Statistics and Computing (2020) 30:93–112
https://doi.org/10.1007/s11222-019-09867-z

Optimal exact designs of experiments via Mixed Integer Nonlinear Programming

Belmiro P. M. Duarte[1,2] · José F. O. Granjo[2] · Weng Kee Wong[3]

Received: 19 January 2018 / Accepted: 6 March 2019 / Published online: 13 March 2019

Abstract

Optimal exact designs are problematic to find and study because there is no unified theory for determining them and studying their properties. Each has its own challenges and when a method exists to confirm the design optimality, it is invariably applicable to the particular problem only. We propose a systematic approach to construct optimal exact designs by incorporating the Cholesky decomposition of the Fisher Information Matrix in a Mixed Integer Nonlinear Programming formulation. As examples, we apply the methodology to find D- and A-optimal exact designs for linear and nonlinear models using global or local optimizers. Our examples include design problems with constraints on the locations or the number of replicates at the optimal design points.

Keywords Model-based optimal designs · Exact designs · Constrained designs · Mixed Integer Nonlinear Programming · Global Optimization

Mathematics Subject Classification 62K05 · 90C47

1 Motivation

Optimal design of experiments (DoE) is an old yet increasingly important subfield of statistics. Running experiments is costly, and users want to rein in cost without sacrificing statistical efficiency in their inference. Finding a formula or an analytic description of an optimal exact design for a nonlinear model is usually impossible due to technical difficulties. A practical approach is to develop efficient algorithms for finding optimal exact designs. In DoE, given a statistical model, a fixed total number of observations N and an optimality criterion, we seek the optimal number of design points, k, their locations from a pre-specified compact design space $\mathbf{X}$ and the number of replicates at each design point subject to the constraint that the number of replicates sum to N. Such an optimal design provides maximal precision for statistical inference at minimum cost (Fedorov and Leonov 2014).

There are two types of designs: large sample or approximate designs and small sample or exact designs. The former are essentially probability measures on the design space and are easier to find. In particular, when the optimality criterion is convex over the design space, we have a convex optimization problem (Kiefer 1974) and there are algorithms for searching the optimal approximate designs, including analytical tools for studying their properties and confirming optimality of the design. In practice, each of the weights at an optimal design point in an approximate design has to be multiplied by N and rounded to an integer such that they all sum to N before such an optimal approximate design can

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11222-019-09867-z) contains supplementary material, which is available to authorized users.

✉ Belmiro P. M. Duarte
bduarte@isec.pt

José F. O. Granjo
josegranjo@eq.uc.pt

Weng Kee Wong
wkwong@ucla.edu

[1] Department of Chemical and Biological Engineering, Instituto Politécnico de Coimbra, Instituto Superior de Engenharia de Coimbra, Rua Pedro Nunes, Quinta da Nora, 3030–199 Coimbra, Portugal

[2] Department of Chemical Engineering, CIEPQPF, University of Coimbra, Rua Sílvio Lima – Pólo II, 3030-790 Coimbra, Portugal

[3] Department of Biostatistics, Fielding School of Public Health, UCLA, 10833 Le Conte Ave., Los Angeles, CA 90095–1772, USA

be implemented; Kiefer (1974), Pukelsheim (1993), Vandenberghe and Boyd (1999) provide details.

In optimal exact design problems, the numbers of observations at design points are integers and they sum to N. Consequently, we do not have a convex optimization problem in general and, so, finding optimal exact designs are computationally more challenging than finding approximate optimal designs (Boer and Hendrix 2000). Each optimal exact design problem has its own unique technical features that depend on the model, design criterion and N; analytic solutions are available only for some simple models, and there are no general algorithms for finding optimal exact designs (Gribik and Kortanek 1977). However, optimal exact designs are important in their own right because (i) simply rounding the approximate number of observations to take at each point from an optimal approximate design to obtain an optimal exact design may result in an inefficient design when N is small (Imhof et al. 2001); and (ii) even with a moderate sample size, a rounded optimal approximate design may not be implementable when some of the optimal proportions at some design points are very small.

There are various numerical algorithms for finding optimal *exact designs* based on exchange methods. They were initially proposed for the D-optimality criterion (Mitchell and Miller 1970; Wynn 1970; Fedorov 1972) and mainly for linear models. An appropriate starting design is required to initialize these algorithms and afterward, at each iteration, they delete points from the current design and add new points from a specified grid until a user specified stopping rule is met. The algorithms do not guarantee global optimality, and so they are run a few times using different starting designs to ascertain that they converge to the same optimal design. Refinements to these algorithms were continually made and include the DETMAX algorithm (Mitchell 1974), the modified Fedorov algorithm (Cook and Nachtsheim 1980) and the KL-exchange algorithm (Atkinson and Donev 1989). These are generally referred to as point-exchange algorithms, and they have been used to construct response surface designs involving random blocks (Goos and Vandebroek 2001; Goos and Donev 2006), D-optimal split-plot designs (Goos and Vandebroek 2003) or crossover designs (Donev 1998). The coordinate-exchange algorithm proposed by Meyer and Nachtsheim (1995) tackled some of the problems the point-exchange algorithms have by avoiding the explicit enumeration of candidate points where this problem occurred for both continuous and discrete and continuous factors. However, like other exchange algorithms, it still has the tendency that it gets trapped in a locally optimal design (Mandal et al. 2015; Palhazi Cuervo et al. 2016).

Mathematical programming methods provide alternative approaches to generate optimal exact designs and have been successfully applied to search for both approximate and factorial designs. Some examples for finding approximate designs include Linear Programming (Harman and Jurík 2008), Second-Order Cone Programming (Sagnol 2011), Semidefinite Programming (Vandenberghe and Boyd 1999; Duarte and Wong 2015), Semi-Infinite Programming (Duarte et al. 2015) and Nonlinear Programming (NLP) (Molchanov and Zuyev 2002). Recently, Mixed Integer Programming was also used to find mixed-level and two-level orthogonally blocked designs (Sartono et al. 2015b), orthogonal fractional factorial split-plot designs (Sartono et al. 2015a; Vo-Thanh et al. 2018) and trend robust run-order designs (Nuñez Ares and Goos 2008).

Applications of mathematical programming methods to find optimal exact designs in a general regression setting are less numerous due to the additional complexity. In Welch (1982), the design space is discretized and a convex optimization algorithm based on branch and bound is used to ensure that the optimal numbers of replicates of the D-optimal exact designs are integers. Similarly, Harman and Filová (2014) and Sagnol and Harman (2015) used, respectively, Mixed Integer Quadratic Programming (MIQP) and Mixed Integer Second-Order Cone Programming techniques (MISOCP) to find D-optimal exact designs. Both methods also require discretizing the design space, ensuring that the global optimal design is found on the discretized space. The resulting optimization problem can be solved efficiently using state-of-the-art solvers employing branch-and-bound techniques. Potential issues with the aforementioned methods include the exponential increase in the size of the optimization problem when the number of factors and the size of the candidate set of design points increase. Although coarser grids could reduce the size of the optimization problem, the resulting design may not be supported at the correct design points and can become inefficient.

Esteban-Bravo et al. (2017) showed that NLP formulations can be used to find unconstrained and constrained exact designs, and that Newton-based methods using Interior Point or Filter techniques performed well for the problem. Their formulation for finding optimal exact designs is that for finding approximate designs with the caveat that the weights are rational numbers so that all replicates are integers that sum to N, see also Leszkiewicz (2014). A general drawback of the NLP formulations is that the numerical calculation of the design criteria, such as those involving the determinant or trace of the inverse of a matrix, may suffer from numerical instabilities that result in multiple optima.

Our aim in this paper is to provide a general mathematical programming framework based on Mixed Integer Nonlinear Programming (MINLP) to find various types of optimal exact designs for a broad class of models. Some key advantages of our approach are: (i) unlike the MIQP and MISOCP formulations, our method does not require the design space to be discretized before optimization; (ii) the design points and the replicates are optimized simultaneously which leads to globally optimal designs in the original design space; and (iii) the method is flexible in that it can readily include additional user-defined constraints, such as restrictions on the distribution of the locations or the number of replicates at the design points. Moreover, the Cholesky decomposition to compute the determinant of the Fisher Information Matrix (FIM) is directly embedded into the optimization problem. From our knowledge, this is also new and, as it will be shown, improves the numerical stability and accuracy of the algorithm.

Section 2 presents the statistical background and reviews the general MINLP problem. In Sect. 3, we provide the formulations for D- and A-optimal exact design problems and extend them to find constrained optimal exact designs with various types of constraints. Section 4 implements the algorithm to generate optimal exact designs for linear models and locally optimal exact designs for several nonlinear models. Section 5 concludes with a summary.

2 Background

All our regression models addressed have a univariate response with n_x independent variables $\boldsymbol{x} \in \mathbf{X} \subset \mathbb{R}^{n_x}$, and the mean response at $\boldsymbol{x}$ is

$$\mathbb{E}[y|\boldsymbol{x}, \boldsymbol{p}] = f(\boldsymbol{x}, \boldsymbol{p}). \tag{1}$$

Here, $f(\boldsymbol{x}, \boldsymbol{p})$ is a known differentiable function, apart from a vector of unknown model parameters $\boldsymbol{p} \in \mathbf{P} \subset \mathbb{R}^{n_\theta}$ where n_θ is number of parameters in the model, and $\mathbb{E}[\bullet]$ is the expectation operator with respect to the error distribution. The design space $\mathbf{X}$ is a known compact domain from which the design points are selected to observe the N outcomes.

Let ξ be a k-point exact design supported at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_k$ in $\mathbf{X}$ with n_i replicates at $\boldsymbol{x}_i$ subject to $\sum_{i=1}^{k} n_i = N$. Henceforth, we assume the number k of support points in the design sought is user specified, and an initial estimate for k is the number of parameters in the model, n_θ. In what follows, let $\boldsymbol{n}$ be the vector of all possible replicates at the design points, let $\Omega_k^N = \{n_i : \ 0 \leq n_i < N, \ \sum_{i=1}^{k} n_i = N, \ 1 \leq i \leq k\}$ and let $\Xi_k^N \equiv \mathbf{X}^k \times \Omega_k^N$ be the set of all k-point feasible designs on $\mathbf{X}$. We assume $k \geq n_\theta$; otherwise, there are not enough support points to estimate all the parameters in the model. The assumption that the number of support points is pre-specified by the user is a limitation, but it is sometimes not the case in practice. For instance, it can be expensive to take observations at a new design point and so it is desirable to limit the number of design points in the study; at other times, this fixed number of design points is naturally restricted. In biomedical studies, examples include the number of times blood samples can be drawn from an infant or an experimental drug is only available in a fixed number of dosages. Alternatively, since running the algorithm with various values of k presents no new technical difficulties, one may run the algorithm using different values of k and compare properties of the different optimal designs before implementation.

The quality of the design ξ is measured by a convex function of its FIM, which is the matrix with elements equal to the negative of the expectation of the second-order derivatives of the log-likelihood of all observed data with respect to the parameters. When responses are independent, the global FIM of design ξ is

$$\begin{aligned}\mathscr{M}(\xi, \boldsymbol{p}) &= -\mathbb{E}\left[\frac{\partial}{\partial \boldsymbol{p}}\left(\frac{\partial \mathscr{L}(\xi, \boldsymbol{p})}{\partial \boldsymbol{p}^{\mathrm{T}}}\right)\right] = \sum_{i=1}^{k} \frac{n_i}{N} M(\delta_{\boldsymbol{x}_i}, \boldsymbol{p}) \\ &= \sum_{i=1}^{k} \frac{n_i}{N} \boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{p})\, \boldsymbol{h}^{\mathsf{T}}(\boldsymbol{x}_i, \boldsymbol{p})\end{aligned} \tag{2}$$

where $\mathscr{L}(\xi, \boldsymbol{p})$ is the log-likelihood function of the observed responses using design ξ, $\delta_{\boldsymbol{x}}$ is the degenerate design that puts all its mass at $\boldsymbol{x}$, $M(\delta_{\boldsymbol{x}_i}, \boldsymbol{p})$ is the elemental FIM at $\boldsymbol{x}_i$ and $\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{p})$ is the vector of first-order derivatives of the log-likelihood with respect to the parameters $\boldsymbol{p}$ at $\boldsymbol{x}_i$.

2.1 Optimality criteria

Here, we describe the design criteria and, when applicable, the theoretical tools to confirm the optimality of a design.

