol{\theta}$, the Bayesian D -optimal design ξ_{BayesD} is defined by

$$\xi_{BayesD} = \arg \max_{\xi \in \Xi} \int_{\Theta} \{\log(\det[\mathcal{M}(\xi, \boldsymbol{\theta})])\} \pi(\boldsymbol{\theta}) d(\boldsymbol{\theta}).$$

Similar representations apply for Bayesian A -, and E -optimal designs from equations (5–6). Gaussian Quadrature Formulas (GQF) can be used to approximate the expectation integral of the optimality criterion by first discretizing the parameter space Θ . For each dimension, the integration points are the roots of the $(\kappa - 1)^{\text{th}}$ order Legendre polynomials and κ is the number of points used to approximate the integral. The roots and weights of the integration scheme are presented in Atkinson (1989) [29] and for simplicity, we use the same number of points for all dimensions of Θ . Other discretization schemes may also be employed.

All computation in this paper were carried using on an Intel Core i7 machine (Intel Corporation, Santa Clara, CA) running 64 bits Windows 7 operating system with 2.80GHz.

2.3. Semidefinite programming

SDP is a subfield of mathematical programming for solving a class of optimization problems that enables us to specify in addition to a set of linear constraints, *semi-definite* constraints, a special form of nonlinear constraints [30]. Generally, the formulation seeks to minimize a linear combination of decision variables aggregated in a matrix that must lie in the (closed convex) cone of positive semidefinite symmetric matrices, subject to a set of linear matrix inequalities. Because both the objective function and the constraints are convex, a semidefinite program is a convex optimization problem. This requires numerical algorithms to solve the problem efficiently which is done using appropriate solvers. Among the most efficient methods used nowadays is Interior Point, see [31]. Applications of SDP to solve various optimization problems in different disciplines are given in Vandenberghe and Boyd (1996) [32], including details on implementing SDP to search for optimal designs. Specific applications to solve optimal design problems include finding (i) D -optimal designs for multi-response linear models [33], (ii) c -optimal designs for single-response trigonometric regression models [34] and, (iii) D -optimal designs for polynomial models and rational functions [12]. Second order conic programming (SOCP) formulations share conic properties with SDP representations and this feature was recently exploited to find c -optimal designs for linear models with multiple responses [11]. Collectively, these papers emphasize the simplicity and efficiency of the SDP based approach for finding optimal designs.

The SDP-based approach requires the design space to be discretized into a user-specified grid of points $\mathbb{X}$. Given the fully parametrized model, we compute the Fisher Information Matrix (FIM) at each discretized point and sum them up to obtain the total information matrix. The design criterion is formulated as a convex function of this matrix, and SDP is

applied to solve the the optimal design problem using an appropriate SDP solver. The essential ingredients of solving an optimal design problem using SDP are as follows:

Let $\mathbb{S}^m$ be the space of $m \times m$ symmetric matrices and let $\zeta = (\zeta_1, \dots, \zeta_{m_1})^T \in \mathbb{R}^{m_1}$ be the vector of variables to be optimized in the the semidefinite program. A function $\phi: \mathbb{R}^{m_1} \mapsto \mathbb{R}$ is called semidefinite representable (SDr) if and only if inequalities of the form $u - \phi(\zeta) \succeq 0$ can be expressed by *linear matrix inequalities* (LMI) [12, 35]. That is, $\phi(\zeta)$ is SDr if and only if $u - \phi(\zeta) \succeq 0$ is equivalent to the existence of $m \times m$ symmetric matrices $M_0, \dots, M_{m_1}, \dots, M_{m_1+m_2} \in \mathbb{S}^m$ and a vector $v = (v_1, \dots, v_{m_2})^T$ such that

$$u - M_0 + \sum_{i=1}^{m_1} \zeta_i M_i + \sum_{j=1}^{m_2} v_j M_{m_1+j} \succeq 0. \quad (10)$$

Here, the notation $\succeq$ means that the matrix left of it *must be* non negative definite. Given the optimality criterion and the design problem, real numbers $c_1, \dots, c_{m_1}$ to be used in the linear combination of ζ are internally generated and the optimal values for ζ of the SDr functions are determined from the *semidefinite programs* of the form:

$$\max_{\zeta} \sum_{i=1}^{m_1} c_i \zeta_i \quad (11a)$$

$$\text{s. t. } \sum_{i=1}^{m_1} \zeta_i M_i - M_0 \succeq 0. \quad (11b)$$

In our design context, $m = n_p$, the known number of parameters in the model. However, the integers m_1 and m_2 are not user-specified and are set by the SDP solver. They depend on the number of points used in the discretization of the design space, which is known, and the operators used to codify the LMI in the SDP formulations. The vector $c = (c_1, \dots, c_{m_1})^T$ are also dependent on the design problem, the operators and the discretization scheme, and is generated internally before the optimization problem is solved. The matrices M_i , $i = 0, \dots, m_1$ are the local FIMs and other matrices used to reformulate SDr functions, and the vector ζ that includes the weights w_i , $i \in [k]$ of the optimal design. The matrices M_i , $i = m_1 + 1, \dots, m_1 + m_2$ are required to represent the SDr functions but are not included in the optimization problem. Optimal designs are found from formulation (11), which frequently contains many semidefinite constraints similar to (11b), along with the obvious linear constraints on w , i.e. its components w_i 's are non negative and they sum to unity.

A list of SDr functions was compiled by Ben-Tal and Nemirovski (2001) [36, Chap. 2–3], and used for deriving SDP formulations for the M-bODE problem, see Boyd and Vandenberghe (2004) [30, Sec. 7.3]. Sagnol (2013) showed that each criterion in the Kiefer's class of optimality criteria defined by

$$\Phi_p[\mathcal{M}(\xi, \theta)] = \left[\frac{1}{n_p} \text{tr}(\mathcal{M}(\xi, \theta)^p) \right]^{1/p}$$

is SDr for all rational values of $p \in (-\infty, 1]$ and general SDP formulations exist [35]. This result is also applicable to the case when $p \rightarrow 0$, whereupon $\Phi_p[\mathcal{M}(\xi, \theta)] = \det[\mathcal{M}(\xi, \theta)]^{1/n_p}$ and D -optimality obtains [37]. The maximization of this SDr function is clearly equivalent to maximizing the geometric mean of the determinant [36, Chap. 3].

2.4. Global optimization

NLP formulations for Bayesian D -optimal designs were used by Chaloner (1989) but they reported difficulty in applying the Federov-Wynn algorithm to find the optimal designs. They had to resort to a Nelder-Mead simplex based method to solve several design problems for the logistic model. Boer and Hendrix (2000) observed that D -optimality NLP formulation can have multiple optimal solutions so that global optimization tools are required. To make the problem numerically tractable, it is helpful search for the optimal design by fixing the number of support points and let the design evolve until the general equivalence theorem is validated.

Global optimization (GO) seeks to find the global optimum $\mathbf{x}$ of a nonconvex function $f: X \mapsto \mathbb{R}$ in a compact domain X . The general structure of the GO problems is:

$$\min_{\xi} \{f(\xi): \mathbf{r}(\xi)=0, \mathbf{g}(\xi) \leq 0, \xi \in \Xi\} \quad (12)$$

where $\mathbf{r}$ is a set of m_e equality constraints and $\mathbf{g}$ is a set of m_i inequality constraints [38]. If all the decision variables ξ are continuous, as we will assume here for w_i and $\mathbf{x}_i$, the problem is a *Nonlinear Program*. When the problem (12) is nonconvex, locally optimal solutions might not be global. The algorithms to handle GO problems fall into the deterministic methods [39] and stochastic methods [40]. The former group of approaches partitions the original domain into sub-domains, determines the local optima and uses theoretical concepts to discard subdomains. Some examples of deterministic algorithms are branch and bound, inner, outer and interval algebra-based methods; a complete overview of GO algorithms is available in [41, 39].

