mathematics

MDPI

Article

Exact Optimal Designs of Experiments for Factorial Models via Mixed-Integer Semidefinite Programming

Belmiro P. M. Duarte [1,2]

[1] Department of Chemical and Biological Engineering, Coimbra Engineering Academia, Polytechnic Institute of Coimbra, Rua Pedro Nunes, Quinta da Nora, 3030-199 Coimbra, Portugal; bduarte@isec.pt

[2] Centro de Investigação em Engenharia dos Processos Químicos e dos Produtos da Floresta, Department of Chemical Engineering, University of Coimbra, Rua Sílvio Lima, Pólo II, 3030-790 Coimbra, Portugal

Abstract: The systematic design of exact optimal designs of experiments is typically challenging, as it results in nonconvex optimization problems. The literature on the computation of model-based exact optimal designs of experiments via mathematical programming, when the covariates are categorical variables, is still scarce. We propose mixed-integer semidefinite programming formulations, to find exact D-, A- and I-optimal designs for linear models, and locally optimal designs for nonlinear models when the design domain is a finite set of points. The strategy requires: (i) the generation of a set of candidate treatments; (ii) the formulation of the optimal design problem as a mixed-integer semidefinite program; and (iii) its solution, employing appropriate solvers. For comparison, we use semidefinite programming-based formulations to find equivalent approximate optimal designs. We demonstrate the application of the algorithm with various models, considering both unconstrained and constrained setups. Equivalent approximate optimal designs are used for comparison.

Keywords: factorial experiments; exact designs; mixed-integer semidefinite programming; model-based optimal designs

MSC: 62K05; 90C47

Citation: Duarte, B.P.M. Exact Optimal Designs of Experiments for Factorial Models via Mixed-Integer Semidefinite Programming. *Mathematics* **2023**, *11*, 854. https://doi.org/10.3390/math11040854

Academic Editor: Victor Orlov

Received: 8 January 2023
Revised: 3 February 2023
Accepted: 6 February 2023
Published: 7 February 2023

1. Motivation

We consider the problem of systematically constructing exact optimal designs of experiments for surface response models. The innovation of the paper is that it proposes a mixed-integer semidefinite programming (MISDP) formulation for such a purpose, and demonstrates its application to D-, A- and I-optimality criteria. MISDP enables the obtaining of globally optimal designs, and the addressing of both constrained and unconstrained design setups. Because equivalent approximate designs serve to assess the optimality of the exact designs, they are obtained with semidefinite-programming-based formulations.

Optimal design of experiments seeks to generate efficient plans for experiments used to conduct experimental work for model parametrization and model discrimination, and is an interesting field for the application of optimization algorithms. The topic has gained importance due to the need to judiciously choose the most informative experiments, given the resources available, mainly due to budget constraints. Optimal design of experiments allows important cost savings in scientific studies, by determining the most efficient design from which to make an accurate inference [1]. Applications range from biostatistics to engineering, social sciences and marketing. The paradigm behind optimal design of experiments is that, 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 and the number of replicates at each design point.

There are two types of designs: (i) continuous or *approximate designs*; (ii) discrete or *exact designs*. In both kinds of designs, the relative effort at each experimental point x_i, also designated *weight*, is represented by $w_i = n_i/N$, where n_i is the number of replications taken

at the ith experimental point, and N is the number of observations of the experimental plan. Approximate designs have all the n_i's constrained to the set of non-negative real numbers. Exact designs appear when all the n_i's are non-negative integer numbers, i.e., $n_i \in \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of non-negative integer numbers [2]. In both cases, the weights form a discrete probability measure (i.e., $\sum_{i=1}^{k} w_i = 1$): they are limited to $[0, 1]$ for continuous designs, and to the set of rational numbers in $[0, 1]$ for discrete designs, which is finite for a given N. The resulting optimization problem to find approximate designs is convex (or can be reformulated as convex) if the design criterion is a convex function of the Fisher Information Matrix (FIM). Consequently, global optimality is guaranteed [3,4] by equivalence theorems, and tailored-based optimization algorithms can be employed [5].

Exact designs are characterized by an integer number of observations for all design points whose sum is exactly N. Thus, the optimal design problem can be cast as a combinatorial (non-convex) problem. Consequently, they are computationally more challenging than approximate optimal designs [6]. This complexity explains why finding and studying properties of exact optimal designs is harder than for approximate optimal designs [7]. However, exact optimal designs are important in their own right because they are the only practically useful designs, in the sense that: (i) simply rounding the number of observations at each point from the corresponding approximate optimal design to an optimal exact design may cause the loss of substantial efficiency when N is small [8]; (ii) even with a moderate sample size, finding an approximate optimal design does not assure that it can be implementable when some of the optimal proportions at some design points are small. The systematic algorithms to handle exact optimal designs are still scarce, and mathematical programming-based tools provide sophisticated state-of-the-art algorithms to address the problem at hand. Algorithms for convex optimization are desirable, as they run in polynomial time. Even considering exact designs require two levels of optimization (one for solving the relaxed semidefinite programming (SDP) problems, and the other for integer variables) a tool including an inner solver relying on SDP (a convex class of problems) may be beneficial. Thus, we believe there is room to exploit MISDP to handle exact designs, especially in problems where the number of candidate points is limited, as is the case with factorial experiments.

Our aim in this paper was to develop/adapt systematic general MISDP formulations to find D-, A- and I-exact optimal factorial designs for linear models and locally optimal factorial designs for nonlinear models. We believe this is the first time that MISDP formulations have been proposed and applied to finding exact optimal designs. Our approach required the previous construction of all candidate treatments: the subsequent resolution found the number of replicates at each one. The applicability of the proposed methodology is demonstrated by considering the A-, D- and I-optimal factorial designs for different polynomial models, where both constrained and unconstrained setups are considered. The use of a mathematical programming-based approach enabled finding optimal designs for every possible number of experiments, and including additional constraints into the design problem that could, for instance, represent non-regular domains or forbidden levels.

1.1. Algorithms for Finding Optimal Experimental Designs

There are several deterministic algorithms for finding approximate designs of experiments, where by "deterministic" we mean algorithms that use a sequential set of rules to evolve the solution to the optimum. Those algorithms do not include randomness sources, and a proof of convergence can be theoretically derived. Equivalence theorems are used to assess the optimality of the solutions, and to iteratively choose new candidate points. Common examples are the Wynn–Fedorov algorithm [9,10] and the exchange algorithm [11]. Overall, they are found to be effective. Two common types of exchange algorithms are: (i) the point exchange algorithm [11]; (ii) the coordinate exchange algorithm [12]. Another approach to finding continuous optimal designs is based on multiplicative algorithms, which have found broad application due to their simplicity [13,14]. Recently, cocktail algorithms, that rely on both exchange and multiplicative algorithms, have been proposed [15,16].