When errors are identically and normally distributed with independent errors, the volume of the confidence region for $\boldsymbol{p}$ is inversely proportional to $\det[\mathscr{M}^{1/2}(\xi, \boldsymbol{p})]$. Consequently, minimizing the determinant of the inverse of the FIM by choice of a design leads to the most accurate estimates for the parameters. If the interest is in finding the best k-point D-optimal exact design, the optimization problem is

$$\xi_D = \arg \min_{\xi \in \Xi_k^N} \log\left\{\det[\mathscr{M}(\xi, \boldsymbol{p})^{-1}]\right\}, \tag{3}$$

which is equivalent to maximizing the determinant of the FIM,

$$\xi_D = \arg \max_{\xi \in \Xi_k^N} \log\left\{\det[\mathscr{M}(\xi, \boldsymbol{p})]\right\}. \tag{4}$$

Another common design criterion is A-optimality that minimizes the sum of the variances of all the estimated parameters in the mean function. Such a design ξ_A satisfies

$$\xi_A = \arg \min_{\xi \in \Xi_k^N} \left\{\mathrm{tr}[\mathscr{M}(\xi, \boldsymbol{p})^{-1}]\right\}. \tag{5}$$

For a linear model, the FIM does not depend on the unknown parameters and so the optimal design does not depend on $\boldsymbol{p}$. If it does, as when the model is nonlinear, nominal values based on experts' opinion, previous or similar studies are used to replace $\boldsymbol{p}$ before the criterion is optimized. The resulting design is called a locally optimal design because it depends on the nominal values (Chernoff 1953).

Approximate designs are similar to exact designs except that each n_i is replaced by w_i, the proportion of observations to be taken at $\boldsymbol{x}_i$. They are implemented by taking $N \times w_i$ observations at $\boldsymbol{x}_i$ subject to each $N \times w_i$ is an integer, and they sum to N. When an optimal design is sought among all approximate designs on the design space and the design criterion is convex, the optimality of an approximate design ξ can be verified using an equivalence theorem based on directional derivative considerations (Fedorov 1972; Kiefer 1974; Pukelsheim 1993). If we let $\delta_{\boldsymbol{x}}$ be the degenerate design at the point $\boldsymbol{x} \in \mathbf{X}$, the equivalence theorem says ξ_D is a D-optimal approximate design among all designs on $\mathbf{X}$ if and only if

$$\Psi(\boldsymbol{x}|\xi_D, \boldsymbol{p}) = \text{tr}\left\{[\mathscr{M}(\xi_D, \boldsymbol{p})]^{-1} M(\delta_{\boldsymbol{x}}, \boldsymbol{p})\right\} \leq n_\theta, \quad \forall \boldsymbol{x} \in \mathbf{X}, \tag{6}$$

with equality at every design point of ξ_D. The function $\Psi(\boldsymbol{x}|\xi, \boldsymbol{p})$ is the *dispersion function* of the design ξ. In Sect. 4.2, we display selected dispersion functions of the optimal exact designs we found and show that they are equivalent to optimal approximate designs when N is a multiple of n_θ.

To compare two designs ξ_1 and ξ_2, we use their relative efficiencies. For the D-optimality criterion, the D-efficiency of ξ_1 relative to ξ_2 is defined by

$$\text{Eff}_D(\xi_1, \xi_2) = \left(\frac{\exp\left[\log(\det[\mathscr{M}(\xi_1, \boldsymbol{p})])\right]}{\exp\left[\log(\det[\mathscr{M}(\xi_2, \boldsymbol{p})])\right]}\right)^{\frac{1}{n_\theta}}, \tag{7}$$

and, similarly, for A-optimality criterion, the efficiency of ξ_1 relative to ξ_2 is defined by

$$\text{Eff}_A(\xi_1, \xi_2) = \frac{\text{tr}[\mathscr{M}^{-1}(\xi_2, \boldsymbol{p})]}{\text{tr}[\mathscr{M}^{-1}(\xi_1, \boldsymbol{p})]}. \tag{8}$$

If ξ_2 is the optimal design sought, the above ratios become the D-efficiency and A-efficiency of ξ_1, respectively. The interpretation of an efficiency is that if a design has an efficiency of 50%, it needs to be replicated twice to do as well as the optimal design (Fedorov 1972; Pukelsheim 1993). In practice, designs with high efficiencies are desired.

2.2 Mixed Integer Nonlinear Programming

MINLP is a class of tools to optimize a nonlinear objective function $\psi(\boldsymbol{x}, \boldsymbol{y})$ with possibly nonlinear constraints where some decision variables are integer. MINLPs arise in a wide range of applications, including chemical engineering, finance and management. If there are n_x continuous variables and n_y discrete variables to optimize, the general form of a MINLP is

$$\min_{\boldsymbol{x},\boldsymbol{y}} \ \psi(\boldsymbol{x}, \boldsymbol{y}) \tag{9a}$$

$$\text{s.t. } v_i(\boldsymbol{x}, \boldsymbol{y}) = 0, \quad \forall i \in \mathscr{E} \tag{9b}$$

$$g_i(\boldsymbol{x}, \boldsymbol{y}) \leq 0, \quad \forall i \in \mathscr{I} \tag{9c}$$

$$\boldsymbol{x} \in \mathbf{X}, \ \ \boldsymbol{y} \in \mathbf{Y}. \tag{9d}$$

Here, the functions $v_i(\boldsymbol{x}, \boldsymbol{y})$ and $g_i(\boldsymbol{x}, \boldsymbol{y})$ map $\mathbb{R}^{n_x+n_y}$ to $\mathbb{R}$, $\mathscr{E}$ is the set of equality constraints, $\mathscr{I}$ the set of inequalities, $\mathbf{X}$ is a compact set containing continuous variables $\boldsymbol{x}$, and $\mathbf{Y}$ is the set containing discrete variables $\boldsymbol{y}$.

Some common algorithms to solve Mixed Integer Nonlinear Programs are the outer-approximation method (Duran and Grossmann 1986), the branch-and-bound method (Fletcher and Leyffer 1998) and the extended cutting plane method (Westerlund and Pettersson 1995). Floudas (2002) reviews the fundamentals of using MINLP to solve optimization problems and note that traditional MINLP algorithms guarantee the global optima under certain convexity assumptions.

Our design problems may have multiple local optima and to guarantee that a global optimum is found, a global solver must be employed. An example of a global solver is BARON. It implements deterministic global optimization algorithms that combine spatial branch-and-bound procedures and bound tightening methods via constraint propagation and interval analysis in a *branch-and-reduce* technique (Tawarlamani and Sahinidis 2002). Sahinidis (2014) showed that these techniques work quite well under fairly general assumptions. In our formulations, those assumptions are satisfied by construction as all decision variables are bounded. However, global optimization solvers still require a long computational time compared to local solvers (Lastusilta et al. 2007) and this may limit their utilization to small and average-sized problems.

A potential way to shorten the CPU time is to use a local MINLP solver, such as, SBB (GAMS Development Corporation 2013b), for handling several design problems addressed in the paper, see Sect. 4.2. In Sect. 4.1, we compare optimal designs obtained with local and global MINLP solvers, where BARON represents the latter class and SBB the former. Both MINLP tools use CONOPT as a NLP solver to handle the relaxed nonlinear programs (Drud 1985) and CPLEX to solve the Mixed Integer Linear Programs (GAMS Development Corporation 2013b).

3 Optimal design MINLP formulations

We introduce MINLP formulations for finding a k-point D- and A-optimal exact design in Sects. 3.1 and 3.2, respectively. We first focus on solving unconstrained optimal designs before we demonstrate how MINLP formulations can solve constrained optimization problems with user-selected restrictions. In Sect. 3.3, we review a common approach to initialize consistently the MINLP problems, tools for solving the optimization problems and discuss limitations of the formulations.

3.1 *D*-optimal designs

Given a statistical model and values for k and N, let $\mathscr{M}(\xi, \boldsymbol{p})$ be the FIM in (2). The formulation for finding a k-point D-optimal exact designs on Ξ_k^N may be equivalently formulated as an optimization problem as follows:

$$\max_{\xi\in\Xi_k^N} \ \log\left(\det[\mathscr{M}(\xi, \boldsymbol{p})]\right) \tag{10a}$$

$$\text{s.t.} \ \sum_{l=1}^{k} n_l = N$$
$$\boldsymbol{x} \in \mathbf{X}, \ \boldsymbol{n} \in \Omega_k^N. \tag{10b}$$

To maximize $\log\left(\det[\mathscr{M}(\xi, \boldsymbol{p})]\right)$, we apply the Cholesky decomposition to the FIM and write

$$\mathscr{M}(\xi, \boldsymbol{p}) = \mathscr{U}^{\mathsf{T}}(\xi, \boldsymbol{p})\ \mathscr{U}(\xi, \boldsymbol{p}) \tag{11}$$

where $\mathscr{U}(\xi, \boldsymbol{p})$ is an upper triangular matrix and has positive diagonal elements $u_{i,i}$ when FIM is positive definite. It follows that

$$\det(\mathscr{M}(\xi, \boldsymbol{p})) = \prod_{i=1}^{n_\theta} u_{i,i}^2, \tag{12}$$

$\log[\det(\mathscr{M}(\xi, \boldsymbol{p}))] = 2\sum_{i=1}^{n_\theta}\log(u_{i,i})$ and maximizing $\det(\mathscr{M}(\xi, \boldsymbol{p}))$ is equivalent to maximizing the sum of the logarithm of the diagonal elements of $\mathscr{U}(\xi, \boldsymbol{p})$.

Hereafter, we write $i, j \in [\![n_\theta]\!]$ to represent indices i, j running from 1 to n_θ. Similarly, the notation $i \in [\![n_\theta]\!]$ means the single index i runs from 1 to n_θ. If $u_{i,j}$ is the (i, j)th element of $\mathscr{U}(\xi, \boldsymbol{p}), i, j \in [\![n_\theta]\!]$, our MINLP formulation for finding an D-optimal exact design is

$$\max_{\boldsymbol{x},\boldsymbol{n}} \sum_{i=1}^{n_\theta} \log(u_{i,i}) \tag{13a}$$

$$\text{s.t.}\ m_{i,j} = \sum_{l=1}^{k} \frac{n_l}{N} h_{l,i}(\boldsymbol{x}, \boldsymbol{p})\ h_{l,j}(\boldsymbol{x}, \boldsymbol{p}), \quad i,\ j \in [\![n_\theta]\!] \tag{13b}$$

$$m_{i,j} = \sum_{l=1}^{k} u_{l,i} u_{l,j}, \quad i,\ j \in [\![n_\theta]\!],\ i \le j \tag{13c}$$

$$u_{i,i} \ge \epsilon, \quad i \in [\![n_\theta]\!] \tag{13d}$$

$$u_{i,j} = 0, \quad i,\ j \in [\![n_\theta]\!],\ i \ge j+1 \tag{13e}$$

$$m_{i,i} \ge u_{i,j}^2, \quad i, j \in [\![n_\theta]\!] \tag{13f}$$

$$\sum_{l=1}^{k} n_l = N$$
$$\boldsymbol{x} \in \mathbf{X},\ \boldsymbol{n} \in \Omega_k^N. \tag{13g}$$

Here, ϵ is a small positive constant to ensure that the FIM is positive definite. For all examples in Sect. 4, $\epsilon = 1 \times 10^{-5}$. Equation (13b) follows from (2), (13c) represents the Cholesky decomposition, (13d) guarantees that all diagonal elements of $\mathscr{U}(\xi, \boldsymbol{p})$ are positive and (13e) assures that $\mathscr{U}(\xi, \boldsymbol{p})$ is upper triangular. Equation (13f) is a numerical stability condition imposed on the Cholesky factorization of positive semidefinite matrices (Golub and van Loan 2013, Theorem 4.2.8) and constraint (13g) restricts the total number of observations to be N and ensures that the design points belong to design space $\mathbf{X}$.

3.2 *A*-optimal design

A-optimal exact experimental plans may be formulated as follows:

$$\min_{\xi\in\Xi_k^N} \ \mathrm{tr}[\mathscr{M}^{-1}(\xi, \boldsymbol{p})] \tag{14a}$$

$$\text{s.t.} \ \sum_{l=1}^{k} n_l = N$$
$$\boldsymbol{x} \in \mathbf{X}, \ \boldsymbol{n} \in \Omega_k^N. \tag{14b}$$

The optimization problem (14) requires inverting $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ which is a potentially numerically unstable operation when the FIM is ill-conditioned. To avoid explicit computation of the inverse matrix, we apply the Cholesky decomposition to invert the resulting upper diagonal matrix $\mathscr{U}(\xi, \boldsymbol{p})$ that results from the decomposition of $\mathscr{M}(\xi, \boldsymbol{p})$; the rationale is that inverting an upper triangular matrix obtained by Cholesky factorization is numerically more stable than inverting the original matrix (Du Croz and Higham 1992). The procedure has three steps that are handled simultaneously within the optimization problem: (i) apply the Cholesky decomposition to the FIM, cf. Sect. 3.1; (ii) invert the upper triangular matrix $\mathscr{U}(\xi, \boldsymbol{p})$ using the relation $\mathscr{U}(\xi, \boldsymbol{p})\ \mathscr{U}^{-1}(\xi, \boldsymbol{p}) = I_{n_\theta}$, where I_{n_θ} is the n_θ-dimensional identity matrix; and (iii) compute $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ via $\mathscr{U}^{-1}(\xi, \boldsymbol{p})$, i.e., $\mathscr{M}^{-1}(\xi, \boldsymbol{p}) = \mathscr{U}^{-1}(\xi, \boldsymbol{p}) \times [\mathscr{U}^{-1}(\xi, \boldsymbol{p})]^{\mathsf{T}}$ (Du Croz and Higham 1992), and, finally, compute $\mathrm{tr}[\mathscr{M}^{-1}(\xi, \boldsymbol{p})]$.