3. SDP-generated Optimal Designs

In this section, we use SDP formulations to solve M-bODE problems. In subsection 3.1 we have linear models and in subsection 3.2 we have nonlinear models. We demonstrate the two strategies using D -optimality and omit details for the A - and E -optimality. We present and compare the D -, A - and E -optimal designs from the two methods and the effect of the discretization scheme on the search and quality of the optimal designs.

3.1. Linear models

The SDP formulations used to find optimal designs for linear models are based on the representations of Boyd and Vandenberghe (2004) [30]. Recalling that the FIM for a linear

model does not depend on the model parameter Θ , we now write the FIM simply as $\mathcal{M}(\xi)$ and ignore the Θ in the argument. As an illustration, consider using the SDP formulation for finding a D -optimal design for a linear model. Recalling that n_p is number of parameters in θ the LMI $\tau \leq (\det[\mathcal{M}(\xi)])^{1/n_p}$ holds if and only if there exists a $n_p \times n_p$ -lower triangular matrix $\mathcal{C}$ such that

$$\begin{bmatrix} \mathcal{M}(\xi) & \mathcal{C}^T \\ \mathcal{C} & \text{Diag}(\mathcal{C}) \end{bmatrix} \succeq 0 \quad \text{and} \quad \tau \leq \left(\prod_{j=1}^{n_p} \mathcal{C}_{j,j} \right)^{1/n_p},$$

where $\text{Diag}(\mathcal{C})$ is the diagonal matrix with diagonal entries $\mathcal{C}_{j,j}$ and the geometric mean of the $\mathcal{C}_{j,j}$ on the extreme right can, in turn, be expressed as a series of 2×2 LMIs [36].

To handle SDP problems we can use user-friendly interfaces, such as *cvx* [42] or *Picos* [43], that automatically transform constraints of the form $\tau \leq \phi(\xi)$ as a series of LMIs, and pass them to SDP solvers such as *SeDuMi* [44] or *Mosek* [45]. In what is to follow, we present SDP formulations for finding A -, E - and D -optimal designs in a compact form, so that they can be directly applied for high-level interfacing. This means that instead of using the LMIs generated by reformulating the optimization problem, we use the operators themselves with additional constraints to ensure the matrix $\mathcal{M}(\xi)$ positive semidefinite.

Consider the SDP formulation for the D -optimal design problem (4) with $\Phi_p[\mathcal{M}(\xi)] = \det[\mathcal{M}(\xi)]^{1/n_p}$ as the criterion. The optimization problem is now compactly represented by

$$\max_{\tau \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^q} \tau \quad (13a)$$

$$\text{s. t. } \tau \leq [\det(\mathcal{M}(\xi))]^{1/n_p} \quad (13b)$$

$$w_i \geq 0, \quad \forall i \in [q] \quad (13c)$$

$$\sum_{i=1}^q w_i = 1. \quad (13d)$$

Similar formulations also apply for the A - and E -optimal design problems (5–6) using the criteria $[\text{tr}(\mathcal{M}^{-1}(\xi))]^{-1}$, and $\lambda_{\min}(\mathcal{M}(\xi))$, respectively. These SDP problems are then solved using the *cvx* environment combined with the SDP solver *Mosek*. The relative and absolute tolerances were set to 10^{-5} in all problems.

3.1.1. Example 1: Optimal Designs for a Linear Mixture Model—We evaluate the SDP formulation for finding optimal designs for a linear model using an empirical model representing the influence of the composition of water/acetone/ethanol mixture on the size of amphiphilic β -cyclodextrin nanoparticle [46]. The experiment is in part motivated by the

known effect of the composition of the solvent mixture in the size of the nanoparticles produced by precipitation. We use a quadratic mixture model with mean function given by

$$\mathbb{E}[y|\mathbf{x}, \boldsymbol{\beta}] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{1,2} x_1 x_2 + \beta_{1,3} x_1 x_3 + \beta_{2,3} x_2 x_3 + \beta_{1,1} x_1^2 + \beta_{2,2} x_2^2 + \beta_{3,3} x_3^2 \quad (14a)$$

$$\text{s. t. } x_1 + x_2 + x_3 = 1 \quad (14b)$$

$$0.4 \leq x_1 \leq 0.7, 0.0 \leq x_2 \leq 0.6, 0.0 \leq x_3 \leq 0.6. \quad (14c)$$

Here x_1 is the fraction of water, x_2 the fraction of ethanol, x_3 the fraction of acetone in the solvent, and the response y is the average size of the nanoparticles. The inequalities (14c) are physical constraints that the fraction of each component has to satisfy. Using the equality constraint (14b) to eliminate x_3 , we obtain

$$\mathbb{E}[y|\mathbf{x}, \mathbf{b}] = b_0 + b_1 x_1 + b_2 x_2 + b_{1,2} x_1 x_2 + b_{2,1} x_1^2 + b_{2,2} x_2^2 \quad (15a)$$

$$\text{s. t. } 0.4 \leq x_1 \leq 0.7, 0.0 \leq x_2 \leq 0.6, x_1 + x_2 \leq 1.0, \quad (15b)$$

where all the parameters b_i and $b_{i,j}$'s are functions of the original β_i 's. Our goal is to find D -, A - and E -optimal designs for estimating all the parameters in the re-parameterized model (15a), which includes the parameters b_i and $b_{i,j}$.

The design space is $\mathbf{X} \equiv [0.4, 0.7] \times [0.0, 0.6]$ and we discretize this two-dimensional space using equally spaced grid points with $x_1 = x_2 = 0.01$. The discretization setup produces a discrete design space, denoted by $\mathbb{X}$ with 1426 candidate points. From equation (3), we note that the mean vector function is $[1, x_{1,i}, x_{2,i}, x_{1,i} x_{2,i}, x_{1,i}^2, x_{2,i}^2]$ and because the model is linear, the FIM is independent of model parameters. The solution of the SDP problem (13) determines the optimal set of k support points in ξ .

Table 1 displays the D -, A - and E -optimal designs obtained for all the optimality criteria. The D - and A -optimal designs have 9 support points, and the E -optimal design requires 35 points. Table 1 also reports the CPU time required to solve each problem. In all cases, our CPU times are relatively short.

To analyze the impact of the discretization scheme on the generated design, we now use two coarser equally spaced grid sets to discretize the design space with $x_1 = x_2 = 0.02$ and $x_1 = x_2 = 0.1$. These choices are somewhat arbitrary but seem adequate for the purpose of the comparison here. The generated designs using these coarser grid sets are shown in Table 2 and are slightly different from those found with a finer grid in Table 1). Since the support points of the optimal design are *dependent* on the grid used to discretize the design space, thinner grids might produce more efficient designs since they are closer to optimal designs obtained for continuous domain $\mathbf{X}$. In section 4 we investigate the efficiency differences assuming that the design space is continuous and global optimization tools are used.

The efficiencies obtained for both discretization schemes are listed in Table 2, and were determined with equations (7–9) assuming that the optimal designs, ξ_D , ξ_A , and ξ_E , were generated with the thinner grid (cf. Table 1). We note that the grid generated with $x_1 = x_2 = 0.02$ has four times fewer nodes than the original ($x_1 = x_2 = 0.01$), but the decrease on the efficiency is below 1%. The grid produced with $x_1 = x_2 = 0.1$ has 100 times fewer nodes, and the reduction on the efficiency is between 10 and 20%. The differences observed in the efficiency of designs, due to support points placement, are compensated by the optimal choice of the weights, w_i . From the practical point of view, a discrete grid is realistic since the design space may not allow, due to physical and economic limitations, all arbitrary combinations of regressors, and its implementation is also constrained to rational values of $N \times w_i$.

3.2. Nonlinear models

We now extend the SDP-based formulation to find optimal designs for estimating the model parameters θ in a nonlinear model. Specifically, we find Bayesian optimal designs by first eliciting a prior distribution $\pi(\theta)$ for the model parameters and then average the criterion over the prior distribution. The resulting expectation integral is then approximated using GQF based on $(\kappa - 1)^{\text{th}}$ degree Legendre polynomials, see Duarte and Wong (2014) [47].