Mathematical programming methods provide alternative approaches to generating approximate designs. Some examples for finding approximate designs include linear

programming [17], second-order cone programming [18], semidefinite programming [5,19], semi-infinite programming [20], and nonlinear programming (NLP) [21].

Rounding approaches are the most common way to convert approximate designs, that are numerically easier to compute, to equivalent exact designs (see [22]). For an implementation based on the cocktail algorithm, the reader is referred to de la Calle-Arroyo et al. [23].

Various numerical algorithms specifically developed for the construction of exact designs are based on exchange methods, and were initially proposed for the D-optimality criterion [9,10,24]: they are initialized with a feasible initial design, and iteratively delete/include new points from a grid of candidates, until a convergence criterion is attained. The algorithm does not assure global optimality, and to overcome this issue the search is run from various starting points. Later refinements of this approach include the DETMAX algorithm [25], the modified Fedorov algorithm [11], and the KL-exchange algorithm [26]. Point-exchange algorithms have been used to construct response surface designs in random blocks [27], D-optimum split-plot designs [28] and crossover designs [29]. The coordinate-exchange algorithm [12] overcomes some of the problems of point-exchange, by avoiding the explicit enumeration of candidate design sets for discrete factors: nevertheless, as with other exchange algorithms, it has the tendency to get trapped in locally optimal designs.

Recently, mixed-integer (linear) programming has been extensively used to find factorial designs for screening purposes, specifically in the construction of mixed-level orthogonal and two-level orthogonally blocked designs [30], orthogonal fractional factorial split-plot designs [31], trend robust run-order designs [32] and for breaking the symmetry of blocking two-level orthogonal experimental designs, which helps in finding optimal orthogonal blocking patterns [33].

Application examples of mathematical programming for solving the exact optimal design problem for model parametrization are more scarce, due to the additional complexity involved. In Welch [34], the design space was discretized, and a convex optimization algorithm based on branch-and-bound was used, to ensure that optimal replicates of D-exact designs were integer. Similarly, Harman and Filová [35] and Sagnol and Harman [36] used, respectively, mixed-integer quadratic programming and mixed-integer second-order cone programming techniques to find exact D-optimal designs. Both methods required discretizing the design space, ensuring that the global optimal design was found in the discretized space. The extension of the mixed-integer second-order cone programming formulation to nonlinear models of the Generalized Linear Model class, using Bayesian frameworks combined with quasi-Monte Carlo integration techniques, is addressed in Duarte and Sagnol [37].

The resulting optimization problems can be solved efficiently by using state-of-the-art solvers employing branch-and-bound techniques, such as `Mosek` [38] and `PICOS` [39]. Potential issues with the aforementioned methods include the exponential increase of the optimization problem with the model dimension and candidate points. Although coarser grids could reduce the size of the optimization problems, they lead to designs with lower efficiency, because the discretized design space may not capture all the model features in the original space.

Esteban-Bravo et al. [40] showed that NLP formulations can be used to find unconstrained and constrained exact designs, and Newton-based methods using Interior Point or Filter techniques perform well in this context: their formulation for exact designs considers that an appropriate probability measure, formed by rational numbers, is to be obtained for an approximate design. Duarte et al. [41] have developed explicit mixed-integer nonlinear programming (MINLP) formulations, based on algebra-based strategies, that enable the construction of exact optimal designs for linear and nonlinear models within constrained and unconstrained design domains. An issue of NLP and MINLP formulations is that they require setting the number of support points of the design.

1.2. Novelty and Organization

This paper contains three elements of novelty: (i) MISDP formulations for finding D-, A- and I-exact optimal factorial designs or locally optimal factorial designs of experiments for linear models (within the parameters), and nonlinear models, respectively; (ii) the extension of the formulations to include additional linear constraints on the replicates;

(iii) the comparison of exact designs obtained with MISDP and equivalent approximate designs obtained with SDP.

The paper is organized as follows: Section 2 presents the statistical background, and reviews the general problem; in Section 3, we provide the formulations for exact D- and A-optimal design problems, and discuss the modifications required to find constrained designs; Section 4 implements the algorithm to generate exact optimal full and fractional factorial designs, and extends the formulation to the I-optimality criterion; Section 5 concludes with a summary.

2. Background

This Section provides the background material required by the mathematical formulation for obtaining approximate and exact optimal designs for factorial experiments using SDP and MISDP (see Section 2.1). Section 2.2 introduces the fundamentals of the two classes of problems.

Here and throughout, we use bold-face lowercase letters to represent vectors, bold-face capital letters for continuous domains, blackboard bold capital letters for discrete domains, and capital letters for matrices. Let $[\![k]\!] = \{1, \cdots, k\}$ be the set containing all integer elements from 1 to k. The acronyms used throughout the paper are identified in back part.

2.1. Nomenclature and Design Setup

All the regression models addressed in this study are linear, and have a univariate response, with n_x independent variables $\mathbf{x} \in \mathbf{X}$, where $\mathbf{X}$ is a discrete set containing all possible combinations of covariates, and the mean response at $\mathbf{x}$ is

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

Here, $\mathbf{p} \in \mathbf{P} \subset \mathbb{R}^{n_p}$ is a vector of unknown model parameters, where n_p is the number of parameters of the model, $\mathbf{f}(\bullet)$ is an n_p-size vector of functions limited to polynomials on the covariates and constant terms and $\mathbb{E}[\bullet]$ is the expectation operator with respect to the error distribution. In Section 4.1, we determine locally optimal designs for nonlinear models which, after linearization, fall into Equation (1).

We focus on factorial designs where each covariate x_i varies over a finite (ordered) set of values that correspond to different levels in the experimental design. Specifically, the covariates are *ordinal variables*. Let us assume that the ith covariate x_i has l_i levels, i.e.

$$x_i \in \Lambda_i \equiv \{\underbrace{\alpha_i, \cdots, \beta_i}_{l_i}\}$$

where α_i and β_i are the lower and upper levels, respectively, at which x_i can be set in the experiment. The ordered set Λ_i contains all the possible levels for the ith covariate, and the jth level of covariate i is denoted by $\lambda_{i,j}$. A *treatment* is a vector containing a combination of values $\lambda_{i,j}$ for each covariate, and the design space is $\mathbf{x} \in \mathbf{X} \equiv \bigotimes_{i=1}^{n_x} x_i$, $x_i \in \Lambda_i$. To distinguish the covariates' mathematical representation from the specific values they have in $\mathbf{X}$, we use $\mathbf{z}_j = \bigotimes_{i=1}^{n_x} \{\exists! x_i : \; x_i = \lambda_{i,j}, \; j \in [\![l_i]\!]\}$ to designate each treatment j. The complete set of candidate treatments is obtained from permutation of the levels of the covariates, and is represented by $\mathbf{Z} \equiv \{\mathbf{z}_j, \; j \in [\![k]\!]\}$, where $k = \prod_{i=1}^{n_x} l_i$ is the number of candidate treatments.