Let $\overline{m}_{i,j}$ be the (i, j)th entry of $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ and $\overline{u}_{i,j}$ be the (i, j)th entry of $\mathscr{U}^{-1}(\xi, \boldsymbol{p})$ where $i,\ j \in [\![n_\theta]\!]$. By construction, $\mathscr{U}(\xi, \boldsymbol{p})$ is positive definite and invertible if all the diagonal elements are positive. The same also holds for $\mathscr{U}^{-1}(\xi, \boldsymbol{p})$. Step (i) is the Cholesky decomposition of the FIM represented by (13c), and the second step corresponds to inverting $\mathscr{U}(\xi, \boldsymbol{p})$:

$$\begin{cases} \sum_{l=1}^{k} u_{i,l}\, \overline{u}_{l,j} = 1 & \text{if } i = j \\ \sum_{l=1}^{k} u_{i,l}\, \overline{u}_{l,j} = 0 & \text{if } i \neq j, \end{cases} \tag{15}$$

with step (iii) represented by

$$\overline{m}_{i,j} = \sum_{l=1}^{k} \overline{u}_{i,l} \overline{u}_{l,j}, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \leq j. \tag{16}$$

A-optimal designs minimize $\mathrm{tr}(\mathscr{M}^{-1}(\xi, \boldsymbol{p}))$ or equivalently, minimizes the sum of all $\overline{m}_{i,i},\ i \in \llbracket n_\theta \rrbracket$. The complete MINLP for computing the A-optimal designs is

$$\min_{\boldsymbol{x},\boldsymbol{n}} \sum_{i=1}^{n_\theta} \overline{m}_{i,i} \tag{17a}$$

$$\text{s.t. } m_{i,j} = \sum_{l=1}^{k} \frac{n_l}{N} h_{l,i}(\boldsymbol{x}, \boldsymbol{p})\, h_{l,j}(\boldsymbol{x}, \boldsymbol{p}), \quad i,\ j \in \llbracket n_\theta \rrbracket \tag{17b}$$

$$m_{i,j} = \sum_{l=1}^{n_\theta} u_{l,i} u_{l,j}, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \leq j \tag{17c}$$

$$\sum_{l=1}^{k} u_{l,i}\, \overline{u}_{l,j} = 1, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i = j \tag{17d}$$

$$\sum_{l=1}^{k} u_{i,l}\, \overline{u}_{l,j} = 0, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \neq j \tag{17e}$$

$$\overline{m}_{i,j} = \sum_{l=1}^{k} \overline{u}_{i,l} \overline{u}_{l,j}, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \leq j \tag{17f}$$

$$u_{i,i} \geq \epsilon, \quad i \in \llbracket n_\theta \rrbracket \tag{17g}$$

$$\overline{u}_{i,i} \geq \epsilon, \quad i \in \llbracket n_\theta \rrbracket \tag{17h}$$

$$u_{i,j} = 0, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \geq j+1 \tag{17i}$$

$$\overline{u}_{i,j} = 0, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \geq j+1 \tag{17j}$$

$$\overline{m}_{i,j} = \overline{m}_{j,i}, \quad i,\ j \in \llbracket n_\theta \rrbracket,\ i \leq j-1 \tag{17k}$$

$$m_{i,i} \geq u_{i,j}^2, \quad i, j \in \llbracket n_\theta \rrbracket \tag{17l}$$

$$\overline{m}_{i,i} \geq \overline{u}_{i,j}^2, \quad i, j \in \llbracket n_\theta \rrbracket \tag{17m}$$

$$\sum_{l=1}^{k} n_l = N$$

$$\boldsymbol{x} \in \mathbf{X},\ \boldsymbol{n} \in \Omega_k^N. \tag{17n}$$

Equations (17b, 17c, 17g, 17i, 17l) and (17n) are similar to those in the D-optimal design formulation. Equations (17d–17e) reflect relationship (15) and generate $\mathscr{U}^{-1}(\xi, \boldsymbol{p})$, equation (17k) captures constraint (16) to produce $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ and equations (17i) and (17j), respectively, impose the lower triangular structure of $\mathscr{U}(\xi, \boldsymbol{p})$ and $\mathscr{U}^{-1}(\xi, \boldsymbol{p})$. Equation (17k) imposes the symmetry on $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ and both equations (17g) and (17h) ensure that the diagonal elements of $\mathscr{U}^{-1}(\xi, \boldsymbol{p})$ and $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ are positive, respectively. Condition (17m) is the numerical stability insurance for the Cholesky factorization of $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$. We impose the symmetry of $\mathscr{M}^{-1}(\xi, \boldsymbol{p})$ in Equation (17k) to reduce the feasibility region which may then help to improve the convergence rate of the solver. The symmetry of the FIM and its inverse are guaranteed by (17b) and (17k), respectively.

An advantage of our approach is that when there are additional constraints, such as restrictions on the number of replicates at each design point, they can be incorporated into our design formulation problem as linear or nonlinear inequalities or equalities. Specifically, we apply formulations (13) and (17) to search for k-point unconstrained optimal designs in Ξ_k^N, the set of feasible designs on $\mathbf{X}$. By including additional constraints in problems (13) and (17), our method can also find a constrained optimal exact designs from the set $\Xi_k^N \cap \{(\boldsymbol{x}, \boldsymbol{n}) :\ \gamma(\boldsymbol{x}, \boldsymbol{n}) \leq \mathbf{0},\ \phi(\boldsymbol{x}, \boldsymbol{n}) = \mathbf{0}\}$. Here, γ and ϕ are user-selected differentiable functions in the inequality and equality constraints, respectively, on the design space $\mathbf{X}$ or on the replicates space Ω_k^N, or both.

3.3 Initialization and limitations

To reduce CPU time, we provide *consistent initial* guesses to the MINLP solver. This means that the initial solution $\xi^{(0)}$ has to be consistent and satisfy all constraints of the problem (Pantelides 1988). To construct $\xi^{(0)}$, we first choose a point centrally located in $\mathbf{X}$ and then select the other grid points using the relation $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \Delta\boldsymbol{x}$ where $\Delta\boldsymbol{x} = (\max \boldsymbol{x} - \min \boldsymbol{x})/(k-1)$ and k is the number of support points selected by the user. The replicates are then distributed so that the values of n_i are equal while ensuring that their sum is N. Next, we compute the elemental and the global FIMs for $\xi^{(0)}$, $\mathscr{U}^\intercal(\xi^{(0)}, \boldsymbol{p})$ and $\mathscr{M}^{-1}(\xi^{(0)}, \boldsymbol{p})$ and let the solver iterate until it converges to the optimum.

The formulations in Sect. 3 are coded in the GAMS environment (GAMS Development Corporation 2013a). GAMS is a general modeling system that supports mathematical programming applications in several areas. Upon execution, the code describing the mathematical program is automatically compiled, symbolically transcribed into a set of numerical structures, and all the information regarding the gradient and matrix Hessian is generated using the automatic differentiation tool and made available to the solver. We provide a sample of such a code in the Supplementary Material.

A drawback of the proposed formulations is their limited ability to find a global optimum in highly nonlinear problems, specifically. Typically, global optimizers can guarantee global optimality, but the computational burden required for finding optimal designs for models with several covariates can be massive or may not even converge within a realistic time frame. Using local optimizers lowers the computational effort but does not guarantee that the solution found is a global optimum. Therefore, the choice of using a global or a local optimizer involves a trade-off. One common strategy to further reduce the CPU time is to tighten the bounds for all the decision variables in the MINLP problem and provide an accurate initial solution. This is a particularly useful step when global optimizers are used.

Generally, our formulations (13) and (17) work well even when N or k are large unless the FIM becomes nearly singular and the model parameters have sensitivities [see Eq. (2)] of very different magnitude or the design space is multi-dimensional and the matrix inversion procedure becomes numerically unstable. However, even in these cases, we can scale properly the variables and equations to handle the problems. From our experiments in Sect. 4, the main limitations are related with the MINLP solvers currently available, and we were not able to identify a cutoff point, either on the size of the optimization problem or the FIM condition number, where the formulations fail. In this study, local solvers work surprisingly well to find a optimum for all problems addressed.

4 Numerical results

We now report D- and A-optimal exact designs for linear and nonlinear models found from our formulations. Some models were chosen for comparison with published results obtained using different numerical approaches. All computations were done using an Intel Core i7 machine running a 64 bits Windows 10 operating system with 2.80 GHz processor. In all problems, the relative and absolute tolerances used to solve the MINLP problems were set to 1×10^{-5} and the absolute tolerance is set equal to ϵ, the minimum value allowed for the diagonal entries in the FIM or its inverse so that they are positive definite matrices. The optima reported for each design in all the tables are for $\log[\det(\mathscr{U}(\xi, \boldsymbol{p}))]$ and $\text{tr}[\mathscr{M}^{-1}(\xi^{(0)}, \boldsymbol{p})]$ for D- and A-optimality criteria, respectively (note the first is a maximizer and the last a minimizer). The efficiency of D-optimal designs is determined from (7) using the relation $\log[\det(\mathscr{M}(\xi, \boldsymbol{p}))] = 2\ \log[\det(\mathscr{U}(\xi, \boldsymbol{p}))]$.

We employ the formulations in Sect. 3 to find optimal designs for the models in Table 1. Models 1–2 are linear, and the design space is $\mathbf{X} = [-1, 1]$ and $\mathbf{X} = [0.5, 2.5]$, respectively. Some of the Models 3–8 have the same design spaces as in the original problems discussed in the literature, and the set of parameters used for constructing local designs was also preserved whenever possible. Models 3–8 are nonlinear, and we report locally optimal designs for the set of parameters listed below each model. The mean responses in Models 6 and 7 are structurally the same, except for the nominal value of the parameter m. Models 9–14 have multiple regressors and so they allow us to test our algorithm for multi-factor experiments. Model 14 is a four-factor second-order response surface model with pairwise interactions. We note that Gotwalt et al. (2009) obtained Bayesian optimal designs for models subsumed in Model 10. Specifically, they addressed optimal exact design problems by first setting $n_i = 1,\ \forall i \in [\![N]\!]$, and allowing that, in result of the optimization, certain rows of the model matrix are identical. Consequently, here, the replication will become visible when multiple rows of the design are identical. Further, they set N but do not impose constraints on the number of distinct support points of the optimal design. Our framework is different as we set first set both N and the number of support points, k, and then optimize the design points and the replication, $n_i \in \Omega_k^N,\ \forall i \in [\![k]\!]$. Rasch et al. (1997) found replication-free D-optimal designs for models 3–4, and we compare some of their optimal exact designs with ours.

We applied our algorithm to find optimal exact designs for a battery of models with one or more factors. We select these models to test our algorithms either because they are widely used in practice or their optimal designs have been analytically or numerically determined and available for comparison. Tables 7 and 8 show that models with more factors (i.e., higher dimension) may not necessarily pose additional numerical difficulty; the crucial issue usually lies in the nonlinearity of the MINLP problem. Smaller dimensional models such as Models 3 and 4 below can be more challenging for our methods to solve than models with more factors involving, for example, linear polynomial terms. That, combined with the existence of reference optimal designs, is the rationale for using Models 3–4 for comparison in next sections.

We first discuss minimally supported optimal designs where $k = n_\theta$ without constraints. Minimally supported designs are desirable when taking observations at a new location is expensive. Designs that have equal masses n_θ/N at every design point are called *uniform designs* (Pukelsheim 1993, Chap. 4). They are popular because of their simplicity and when appropriate, we compare our optimal designs with uniform designs. In Sect. 4.1, we first compare the optimal designs obtained with our formulations for Models 3 and 4 and use different N values to analyze the effect of different sample sizes on the design. We compare optimal designs obtained from a locally optimal MINLP solver (`SBB`) with those generated from a global optimization solver (`BARON`). In addition, we compare designs obtained from our algorithm with analytical solutions in the literature when the latter are available.