Let $\mathcal{M}(\xi, \theta)$ be the FIM from an approximate design ξ . The Bayesian optimal design problem is to find a design that satisfies

$$\xi_{\text{Bayes}\Phi_p} = \arg \max_{\xi \in \Xi} \int_{\Theta} \Phi_p[\mathcal{M}(\xi, \theta)] \pi(\theta) d(\theta). \quad (16)$$

Here we assume the design criterion is one of Kiefer's Φ_p optimality criteria, but other criteria may be used. For D-optimality, $\Phi_p[\mathcal{M}(\xi, \theta)] = (\det[\mathcal{M}(\xi, \theta)])^{1/n_p}$, which is equivalent to $\log(\det[\mathcal{M}(\xi, \theta)])$. For A-optimality, we have $(\text{tr}[\mathcal{M}^{-1}(\xi, \theta)])^{-1}$, and for E-optimality, we have $\lambda_{\min}[\mathcal{M}^{-1}(\xi, \theta)]$.

Let ι be the number of points used in the integral approximation over Θ and let $[\iota] = \{1, \dots, \iota\}$ denote the set of points. Because we use Legendre polynomials of the same degree to approximate the expectation in each dimension of Θ , we have $\iota = \kappa^{n_p}$. The discrete set $\Theta \in \Theta$ contains ι parameter combinations $\theta_i, i \in [\iota]$. Each element $\theta_i \in \mathbb{R}^{n_p}$ of Θ is the cartesian product of the set containing GQF points from each dimension of Θ . If $\rho^T \equiv [\rho_1, \dots, \rho_\kappa]$ is the vector of roots of the $(\kappa - 1)^{\text{th}}$ order Legendre's polynomial on the interval $[-1, 1]$, then

$$\Theta = \times_{i=1}^{n_p} \left\{ \rho_k \left(\frac{\theta_i^U - \theta_i^L}{2} \right) + \left(\frac{u_i + l_i}{2} \right), \quad k \in [\kappa] \right\}.$$

Let i^{th} be the tuple containing elements of $[\kappa] = \{1, \dots, \kappa\}$, let $[k_1^i, \dots, k_{n_p}^i] \in [\kappa]^{n_p}, i \in \iota$ and let $\gamma^T = [\gamma_{k_1}, \dots, \gamma_{k_{n_p}}] \in \mathbb{R}^\kappa$ be the vector of weights of the Legendre polynomials on the interval $[-1, 1]$. The weight of the i^{th} point $\theta_i \in \Theta$ in the GQF is

$$\omega_i = \prod_{j=1}^{n_p} \gamma_{k_j^i} \left(\frac{u_j - l_j}{2} \right)$$

and the expectation in (16) is now approximated using the GQF. The sought Bayesian optimal design for the prior $\pi(\boldsymbol{\theta})$ is

$$\xi_{B\Phi_p} = \arg \max_{\xi \in \Xi} \sum_{i=1}^l \Phi_p[\mathcal{M}(\xi, \boldsymbol{\theta}_i)] \pi(\boldsymbol{\theta}_i) \omega_i, \quad \boldsymbol{\theta}_i \in \Theta. \quad (17)$$

We observe that the Bayesian optimal design in (17) is obtained by optimizing a linear combination of the criteria Φ_p and because $0 \leq \pi(\boldsymbol{\theta}_i) \leq 1$ and $0 \leq \omega_i \leq 1$, $i \in [l]$, it is also a linear combination of SDr functions. This implies that the sum is a SDr function since each atomic element $\Phi_p[\mathcal{M}(\xi, \boldsymbol{\theta}_i)]$ is SDr by definition. Consequently, the SDP formulations for finding Bayesian A- and E-optimal designs follow directly as follows. For A-optimality, we note that $\Phi_p[\mathcal{M}(\xi, \boldsymbol{\theta}_i)] = (\text{tr}[\mathcal{M}^{-1}(\xi, \boldsymbol{\theta}_i)])^{-1}$ is SDr for all $i \in [l]$ and the optimization problem is

$$\max_{\tau \in \mathbb{R}^l, \mathbf{w} \in \mathbb{R}^q} \sum_{j=1}^l \tau_j \pi(\boldsymbol{\theta}_j) \omega_j \quad (18a)$$

$$\text{s. t. } \tau_j \leq [\text{tr}(\mathcal{M}^{-1}(\xi, \boldsymbol{\theta}_j))]^{-1}, \quad \forall j \in [l] \quad (18b)$$

$$w_i \geq 0, \quad \forall i \in [q] \quad (18c)$$

$$\sum_{i=1}^q w_i = 1. \quad (18d)$$

For E-optimality, the SDP problem may be similarly formulated as follows:

$$\max_{\tau \in \mathbb{R}^l, \mathbf{w} \in \mathbb{R}^q} \sum_{j=1}^l \tau_j \pi(\boldsymbol{\theta}_j) \omega_j \quad (19a)$$

$$\text{s. t. } \tau_j \leq \lambda_{\min}(\mathcal{M}(\xi, \boldsymbol{\theta}_j)), \quad \forall j \in [l] \quad (19b)$$

$$w_i \geq 0, \quad \forall i \in [q] \quad (19c)$$

$$\sum_{i=1}^q w_i = 1. \quad (19d)$$

The SDP formulation for the D -optimal design problem is more complicated because as noted earlier, $\Phi_p[\mathcal{M}(\xi, \theta)] = \log(\det[\mathcal{M}^{-1}(\xi, \theta)])$ is not SDr. However, the exponentiation of the sum of logarithm terms produces an equivalent problem

$$\xi_{BayesD} = \arg \max_{\xi \in \Xi} \prod_{i=1}^l \left\{ (\det[\mathcal{M}(\xi, \theta_i)])^{1/n_p} \right\}^{n_p \pi(\theta_i) \omega_i}, \quad \theta_i \in \Theta,$$

where the function now to optimize is SDr. Specifically, the terms $(\det[\mathcal{M}(\xi, \theta_i)])^{1/n_p}$, by construction, are SDr, and the product has the form of a concave monomial, which is SDr if the power terms $\alpha_i = n_p \pi(\theta_i) \omega_i, \forall i \in [l]$ are rational; see Ben-Tal and Nemirovski (2001) [36, Chap. 3]. If the power α_i is irrational, a nearby rational value is used instead [42]. The upshot is that the Bayesian D -optimal design formulation is

$$\max_{v \in \mathbb{R}^l, \tau \in \mathbb{R}^l, w \in \mathbb{R}^q} \left(\prod_{j=1}^l v_j \right)^{1/l} \quad (20a)$$

$$\text{s. t. } \tau_j \leq [\det(\mathcal{M}(\xi, \theta_j))]^{1/n_p}, \quad \forall j \in [l] \quad (20b)$$

$$v_j \leq \tau_j^{\alpha_j}, \quad \forall j \in [l] \quad (20c)$$

$$w_i \geq 0, \quad \forall i \in [q] \quad (20d)$$

$$\sum_{i=1}^q w_i = 1. \quad (20e)$$

3.2.1. Example 2: Optimal Designs for Estimating the Kinetics of Alcohol

Dehydration—We now present two examples of SDP formulations of design problems for nonlinear models. The first considers the case where the FIM is obtained from the mean function f directly. In the second case, we consider a more complex model where the mean response function is only implicitly defined.