For convenience, let us consider a general design of experiments represented by a probability measure ξ on a continuous domain $\mathbf{X}$. By definition, ξ satisfies the equality $\int_{\mathbf{X}} \xi(\mathrm{d}x) = 1$. Equivalently, probability measures supported on a finite (discrete) subset of $\mathbf{X}$ validate the equality $\sum_{x_j \in \mathbf{X}} \xi(x_j) = \sum_{j=1}^{k} w_j = 1$, where $\xi(x_j) = w_j$, $\forall j \in [\![k]\!]$ is the relative experimental effort at x_j [42] (Chapter 1). In our context, *approximate designs*, ξ^a, are discrete probability measures that allocate a weight $w_j \geq 0$ to jth treatment $\mathbf{z}_j \in \mathbf{Z}$. A design ξ^a on $\mathbf{Z}$ is described by a vector of non-negative weights $\mathbf{w}$ summing to 1, where each element represents the relative replication of the treatments in $\mathbf{Z}$. Hence, the set of all feasible approximate

designs, Ξ^a, is the $k-1$-dimensional simplex $\left\{\mathbf{w} \in \mathbb{R}^k : \ w_j \geq 0, \ \forall j \in [\![k]\!], \ \sum_{j=1}^k w_j = 1\right\}$ where $\xi^a = \{(\mathbf{z}_j^\intercal, w_j)\}, \ j \in [\![k]\!]$.

By using the same rationale, an $N-$point *exact design* on $\mathbf{Z}$, ξ_N^e, is also a discrete probability measure, where the weights $w_j = n_j/N$ are constrained to rational values in $[0,1]$. In practice, the number of replications $n_j \in \mathbb{N}_0$ at $\mathbf{z}_j \in \mathbf{Z}$ satisfy the constraint $\sum_{j=1}^k n_j = N$. The set of feasible $N-$point exact designs $\xi_N^e = \{(\mathbf{z}_j^\intercal, w_j)\}, \ j \in [\![k]\!]$ is

$$\Xi_N^e \equiv \left\{\mathbf{w} \in \mathbb{R}^k : \ \exists n_1, \ldots, n_k \in \mathbb{N}_0, \ w_j = \frac{n_j}{N}, \ \sum_{j=1}^k n_j = N\right\}.$$

The worth of the design ξ^a (or ξ_N^e) is measured by a convex functional of its FIM. The elements of the normalized FIM obtained after adjusting for the sample size are the negative of the expectation of the second order derivatives of the log-likelihood of (1), given the set of candidate treatments $\mathbf{Z}$, $\mathcal{L}(\xi^a|\mathbf{Z})$, with respect to the parameters. This matrix is proportional to

$$\begin{aligned}\mathcal{M}(\xi|\mathbf{Z}) &= -\mathbb{E}\left[\frac{\partial}{\partial \mathbf{p}}\left(\frac{\partial \mathcal{L}(\xi|\mathbf{Z})}{\partial \mathbf{p}}\right)\right] = \int_{\mathbf{x}\in\mathbf{X}} M(\mathbf{x}|\mathbf{Z}) \ \mathrm{d}\,\xi(\mathbf{x}) \\ &= \sum_{j=1}^k w_j \ \mathbf{f}(\mathbf{z}_j) \ \mathbf{f}^\intercal(\mathbf{z}_j), \end{aligned} \tag{2}$$

where $\mathcal{M}(\xi|\mathbf{Z})$ is the *global* FIM of the design ξ, and $M(\mathbf{x}|\mathbf{Z})$ is the *elemental* FIM at $\mathbf{x}$.

When errors are normally and independently distributed, the volume of the confidence region for the model parameters $\mathbf{p}$ is inversely proportional to $\det[\mathcal{M}^{1/2}(\xi|\mathbf{Z})]$. Consequently, maximizing the determinant of the FIM, or equivalently its geometric mean, by choice of a design, leads to the most accurate estimates for the parameters. If the interest is in finding the exact D-optimal design, the optimization problem is

$$\xi_D = \arg\max_{\xi\in\Xi_N^e} \{\det[\mathcal{M}(\xi|\mathbf{Z})]\}^{1/n_p}. \tag{3}$$

Other design criteria optimize the FIM in different ways: for example, if the goal is minimizing the sum of the lengths of the axes of the confidence ellipsoid, one minimizes the nuclear norm of the parametric covariance matrix which, in turn, is proportional to the inverse of the FIM. This criterion is commonly designated A-optimality, and is formalized as the minimization of the trace of the inverse of the FIM [4]:

$$\xi_A = \arg\min_{\xi\in\Xi_N^e} \left\{\mathrm{tr}[\mathcal{M}(\xi|\mathbf{Z})^{-1}]\right\}. \tag{4}$$

The formulations for approximate designs are constructed similarly. For linear models, which we have, the FIM is independent of the unknown parameters $\mathbf{p}$, and the optimal design does not depend on $\mathbf{p}$; otherwise, the optimal design is locally optimal, as it depends on $\mathbf{p}$, which we want to estimate (see Model 5 in Section 4).

2.2. Semidefinite Programming and Mixed-Integer Semidefinite Programming

Semidefinite Programming and MISDP are employed to solve the optimal design problems for D- and A-optimality criteria, over a given treatment candidate set $\mathbf{Z}$. In this section, we introduce the fundamentals of these classes of mathematical programs.