Section 4.2 presents unconstrained optimal designs obtained for the models in Table 1. Here, `SBB` was used to solve the

Table 1 Battery of statistical models used for testing the MINLP formulations

Model	Regression function	Design space (**X**)
1	$\beta_0 + \beta_1\, x + \beta_2\, x^2 + \beta_3\, x^3 + \beta_4\, x^4$	$[-1, 1]$
2	$\beta_0 + \beta_1\, x + \beta_2\, x^{-1} + \beta_3\, \exp(-x)$	$[0.5, 2.5]$
3[a]	$\alpha + \beta\, \exp(\gamma x)$ $\alpha = 1, \beta = -1.4, \gamma = -0.2$	$[0, 25]$
4[b]	$\alpha\, \exp[\beta\, \exp(\gamma x)]$ $\alpha = 1, \beta = -1.4, \gamma = -0.2$	$[0, 150]$
5[c]	$\frac{V\, x}{k+x} + F\, x$ $V = 2.0, k = 0.5, F = 10$	$[0.001, 2]$
6[d]	$E_0 + \frac{(E_\infty - E_0)\, x^m}{k_d^m + x^m}$ $E_\infty = 1.70, E_0 = 0.137, k_d^m = 1, m = 1.5$	$[1 \times 10^{-5}, 10]$
7[d]	$E_0 + \frac{(E_\infty - E_0)\, x^m}{k_d^m + x^m}$ $E_\infty = 1.70, E_0 = 0.137, k_d^m = 1, m = -1.5$	$[1 \times 10^{-5}, 10]$
8[e]	$\gamma\, \exp[-\alpha\, (x - \theta) - \exp(-\lambda\, (x - \theta))]$ $\gamma = 1.946, \theta = 6.06, \alpha = 0.174, \lambda = 0.288$	$[0, 30]$
9[f]	$\beta_0 + \beta_1\, x_1 + \beta_1\, x_2 + \beta_3\, x_1^2 + \beta_4\, x_2^2 + \beta_5\, x_1\, x_2$	$[-1, 1]^2$
10[g]	$\frac{1}{1+\exp(\beta_0+\beta_1\, x_1+\beta_2\, x_2+\beta_3\, x_1\, x_2)}$ $\beta_0 = -2, \beta_1 = 0.5, \beta_2 = 0.5, \beta_3 = 0.1$	$[0, 5] \times [0, 1]$
11[h]	$\frac{\theta_1\, \theta_3\, x_1}{1+\theta_1\, x_1+\theta_2\, x_2}$ $\theta_1 = 2.9, \theta_2 = 12.2, \theta_3 = 0.69$	$[0, 3]^2$
12[i]	$\beta_0 + \beta_1\, x_1 + \beta_1\, x_2 + \beta_3\, x_3 + \beta_4\, x_1^2 + \beta_5\, x_2^2 + \beta_6\, x_3^2 +$ $\beta_7\, x_1\, x_2 + \beta_8\, x_1\, x_3 + \beta_9\, x_2\, x_3$	$[-1, 1]^3$
13[j]	$\beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_3 + \beta_4\, x_1\, x_2 + \beta_5\, x_1\, x_3 + \beta_6\, x_2\, x_3 +$ $\beta_7/x_1 + \beta_8/x_2 + \beta_9/x_3$	$[0.5, 2.0]^3$
14[k]	$\beta_0+\beta_1\, x_1+\beta_2\, x_2+\beta_3\, x_3+\beta_4\, x_4\, x_2+\beta_5\, x_1\, x_2+\beta_6\, x_1\, x_3+\beta_7\, x_1\, x_4+$ $\beta_8\, x_2\, x_3 + \beta_9\, x_2\, x_4 + \beta_{10}\, x_3\, x_4 + \beta_{11}\, x_1^2 + \beta_{12}\, x_2^2 + \beta_{13}\, x_3^2 + \beta_{14}\, x_4^2$	$[-1, 1]^4$

[a] Three parameter exponential function.
[b] Gompertz model (Laird 1964)
[c] Modified Michaelis–Menten model (López-Fidalgo et al. 2008)
[d] Four-parameter Hill model (Hill 1910)
[e] Coale–McNeil model (Coale and McNeil 1972)
[f] Second-order response surface model with two factors (Atkinson et al. 2007, p. 170)
[g] Logistic model with two factors and all pairwise interactions
[h] Kinetics of the catalytic dehydrogenation of n-hexyl alcohol model (Box and Hunter 1965)
[i] Three-factor quadratic polynomial mixture model
[j] Johnson and Nachtsheim (1983) model including nonlinear terms (on covariates)
[k] Four-factor surface response model

MINLP optimization problems, where $N = 3 \times n_\theta$ for all cases. In Sect. 4.3, we find constrained optimal designs where the design space and/or the number of replicates are subject to restrictions. These constraints can be linear or nonlinear, and k is at least as large as n_θ. For designs with $k > n_\theta$, we also set the value of k as in the minimally supported designs and then include an additional constraint in problems (13) and (17) imposing that all support points have at least one observation (i.e., $n_i \geq 1$). Practically, in our formulation these are constrained designs as they include additional constraints and will be addressed in Sect. 4.3. The D-optimal exact designs for most of the cases in Table 1 are new and so are all the A-optimal exact designs.

4.1 Comparison of results

In this subsection, we compare optimal exact designs found from our algorithm in a variety of ways: in Sect. 4.1.1, we study them using different sample sizes; in Sect. 4.1.2, we compare them with those found using a global optimal MINLP solver, and in Sect. 4.1.3, we compare them with analytical results when they are available.

4.1.1 Impact of the sample size

Table 2 presents optimal exact designs of different sizes for Models 3 and 4 found by BARON. For Model 3, $N = 3 \times$

Table 2 D- and A-optimal exact designs for Models 3 and 4 in Table 1 obtained using a global MINLP solver

Model	Criterion	N	Design	Optimum value	CPU (s)
3	D	9	$\begin{pmatrix}0.0000 & 4.8304 & 25.0000\\ 3 & 3 & 3\end{pmatrix}$	– 0.7682	2.9
		10	$\begin{pmatrix}0.0000 & 4.8396 & 25.0000\\ 4 & 3 & 3\end{pmatrix}$	– 0.7824	2.8
		11	$\begin{pmatrix}0.0000 & 4.8364 & 25.0000\\ 4 & 3 & 4\end{pmatrix}$	– 0.7815	2.4
	A	9	$\begin{pmatrix}0.0000 & 4.3005 & 25.0000\\ 3 & 1 & 5\end{pmatrix}$	8.7943	3.5
		10	$\begin{pmatrix}0.0000 & 3.9398 & 25.0000\\ 3 & 2 & 5\end{pmatrix}$	8.8053	4.7
		11	$\begin{pmatrix}0.0000 & 3.9756 & 25.0000\\ 4 & 2 & 5\end{pmatrix}$	8.8035	5.2
4	D	15	$\begin{pmatrix}0.0000 & 7.3638 & 150.0000\\ 5 & 5 & 5\end{pmatrix}$	– 2.5086	26.5
		16	$\begin{pmatrix}0.0000 & 7.2868 & 84.0576\\ 5 & 5 & 6\end{pmatrix}$	– 2.5143	21.0
		17	$\begin{pmatrix}0.0000 & 7.3418 & 86.7134\\ 5 & 6 & 6\end{pmatrix}$	– 2.5140	21.4
	A	15	$\begin{pmatrix}0.0000 & 7.2623 & 150.0000\\ 10 & 1 & 4\end{pmatrix}$	37.8972	103.1
		16	$\begin{pmatrix}0.0000 & 7.2028 & 117.9004\\ 10 & 2 & 4\end{pmatrix}$	37.7017	105.8
		17	$\begin{pmatrix}0.0000 & 7.1710 & 85.2452\\ 11 & 2 & 4\end{pmatrix}$	37.5276	160.2

$n_\theta + q,\ q \in \{0, 1, 2\}$, and for Model 4, $N = 4 \times n_\theta + q,\ q \in \{0, 1, 2\}$. The generated optimal exact designs are presented as $2 \times k$ matrices where the first line shows the design points and the second line shows their replicates.

The D-optimal exact designs found for $N = 9$ and $N = 15$ for Model 3 and 4 in Table 2, respectively, are similar to those of Rasch et al. (1997) found from an exchange algorithm. The exact D-optimal designs are uniform when N is a multiple of n_θ, which is a general observation already made by others (Yang et al. 2012, 2016). When N is a multiple of n_θ, the values of the optimum suggest that D-optimal designs are slightly more efficient than those found for other sample sizes. Both examples in Table 2 show that the optimum has slightly larger values of N that lead to exact designs equal to those obtained by rounding optimal approximate designs and the weights are equal for all support points. Practically, exact k-point optimal designs with unequal ratios n_i/N can be slightly poorer than those where n_i/N is equal $\forall i \in [\![n_\theta]\!]$. This finding is consistent with earlier observations for N-point D-optimal designs (Pukelsheim 1993, Chap. 4). Table 2 shows that changes in the support points of the D-optimal designs are marginal for small changes in the value of N.

The A-optimal exact designs are not uniform even when N is multiple of n_θ and so finding an optimal approximate design first and rounding it to obtain an exact design may result in a non-optimal exact design.

Figure 1 shows the plots of the dispersion functions of the D-optimal exact designs for Model 3 when $N = 9, 10, 11$. It shows that the D-optimal exact design ξ for $N = 9$ has 3 support points and the function $\Psi(\boldsymbol{x}|\xi_D, \boldsymbol{p})$ is bounded from above by n_θ with equality at the support points. The maxima of the dispersion functions for $N = 10$ and $N = 11$ are not n_θ implying that the D-optimal exact designs do not match the D-optimal approximate designs.

We report optimal design points to four decimal places for accuracy. In scenarios where the design variables cannot be controlled with this level of certainty, the optimal design will serve as a reference design for the implemented design, which will have a slightly lower efficiency.

4.1.2 Impact of the MINLP solver

In this section, we determine optimal designs for Models 3–4 using formulations in Sect. 3 with a local optimization MINLP solver (`SBB`). Table 3 displays the D- and

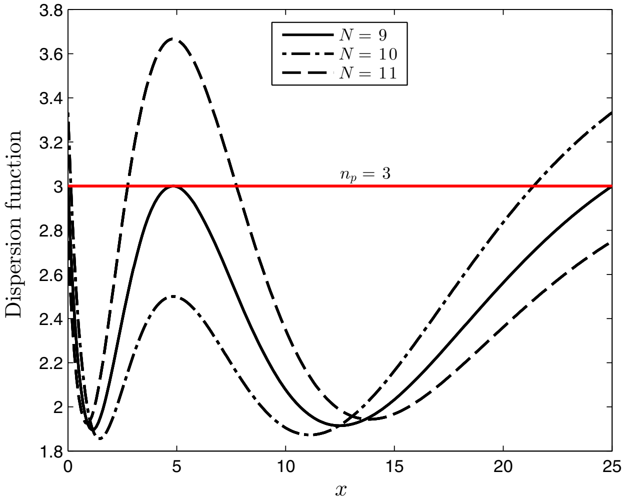

Fig. 1 Dispersion functions of the D-optimal designs for Model 3 in Table 2 when $k = n_\theta = 3$ for different values of N

A-optimal exact designs. When we compare them with the optimal exact designs in Table 2 found with a global optimization solver, we observe that (i) the CPU time required by the local optimization MINLP solver is on average 382 times shorter than that required by the global optimization MINLP solver; (ii) both solvers produce very similar A- and D-optimal exact designs with almost identical optimality values; (iii) both solvers produce D-optimal exact designs with replicates as equally distributed as possible and A-optimal exact designs with very different numbers of replicates at the design points; (iv) D- and A-optimal exact designs are supported at the extreme ends of the design space when the design space is small (e.g., Model 3), but this is no longer the case when the design space is large (Model 4); and (v) the largest support points of the A-optimal exact designs found by local and global optimization solvers are noticeably different.

We calculated the D- and A-efficiencies of the optimal exact designs obtained with the locally optimal MINLP solver relative to those obtained with the global optimization solver using Eqs. (7) and (8), and they are all close to 1.0000. That implies that for this set of problems, the local optimization solver is able to find designs very close to those obtained by a global optimization solver in a small fraction of time. We therefore use the local optimization MINLP solver `SBB` to generate all the optimal exact designs in the rest of our work. Typically, the optimal designs obtained with global optimizers are equally or more efficient than those obtained with local optimizers, an exception is the A-optimal design for Model 4 and $N = 17$ in Table 3 where the design found by the global solver has an efficiency slightly below 100 %.