The first example is taken from Box (1965) [24], where interest was fitting the kinetics of the catalytic dehydration of n-hexyl alcohol using the model:

$$\mathbb{E}[y|\mathbf{x}, \mathbf{b}] = f(\mathbf{x}, \boldsymbol{\theta}) = \frac{b_3 x_1}{1 + b_1 x_1 + b_2 x_2}, \quad \mathbf{x} = (x_1, x_2)^T, \quad \boldsymbol{\theta} = (b_1, b_2, b_3)^T. \quad (21)$$

Here y is the reaction rate, b_1, b_2, b_3 are parameters to estimate, and x_1 and x_2 are the partial pressures of alcohol and olefin used in the experiment, respectively. The plausible range of

values for the regressors $\mathbf{x} = (x_1, x_2)^T$ are in a set $\mathbf{X} \equiv [0.0, 2.0] \times [0.0, 2.0]$, and the vector of possible values for the parameters is contained in $\Theta \equiv [b_1^L, b_1^U] \times [b_2^L, b_2^U] \times [b_3^L, b_3^U]$. The FIM for a “single observation” $\mathbf{x}_i$ is constructed by first differentiating $f(\mathbf{x}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ to obtain the vector $\mathbf{h}^T(\mathbf{x}_i, \boldsymbol{\theta})$ given by

$$\left[\frac{-b_3 x_{1,i}^2}{(1+b_1 x_{1,i}+b_2 x_{2,i})^2}, \frac{-b_3 x_{1,i} x_{2,i}}{(1+b_1 x_{1,i}+b_2 x_{2,i})^2}, \frac{x_{1,i}}{1+b_1 x_{1,i}+b_2 x_{2,i}} \right].$$

Our goals are to find A -, D - and E -optimal designs for estimating model parameters. To this end, we used a grid with equally spaced points of width $x_1 = x_2 = 0.1$ to first discretize the design space. This results in 441 grid points as candidate support points for the design. The expectation is determined by using a 6 point-based GQF in each dimension of Θ , resulting in $6^3 = 216$ points. We used Box (1965) [24] nominal values for the parameters and set the plausible region $\Theta \equiv [1.9, 3.9] \times [9.2, 15.2] \times [1.14, 2.34]$. Our interest is to determine various optimal designs for estimating the model parameters assuming that we have a three-dimensional uniform prior distribution on Θ , i.e. $\boldsymbol{\theta} \stackrel{\text{iid}}{\sim} \mathcal{U}(\Theta)$.

Table 3 presents the Bayesian optimal designs obtained with SDP formulations using different grid sets. We observe that all optimal designs obtained with the grid constructed with $x_1 = x_2 = 0.1$ have 4 support points, and 3 support points when a coarse grid with size twice that of the original is employed. The CPU time required is much longer than that required in Example 1 because the size of the SDP problem needed to solve here has increased by a few orders of magnitude. We also notice that (i) the CPU time increases significantly when the grid set used to discrete $\mathbf{X}$ is finer and, (2) the Bayesian D -optimal design problem is computationally more challenging because of the number of LMIs in the problem.

To investigate the impact of choice of the prior distribution $\pi(\boldsymbol{\theta})$ on the generated design, we now suppose we have a three-dimensional uncorrelated normal distribution with mean $\boldsymbol{\mu} = (2.9, 12.2, 1.74)^T$, and the ordered diagonal elements in the covariance matrix Υ_1 are $1/3^2$, 1.0^2 and 0.2^2 , respectively. Table 4 presents the the A -, D - and E -optimal designs when the prior distribution is $\boldsymbol{\theta} \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \Upsilon_1)$. The number of support points of the A -optimal design is the same as that of the uniform prior, and but the E -optimal design has one additional support point compared with the design found under the uniform prior distribution. Tables 3 and 4 compare the effect of the fineness of discretization scheme on the grid used to discretize the design space and they reveal that the prior distribution seems to have little influence in the generated designs. Of course, this observation cannot be generalized to other design problems but techniques discussed here can be similarly applied to ascertain the impact of the choice of the grid set on the generated design.

3.2.2. Example 3: Optimal Designs for Estimating Activity Coefficients in a UNIFAC Model—This application finds optimal designs for estimating group interaction parameters in the UNIFAC model. Recalling that UNIQUAC (short for UNIVERSAL

QUAsiChemical) is an activity coefficient model used to describe phase equilibria, UNIFAC stands for UNIQUAC Functional-group Activity Coefficients and the method is a semi-empirical system for estimating non-electrolyte activity in non-ideal mixtures. For example, [48] used the model for estimating the activity coefficient in the liquid-vapor equilibrium. Our model comprises a binary mixture of n-pentane and acetone where the mean response is given by

$$\mathbb{E}[y|\mathbf{x}, \mathbf{b}] = f(x_1, x_2, b_1, b_2), \quad \mathbf{x} = (x_1, x_2)^T, \quad \boldsymbol{\theta} = (b_1, b_2)^T. \quad (22)$$

Here y is the activity coefficient and ζ is experimentally measured using the ratio $\gamma^l P_v / (\gamma^v P)$, where γ^l the molar fraction of a mixture component in the liquid phase, P_v is the vapor pressure estimated by the Antoine equation, γ^v is the molar fraction of such a component in the vapor phase in equilibrium, P is the pressure at which the experiment is carried out, and both b_1 and b_2 are group interaction parameters. The Antoine equation is a simple 3-parameter regression model commonly used to fit experimental vapor pressures measured over a restricted temperature range, and we assume the parameters in the equation are known. The function f , formalized as the UNIFAC model, is continuous and differentiable, and can be found in several textbooks. For a complete overview of the model and their technicalities the readers are referred to [48, pages 8.75–8.77]. We assume that the regressors in the design of experiments are the molar fraction of one of the components in the liquid phase, here called x_1 , and the temperature of the experiment (expressed in K), designated by x_2 . Once the mixture becomes binary, the composition of the second component depends only on that of the first, and is not considered a factor in the design of experiments.

Let us consider a binary mixture formed by n-pentane/acetone, with three different functional groups: CH_3- , $-\text{CH}_2-$, and $\text{CH}_3\text{CO}-$. The group interaction parameters between the groups CH_3- and $-\text{CH}_2-$ are 0, since both belong to the same *Main groups*, see Hansen (1991) [49]. An optimal design is sought to estimate, as accurate as possible, the interaction parameter between the groups $\text{CH}_3\text{CO}-$ and CH_3- , b_1 , and the interaction parameter between the groups CH_3- and $\text{CH}_3\text{CO}-$, b_2 . The domain $\boldsymbol{\Theta}$ is $[426.40, 526.40] \times [20.76, 32.76]$, the design space is $\mathbf{X} \equiv [0.0, 1.0] \times [298.0, 318.0]$ and the nominal values for b_1 and b_2 are 476.40 and 26.76, respectively, see Poling et al. (2001) [48]. The grid employed to discretize the design space $\mathbf{X}$ is equally spaced in each dimension with $x_1 = 0.01$ and $x_2 = 1.0$. The information matrix is found in the same way, except that the vector of the derivatives of f , $\mathbf{h}^T(\mathbf{x}_i, \boldsymbol{\theta})$, is now determined using a *numerical differentiation* scheme based on central differences approximation with a perturbation technique and a step size equal to 10^{-5} .

Table 5 shows the optimal designs found when the uniform prior $\boldsymbol{\theta} \stackrel{\text{iid}}{\sim} \mathcal{U}(\boldsymbol{\Theta})$ is used. The A— and E—optimal designs have 2 support points, and the D—optimal design has 3 points. In all the designs, the optimal temperature settings to carry out the experiment are at the extreme ends in $\boldsymbol{\Theta}$. Table 5 also presents the designs obtained for a larger discretization grid where $x_1 = 0.2$ and $x_2 = 1.0$. Both grids produce similar designs with about the same

efficiency, except for A -optimality where both discretization schemes produce the same design.

Table 6 displays the optimal designs determined for the normal prior $\boldsymbol{\theta} \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \Upsilon_2)$ with $\boldsymbol{\mu} = (476.40, 26.76)^T$ and $\Upsilon_2 = \text{Diag}(16.67^2, 2.00^2)$. The results in Tables 5 and 6 follow the trends observed in Example 2, that is, the prior distribution seems to only marginally affect the generated design. In this example, there are only differences in the weights of the support points, and even then, they are not very different.

4. Global Optimization-generated Optimal Designs

We provide in this section, NLP formulations for the D -optimal design problem for linear and nonlinear models and note that because the function to be optimized is no longer convex, there may have multiple local optima. Accordingly, we require Global Optimization (GO) solvers. For space consideration, we illustrate the procedure using the D -optimality criterion. The interest in this methodology is that we can search for the global optimal design over a continuous design space $\mathbf{X}$ and use it as a reference to assess the efficiency of SDP generated designs.