Let $\mathbb{S}_+^{n_p}$ be the space of $n_p \times n_p$ symmetric positive semidefinite matrices, and $\mathbb{S}^{n_p}$ the space of $n_p \times n_p$ symmetric matrices. A convex set $\mathbf{S} \in \mathbb{R}^n$ is semidefinite representable (SDr) if all elements $\boldsymbol{\zeta} \in \mathbf{S}$ can be projected on to a higher dimensional set $\mathbf{S}^{\mathrm{exp}}$, which can

be described by linear matrix inequalities (LMIs) [43]: that is, $\mathbf{S}$ is SDr if and only if there exists some symmetric matrices, $M_0, \cdots, M_{m_1}, \cdots, M_{m_1+m_2} \in \mathbb{S}^{n_p}$, such that [44]

$$\mathbf{S}^{\text{exp}} \equiv \{(\boldsymbol{\zeta}, \mathbf{v}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} : \; M_0 + \sum_{i=1}^{m_1} \zeta_i \, M_i + \sum_{j=1}^{m_2} v_j \, M_{m_1+j} \succeq 0\} (\subset \mathbb{R}^{m_1+m_2}). \quad (5)$$

Here, $\succeq$ is the semidefinite operator, i.e., $A \succeq 0 \iff \langle A, \Omega \rangle \geq 0, \; \forall \Omega \in \mathbb{S}_+^{n_p}$, where $\langle ., . \rangle$ is the Frobenius inner product operator, $\boldsymbol{\zeta} \in \mathbb{R}^{m_1}$ is an element of the original set $\mathbf{S}$, and $\mathbf{v}$ is a point of the incremental subspace $\mathbb{R}^{m_2}$.

On its turn, a convex (or concave) function $\varphi : \mathbb{R}^{m_1} \mapsto \mathbb{R}$ is SDr if and only if the epigraph of φ, $\{(t, \boldsymbol{\zeta}) : \varphi(\boldsymbol{\zeta}) \leq t\}$ (or the hypograph, represented by $\{(t, \boldsymbol{\zeta}) : \varphi(\boldsymbol{\zeta}) \geq t\}$) is SDr, and can be cast by LMIs [45,46]. The optimal values, $\boldsymbol{\zeta}$, of SDr functions are then formulated as a *semidefinite program*, with the form:

$$\max_{\boldsymbol{\zeta}} (\text{or } \min_{\boldsymbol{\zeta}}) \left\{ \mathbf{c}^\intercal \, \boldsymbol{\zeta}, \sum_{i=1}^{m_1} \zeta_i \, M_i - M_0 \succeq 0 \right\}. \quad (6)$$

In our design context, $\mathbf{c}$ is a vector of known constants that depends on the design problem, and matrices M_i, $i = \{0, \ldots, m_1\}$ contain local FIMs and other matrices produced by the reformulation of the functions $\varphi(\boldsymbol{\zeta})$ into LMIs. The decision variables in vector $\boldsymbol{\zeta}$ are the weights w_j, $j \in [\![k]\!]$ of the optimal design and the other auxiliary variables required. The problem of calculating a design for a pre-specified set $\mathbf{Z}$ of treatments is solved by formulation (6) complemented with the linear constraints on $\mathbf{w}$ ($\mathbf{w} \geq 0$ and $\mathbf{1}^\intercal \, \mathbf{w} = 1$) that assure the feasibility of the probability measure. In the remaining sections of the paper, the problem of (6) is the general SDP problem, which can be adapted to find experimental designs employing different optimality criteria.

Ben-Tal and Nemirovski [45] (Chapters 2 and 3) have provided a list of SDr functions useful for solving continuous optimal design problems—see Boyd and Vandenberghe [46] (Section 7.3) for specific applications. Recently, Sagnol [47] showed that each criterion in the Kiefer's class of optimality criteria defined by

$$\Phi_\delta[\mathcal{M}(\xi)] = \left[\frac{1}{n_p} \; \text{tr}(\mathcal{M}(\xi)^\delta) \right]^{1/\delta} \quad (7)$$

is SDr for all rational values of $\delta \in (-\infty, -1]$, and that general SDP formulations exist: this result also applies to cases where $\delta \to 0$, and both criteria addressed in this work (i.e., problems (3) and (4)) fall into this class. Practically, the problem of optimally designing approximate plans of experiments for the most common (convex) criteria can be formulated as a semidefinite programming problem falling into the general representation category (5)—see Vandenberghe and Boyd [5], Duarte and Wong [19], among others.

The extension of formulations (3) and (4) to handle exact designs is straightforward, except that the feasibility domain Ξ^a is replaced by Ξ_N^e, the weights $\mathbf{w} = \mathbf{n}/N$ must be chosen, so that all the elements in $\mathbf{n}$ are integers, and $\mathbf{1}_k^\intercal \, \mathbf{n} = N$: consequently, some of the variables in $\boldsymbol{\zeta}$ are integers, and the resulting problem is of an integer conic nature. Typically, MISDP falls outside of the scope of traditional convex mixed-integer nonlinear programming, because (i) the SDP constraints are not differentiable everywhere; (ii) the MISDP problems are more efficiently solvable by using cone optimization algorithms, even requiring branch-and-bound techniques to handle integer variables, as the relaxed (inner) problems are convex. Due to the specificities of MISDP, and the advantages it offers in handling well-characterized design problems, it is currently getting increased attention (see Gally et al. [48]). Several applications of MISDP have been addressed, including: the truss topology design [49]; the optimal placement of metering systems in distribution grids [50], where the authors solve the measurement placement problem by exploiting the M-optimality design criterion; and optimal sensor placement, using the A- and D-optimality criteria (see Schäfer [51]). Here, we consider another related application of MISDP: the construction of exact optimal designs of experiments. As in the construction of

approximate optimal designs of experiments, the problem falls into (6), except that some of the variables are constrained to integer values.

3. MISDP Formulations for Exact Optimal Design of Experiments

This section presents the formulations for finding exact optimal designs via MISDP. The D-optimality criterion is addressed in Section 3.1, and the A-optimal designs in Section 3.2. In Section 3.3, we overview the approach followed to solve the MISDP problems, describing the optimal design of experiments.

In Section 4, we compare the exact optimal designs obtained from MISDP with equivalent approximate designs obtained using SDP formulations. The SDP formulations for linear models are currently state-of-the-art, and are presented in Vandenberghe and Boyd [5], Vandenberghe et al. [52], and Boyd and Vandenberghe [46] (Section 7.5), among others. Due to space limitations, we have centered our discussion on MISDP formulations.