4.1.3 Models with analytically derived optimal designs

This section compares D-optimal designs obtained from our formulations with known D-optimal designs for some models, which are presented in Table 4. The first selected model with known D-optimal exact designs is the one with mean function equal to the first-order trigonometric regression model on a partial circle of different size and different values of N (Models 15–19 in Table 4). The D-optimal exact designs were reported in Chang et al. (2013, Theorem 4.1). Table 5 shows D-optimal exact designs found for Models 15–19 with formulation (13). Our results agree with the analytical D-optimal designs for the first-order trigonometric regression model. There are two optimal designs for Model 17, and our algorithm is able to find one of them. Further numerical experiences revealed that the other optimum can be obtained with a different starting point.

The second selected model to compare optimal designs obtained from our algorithm and the theoretical designs is the one with a quadratic mean response function on $[-1, 1]$ for $N = 11,\ 12$ and $N = 13$ (Models 20–22). The theoretical D-optimal exact designs were reported in Gaffke (1987). In particular, they showed that the D-optimal designs are all minimally supported with $k = 3$. The D-optimal designs obtained by our algorithm for Models 20–22 in Table 5 agree with their theoretical results. Chang and Yeh (1998) and Imhof (1998) provided corresponding results for A-optimal exact designs, and we used them to verify the accuracy of the results from our formulations. Table 6 displays the A-optimal designs obtained from (17) for Models 20–22. Our A-optimal exact designs also agree with those obtained analytically by Chang and Yeh (1998, Theorem 2.1).

4.2 Unconstrained optimal exact designs

In this section, we determine minimally supported optimal designs for all the models in Table 1 assuming that $N = 3 \times n_\theta$. Other design setups are considered in Sect. 4.3.

Tables 7 and 8 list D- and A-optimal exact designs obtained from our algorithm, respectively. The D-optimal designs are uniform in all cases, and the CPU time required to solve the MINLP problems is generally short in part due to our initialization strategy, see Sect. 3.2, and because of the locally optimal MINLP solver. This shows that our formulation herein is effective for the problems we have tackled. We recall that Models 6 and 7 are structurally the same, except for the parameter value in the power exponent m, and the optimal designs obtained for both are equal. Our optimal designs for Model 9 are similar to those obtained using an exchange algorithm in Atkinson et al. (2007, Chap. 12). Unlike the D-optimal exact designs, our A-optimal designs are not uniform.

Table 3 *D*- and *A*-optimal exact designs for Models 3–4 in Table 1 obtained using a locally optimal MINLP solver

Model	Criterion	N	Design	Optimum value	CPU (s)
3	*D*	9	$\begin{pmatrix} 0.0000 & 4.8304 & 25.0000 \\ 3 & 3 & 3 \end{pmatrix}$	– 0.7682	0.03
		10	$\begin{pmatrix} 0.0000 & 4.8304 & 25.0000 \\ 4 & 3 & 3 \end{pmatrix}$	– 0.7824	0.03
		11	$\begin{pmatrix} 0.0000 & 4.8304 & 25.0000 \\ 4 & 4 & 3 \end{pmatrix}$	– 0.7815	0.14
	A	9	$\begin{pmatrix} 0.0000 & 4.3024 & 25.0000 \\ 3 & 1 & 5 \end{pmatrix}$	8.7943	0.14
		10	$\begin{pmatrix} 0.0000 & 3.9613 & 25.0000 \\ 3 & 2 & 5 \end{pmatrix}$	8.8053	0.14
		11	$\begin{pmatrix} 0.0000 & 3.9765 & 25.0000 \\ 4 & 2 & 5 \end{pmatrix}$	8.8035	0.14
4	*D*	15	$\begin{pmatrix} 0.0000 & 7.3638 & 118.578 \\ 5 & 5 & 5 \end{pmatrix}$	– 2.5086	0.03
		16	$\begin{pmatrix} 0.0000 & 7.3638 & 150.0000 \\ 6 & 5 & 5 \end{pmatrix}$	– 2.5143	0.05
		17	$\begin{pmatrix} 0.0000 & 7.3638 & 150.0000 \\ 6 & 6 & 5 \end{pmatrix}$	– 2.5141	0.05
	A	15	$\begin{pmatrix} 0.0000 & 7.1472 & 150.000 \\ 10 & 1 & 4 \end{pmatrix}$	37.8972	0.14
		16	$\begin{pmatrix} 0.0000 & 7.1991 & 150.0000 \\ 10 & 2 & 4 \end{pmatrix}$	37.7017	0.13
		17	$\begin{pmatrix} 0.0000 & 7.1710 & 150.0000 \\ 11 & 2 & 4 \end{pmatrix}$	37.4915	0.16

Table 4 Battery of statistical models with known *D*-optimal exact designs used for testing the MINLP formulations

Model	Regression function	Design space (**X**)	N
15[j]	$\beta_1 \cos(x) + \beta_2 \sin(x)$	$[-\pi/4, \pi/4]$	4
16[j]	$\beta_1 \cos(x) + \beta_2 \sin(x)$	$[-2\pi/3, 2\pi/3]$	4
17[j]	$\beta_1 \cos(x) + \beta_2 \sin(x)$	$[-\pi/4, \pi/4]$	5
18[j]	$\beta_1 \cos(x) + \beta_2 \sin(x)$	$[-1/2 \ \arccos(-1/10), 1/2 \ \arccos(-1/10)]$	5
19[j]	$\beta_1 \cos(x) + \beta_2 \sin(x)$	$[-2\pi/3, 2\pi/3]$	5
20[k]	$\beta_0 + \beta_1 x + \beta_2 x^2$	$[-1, 1]$	11
21[k]	$\beta_0 + \beta_1 x + \beta_2 x^2$	$[-1, 1]$	12
22[k]	$\beta_0 + \beta_1 x + \beta_2 x^2$	$[-1, 1]$	13

[j] First-order trigonometric regression model on a partial circle.
[k] Quadratic regression model

The designs obtained are in good agreement with the optimal approximate designs found in Duarte et al. (2018) using an adaptive semidefinite programming (ASP)-based algorithm. We observe that the exact designs are equally or slightly more efficient than those in Duarte et al. (2018) when $N = 3 \times n_\theta$ partly because they first discretized the design space and the optimum is found from this finite set of candidate points; in contrast, we work with continuous design spaces. In practice, the trend observed in Sect. 4.1.1 extends to models in Table 1 when different values of N are tested. The exact optimal designs for Model 14 were also determined by the global solver BARON, and the results obtained are similar to those in Table 7 and 8 (last line) except that the CPU time has increased to 3.12 s for *D*-optimal designs and to 138.35 s for *A*-optimal designs.

To further validate the formulations in Sect. 3, we compared the optimal designs found for surface response models, such as, Models 1, 9, 12 and 14 in Table 1 with those obtained from the commercial software JMP® 14.0.0 (64 bit) (SAS Institute Inc. 2017). This software implements

Table 5 D-optimal exact designs for Models 15–22 in Table 4 obtained using a locally optimal MINLP solver

Model	N	Design	Optimum value	CPU (s)
15	4	$\begin{pmatrix} -0.7854 & 0.7854 \\ 2 & 2 \end{pmatrix}$	− 0.6931	0.09
16	4	$\begin{pmatrix} -0.8365 & 0.7343 \\ 2 & 2 \end{pmatrix}$	− 0.6931	0.02
17	5	$\begin{pmatrix} -0.7854 & 0.7854 \\ 2 & 3 \end{pmatrix}$	− 0.7136	0.05
18	5	$\begin{pmatrix} -0.8355 & 0.0000 & 0.8355 \\ 2 & 1 & 2 \end{pmatrix}$	− 0.7004	0.04
19	5	$\begin{pmatrix} -1.1780 & 0.1401 & 1.0518 \\ 2 & 2 & 1 \end{pmatrix}$	− 0.6931	0.03
20	11	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 3 & 4 & 4 \end{pmatrix}$	− 0.9681	0.05
21	12	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 4 & 4 & 4 \end{pmatrix}$	− 0.9548	0.03
22	13	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 4 & 5 & 4 \end{pmatrix}$	− 0.9633	0.03

Table 6 A-optimal exact designs for Models 19–21 in Table 4 obtained using a locally optimal MINLP solver

Model	N	Design	Optimum value	CPU (s)
20	11	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 3 & 5 & 3 \end{pmatrix}$	8.0667	0.03
21	12	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 3 & 6 & 3 \end{pmatrix}$	8.0000	0.02
22	13	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 3 & 7 & 3 \end{pmatrix}$	8.0476	0.13

the coordinate-exchange algorithm (CEA) using an initial design to iterate from multiple initial points and keeping the best solution found. Tables S1 and S2 in the Supplementary Material report the D- and A-optimal designs obtained from CEA, respectively. They were obtained using 1×10^6 re-initializations of the algorithm, and we limit the analysis to linear (on covariates and parameters) surface response models and minimally supported designs so that they are equivalent to those found with MINLP formulations and can be compared. We note that CPU time required by JMP® 14.0.0 (64 bit) for the examples addressed was of 8 min to 15 min and our formulations with a locally optimal solver require a few seconds. If a global optimizer were used with our formulations, we expect the CPU time would have increased 2 to 3 orders of magnitude even though in some cases the CPU times are still competitive.

Table 9 lists the D-optimal values found from the MINLP formulation and those found from the CEA along with the efficiencies of the latter relative to the former. We note that the optimal designs obtained with the formulations proposed herein have about the same efficiencies as those from the CEA. For a fair comparison between our algorithm and CEA, we set the number of re-initializations of the latest to 1×10^3 which allows reducing the CPU time needed to about 1 s for all the examples addressed. We note that the optimal designs obtained with CEA for this setup are close to the optimum having efficiencies higher than 0.94 (relative to optimal designs obtained with MINLP formulation).

4.3 Constrained optimal exact designs

Here, we test the algorithm to find D- and A-optimal designs when there are linear or nonlinear constraints compactly represented by inequalities $\boldsymbol{\gamma}(\boldsymbol{x}, \boldsymbol{n}) \leq \mathbf{0}$ and equalities $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{n}) = \mathbf{0}$ where $\boldsymbol{\gamma} \in \mathbb{R}^{n_i}$ and $\boldsymbol{\phi} \in \mathbb{R}^{n_e}$, n_i standing for the number inequality constraints and n_e for the number of equality constraints. To fix ideas, in Sect. 4.3.1 we impose constraints only on the support points and search for optimal designs in the constrained space $\Xi_k^{N,x} \equiv \Xi_k^N \cap \{\boldsymbol{x} : \boldsymbol{\gamma}(\boldsymbol{x}) \leq \mathbf{0},\ \boldsymbol{\phi}(\boldsymbol{x}) = \mathbf{0}\}$. In Sect. 4.3.2, we impose constraints only on the number of replicates and search for k-point optimal designs in the space $\Xi_k^{N,n} \equiv \Xi_k^N \cap \{\boldsymbol{n} : \boldsymbol{\gamma}(\boldsymbol{n}) \leq \mathbf{0},\ \boldsymbol{\phi}(\boldsymbol{n}) = \mathbf{0}\}$. Finally, Sect. 4.3.3 allows constraints on both the support points and the number of replicates and

Table 7 *D*-optimal exact designs for Models 1–14 in Table 1 with $N = 3 \times n_\theta$