The D -optimality criterion seeks to find a design that minimizes the determinant of the inverse of the FIM. This problem is equivalent to finding a design that minimizes the

product of the inverse of the eigenvalues, $\prod_{i=1}^k \frac{1}{\lambda_i}$ [26]. The number of support points of the optimal design is not known a priori and so we used an iterative procedure to find it. Following Dette and Tittof (2009) [50], we set the number of support points, k , in the starting design equal to n_p and solve the problem; if necessary we update the value of k and re-solve the problem until there is no improvement in the objective function for two consecutive values of k or one of the k support points of the design has a null weight. The latter strategy is inspired by the cutting plan algorithm to determine optimal designs of experiments [51].

Using Cholesky decomposition, we write the information matrix $\mathcal{M}(\xi, \boldsymbol{\theta})$ as a product of a unique lower triangular matrix $\mathcal{D}(\xi, \boldsymbol{\theta}) \in \mathbb{R}^{n_p \times n_p}$ and its transpose as

$$\mathcal{M}(\xi, \boldsymbol{\theta}) = \mathcal{D}(\xi, \boldsymbol{\theta}) \mathcal{D}^T(\xi, \boldsymbol{\theta}).$$

Let $\mathcal{D}_{i,i}$ be the i^{th} diagonal entry of $\mathcal{D}(\xi, \boldsymbol{\theta})$. It follows that the determinant of $\mathcal{M}(\xi, \boldsymbol{\theta})$ is

$$\det[\mathcal{M}(\xi, \boldsymbol{\theta})] = \prod_{i=1}^{n_p} \mathcal{D}_{i,i}^2$$

and the D -optimality criterion becomes

$$\log(\det[\mathcal{M}^{-1}(\xi, \boldsymbol{\theta})]) = -2 \sum_{i=1}^{n_p} \log(\mathcal{D}_{i,i}). \quad (23)$$

4.1. Linear models

We now introduce the NLP formulation to find D -optimal designs for linear models. The FIM does not depend on the model parameters and we may denote the derivative of the mean function with respect to $\boldsymbol{\theta}$ by $\mathbf{h}(\mathbf{x})$ with elements $h_j(\mathbf{x})$, $j = 1, \dots, n_p$. If $[n_p] = \{1, \dots, n_p\}$ is the set of indices of the regressors, the formulation of the optimization problem is:

$$\max_{A, \mathcal{D}, \mathbf{x}, \mathbf{w}} 2 \sum_{m=1}^{n_p} \log(\mathcal{D}_{m,m}) \quad (24a)$$

$$\text{s. t. } A_{m,j} = \sum_{i=1}^k w_i h_m(x_i) h_j(x_i), \quad \forall m \in [n_p], j \in \{1, \dots, m\} \quad (24b)$$

$$A_{j,m} = A_{m,j}, \quad \forall m \in [n_p], j \in \{1, \dots, m-1\} \quad (24c)$$

$$A_{m,j} = \sum_{l=1}^{\min(m,j)} \mathcal{D}_{m,l} \mathcal{D}_{j,l}, \quad \forall m, j \in [n_p] \quad (24d)$$

$$\mathcal{D}_{m,j} = 0, \quad \forall m \in [n_p], j \in \{m+1, \dots, n_p\} \quad (24e)$$

$$\sum_{i=1}^k w_i = 1 \quad (24f)$$

$$A \in \mathbb{R}^{n_p \times n_p}, \mathcal{D} \in \mathbb{R}^{n_p \times n_p}, \mathbf{x} = (x_1, \dots, x_k) \in \mathbf{X}^k, \mathbf{w} = (w_1, \dots, w_k) \in [0, 1]^k. \quad (24g)$$

Other constraints, such as symmetry of the design or additional equalities that the design should satisfy, can also be explicitly included in the mathematical program. As always, in our code here and elsewhere, we stipulate a small positive constant, ε , say, equal to 10^{-8} , to ensure the semidefinite positiveness of the FIM during the iteration process.

The problem (24) may have multiple optima even when the number of support points, k , is imposed at the onset. To determine a global optimal design we codified the problem in GAMS [52] and used a multistart heuristic algorithm based solver, OQNLP, to find the global optimum. The algorithm calls a NLP solver from multiple starting points, keeps all the feasible solutions found, and picks the best as the optimum of the problem [53]. The starting points are computed with a random sampling driver that uses normal independent

probability distribution functions for each decision variable. OQNLP does not guarantee that the final solution is a global optimum, but it has been successfully tested in a large set of problems. To build the initial sampling points the variables need to be bounded, which is what we have since the design space and the region of plausible values are all compact by assumption. The NLP solver called by OQNLP is CONOPT, which in turn uses the Generalized Reduced Gradient (GRG) algorithm [54]. The maximum number of starting points allowed is set to 1000 and the procedure terminates when 100 consecutive NLP solver calls result in a only tiny improvement in the criterion value, say, less than 10^{-4} . The absolute and relative tolerances of the solvers were all set equal to 10^{-8} and 10^{-7} , respectively, in all our problems.

We now apply the procedure to Example 1 in §3.1.1, previously handled with the SDP based strategy, and use it to test the NLP formulation (24). NLP formulation produces a design with 8 support points, one point fewer than the D -optimal design resulting from the SDP based approach, see Table 7. We observed that two of the support points obtained with the SDP formulation, the discrete points (0.5300,0.2300,0.2400) and (0.5300,0.2400,0.2300), are collapsed into a single point in the NLP-based design. The weight of the support point that replaces the former two is equal to the sum of the weights of the collapsed points in the SDP design, see the 4th. point on the right column of the Table 7.

From (7), a direct calculation shows the D -efficiency of the design found from the SDP formulation with $x_1 = x_2 = 0.01$ relative to the global optimal design found from the NLP method is 0.9949. This suggests that the SDP-generated design is very close to the optimal design found with the NLP formulation, and should be adequate for practical purposes. Another aspect to mention is that the NLP formulation is computationally intensive; the CPU time is more than 100 times greater than that required by SDP.

4.2. Nonlinear models

To extend the NLP formulation for finding D -optimal designs for nonlinear models, we use nomenclature in §3.2 and apply the Bayesian framework and GQF to compute the expectation. The roots, $\theta_\ell \in \Theta$, and weights, ω_ℓ , $\ell \in [\iota]$ included in the integration are computed in similar way. Different types of prior distributions are considered. A main difference here, unlike the linear model case discussed in §4.1, is that the determinants of the FIMs for all elements in $\theta_\ell \in \Theta$ need to be computed, and consequently we need to factorize the ι FIM matrices via Cholesky decomposition. The NLP is as follows:

$$\max_{A, \mathcal{D}, \mathbf{x}, \mathbf{w}} 2 \sum_{\ell=1}^{\iota} \sum_{m=1}^{n_p} \log(\mathcal{D}_{m,m}^{\ell}) \pi(\theta_{\ell}) \omega_{\ell} \quad (25a)$$

$$\text{s. t. } A_{m,j}^{\ell} = \sum_{i=1}^k w_i h_m(x_i, \theta_{\ell}) h_j(x_i, \theta_{\ell}), \quad \forall m \in [n_p], j \in \{1, \dots, m\}, \ell \in [\iota] \quad (25b)$$

$$A_{j,m}^{\ell} = A_{m,j}^{\ell}, \quad \forall m \in [n_p], j \in \{1, \dots, m-1\}, \ell \in [\iota] \quad (25c)$$

$$A_{m,j}^{\ell} = \sum_{l=1}^{\min(m,j)} \mathcal{D}_{m,l}^{\ell} \mathcal{D}_{j,l}^{\ell}, \quad \forall m, j \in [n_p], \ell \in [\iota] \quad (25d)$$

$$\mathcal{D}_{m,j}^{\ell} = 0, \quad \forall m \in [n_p], j \in \{m+1, \dots, n_p\}, \ell \in [\iota] \quad (25e)$$

$$\sum_{i=1}^k w_i = 1 \quad (25f)$$

$$A^{\ell} \in \mathbb{R}^{n_p \times n_p}, \mathcal{D}^{\ell} \in \mathbb{R}^{n_p \times n_p}, \forall \ell \in [\iota], \mathbf{x} \in \mathbf{X}^k, \mathbf{w} \in [0, 1]^k. \quad (25g)$$

To test the formulation (25) we use the nonlinear model (21) in §3.2.1. The expectation is computed with a six-point GQF, and we adopt both priors used earlier: *i.* three-dimensional uniform distribution $\boldsymbol{\theta} \stackrel{\text{iid}}{\sim} \mathcal{U}(\boldsymbol{\Theta})$ with $\boldsymbol{\Theta} \equiv [1.9, 3.9] \times [9.2, 15.2] \times [1.14, 2.34]$; and *ii.* multivariate normal distribution $\boldsymbol{\theta} \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \Upsilon)$ with $\boldsymbol{\mu} = (2.9, 12.2, 1.74)^T$ and $\Upsilon_1 = \text{Diag}(1/3^2, 1.0^2, 0.2^2)$.