3.1. Exact D-Optimal Designs

Here, we recall the exact D-optimal design problem (3). We extended the formulation proposed by Vandenberghe and Boyd [5] for approximate designs, to determine exact optimal designs: that is, given a statistical model, the set of candidate treatments $\mathbf{Z}$, the *elemental* FIMs for each candidate treatment and the total number of experiments N, the formulation for finding exact D–optimal designs on Ξ_N^e was formulated as follows:

$$\max_{\mathbf{n},\mathcal{B}} \prod_{j=1}^{n_p} \mathcal{B}_{j,j}^{1/n_p} \tag{8a}$$

$$\text{s.t. } \mathcal{M}(\xi|\mathcal{Z}) = \sum_{j=1}^{k} \frac{n_j}{N} M(\mathbf{z}_j|\mathcal{Z}) \tag{8b}$$

$$\left(\begin{array}{c|c} \mathcal{M}(\xi|\mathcal{Z}) & \mathcal{B} \\ \hline \mathcal{B}^\intercal & \text{diag}(\mathcal{B}) \end{array}\right) \succeq 0 \tag{8c}$$

$$\sum_{j=1}^{k} n_j = N \tag{8d}$$

$$n_j \in \mathbb{N}_0, \ \forall j.$$

Equation (8b) represents the generation of the global FIM. Equation (8c) is the linear matrix inequality (LMI) representing the LDL factorization of $\mathcal{M}(\xi|\mathbf{Z})$ [45]. $\mathcal{B} \in \mathbb{R}^{n_p \times n_p}$ is a lower-triangular matrix, and $\text{diag}(\mathcal{B})$ is a positive definite diagonal matrix containing the (ordered) elements of $\mathcal{M}(\xi|\mathbf{Z})$. Equation (8d) restricts the total number of observations to N. The determinant root in Equation (8a) is represented as the geometric mean of the diagonal elements of $\mathcal{B}$, which is also an SDr function, and can, in turn, be expressed as a series of 2×2 LMIs [53,54]. The FIMs of the candidate treatments, $M(\mathbf{z}_j|\mathbf{Z})$, $\mathbf{z}_j \in \mathcal{Z}$, are constructed with Equation (2).

3.2. Exact A-Optimal Designs

Now, we propose a MISDP formulation for determining exact A-optimal designs solving problem (4), where $\text{tr}[\mathcal{M}(\xi|\mathbf{Z})^{-1}]$ is minimized. We extend the formulation of Fedorov and Lee [55], proposed for approximate designs. The complete MISDP for computing the A-optimal designs is

$$\min_{\mathbf{n},\mathcal{F}} \sum_{j=1}^{n_p} \mathcal{F}_{j,j} \tag{9a}$$

$$\text{s.t. } \mathcal{M}(\xi|\mathbf{Z}) = \sum_{j=1}^{k} \frac{n_j}{N} M(\mathbf{z}_j|\mathbf{Z}) \tag{9b}$$

$$\left(\begin{array}{c|c} \mathcal{M}(\xi|\mathbf{Z}) & I_{n_p} \\ \hline I_{n_p} & \mathcal{F} \end{array}\right) \succeq 0 \tag{9c}$$

$$\sum_{j=1}^{k} n_j = N \tag{9d}$$
$$n_j \in \mathbb{N}_0,\ \forall j.$$

Equations (9b) and (9d) are similar to those in the D-optimal design problem. Equation (9c) is an LMI representing the Schur complement, and holds if and only if $\mathcal{M}(\xi|\mathbf{Z}) - I_{n_p}\mathcal{F}^{-1}I_{n_p} \succeq 0$; $\mathcal{F} \in \mathbb{R}^{n_p \times n_p}$, being a positive definite matrix, and I_{n_p} the n_p-size identity matrix; then, $\mathcal{F} \succeq \mathcal{M}^{-1}(\xi|\mathbf{Z})$. Consequently, the nuclear norm of $\mathcal{F}$ is larger than that of $\mathcal{M}^{-1}(\xi|\mathbf{Z})$. As the nuclear norm of a matrix is the sum of its eigenvalues, the upper bound of the nuclear norm of $\mathcal{M}^{-1}(\xi|\mathbf{Z})$ is $\mathrm{tr}(\mathcal{F})$, which we want to minimize by choosing $\mathbf{n}$. Finally, Equation (9a) represents the objective function, which is equivalent to the sum of all the elements in the diagonal of $\mathcal{F}$.

3.3. Numerical Strategy

Here, we describe the numerical approach employed to handle the optimal design problems previously formulated as MISDPs. The formulations (8) and (9) were applied, to find unconstrained optimal designs in Ξ_N^e. An advantage of our formulations is that when there are additional constraints, such as restrictions on the number of replicates at each design point, they can be incorporated into the design problems. Here, we limited the constraints to linear inequalities or equalities; hence, our method could also find constrained exact optimal designs from the set $\Xi_N^e \cap \{(\mathbf{n}|\mathbf{Z}) : A\mathbf{n}^\intercal \leq \mathbf{b}_{in}^\intercal,\ B\mathbf{n}^\intercal = \mathbf{b}_{eq}^\intercal\}$, where $A \in \mathbb{R}^{n_{in} \times k}$ and $B \in \mathbb{R}^{n_{eq} \times k}$ were user-specified matrices containing the coefficients of linear inequalities and equalities, respectively, on the replicates space, where $\mathbf{b}_{in} \in \mathbb{R}^{n_{in}}$ and $\mathbf{b}_{eq} \in \mathbb{R}^{n_{eq}}$ were vectors of constants, and where n_{in} was the number of inequalities and n_{eq} the number of equality constraints of the problem.

To handle the approximate designs via SDP there are user-friendly interfaces, such as `cvx` [54] or `YALMIP` [56], that automatically transform the constraints into a series of LMIs, before passing them to SDP solvers, such as `SeDuMi` [57] or `Mosek` [38]: this is possible if Φ_δ is SDr, which was true for our design criteria of interest. In our work, we solved all approximate design problems using the `cvx` environment combined with the solver `Mosek`, which uses an efficient interior-point-based algorithm. The reason `cvx` was chosen was because it is a high-level environment for convex programming, which includes parser functions that convert the optimality criteria in SDP formulations into a series of LMIs, and it supports the explicit modeling of the remaining (linear and conic) constraints. `Mosek` offers a significant speed-up relative to other alternatives.

A similar approach was used to solve the MISDP problems that represented exact optimal designs. The computational tool employed to solve this class of programs was implemented in Matlab, using `YALMIP`. A branch-and-bound method was used for handling the integer variables, and an SDP solver for handling the problems at the-branch and-bound nodes constructed by relaxing the integer variables. Here, we used `BNB` [58] for branch-and-bound, and `Mosek` [38] for solving the SDP relaxed problems. We chose `BNB` because it includes a MISDP solver standing on a branch-and-bound method that can be combined with a convex solver. The code is available by request.

All computations in Section 4 were done using an Intel Core i7 machine running a 64 bits Windows 10 operating system with a 2.80 GHz processor. In all problems, the relative and absolute tolerances used to solve the SDP and MISDP problems were set to 1×10^{-5}.

4. Numerical Results

We now report the D- and A-exact optimal designs for the linear and nonlinear models found from our formulations in Section 3. The literature on the construction of factorial designs is vast, and methods different to those based on the maximization of information criteria are described in various references, such as Cheng [59], and Atkinson et al. [60]. Various practical examples similar to our models can be found in applied textbooks (see Antony [61], Goos and Jones [62], Box [63] among others). The models used for testing are presented in Table 1. We note that the regression factors were inherently ordinal variables, each one having multiple levels.