Model	N	Design	Optimum value	CPU (s)
1	15	$\begin{pmatrix} -1.0000 & -0.6547 & 0.0000 & 0.6547 & 1.0000 \\ 3 & 3 & 3 & 3 & 3 \end{pmatrix}$	− 5.0275	0.06
2	12	$\begin{pmatrix} 0.5000 & 0.7852 & 1.6148 & 2.5000 \\ 3 & 3 & 3 & 3 \end{pmatrix}$	− 5.9706	0.02
3	9	$\begin{pmatrix} 0.0000 & 4.8304 & 25.0000 \\ 3 & 3 & 3 \end{pmatrix}$	− 0.7682	2.1
4	9	$\begin{pmatrix} 0.0000 & 7.3638 & 118.578 \\ 3 & 3 & 3 \end{pmatrix}$	− 2.5086	0.11
5	9	$\begin{pmatrix} 0.2142 & 0.9780 & 2.0000 \\ 3 & 3 & 3 \end{pmatrix}$	− 3.4333	0.03
6	12	$\begin{pmatrix} 1.0 \times 10^{-5} & 0.4535 & 1.7253 & 10.0000 \\ 3 & 3 & 3 & 3 \end{pmatrix}$	− 5.0864	0.45
7	12	$\begin{pmatrix} 1.0 \times 10^{-5} & 0.4535 & 1.7253 & 10.0000 \\ 3 & 3 & 3 & 3 \end{pmatrix}$	− 6.08644	0.05
8	12	$\begin{pmatrix} 2.3129 & 5.5988 & 10.1971 & 18.3971 \\ 3 & 3 & 3 & 3 \end{pmatrix}$	− 5.1976	0.11
9	18	$\begin{pmatrix} -1.0000 & -1.0000 & -0.1315 & 0.3945 & 1.0000 & 1.0000 \\ 1.0000 & -1.0000 & -0.1315 & 1.0000 & 0.3945 & -1.0000 \\ 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix}$	− 2.5803	0.13
10	12	$\begin{pmatrix} 0.1412 & 2.8208 & 2.8208 & 5.0000 \\ 0.0000 & 0.0000 & 1.0000 & 1.0000 \\ 3 & 3 & 3 & 3 \end{pmatrix}$	− 7.9789	0.09
11	9	$\begin{pmatrix} 0.2804 & 3.0000 & 3.0000 \\ 0.0000 & 0.0000 & 0.7951 \\ 3 & 3 & 3 \end{pmatrix}$	− 9.1640	0.03
12	30	$\left(\begin{array}{ccccc} -1.0000 & -1.0000 & -1.0000 & -1.0000 & 0.0387 \\ -1.0000 & -1.0000 & 0.2878 & 1.0000 & 0.1262 \\ 1.0000 & -1.0000 & 0.2711 & -1.0000 & -1.0000 \\ 3 & 3 & 3 & 3 & 3 \\ \hline 0.0387 & 0.1965 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & -1.0000 & -0.3415 & 1.0000 & -1.0000 \\ 1.0000 & 0.1658 & 1.0000 & -0.4429 & -1.0000 \\ 3 & 3 & 3 & 3 & 3 \end{array}\right)$	− 4.3187	0.19
13	27	$\left(\begin{array}{ccccc} 0.0500 & 0.0500 & 0.0500 & 0.2067 & 2.0000 \\ 2.0000 & 0.0500 & 0.2394 & 2.0000 & 0.1955 \\ 0.0500 & 0.0500 & 2.0000 & 2.0000 & 0.0500 \\ 3 & 3 & 3 & 3 & 3 \\ \hline 2.0000 & 2.0000 & 2.0000 & 2.0000 & \\ 2.0000 & 2.0000 & 0.0500 & 2.0000 & \\ 0.3162 & 2.0000 & 2.0000 & 0.0500 & \\ 3 & 3 & 3 & 3 & \end{array}\right)$	− 5.3790	0.23
14	45	$\left(\begin{array}{ccccc} -1.0000 & -1.0000 & -1.0000 & -1.0000 & -1.0000 \\ 1.0000 & 1.0000 & -1.0000 & 1.0000 & -1.0000 \\ -1.0000 & 1.0000 & 1.0000 & 0.6417 & -1.0000 \\ -0.5926 & -1.0000 & 0.0308 & 1.0000 & 1.0000 \\ 3 & 3 & 3 & 3 & 3 \\ \hline -1.0000 & -0.0198 & -0.0198 & 0.1888 & 1.0000 \\ -0.2514 & -1.0000 & -0.2389 & 1.0000 & 1.0000 \\ -0.0644 & -1.0000 & 1.0000 & 0.1379 & -1.0000 \\ -1.0000 & -1.0000 & 1.0000 & -0.1470 & -1.0000 \\ 3 & 3 & 3 & 3 & 3 \\ \hline 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ -1.0000 & 1.0000 & -0.3446 & -1.0000 & 1.0000 \\ 0.1008 & -1.0000 & -1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & -0.0357 & -1.0000 & 1.0000 \\ 3 & 3 & 3 & 3 & 3 \end{array}\right)$	− 6.3017	0.52

Designs obtained using a locally optimal MINLP solver

Table 8 A-optimal exact designs for Models 1–14 in Table 1 with $N = 3 \times n_\theta$

Model	N	Design	Optimum value	CPU (s)
1	15	$\begin{pmatrix} -1.0000 & -0.6983 & -0.0392 & 0.6540 & 1.0000 \\ 2 & 3 & 4 & 4 & 2 \end{pmatrix}$	1.976×10^2	0.50
2	12	$\begin{pmatrix} 0.5000 & 0.7511 & 1.6680 & 2.5000 \\ 2 & 4 & 4 & 2 \end{pmatrix}$	5.300×10^3	0.16
3	9	$\begin{pmatrix} 0.0000 & 4.3024 & 25.0000 \\ 3 & 1 & 5 \end{pmatrix}$	8.7943	0.05
4	9	$\begin{pmatrix} 0.0000 & 7.1472 & 150.0000 \\ 6 & 1 & 2 \end{pmatrix}$	37.4322	0.05
5	9	$\begin{pmatrix} 0.1505 & 1.0274 & 2.0000 \\ 3 & 4 & 2 \end{pmatrix}$	4.543×10^2	0.16
6	12	$\begin{pmatrix} 1.0 \times 10^{-5} & 0.4693 & 1.7391 & 10.0000 \\ 2 & 3 & 4 & 3 \end{pmatrix}$	1.373×10^2	0.14
7	12	$\begin{pmatrix} 1.0 \times 10^{-5} & 0.4693 & 1.7301 & 10.0000 \\ 2 & 3 & 4 & 3 \end{pmatrix}$	1.373×10^2	0.09
8	12	$\begin{pmatrix} 1.3817 & 5.3579 & 10.7974 & 22.1334 \\ 2 & 2 & 3 & 5 \end{pmatrix}$	4.651×10^3	0.08
9	18	$\begin{pmatrix} -1.0000 & -1.0000 & 0.0000 & 0.0000 & 1.0000 & 1.0000 \\ 1.0000 & -0.5016 & 0.1879 & -1.0000 & -0.5016 & 1.0000 \\ 2 & 3 & 5 & 3 & 3 & 2 \end{pmatrix}$	21.163	0.23
10	12	$\begin{pmatrix} 0.0000 & 2.0033 & 2.0033 & 5.0000 \\ 0.0000 & 0.0000 & 1.0000 & 1.0000 \\ 6 & 1 & 3 & 2 \end{pmatrix}$	9.520×10^2	0.11
11	9	$\begin{pmatrix} 0.2685 & 3.0000 & 3.0000 \\ 0.0000 & 0.0000 & 0.8074 \\ 4 & 1 & 4 \end{pmatrix}$	2.994×10^4	0.05
12	30	$\left(\begin{array}{ccccc} -1.0000 & -1.0000 & -1.0000 & -1.0000 & 0.0585 \\ 0.0577 & -1.0000 & 0.1457 & 1.0000 & -1.0000 \\ 1.0000 & -0.0284 & -1.0000 & -0.0256 & 1.0000 \\ 3 & 3 & 3 & 3 & 3 \\ \hline 0.0585 & 0.4940 & 1.0000 & 1.0000 & 1.0000 \\ 0.2141 & -1.0000 & 1.0000 & -0.5446 & 1.0000 \\ -0.0283 & -1.0000 & 1.0000 & 0.2422 & -1.0000 \\ 5 & 2 & 2 & 4 & 2 \end{array}\right)$	36.565	3.85
13	27	$\left(\begin{array}{ccccc} 0.0500 & 0.2513 & 0.2513 & 0.3604 & 2.0000 \\ 2.0000 & 0.2513 & 2.0000 & 2.0000 & 0.2513 \\ 2.0000 & 2.0000 & 0.2513 & 2.0000 & 0.2513 \\ 2 & 5 & 5 & 2 & 5 \\ \hline 2.0000 & 2.0000 & 2.0000 & 2.0000 & \\ 2.0000 & 0.3604 & 0.0500 & 2.0000 & \\ 0.3604 & 2.0000 & 2.0000 & 0.0500 & \\ 2 & 2 & 2 & 2 & \end{array}\right)$	16.468	0.81
14	45	$\left(\begin{array}{ccccc} - - 1.0000 & -1.0000 & -1.0000 & -1.0000 & -1.0000 \\ 1.0000 & 0.5863 & -1.0000 & 1.0000 & -0.3197 \\ -1.0000 & 1.0000 & 1.0000 & 0.8114 & -0.7094 \\ -0.1290 & -1.0000 & 0.3348 & 1.0000 & 1.0000 \\ 3 & 2 & 2 & 2 & 3 \\ \hline -1.0000 & -0.0636 & -0.0269 & 0.1687 & 1.0000 \\ -1.0000 & -1.0000 & 0.0464 & 1.0000 & 1.0000 \\ -0.3694 & -1.0000 & 0.0545 & -1.0000 & 0.4075 \\ -1.0000 & 0.3826 & 0.0905 & -1.0000 & -0.3641 \\ 2 & 4 & 8 & 3 & 3 \\ \hline 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ -1.0000 & 1.0000 & -0.1955 & -1.0000 & 0.1943 \\ -0.0435 & -1.0000 & -1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.0000 & -0.4484 & -1.0000 & 1.0000 \\ 3 & 2 & 3 & 2 & 3 \end{array}\right)$	55.496	65.87

Designs obtained using a locally optimal MINLP solver

Table 9 D-optimality values of the designs found with MINLP formulations and CEA for selected models

Model	Optimum[a]	Optimum[b]	Eff$_D$
1	− 5.0275	− 5.0275	1.0000
9	− 2.5803	− 2.5818	0.9995
12	− 4.3187	− 4.4084	0.9822
14	− 6.3017	− 6.3267	0.9967

[a] Optimal design obtained from the MINLP formulations using a local optimizer.
[b] Optimal design obtained from the CEA using JMP® 14.0.0 (64 bit)

we search for a k-point optimal design among the designs in space $\Xi_k^{N,n,x} \equiv \Xi_k^N \cap \{(\boldsymbol{x}, \boldsymbol{n}) : \boldsymbol{\gamma}(\boldsymbol{x}, \boldsymbol{n}) \leq \boldsymbol{0},\ \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{n}) = \boldsymbol{0}\}$.

In this section, we apply the algorithm to find minimally supported A- and D-optimal exact designs for the Model 13 in Table 1 defined on the design space $\mathbf{X} = [0.5, 2.0] \times [0.5, 2.0] \times [0.5, 2.0]$. We present results only for this model when $N = 3 \times n_\theta = 27$ for space consideration and omit results for other models in Table 1. This model has 9 parameters and so $k = n_\theta = 9$. For values of N not a multiple of n_θ, we obtained optimal exact designs that exhibit a trend similar to that observed for Models 3–4 in Sect. 4.1.1.

4.3.1 Constraints on the design space

We now test our algorithm when the design problem has equality or inequality constraints on the design space and the constraints are of linear or nonlinear kind.

Linear constraints arise naturally in some experimental design problems. For example, in mixture experiments, a support point represents the proportions from the various components in the study. Naturally, each support point has nonnegative components that add to unity, see for example, Atkinson et al. (2007, Chap.16). At other times, there are physical and linear constraints on the support points. As an illustration, suppose that we require that the sum of the components in each support point $\boldsymbol{x}$ in the design is 1.0, i.e.,

$$\phi_1(\boldsymbol{x}) := \sum_{j=1}^{3} x_{j,i} = 1.0, \quad \forall i \in [\![k]\!]. \tag{18}$$

The set of candidate designs are found from the given design space $\mathbf{X} = [0.05, 1.0] \times [0.05, 1.0] \times [0.0, 1.0]$ that incorporates the additional constraints.

Nonlinear constraints on the design problem vary and occur frequently, depending on the problem at hand. For example, Berman's model (Berman 1962) is typically defined on an arc in brain mapping studies; consequently, it is likely more appropriate that the design space is spherical in which case the support points are contained inside a sphere of radius r centered at $\boldsymbol{x} = [c_1, c_2, c_3]^\intercal \in \mathbf{X}$, i.e.,

$$\gamma_1(\boldsymbol{x}) := \sum_{j=1}^{3} (x_{j,i} - c_j)^2 \leq r^2, \quad \forall i \in [\![k]\!]. \tag{19}$$

Another example is requiring a minimum distance between the support points because it may not be practical or meaningful to take observations from locations too close together. Such a constraint may take the form

$$\gamma_1(\boldsymbol{x}) := \delta^2 \leq \sum_{j=1}^{3} (x_{j,i} - x_{j,l})^2, \quad \forall i,\ l \in [\![k]\!],\ l \geq i + 1 \tag{20}$$

where δ is the minimum distance allowed between any two support points. In our algorithm, we set $r = 0.75$, $c_1 = c_2 = c_3 = 1.25$ in (19) and $\delta = 0.2$ in (20).