Table 8 displays the D -optimal designs resulting from the NLP-formulation for the uniform and normal prior distributions. The support points of both designs found from the NLP formulations are close even though they have different types of prior distributions. We also observed that two neighboring support points of the SDP-generated design, i.e. (2.0000, 0.5000) and (2.0000, 0.6000), have been collapsed into one in the design found by the NLP-formulation. The NLP-generated design is equally supported at 3 support points, which is the minimum required for the kinetic rate model.

As noted before, the average CPU required by the NLP-formulation is about 12 times greater than that of the SDP-based setup, compare the first column of Tables 3 and 4 with the results in Table 8. Considering that the NLP-generated design is the reference, ξ_D , in (7), the efficiencies of the SDP-generated Bayesian designs for uniform and normal priors, presented in Tables 3 and 4, are 0.9986 and 0.9968, respectively. These results suggest that the SDP-generated Bayesian D -optimal designs, being sub-optimal, have consistently high enough efficiencies for most practical implementations, and yet require considerably lower computational effort than the NLP formulation.

5. Conclusions

This paper discusses a systematic approach based on mathematical programming to find M-bODE using Semidefinite Programming tools and NLP formulations. The latter method is capable of solving nonconvex optimization problems with multiple local optima. However, a GO solver is required to find a global optimum for the problem. For nonlinear models, we adopt a Bayesian approach and expectation was evaluated using GQF. Unlike SDP formulations which require the design space be discretized, the NLP formulation is based on

the Cholesky decomposition of the FIM and is capable of solving problems over a continuous domain.

We demonstrated the two procedures by applying them to find D -, A - and E -optimal designs for some chemical engineering problems. For D -optimality, which is the most common design criterion, our results consistently demonstrate that the NLP formulation produces more efficient designs than those obtained via SDP. However the differences in their efficiencies are typically quite negligible and the computational effort required is one or two order of magnitude lower in the SDP formulation. Consequently, we recommend SDP formulation for practical purposes. We also observed that designs obtained from the SDP formulations typically contain more points than that from the NLP formulation, where the extra points tend to collapse to the support points from the NLP formulations. This may not be surprising because one method assumes a discrete design space and the other does not. A cautionary note in these procedures is that appropriate and reliable solvers are required to solve the optimization problems efficiently.

Acknowledgments

The research of Wong reported in this paper was partially supported by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R01GM107639.

References

1. Fedorov, VV. Theory of Optimal Experiments. Academic Press; 1972.
2. Wynn HP. Results in the theory and construction of D-optimum experimental designs. Journal of Royal Statistics Soc - Ser B. 1972; 34:133–147.
3. Mitchell TJ. An algorithm for the construction of D-optimal experimental designs. Technometrics. 1974; 16:203–210.
4. Galil Z, Kiefer J. Time- and space-saving computer methods, related to Mitchell's DETMAX for finding D-optimum designs. Technometrics. 1980; 22:301–313.
5. Torsney B, Mandal S. Two classes of multiplicative algorithms for constructing optimizing distributions. Computational Statistics & Data Analysis. 2006; 51(3):1591–1601.
6. Dette H, Pepelyshev A, Zhigljavsky AA. Improving updating rules in multiplicative algorithms for computing D-optimal designs. Computational Statistics & Data Analysis. 2008; 53(2):312–320.
7. Cook RD, Nachtsheim CJ. Model robust, linear-optimal designs. Technometrics. 1982; 24:49–54.
8. Pronzato L. Optimal experimental design and some related control problems. Automatica. 2008; 44:303–325.
9. Kiefer J, Wolfowitz J. The equivalence of two extremum problem. Canadian Journal of Mathematics. 1960; 12:363–366.
10. Vandenberghe L, Boyd S. Applications of semidefinite programming. Applied Numerical Mathematics. 1999; 29:283–299.
11. Sagnol G. Computing optimal designs of multiresponse experiments reduces to second-order cone programming. Journal of Statistical Planning and Inference. 2011; 141(5):1684–1708.
12. Papp D. Optimal designs for rational function regression. Journal of the American Statistical Association. 2012; 107:400–411.
13. Duarte BP, Wong W-K. A semi-infinite programming based algorithm for finding minimax optimal designs for nonlinear models. Statistics and Computing. 2014; 24(6):1063–1080.
14. Chaloner K, Larntz K. Optimal Bayesian design applied to logistic regression experiments. Journal of Statistical Planning and Inference. 1989; 59:191–208.
15. Molchanov I, Zuyev S. Steepest descent algorithm in a space of measures. Statistics and Computing. 2002; 12:115–123.

16. Heredia-Langner A, Montgomery DC, Carlyle WM, Borror CM. Model-robust optimal designs: A Genetic Algorithm approach. *Journal of Quality Technology*. 2004; 36:263–279.
17. Zhang, Y. PhD thesis. Virginia Polytechnic Institute and State University; 2006. Bayesian D-Optimal Design for Generalized Linear Models.
18. Boer EPJ, Hendrix EMT. Global optimization problems in optimal design of experiments in regression models. *Journal Global Optimization*. 2000; 18:385–398.
19. Pazman, A. Foundations of Optimum Experimental Design. Reidel Publ. Company; New York: 1986.
20. Whittle P. Some general points in the theory of optimal experimental design. *Journal of the Royal Statistical Society Ser B*. 1973; 35:123–130.
21. Kiefer J. General equivalence theory for optimum design (approximate theory). *Annals of Statistics*. 1974; 2:849–879.
22. Silvey, S. Optimal Design. Chapman & Hall; 1980.
23. Pukelsheim, F. Optimal Design of Experiments. SIAM; Philadelphia: 1993.
24. Box GEP, Hunter WG. The experimental study of physical mechanisms. *Technometrics*. 1965; 7(1):23–42.
25. Dette, H.; Melas, VB.; Strigul, N. Applied Optimal Designs. John Wiley & Sons; 2005. Design of experiments for microbiological models; p. 137–180.
26. Atkinson, AC.; Donev, AN.; Tobias, RD. Optimum Experimental Designs, with SAS. Oxford University Press; Oxford: 2007.
27. Wong W. A unified approach to the construction of minimax designs. *Biometrika*. 1992; 79:611–620.
28. Chaloner K, Verdinelli I. Bayesian experimental design: A review. *Statist Science*. 1995; 10:273–304.
29. Atkinson, KE. An Introduction to Numerical Analysis. 2. John Wiley & Sons; New York: 1989.
30. Boyd, S.; Vandenberghe, L. Convex Optimization. University Press; Cambridge: 2004.
31. Ye, Y. Interior Point Algorithms: Theory and Analysis. John Wiley & Sons; New York: 1997.
32. Vandenberghe L, Boyd S. Semidefinite programming. *SIAM Review*. 1996; 8:49–95.
33. Filová L, Trnovská M, Harman R. Computing maximin efficient experimental designs using the methods of semidefinite programming. *Metrika*. 2011; 64(1):109–119.
34. Qi H. A semidefinite programming study of the Elfving theorem. *Journal of Statistical Planning and Inference*. 2011; 141:3117–3130.
35. Sagnol G. On the semidefinite representation of real functions applied to symmetric matrices. *Linear Algebra and its Applications*. 2013; 439(10):2829–2843.
36. Ben-Tal, A.; Nemirovski, AS. Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications. Society for Industrial and Applied Mathematics; Philadelphia: 2001.
37. Pronzato L. A delimitation of the support of optimal designs for Kiefer's ϕ_p -class of criteria. *Statistics & Probability Letters*. 2013; 83(12):2721–2728.
38. Horst, R.; Pardalos, PM.; Thoai, NV. Introduction to Global Optimization. 2. Springer; Dordrecht: 2000.
39. Tawarlamani, M.; Sahinidis, N. Convexification and Global Optimization in Continuous and Mixed Integer Nonlinear Programming. 1. Kluwer Academic Publishers; Dordrecht: 2002.
40. Zhigljavsky, A.; Žilinskas, A. Stochastic Global Optimization. 1. Springer; New York: 2007.
41. Floudas, CA. Deterministic Global Optimization: Theory, Methods and Applications. 1. Springer; Dordrecht: 1999.
42. Grant, M.; Boyd, S.; Ye, Y. cvx Users Guide for cvx version 1.22. 1104 Claire Ave., Austin, TX 78703-2502: 2012.
43. Sagnol, G. Tech Rep. ZIB; 2012. Picos, a python interface to conic optimization solvers; p. 12–48. <http://picos.zib.de>
44. Sturm J. Using SeDuMi 1.02, a Matlab toolbox for optimization oversymmetric cones. *Optimization Methods and Software*. 1999; 11:625–653.