Model 1 was linear (in its parameters), and included only two factors with three levels; it is considered herein to assess the accuracy of the algorithm as it was solved in Atkinson [64]. Models 2–4 were also linear (in their parameters), and demonstrated the application of the formulations to find factorial designs for response surface models restricted to second-order interactions with four and three factors, respectively. Models 2 and 3 had two levels, and they only differed because the discrete design space for Model 3 included the center point $(0, 0, 0, 0)$, also known as the star point. Model 4 represented a three-level response surface including three factors, and Model 5 was a non-linear form (i.e., a logistic model with three factors). For this last example, we determined a two-level factorial locally optimal design, where the vector of the parameters considered in the linearization was below the model representation. Section 4.1 reports unconstrained designs, and in Section 4.2 we consider constrained designs in the replicates domain, i.e., when additional constraints on the replicates number are included in the design. Finally, in Section 4.3, we extend the formulation to an I–optimality criterion. Herein, designs that have equal masses k/N at every design point are called *uniform designs* [4] (Chapter 4).

Table 1. Statistical models used for testing the MISDP formulations.

Model	Regression Function	Design Space (X)
1 [a]	$\beta_0 + \beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_1^2 + \beta_4\, x_2^2 + \beta_5\, x_1\, x_2$	$\{-1, 0, +1\}^2$
2 [b]	$\beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_3 + \beta_4\, x_4 + \beta_5\, x_1\, x_2 + \beta_6\, x_1\, x_3 + \beta_7\, x_1\, x_4 + \beta_8\, x_2\, x_3 + + \beta_9\, x_2\, x_4 + \beta_{10}\, x_3\, x_4$	$\{-1, +1\}^4$
3 [c]	$\beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_3 + \beta_4\, x_4 + \beta_5\, x_1\, x_2 + \beta_6\, x_1\, x_3 + \beta_7\, x_1\, x_4 + \beta_8\, x_2\, x_3 + + \beta_9\, x_2\, x_4 + \beta_{10}\, x_3\, x_4$	$\{-1, +1\}^4 \cup (0, 0, 0, 0)$
4 [d]	$\beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_3 + \beta_4\, x_1\, x_2 + \beta_5\, x_1\, x_3 + \beta_6\, x_2\, x_3$	$\{-1, 0, +1\}^3$
5 [e]	$[1 + \exp(\phi(\mathbf{x}))]^{-1}$ $\phi(\mathbf{x}) = \beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_3 + \beta_4\, x_1\, x_2 + \beta_5\, x_1\, x_3 + \beta_6\, x_2\, x_3$ $\beta_1 = 0.8,\ \beta_2 = 1.2, \beta_3 = -1.0, \beta_4 = 0.1, \beta_5 = -0.15, \beta_6 = -0.08$	$\{-1, +1\}^3$

[a]—Two-factor surface response model with three levels (see Atkinson [64]). [b]—Four-factor surface response model. [c]—Four-factor surface response model including the star treatment. [d]—Three-factor surface response model. [e]—Logistic model with three factors and all pairwise interactions.

4.1. Unconstrained Exact Optimal Designs

In this section, we determine optimal designs for all the models in Table 1, using the formulations (8) and (9) for the D- and A-optimality design criteria, respectively. For comparison, we report the approximate and the exact optimal designs for each model, and use different N values to analyze the effect of different sample sizes on the design.

Table 2 presents the optimal designs for Model 1. The design was represented by the complete set of candidate treatments (in the first two columns), followed by the respective weights if approximate designs were sought, and the replicates when exact designs were obtained. We observe that: (i) the use of approximate designs to construct exact designs by rounding $N \times w_i$ may lead to slightly sub-optimal experimental plans; and (ii) the exact designs obtained for the D-optimality criterion were equal to those obtained by Atkinson [64] with the KL algorithm [60] (Chapter 12), for $N = \{9, 13\}$. Similarly, the results for A-optimal designs obtained for these values of N reproduce those obtained with the KL algorithm: this finding demonstrates the accuracy of MISDP formulations.

For Models 2–5, the rules used to set N were: (i) $N = 2\ k$, where the number of experiments was a multiple of the candidate treatments; (ii) $N = k + 4$; (iii) $N = k + 7$, when that did not occur. Typically, the last two rules put additional challenges to solvers, as the resulting designs are not uniform, and the optimization problems may have multiple optima.

Table 2. D- and A-optimal designs for Model 1, where $k = 9$ and $N = \{9, 13, 17\}$.

				Exact Design					
		Approximate Design		$N = 9$		$N = 13$		$N = 17$	
Treatment		D-opt	A-opt	D-opt	A-opt	D-opt	A-opt	D-opt	A-opt
−1	−1	0.1458	0.0940	1	1	2	1	2	1
−1	0	0.0802	0.0978	1	1	1	2	1	2
−1	1	0.1458	0.0940	1	1	2	1	2	1
0	−1	0.0802	0.0978	1	1	1	2	2	2
0	0	0.0962	0.2332	1	1	1	3	2	3
0	1	0.0802	0.0978	1	1	1	1	2	2
1	−1	0.1458	0.0940	1	1	2	1	2	2
1	0	0.0802	0.0978	1	1	1	1	2	2
1	1	0.1458	0.0940	1	1	2	1	2	2

Table 3 lists the optimal designs obtained for Model 2. The approximate designs were uniformly distributed, as were the exact designs generated for N's multiple of k. Both designs required very small CPU times to generate (less than 1 s), and the formulation for the exact designs was, in this case, very efficient. Contrarily, when N was not a multiple of k, the CPU times increased to about 30 s, which was mainly due to difficulties in converging the upper and lower bounds of the MISDP problem, which, in turn, indicated that the branch-and-bound tree got larger. Specifically, it was harder to evolve the lower bound, which was an indication of the difficulty of proving the optimality of the relaxed problems or the occurrence of several optima with similar merit. In general, solving large-scale MISDP problems is still challenging, as the MISDP solvers are not yet as mature as mixed-integer linear programming solvers.

Table 3. D- and A-optimal designs for Model 2, where $k = 16$ and $N = \{20, 23, 32\}$.