Our last example has a discrete design space with three discrete factors x_1, x_2 and x_3, each with three levels and coded as 1, 2, and 3. Let $\boldsymbol{y}$ be an integer variable containing the integer codes for $\boldsymbol{x}$, and we want to find D- and A-optimal exact designs subject to the following constraints in problems (13) and (17), respectively.

$$\gamma_1(\boldsymbol{x}) := x_{j,i} = y_{j,i} + 1, \quad \forall j \in [\![n_x]\!],\ i \in \{1, 2, 3\} \tag{21a}$$

$$\phi_1(\boldsymbol{y}) := y_{j,i} \leq 2, \quad \forall j \in [\![n_x]\!],\ i \in \{1, 2, 3\}, \quad y_{j,i} \in \mathbb{N}_0. \tag{21b}$$

Each of the optimal designs is then obtained by finding the k most informative points from the 27 candidate design points. Tables 10 and 11 report the D- and A-optimal designs, respectively, obtained under constraints (18)–(21). The D-optimal designs are still uniform, and the CPU times required to solve the optimization problems are short for the various constraints. Table 11 shows that A-optimal designs require noticeably longer time to compute than D-optimal designs.

4.3.2 Constraints on the replicates

This section considers a design problem where the number of support points in the optimal design sought exceeds the number parameters in the model. As an example, suppose that we require 10 support points ($k = 10$) and the optimal design must have at least two observations at each support point:

$$\gamma_1(\boldsymbol{n}) := n_i \geq 2, \quad \forall i \in [\![k]\!]. \tag{22}$$

Table 10 Constrained D-optimal exact designs for Model 13 in Table 1 with $N = 27$ and $k = 9$

Constraints	Design	Optimum value	CPU (s)
(18)	$\left(\begin{array}{ccccc} 0.0500 & 0.0500 & 0.0500 & 0.1311 & 0.1660 \\ 0.0500 & 0.9000 & 0.4818 & 0.1408 & 0.7020 \\ 0.9000 & 0.0500 & 0.4682 & 0.7281 & 0.1320 \\ 3 & 3 & 3 & 3 & 3 \\ \hline 0.4239 & 0.6181 & 0.6181 & 0.9000 & \\ 0.0500 & 0.1551 & 0.3319 & 0.0500 & \\ 0.5261 & 0.2268 & 0.0500 & 0.0500 & \\ 3 & 3 & 3 & 3 & \end{array}\right)$	– 8.4728	0.06
(19)	$\left(\begin{array}{ccccc} 0.5011 & 1.0198 & 1.0198 & 1.0198 & 1.0812 \\ 1.2223 & 1.1885 & 0.8669 & 1.8714 & 1.7564 \\ 1.2210 & 0.5389 & 1.8523 & 0.8987 & 1.7768 \\ 3 & 3 & 3 & 3 & 3 \\ \hline 1.2174 & 1.7894 & 1.7990 & 1.7990 & \\ 0.5025 & 1.7638 & 0.9367 & 1.0607 & \\ 1.1984 & 1.1627 & 0.8464 & 1.7246 & \\ 3 & 3 & 3 & 3 & \end{array}\right)$	– 10.1176	0.08
(20)	$\left(\begin{array}{ccccc} 0.5000 & 0.5000 & 0.5000 & 0.9415 & 0.9415 \\ 0.5000 & 0.9939 & 2.0000 & 0.5000 & 2.0000 \\ 0.5000 & 2.0000 & 0.5000 & 2.0000 & 2.0000 \\ \hline 2.0000 & 2.0000 & 2.0000 & 2.0000 & \\ 0.5000 & 0.8843 & 2.0000 & 1.1075 & \\ 1.0102 & 0.5000 & 0.7835 & 2.0000 & \\ 3 & 3 & 3 & 3 & \end{array}\right)$	– 5.6500	0.09
(21)	$\left(\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 & 3 & 3 & 1 & 3 & 1 \\ 1 & 1 & 2 & 3 & 1 & 3 & 1 & 2 & 3 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{array}\right)$	– 6.0040	1.38

Designs obtained using a locally optimal MINLP solver

Another setup is a problem where both spaces $\mathbf{X}$ and Ω_k^N are constrained independently. Here, in addition to requiring having 10 support points and at least two replicates from each support point, a minimum distance between any two support points is required. The complete set of constraints is represented by (23), and each constraint depends on the number of replicates or the location of the support points. Assuming $k = 10$, the 10-point optimal designs sought here satisfies

$$\gamma_1(\boldsymbol{n}) := n_i \geq 2, \quad \forall i \in [\![k]\!] \tag{23a}$$

$$\gamma_2(\boldsymbol{x}) := \delta^2 \leq \sum_{j=1}^{3} (x_{j,i} - x_{j,l})^2, \quad \forall i,\ l \in [\![k]\!],\ l \geq i+1 \tag{23b}$$

where we set $\delta = 0.2$. This design has k distinct support points, and the number of replicates at each support point is 2 or more subject to the constraint (23b) that assures a minimum distance between each pair of support points.

Tables 12 and 13 display the constrained optimal designs found from our algorithm and note that the designs obtained are not uniform due to the value of k.

4.3.3 Constraints on the design and replicates space

In practice, there may be constraints on both the design space and the number of replicates allowed at each point. For example, there are budget constraints to account for, the unitary cost of each factor is different and linearly proportional to the levels chosen in the experiment. An example is

$$\gamma_1(\boldsymbol{x}, \boldsymbol{n}) := c_0 + c_1 \sum_{i=1}^{k} n_i\, x_{1i} + c_2 \sum_{i=1}^{k} n_i\, x_{2i} + c_3 \sum_{i=1}^{k} n_i\, x_{3i} \leq b, \tag{24}$$

where $c_0 = 15$ is the fixed experimental cost, $c_1 = 4$, $c_2 = 5$, $c_3 = 6$ are the variable experimental costs of each factor in the design, and $b = 350$ is the budget available for the complete study. Here, we set $k = n_\theta$ as in the unconstrained design problem.

Tables 12 and 13 present D- and A-optimal exact designs found by our algorithm (third line) and subject to constraint (24). We notice the D-optimal exact design is not uniform in this case, and more replicates are allocated to treatment levels that combine lower levels of factors to satisfy the cost constraint. This trend is more evident for factors with higher

Table 11 Constrained A-optimal exact designs for Model 13 in Table 1 with $N = 27$ and $k = 9$

Constraints	Design	Optimum value	CPU (s)
(18)	$\begin{pmatrix} 0.0500 & 0.0789 & 0.0789 & 0.0789 & 0.1198 \\ 0.4686 & 0.7939 & 0.8711 & 0.0866 & 0.4159 \\ 0.4814 & 0.1271 & 0.0500 & 0.8345 & 0.4642 \\ 3 & 4 & 2 & 3 & 4 \\ \hline 0.4539 & 0.4539 & 0.4539 & 0.8501 & \\ 0.1321 & 0.4796 & 0.0500 & 0.0718 & \\ 0.4141 & 0.0665 & 0.4961 & 0.0781 & \\ 3 & 3 & 3 & 2 & \end{pmatrix}$	4.0689×10^3	13.63
(19)	$\begin{pmatrix} 0.5011 & 0.9533 & 0.9533 & 0.9533 & 0.9533 \\ 1.2508 & 1.1358 & 0.8252 & 1.8378 & 1.7413 \\ 1.2102 & 0.5707 & 1.7922 & 0.8910 & 1.7328 \\ 2 & 3 & 4 & 4 & 3 \\ \hline 1.2026 & 1.7577 & 1.7995 & 1.8199 & \\ 0.5120 & 1.7249 & 0.7940 & 1.1122 & \\ 1.1250 & 1.5314 & 1.4793 & 0.7823 & \\ 2 & 3 & 3 & 3 & \end{pmatrix}$	492.7252	2.39
(20)	$\begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.8924 & 0.8924 \\ 0.5000 & 2.0000 & 1.1455 & 0.6636 & 2.0000 \\ 1.1455 & 2.0000 & 0.5000 & 2.0000 & 0.6636 \\ 3 & 2 & 3 & 4 & 4 \\ \hline 0.9302 & 2.0000 & 2.0000 & 2.0000 & \\ 0.5000 & 1.2153 & 0.5000 & 2.0000 & \\ 0.5000 & 0.5000 & 1.2153 & 2.0000 & \\ 3 & 3 & 3 & 2 & \end{pmatrix}$	135.05121	6.27
(21)	$\begin{pmatrix} 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 2 & 1 & 3 & 1 & 3 & 1 & 3 & 2 & 2 \\ 2 & 1 & 1 & 3 & 3 & 2 & 2 & 1 & 3 \\ 4 & 2 & 3 & 2 & 2 & 4 & 3 & 4 & 3 \end{pmatrix}$	388.7682	12.12

Designs obtained using a locally optimal MINLP solver

Table 12 Constrained D-optimal exact designs for Model 13 in Table 1 with $N = 27$

Constraints	k	Design	Optimum value	CPU (s)
(22)	10	$\begin{pmatrix} 0.5000 & 0.5000 & 0.6682 & 1.0196 & 0.5000 \\ 0.8530 & 2.0000 & 2.0000 & 0.5000 & 0.5000 \\ 2.0000 & 0.5000 & 2.0000 & 2.0000 & 0.5000 \\ 3 & 3 & 2 & 3 & 3 \\ \hline 2.0000 & 2.0000 & 2.0000 & 2.0000 & 2.0000 \\ 0.7929 & 0.8252 & 2.0000 & 0.5000 & 2.0000 \\ 2.0000 & 0.5000 & 0.6388 & 1.0170 & 2.0000 \\ 2 & 3 & 3 & 3 & 2 \end{pmatrix}$	– 5.4073	1.16
(23)	10	$\begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.9024 \\ 0.5000 & 2.0000 & 2.0000 & 0.8305 & 0.5000 \\ 0.8853 & 0.5000 & 2.0000 & 2.0000 & 0.5000 \\ 3 & 3 & 3 & 3 & 2 \\ \hline 0.9024 & 2.0000 & 2.0000 & 2.0000 & 2.0000 \\ 0.5000 & 0.7494 & 2.0000 & 0.5000 & 2.0000 \\ 2.0000 & 0.5000 & 0.7295 & 2.0000 & 2.0000 \\ 2 & 2 & 3 & 3 & 3 \end{pmatrix}$	– 5.4709	0.55
(24)	9	$\begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.7968 & 2.0000 & 2.0000 & 0.5000 & 0.5000 \\ 0.5000 & 2.0000 & 0.5000 & 2.0000 & 0.7789 \\ 5 & 1 & 2 & 2 & 5 \\ \hline 0.8005 & 2.0000 & 2.0000 & 2.0000 & \\ 0.5000 & 0.5000 & 2.0000 & 0.5000 & \\ 0.5000 & 2.0000 & 0.5000 & 0.5000 & \\ 6 & 1 & 2 & 3 & \end{pmatrix}$	– 7.0964	1.60

Designs obtained using a locally optimal MINLP solver

Table 13 Constrained A-optimal exact designs for Model 13 in Table 1 with $N = 27$

Constraints	k	Design	Optimum value	CPU (s)
(22)	10	$\left(\begin{array}{ccccc} 2.0000 & 0.5000 & 2.0000 & 0.9813 & 0.5000 \\ 2.0000 & 0.9993 & 0.8047 & 2.0000 & 2.0000 \\ 0.8047 & 2.0000 & 2.0000 & 0.5000 & 0.9993 \\ 2 & 4 & 2 & 3 & 4 \\ \hline 0.5000 & 0.5000 & 1.4667 & 2.0000 & 0.9813 \\ 0.7908 & 0.5000 & 2.0000 & 0.5941 & 0.5000 \\ 0.5000 & 0.7908 & 2.0000 & 0.5941 & 2.0000 \\ 2 & 2 & 2 & 3 & 3 \end{array}\right)$	130.9651	1.27
(23)	10	$\left(\begin{array}{ccccc} 0.5000 & 0.5000 & 0.7802 & 0.8146 & 0.8146 \\ 1.2342 & 0.5000 & 2.0000 & 2.0000 & 0.5000 \\ 1.2342 & 0.5000 & 2.0000 & 0.5000 & 2.0000 \\ 3 & 2 & 2 & 3 & 3 \\ \hline 2.0000 & 2.0000 & 2.0000 & 2.0000 & 2.0000 \\ 1.0398 & 0.9062 & 2.0000 & 0.5000 & 2.0000 \\ 0.5000 & 2.0000 & 0.9062 & 1.0398 & 2.0000 \\ 4 & 2 & 2 & 4 & 2 \end{array}\right)$	128.6433	0.63
(24)	9	$\left(\begin{array}{ccccc} 0.5000 & 0.5000 & 0.5000 & 0.5000 & 0.8561 \\ 2.0000 & 2.0000 & 0.5542 & 0.5893 & 1.0431 \\ 2.0000 & 0.7417 & 2.0000 & 0.5000 & 0.5000 \\ 1 & 2 & 2 & 8 & 4 \\ \hline 0.8561 & 2.0000 & 2.0000 & 2.0000 & \\ 0.5000 & 2.0000 & 0.6329 & 0.5000 & \\ 0.5701 & 1.2799 & 0.6543 & 2.0000 & \\ 6 & 1 & 2 & 1 & \end{array}\right)$	221.6272	4.47

Designs obtained using a locally optimal MINLP solver

cost in the experimental design. This example shows that D-optimal exact designs generated by our algorithm are not necessarily uniform if additional constraints are involved.