45. Andersen, E.; Jensen, B.; Jensen, J.; Sandvik, R.; Worsøe, U. Tech rep, Technical Report TR–2009–3. MOSEK; 2009. Mosek version 6.
46. Choinsard L, Géze A, Bigan M, Putaux J, Wouessidjewe D. Efficient size control of amphiphilic cyclodextrin nanoparticles through a statistical mixture design methodology. *J Pharm Pharmaceut Sci.* 2005; 8:593–600.
47. Duarte BPM, Wong WK. Finding Bayesian optimal designs for nonlinear models: A semidefinite programming-based approach. *International Statistical Review.* 2015; 83(2):239–262. [PubMed: 26512159]
48. Poling, BE.; Prausnitz, JM.; O’Connel, JP. *The Properties of Gases and Liquids.* 5. McGraw-Hill; New York: 2001.
49. Hansen HK, Rasmussen P, Fredenslund A, Schiller M, Gmehling J. Vapor-liquid equilibria by UNIFAC group contribution. 5. Revision and extension. *Industrial Engineering Chemistry Research.* 1991; 30:2352–2355.
50. Dette H, Titoff S. Optimal discriminating designs. *Annals of Statistics.* 2009; 37:2056–2081.
51. Gribik PR, Kortanek KO. Equivalence theorems and cutting plane algorithms for a class of experimental design problems. *SIAM J Appl Mathematics.* 1977; 32:232–259.
52. Brooke, A.; Kendrick, D.; Meeraus, A.; Raman, R. *GAMS - A Users Guide.* GAMS Development Corporation; Washington: 1998.
53. Ugray, Z.; Lasdon, L.; Plummer, J.; Glover, F.; Kelly, J.; Martí, R. *Metaheuristic Optimization via Memory and Evolution.* Springer; 2005. A multistart scatter search heuristic for smooth nlp and minlp problems; p. 25-51.
54. Drud A. CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming.* 1985; 31:153–191.

HIGHLIGHTS

- SDP-based formulations for optimal design of experiments;
- NLP-based formulation for D-optimal design of experiments;
- Formulations to handle both linear and nonlinear algebraic models;
- Examples from the areas of Chemistry and Chemical Engineering;
- SDP-based formulation is computationally competitive and accurate.

Table 1

SDP-generated D -, A - and E -optimal designs for the linear mixture model with $x_1 = x_2 = 0.01$ in Example 1.

	D -optimal design	A -optimal design	E -optimal design
	(0.40,0.00,0.60,0.1611)	(0.40,0.00,0.60,0.1542)	(0.40,0.00,0.60,0.1980)
	(0.40,0.30,0.30,0.1527)	(0.40,0.32,0.28,0.0623)	(0.55,0.00,0.45,0.0929)
	(0.40,0.60,0.00,0.1610)	(0.40,0.33,0.27,0.0674)	(0.55,0.01,0.44,0.0555)
	(0.53,0.23,0.24,0.0183)	(0.40,0.60,0.00,0.0322)	(0.55,0.02,0.43,0.0408)
	(0.53,0.24,0.23,0.0270)	(0.54,0.46,0.00,0.0083)	(0.55,0.03,0.42,0.0338)
	(0.56,0.00,0.44,0.0961)	(0.55,0.00,0.45,0.1390)	(0.55,0.45,0.00,0.0568)
	(0.56,0.44,0.00,0.0957)	(0.55,0.45,0.00,0.1214)	(0.70,0.29,0.01,0.0436)
	(0.70,0.00,0.30,0.1439)	(0.70,0.00,0.30,0.0818)	(0.70,0.30,0.00,0.2078)
	(0.70,0.30,0.00,0.1439)	(0.70,0.30,0.00,0.1480)	Additional 26 points
CPU (s)	1.7316	1.4040	1.6068
Optimum	0.00569874	4.0727e-005	5.5149e-005

$(x1.xx, x2.xx, x3.xx, w.wwww) \equiv (\text{design point, weight})$

Table 2

SDP-generated A-, E- and D-optimal designs for the linear mixture model in Example 1 using different discretization schemes.

	Grid: $x_1 = x_2 = 0.02$			Grid: $x_1 = x_2 = 0.1$		
	D-optimal design	A-optimal design	E-optimal design	D-optimal design	A-optimal design	E-optimal design
	(0.40,0.00,0.60,0.1604)	(0.40,0.00,0.60,0.1531)	(0.56,0.14,0.30,0.0190)	(0.40,0.00,0.60,0.1603)	(0.40,0.00,0.60,0.1534)	(0.50,0.50,0.00,0.0487)
	(0.40,0.30,0.30,0.1537)	(0.40,0.32,0.28,0.0240)	(0.56,0.16,0.28,0.0181)	(0.40,0.30,0.30,0.1487)	(0.40,0.30,0.30,0.1165)	(0.60,0.00,0.40,0.0701)
	(0.40,0.60,0.00,0.1605)	(0.40,0.34,0.26,0.1091)	(0.70,0.18,0.12,0.0186)	(0.40,0.60,0.00,0.1604)	(0.40,0.40,0.20,0.0200)	(0.60,0.10,0.30,0.0490)
	(0.52,0.24,0.24,0.0102)	(0.40,0.60,0.00,0.0329)	(0.70,0.20,0.10,0.0206)	(0.50,0.00,0.50,0.0202)	(0.40,0.60,0.00,0.0329)	(0.60,0.20,0.20,0.0387)
	(0.54,0.22,0.24,0.0173)	(0.54,0.18,0.28,0.1822)	(0.70,0.22,0.08,0.0233)	(0.60,0.30,0.10,0.0310)	(0.50,0.20,0.30,0.0263)	(0.50,0.00,0.50,0.0648)
	(0.54,0.24,0.22,0.0187)	(0.54,0.46,0.00,0.1303)	(0.70,0.24,0.06,0.0275)	(0.50,0.30,0.20,0.0264)	(0.50,0.20,0.30,0.1723)	(0.60,0.40,0.00,0.0264)
	(0.56,0.00,0.44,0.0962)	(0.56,0.00,0.46,0.1431)	(0.70,0.26,0.04,0.0345)	(0.50,0.50,0.00,0.0197)	(0.50,0.50,0.00,0.0711)	(0.70,0.00,0.30,0.0285)
	(0.56,0.44,0.00,0.0961)	(0.70,0.00,0.30,0.0820)	(0.70,0.28,0.02,0.0497)	(0.60,0.00,0.40,0.0816)	(0.60,0.00,0.40,0.1069)	(0.70,0.10,0.20,0.0357)
	(0.70,0.00,0.30,0.1434)	(0.70,0.30,0.00,0.1433)	(0.70,0.30,0.00,0.1133)	(0.60,0.40,0.00,0.0821)	(0.60,0.40,0.00,0.0537)	(0.70,0.10,0.10,0.0488)
	(0.70,0.30,0.00,0.1434)	Additional 26 points		(0.70,0.00,0.30,0.1372)	(0.70,0.00,0.30,0.0838)	(0.70,0.30,0.00,0.1077)
CPU (s)	1.4352	0.8268	1.2636	1.2168	0.7644	0.6084
Optimum	0.00569745	4.0621e-005	5.4655e-005	0.0055639	3.4523e-005	4.35901e-5
eff [†]	0.9989	0.9983	0.9910	0.8647	0.8486	0.7904

(x_1 , x_2 , x_3 , x_4) $\equiv$ (design point, weight).