						Exact Design					
				Approximate Design		$N = 20$		$N = 23$		$N = 32$	
Treatment				D-opt	A-opt	D-opt	A-opt	D-opt	A-opt	D-opt	A-opt
−1	−1	−1	−1	1/16	1/16	1	1	1	2	2	2
−1	−1	−1	1	1/16	1/16	1	2	2	1	2	2
−1	−1	1	−1	1/16	1/16	1	1	1	1	2	2
−1	−1	1	1	1/16	1/16	1	1	2	2	2	2
−1	1	−1	−1	1/16	1/16	1	1	2	1	2	2
−1	1	−1	1	1/16	1/16	2	2	1	2	2	2
−1	1	1	−1	1/16	1/16	2	1	2	1	2	2
−1	1	1	1	1/16	1/16	1	1	2	1	2	2
1	−1	−1	−1	1/16	1/16	1	2	1	1	2	2
1	−1	−1	1	1/16	1/16	2	1	1	2	2	2
1	−1	1	−1	1/16	1/16	2	1	2	1	2	2
1	−1	1	1	1/16	1/16	1	1	1	2	2	2
1	1	−1	−1	1/16	1/16	1	2	2	2	2	2
1	1	−1	1	1/16	1/16	1	1	1	2	2	2
1	1	1	−1	1/16	1/16	1	1	1	1	2	2
1	1	1	1	1/16	1/16	1	1	1	1	2	2

Table 4 presents the results for Model 3, and in both the approximate and exact designs, the candidate point $(0, 0, 0, 0)$ had null replication. Consequently, the optimal design for $N = 34$ had two treatments with replication 3, and no experiments at the star point.

Table 5 reports the optimal designs for Model 4. The approximate designs only required 8 of the 27 candidate points, and the same held for the exact designs, i.e., the optimal designs were saturated designs (characterized by having a number of support points equal to the number of parameters in the model). The points of the exact designs without replication were the same as those in the approximate design, having weights equal to 0.0, as expected.

Table 4. D- and A-optimal designs for Model 3, where $k = 17$ and $N = \{21, 24, 34\}$.

						Exact Design					
				Approximate Design		$N = 21$		$N = 24$		$N = 34$	
Treatment				D-opt	A-opt	D-opt	A-opt	D-opt	A-opt	D-opt	A-opt
−1	−1	−1	−1	1/16	1/16	1	1	1	2	2	2
−1	−1	−1	1	1/16	1/16	1	2	1	1	3	2
−1	−1	1	−1	1/16	1/16	1	2	2	1	2	2
−1	−1	1	1	1/16	1/16	2	1	2	1	2	2
−1	1	−1	−1	1/16	1/16	1	2	1	2	2	3
−1	1	−1	1	1/16	1/16	2	1	1	2	2	2
−1	1	1	−1	1/16	1/16	1	1	1	1	2	2
−1	1	1	1	1/16	1/16	1	1	1	2	2	2
1	−1	−1	−1	1/16	1/16	1	1	2	2	2	2
1	−1	−1	1	1/16	1/16	1	1	1	2	2	2
1	−1	1	−1	1/16	1/16	1	1	2	1	3	2
1	−1	1	1	1/16	1/16	2	2	2	1	2	2
1	1	−1	−1	1/16	1/16	2	1	2	1	2	2
1	1	−1	1	1/16	1/16	2	2	2	2	2	2
1	1	1	−1	1/16	1/16	1	1	1	1	2	2
1	1	1	1	1/16	1/16	1	1	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0

Table 5. D- and A-optimal designs for Model 4, where $k = 27$ and $N = \{31, 34, 54\}$.

					Exact Design					
			Approximate Design		$N = 31$		$N = 34$		$N = 54$	
Treatment			D-opt	A-opt	D-opt	A-opt	D-opt	A-opt	D-opt	A-opt
−1	−1	−1	1/8	1/8	4	3	4	4	7	7
−1	−1	0	0	0	0	0	0	0	0	0
−1	−1	1	1/8	1/8	4	4	5	4	7	7
−1	0	−1	0	0	0	0	0	0	0	0
−1	0	0	0	0	0	0	0	0	0	0
−1	0	1	0	0	0	0	0	0	0	0
−1	1	−1	1/8	1/8	4	4	4	5	7	7
−1	1	0	0	0	0	0	0	0	0	0
−1	1	1	1/8	1/8	4	4	4	5	7	7
0	−1	−1	0	0	0	0	0	0	0	0
0	−1	0	0	0	0	0	0	0	0	0
0	−1	1	0	0	0	0	0	0	0	0
0	0	−1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	1	−1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0
1	−1	−1	1/8	1/8	4	4	4	4	7	7
1	−1	0	0	0	0	0	0	0	0	0
1	−1	1	1/8	1/8	3	4	4	4	7	7
1	0	−1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
1	1	−1	1/8	1/8	4	4	4	4	7	7
1	1	0	0	0	0	0	0	0	0	0
1	1	1	1/8	1/8	4	4	5	4	7	7

Now, we address Model 5, where $\mathbb{E}[y|\mathbf{x}, \mathbf{p}] = f(\mathbf{x}, \mathbf{p})$, $f(\bullet)$ was the probability of responding to a stimulus, and $\mathbf{x}$ was the set of ordinal covariates that influenced the response. For demonstration, we considered that $f(\mathbf{x}, \mathbf{p})$ was the logistic model in the fifth line of Table 1.

The elemental FIM at the point $\mathbf{z}_j$ depended on the vector $\mathbf{p}$, and we circumvented this roundabout by determining locally optimal designs that were common when previous knowledge of the exact value of the parameters was available: thus, the elemental FIM at $\mathbf{z}_j$ is $M(\mathbf{z}_j|\mathbf{Z},\mathbf{p}) = h(\mathbf{z}_j|\mathbf{Z},\mathbf{p})\; h(\mathbf{z}_j|\mathbf{Z},\mathbf{p})^\intercal$ [60], where

$$h(\mathbf{z}_j|\mathbf{Z},\mathbf{p}) = \frac{1}{\sqrt{y(\mathbf{z}_j,\mathbf{p})\;(1-y(\mathbf{z}_j,\mathbf{p}))}}\left(\frac{\partial y(\mathbf{z}_j,\mathbf{p})}{\partial \mathbf{p}}\right), \quad \frac{\partial y(\mathbf{z}_j,\mathbf{p})}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial y(\mathbf{z}_j,\mathbf{p})}{\partial p_1} \\ \vdots \\ \frac{\partial y(\mathbf{z}_j,\mathbf{p})}{\partial p_{n_p}} \end{pmatrix},$$

and $p_j,\ j \in [\![n_p]\!]$ was the jth parameter in the model.

Table 6 contains the optimal designs for Model 5. Contrarily to the approximate D-optimal design, the A-optimal design is not uniform. The exact designs obtained were in good agreement with the trends observed for the approximate designs. Similarly to the approximate D-optimal designs, the exact D-optimal designs were minimally supported, as they had 6 support points and $n_p = 6$. This finding did not hold for the A-optimal designs.