5 Summary

Our paper is the first to apply MINLP formulations to find D- and A-optimal exact designs for linear models and locally optimal designs for nonlinear models. We use algebra-based concepts to develop MINLP representations of the optimal exact design problems. The resulting problem may have multiple local optima, and we use global and local optimization MINLP solvers to determine the optimum. Our work suggests that there are numerical advantages of working with global optimization solvers for solving optimal exact design problems as they have the ability to assure that globally optimal designs are found. However, in our examples, local optimizers in most of the tests find optimal exact designs that coincide with those obtained using global solvers and in shorter time. Our general recommendation is to try both global and local solvers and compare results. Practically, one advantage of our formulations is that the structure of the optimization problems is transparent to the solver that, through matrix decomposition techniques, can efficiently solve it.

We tested our problem formulations and algorithm for finding A- and D-optimal exact designs for a range of linear and nonlinear models with one or more regressors, and they all performed well with a short CPU time. We recall that there is rarely a theoretical proof that an exact design is optimal for a nonlinear model and so, for comparison purposes, some of our models were chosen so that we can confirm our results with the few theoretical results reported in the literature.

The D-optimal exact designs coincide with optimal approximate designs when N is a multiple of the number of design points which is expected. We note that when the D-optimal exact designs are uniformly supported at their design points, but the same is not true for A-optimal exact designs. Using several examples, we also demonstrated that mathematical programming techniques can easily incorporate additional constraints in the design problems and as the number of constraints grow in complexity, the feasibility region shrinks and the optimum is easier to find, assuming it exists. However, the performance of the algorithm depends on the size of the problem and the constraints imposed on the design and replicates spaces.

Acknowledgements The research of Wong is partially supported by a grant from the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R01GM107639. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors acknowledge two anonymous reviewers that contributed undoubtedly to improve the quality of the paper.

References

Atkinson, A., Donev, A.: The construction of exact D-optimum experimental designs with application to blocking response surface designs. Biometrika **76**(3), 515–526 (1989)

Atkinson, A., Donev, A., Tobias, R.: Optimum Experimental Designs, with SAS. Oxford University Press, Oxford (2007)

Berman, S.: An extension of the arc sine law. Ann. Math. Stat. **33**(2), 681–684 (1962)

Boer, E., Hendrix, E.: Global optimization problems in optimal design of experiments in regression models. J. Glob. Optim. **18**, 385–398 (2000)

Box, G., Hunter, W.: The experimental study of physical mechanisms. Technometrics **7**(1), 23–42 (1965)

Chang, F.C., Imhof, L., Sun, Y.Y.: Exact D-optimal designs for first-order trigonometric regression models on a partial circle. Metrika Int. J. Theor. Appl. Stat. **76**(6), 857–872 (2013)

Chang, F.C., Yeh, Y.R.: Exact A-optimal designs for quadratic regression. Stat. Sin. **8**, 527–534 (1998)

Chernoff, H.: Locally optimal designs for estimating parameters. Ann. Math. Stat. **24**, 586–602 (1953)

Coale, A., McNeil, D.: The distribution by age of the frequency of first marriage in a female cohort. J. Am. Stat. Assoc. **67**(340), 743–749 (1972)

Cook, R., Nachtsheim, C.: Comparison of algorithms for constructing D-optimal design. Technometrics **22**(3), 315–324 (1980)

Donev, A.: Crossover designs with correlated observations. J. Biopharm. Stat. **8**(2), 249–262 (1998). PMID: 9598421

Drud, A.: CONOPT: a GRG code for large sparse dynamic nonlinear optimization problems. Math. Program. **31**, 153–191 (1985)

Du Croz, J., Higham, N.: Stability of methods for matrix inversion. IMA J. Numer. Anal. **12**, 1–19 (1992)

Duarte, B., Wong, W.: Finding Bayesian optimal designs for nonlinear models: a semidefinite programming-based approach. Int. Stat. Rev. **83**(2), 239–262 (2015)

Duarte, B., Wong, W., Atkinson, A.: A semi-infinite programming based algorithm for determining T-optimum designs for model discrimination. J. Multivar. Anal. **135**, 11–24 (2015)

Duarte, B., Wong, W., Dette, H.: Adaptive grid semidefinite programming for finding optimal designs. Stat. Comput. **28**(2), 441–460 (2018)

Duran, M., Grossmann, I.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**(3), 307–339 (1986)

Esteban-Bravo, M., Leszkiewicz, A., Vidal-Sanz, J.: Exact optimal experimental designs with constraints. Stat. Comput. **27**(3), 845–863 (2017)

Fedorov, V.: Theory of Optimal Experiments. Academic Press, New York (1972)

Fedorov, V., Leonov, S.: Optimal Design for Nonlinear Response Models. Chapman and Hall/CRC Press, Boca Raton (2014)

Fletcher, R., Leyffer, S.: Numerical experience with lower bounds for MIQP branch-and-bound. SIAM J. Optim. **8**(2), 604–616 (1998)

Floudas, C.: Mixed-integer nonlinear optimization. In: Pardalos, P., Resende, M. (eds.) Handbook of Applied Optimization, pp. 451–475. Oxford University Press, Oxford (2002)

Gaffke, N.: On D-optimality of exact linear regression designs with minimum support. J. Stat. Plan. Inference **15**, 189–204 (1987)

GAMS Development Corporation: GAMS—a user's guide, GAMS release 24.2.1. GAMS Development Corporation, Washington, DC, USA (2013a)

GAMS Development Corporation: GAMS—the solver manuals, GAMS release 24.2.1. GAMS Development Corporation, Washington, DC, USA (2013b)

Golub, G.H., van Loan, C.F.: Matrix Computations, 4th edn. JHU Press, Baltimore (2013)

Goos, P., Donev, A.: Blocking response surface designs. Comput. Stat. Data Anal. **51**(2), 1075–1088 (2006)

Goos, P., Vandebroek, M.: D-optimal response surface designs in the presence of random block effects. Comput. Stat. Data Anal. **37**(4), 433–453 (2001)

Goos, P., Vandebroek, M.: D-optimal split-plot designs with given numbers and sizes of whole plots. Technometrics **45**(3), 235–245 (2003)

Gotwalt, C., Jones, B., Steinberg, D.: Fast computation of designs robust to parameter uncertainty for nonlinear settings. Technometrics **51**(1), 88–95 (2009)

Gribik, P., Kortanek, K.: Equivalence theorems and cutting plane algorithms for a class of experimental design problems. SIAM J. Appl. Math. **32**, 232–259 (1977)

Harman, R., Filová, L.: Computing efficient exact designs of experiments using integer quadratic programming. Comput. Stat. Data Anal. **71**, 1159–1167 (2014)

Harman, R., Jurík, T.: Computing c-optimal experimental designs using the simplex method of linear programming. Comput. Stat. Data Anal. **53**(2), 247–254 (2008)

Hill, A.: The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. J. Physiol. **40**(Suppl.), 4–7 (1910)

Imhof, L.: A-optimum exact designs for quadratic regression. J. Math. Anal. Appl. **228**, 157–165 (1998)

Imhof, L., Lopez-Fidalgo, J., Wong, W.K.: Efficiencies of rounded optimal approximate designs for small samples. Stat. Neerlandica **55**(3), 301–318 (2001)

Johnson, M., Nachtsheim, C.: Some guidelines for constructing exact D-optimal designs on convex design spaces. Technometrics **25**, 271–277 (1983)

Kiefer, J.: General equivalence theory for optimum design (approximate theory). Ann. Stat. **2**, 849–879 (1974)

Laird, A.: Dynamics of tumor growth. Br. J. Cancer **18**(3), 490–502 (1964)

Lastusilta, T., Bussieck, M., Westerlund, T.: Comparison of some high-performance MINLP solvers. Chem. Eng. Trans. **11**, 125–130 (2007)

Leszkiewicz, A.: Three essays on conjoint analysis: optimal design and estimation of endogenous consideration sets. Ph.D. thesis, Universidad Carlos III de Madrid (2014)

López-Fidalgo, J., Tommasi, C., Trandafir, P.: Optimal designs for discriminating between some extensions of the Michaelis–Menten model. J. Stat. Plan. Inference **138**(12), 3797–3804 (2008)

Mandal, A., Wong, W.K., Yu, Y.: Algorithmic searches for optimal designs. In: Dean, A., Morris, M., Stufken, J., Bingham, D. (eds.) Handbook of Design and Analysis of Experiments, pp. 755–786. CRC Press, Boca Ratton (2015)

Meyer, R., Nachtsheim, C.: The coordinate-exchange algorithm for constructing exact optimal experimental designs. Technometrics **37**, 60–69 (1995)

Mitchell, T.: An algorithm for the construction of D-optimal experimental designs. Technometrics **20**, 203–210 (1974)

Mitchell, T., Miller Jr., F.: Use of Design Repair to Construct Designs for Special Linear Models. Technical Report, pp. 130–131, Oak Ridge National Laboratory (1970)

Molchanov, I., Zuyev, S.: Steepest descent algorithm in a space of measures. Stat. Comput. **12**, 115–123 (2002)

Nuñez Ares, J., Goos, P.: An integer linear programming approach to find trend-robust run orders of experimental designs. J. Qual. Technol. **51**(1), 37–50 (2019). https://doi.org/10.1080/00224065.2018.1545496

Palhazi Cuervo, D., Goos, P., Sörensen, K.: Optimal design of large-scale screening experiments: a critical look at the coordinate-exchange algorithm. Stat. Comput. **26**(1), 15–28 (2016)

Pantelides, C.: The consistent initialization of differential-algebraic systems. SIAM J. Sci. Stat. Comput. **9**(2), 213–231 (1988)

Pukelsheim, F.: Optimal Design of Experiments. SIAM, Philadelphia (1993)

Rasch, D., Hendrix, E., Boer, E.: Replication-free optimal designs in regression analysis. Comput. Stat. **12**, 19–52 (1997)

Sagnol, G.: Computing optimal designs of multiresponse experiments reduces to second-order cone programming. J. Stat. Plan. Inference **141**(5), 1684–1708 (2011)

Sagnol, G., Harman, R.: Computing exact D-optimal designs by mixed integer second order cone programming. Ann. Stat. **43**(5), 2198–2224 (2015)

Sahinidis, N.: BARON 14.3.1: global optimization of mixed-integer nonlinear programs, user's manual. The Optimization Firm, LLC, Pittsburgh (2014)

Sartono, B., Goos, P., Schoen, E.: Constructing general orthogonal fractional factorial split-plot designs. Technometrics **57**(4), 488–502 (2015a)

Sartono, B., Schoen, E., Goos, P.: Blocking orthogonal designs with mixed integer linear programming. Technometrics **57**(3), 428–439 (2015b)

SAS Institute Inc.: JMP® 13 User Guide, 2nd edn. SAS Institute Inc, Cary (2017)

Tawarlamani, M., Sahinidis, N.: Convexification and Global Optimization in Continuous and Mixed Integer Nonlinear Programming, 1st edn. Kluwer Academic Publishers, Dordrecht (2002)

Vandenberghe, L., Boyd, S.: Applications of semidefinite programming. Appl. Numer. Math. **29**, 283–299 (1999)

Vo-Thanh, N., Jans, R., Schoen, E., Goos, P.: Symmetry breaking in mixed integer linear programming formulations for blocking two-level orthogonal experimental designs. Comput. Oper. Res. **97**, 96–110 (2018)

Welch, W.: Branch-and-bound search for experimental designs based on D-optimality and other criteria. Technometrics **24**(1), 41–48 (1982)

Westerlund, T., Pettersson, F.: An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. **19**, 131–136 (1995)

Wynn, H.: The sequential generation of D-optimum experimental designs. Ann. Math. Stat. **41**(5), 1655–1664 (1970)

Yang, J., Mandal, A., Majumdar, D.: Optimal design for two-level factorial experiments with binary response. Stat. Sin. **22**(2), 885–907 (2012)

Yang, J., Mandal, A., Majumdar, D.: Optimal designs for 2^k factorial experiments with binary response. Stat. Sin. **26**, 381–411 (2016)