[†] determined with (7-9), the optimal designs were generated with the thinner grid (cf. Table 1).

SDP generated optimal designs for the kinetics of the catalytic dehydration of n-hexyl alcohol with $\theta \stackrel{\text{id}}{\sim} \mathcal{U}(\Theta)$ for different discretization schemes.

Table 3

	Grid: $x_1 = x_2 = 0.1$				Grid: $x_1 = x_2 = 0.2$			
	<i>D</i> -optimal design	<i>A</i> -optimal design	<i>E</i> -optimal design	<i>un</i>	<i>D</i> -optimal design	<i>A</i> -optimal design	<i>E</i> -optimal design	<i>E</i> -optimal design
	(0.30,0.00,0.3335)	(0.20,0.00,0.4432)	(0.20,0.00,0.0896)		(0.20,0.00,0.3333)	(0.20,0.00,0.4804)	(0.20,0.00,0.4257)	
	(2.00,0.00,0.3331)	(0.30,0.00,0.0373)	(0.30,0.00,0.4125)		(2.00,0.00,0.3333)	(2.00,0.00,0.0663)	(0.40,0.00,0.0859)	
	(2.00,0.50,0.0490)	(2.00,0.00,0.0661)	(2.00,0.00,0.0219)		(2.00,0.60,0.3333)	(2.00,0.60,0.4533)	(2.00,0.00,0.0151)	
	(2.00,0.60,0.2843)	(2.00,0.60,0.4534)	(2.00,0.50,0.4760)				(2.00,0.50,0.4734)	
CPU (s)	168.5435	51.6987	16.4581		42.5103	23.93055	5.5848	
Optimum	0.967486	54041.8	3.02988E-5		0.966133	54033.0	2.97844E-5	

(x_1 - x_2 , x_2 , w , w , w) $\equiv$ (design point, weight).

Table 4

SDP generated optimal designs for the kinetics of the catalytic dehydration of n-hexyl alcohol with $\theta \stackrel{\text{iid}}{\sim} \mathfrak{N}(\mu, \Upsilon_1)$ for Example 2 with a grid width of $x_1 = x_2 = 0.1$.

	<i>D</i> -optimal design	<i>A</i> -optimal design	<i>E</i> -optimal design
	(0.30,0.00,0.3333)	(0.20,0.00,0.4089)	(0.20,0.00,0.2001)
	(2.00,0.00,0.3333)	(0.30,0.00,0.0747)	(0.30,0.00,0.3077)
	(2.00,0.50,0.0411)	(2.00,0.00,0.0619)	(2.00,0.00,0.0134)
	(2.00,0.60,0.2923)	(2.00,0.60,0.4545)	(2.00,0.50,0.3967)
			(2.00,0.60,0.0819)
CPU (s)	133.7084	61.32399	16.9729
Optimum	0.986152	46486.9	2.69544E-5

$(x1.xx, x2.xx, w.wwww) \equiv (\text{design point, weight})$.

SDP generated A-, E- and D-optimal designs for the activity coefficient of a binary mixture of n-pentane/acetone with $\theta \stackrel{\text{iid}}{\sim} \mathcal{U}_2(\Theta)$ for different discretization schemes.

Table 5

	Grid: $x_1 = 0.1, x_2 = 1.0$			Grid: $x_1 = 0.2, x_2 = 1.0$		
	D-optimal design	A-optimal design	E-optimal design	D-optimal design	A-optimal design	E-optimal design
	(0.14,318.0.0.1207)	(0.12,318.0.0.5971)	(0.11,318.0.0.6058)	(0.14,318.0.0.0633)	(0.12,318.0.0.5971)	(0.12,318.0.0.5978)
	(0.48,318.0.0.3844)	(0.90,318.0.0.4029)	(0.90,318.0.0.3942)	(0.48,318.0.0.4398)	(0.90,318.0.0.4029)	(0.90,318.0.0.4022)
	(0.87,318.0.0.4949)			(0.88,318.0.0.4969)		
CPU (s)	257.0740	252.6124	250.3660	123.3812	121.8523	121.5872
Optimum	0.996786	32.9628	0.0324719	0.9967833	32.9628	0.0323780

(x1,xx,x2xx,x,w,wwww) $\equiv$ (design point, weight).

Table 6

SDP generated A -, E - and D -optimal designs for the activity coefficient of a binary mixture of n-pentane/acetone with $\theta \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \Upsilon_2)$, for $x_1 = 0.1$ and $x_2 = 1.0$.

	<i>D</i> -optimal design	<i>A</i> -optimal design	<i>E</i> -optimal design
	(0.1400,318.0,0.1192)	(0.1200,318.0,0.5962)	(0.1100,318.0,0.6068)
	(0.4800,318.0,0.3857)	(0.9000,318.0,0.4038)	(0.9000,318.0,0.3932)
	(0.8700,318.0,0.4951)		
CPU (s)	258.9304	254.4220	253.5640
Optimum	0.9967869	32.4097	0.032287

$(x1.xx, x2xx.x, w.wwww) \equiv (\text{design point, weight})$.

Table 7

SDP (based on a grid with $x_1 = x_2 = 0.01$) and NLP-generated D -optimal designs for the linear mixture model for Example 1.

	SDP-generated design	NLP-generated design
	(0.4000,0.0000,0.6000,0.1605)	(0.4000,0.0000,0.6000,0.1601)
	(0.4000,0.3000,0.3000,0.1528)	(0.4000,0.3000,0.3000,0.1529)
	(0.4000,0.6000,0.0000,0.1605)	(0.4000,0.6000,0.0000,0.1601)
	(0.5300,0.2300,0.2400,0.0234)	(0.5313,0.2343,0.2344,0.0475)
	(0.5300,0.2400,0.2300,0.0236)	(0.5569,0.0000,0.4431,0.0955)
	(0.5600,0.0000,0.4400,0.0961)	(0.5569,0.4431,0.0000,0.0955)
	(0.5600,0.4400,0.0000,0.0961)	(0.7000,0.0000,0.3000,0.1442)
	(0.7000,0.0000,0.3000,0.1435)	(0.7000,0.3000,0.0000,0.1442)
	(0.7000,0.3000,0.0000,0.1435)	
CPU (s)	2.2152	334.6080
Optimum	0.00569874	0.00574001

$(x1.xxxx, x2.xxxx, x3.xxxx, w.wwww) \equiv (\text{design point, weight})$.

Table 8

NLP-generated Bayesian D -optimal designs for the kinetics of the catalytic dehydration of n-hexyl alcohol for different priors.

	Prior	
	$\theta \stackrel{\text{iid}}{\sim} \mathcal{U}(\Theta)$	$\theta \stackrel{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{\mu}, \Upsilon_1)$
	(0.2597,0.0000,0.3333)	(0.2575,0.0000,0.3333)
	(2.0000,0.0000,0.3333)	(2.0000,0.0000,0.3333)
	(2.0000,0.5549,0.3333)	(2.0000,0.5566,0.3333)
CPU (s)	3002.334	2668.5240
Optimum	0.968842	0.989318

$(x1.xxx, x2.xxx, w.wwww) \equiv (\text{design point, weight})$.