Table 6. D- and A-optimal designs for Model 5 where $k = 8$ and $N = \{12, 15, 16\}$.

					Exact Design					
			Approximate Design		$N = 12$		$N = 15$		$N = 16$	
Treatment			**D-opt**	**A-opt**	**D-opt**	**A-opt**	**D-opt**	**A-opt**	**D-opt**	**A-opt**
−1	−1	−1	0.1667	0.1577	2	2	3	2	3	2
−1	−1	1	0.0000	0.1130	0	1	0	2	0	2
−1	1	−1	0.1667	0.1280	2	2	3	2	2	2
−1	1	1	0.1667	0.1427	2	2	3	2	3	2
1	−1	−1	0.1667	0.1141	2	1	2	2	2	2
1	−1	1	0.1667	0.1788	2	2	2	2	3	3
1	1	−1	0.0000	0.0465	0	0	0	1	0	1
1	1	1	0.1667	0.1192	2	2	2	2	3	2

4.2. Constrained Exact Optimal Designs

Here, we tested the algorithm, to find D- and A-optimal designs when there were linear constraints compactly represented by inequalities $A\mathbf{n}^\intercal \leq \mathbf{b}_{in}^\intercal$ or equalities $B\mathbf{n}^\intercal = \mathbf{b}_{eq}^\intercal$. We imposed constraints on the replicates, and searched in the (tightened) set of feasible designs $\Xi_N^e \equiv \Xi_N^e \cap \{(\mathbf{n}|\mathbf{Z}) :\ A\mathbf{n}^\intercal \leq \mathbf{b}_{in}^\intercal,\ B\mathbf{n}^\intercal = \mathbf{b}_{eq}^\intercal\}$. The constrained design problems could be solved using the formulations (8) and (9), complemented with the respective equality and inequality constraints.

To illustrate the application, we considered Model 3 (see Table 1). The model had 10 parameters, and the number of candidate treatments was 17, corresponding to 2^{n_p} factorial experiments plus the star treatment. First, we considered an equality constraint by which the replicates of the star treatment were exactly 2, i.e.,

$$n_k = 2. \tag{10}$$

Here, the matrix B containing the coefficients of the equality constraints was $1 \times k$, with all entries except the k^{th} being 0, and $B_{1,k} = 1.0$. Vector $\mathbf{b}_{eq}$ had only a single element, i.e., $b_{eq,1} = 2$.

Now, we consider constrained designs with constraints of the inequality kind. Let us consider that the experimental study has a limited budget, and that the cost of each treatment includes: (i) a fixed cost term; (ii) a variable cost term that is proportional to the level of the covariates. Furthermore, we assume that the total budget available for implementing the complete design is fixed at, for example, 150, that the fixed cost of each treatment is 1.8, and that the variable costs are linearly dependent on the levels of the covariates in the experiments, where the slopes are 0.5, 0.6, 0.8 and 1.0 for x_1, x_2, x_3 and x_4,

respectively. As an example, the cost of jth treatment is $1.8 + 0.5\,(z_{j,1} + 1) + 0.6\,(z_{j,2} + 1) + 0.8\,(z_{j,3} + 1) + 1.0\,(z_{j,4} + 1)$. Algebraic manipulations lead to a constraint of the kind:

$$A\mathbf{n}^{\intercal} \leq \mathbf{b}_{in} \tag{11}$$

where

$$A = \big(1.8,\ 3.8,\ 3.4,\ 5.4,\ 3.0,\ 5.0,\ 4.6,\ 6.6,\ 2.8,\ 4.8,\ 4.4,\ 6.4,\ 4.0,\ 6.0,\ 5.6,\ 7.6,\ 4.7\big).$$

Furthermore, let $b_{in,1} = 150$ for $N = 34$, and $b_{in,1} = 90$ for $N = 21$.

The results for both constrained designs generated by including Equations (10) and (11) in problems (8) and (9) are in Table 7. Now, the design obtained considering constraint (10) for $N = 34$ is uniform, and the point $(0, 0, 0, 0)$ has replication 2 as was imposed. The optimal designs obtained by including constraint (11) concentrate the replicates at treatments with lower cost. Similarly to unconstrained designs, no replication is required at the star treatment.

Table 7. Constrained D- and A-exact optimal designs for Model 3, where $k = 17$, $N = \{21, 34\}$ and constraints (10) and (11).

				Formulations (8) and (9) + Constraint (10)				**Formulations (8) and (9) + Constraint (11)**			
				$N = 21$		$N = 34$		$N = 21$		$N = 34$	
Treatment				**D-opt**	**A-opt**	**D-opt**	**A-opt**	**D-opt**	**A-opt**	**D-opt**	**A-opt**
−1	−1	−1	−1	1	1	2	2	2	2	4	4
−1	−1	−1	1	1	1	2	2	2	2	2	2
−1	−1	1	−1	1	1	2	2	2	2	2	3
−1	−1	1	1	1	1	2	2	1	1	2	2
−1	1	−1	−1	1	1	2	2	2	2	2	2
−1	1	−1	1	2	1	2	2	1	1	2	2
−1	1	1	−1	2	1	2	2	1	1	2	2
−1	1	1	1	1	2	2	2	1	1	2	2
1	−1	−1	−1	2	1	2	2	2	2	3	2
1	−1	−1	1	1	2	2	2	1	1	2	2
1	−1	1	−1	1	1	2	2	1	1	2	2
1	−1	1	1	1	1	2	2	1	1	2	2
1	1	−1	−1	1	1	2	2	1	1	2	2
1	1	−1	1	1	2	2	2	1	1	2	2
1	1	1	−1	1	1	2	2	1	1	2	2
1	1	1	1	1	1	2	2	1	1	1	1
0	0	0	0	2	2	2	2	0	0	0	0

One of the advantages of using mathematical-programming-based approaches to finding exact designs is that they can easily handle additional constraints and assure maximum efficiency in the feasibility domain of interest. As the rounding procedures are not able to incorporate constraints, its use in this context may lead to inefficient or even infeasible designs.

4.3. Exact I-Optimal Designs

In this section, we extend our framework to find I-optimal designs. Specifically, I-optimal designs minimize the average variance of prediction: consequently, they are more appropriate for a mixture experiments, as we may have, than the D- or A-optimality criteria, the focus of which is on a precise model estimation rather than precise predictions [65]. The I-optimality criterion is SDr, as it falls into A-optimality [36], and the formulation (9) can be easily adapted to compute I-optimal designs.

I-optimal designs are sought, to minimize the integral of the variance of the best linear unbiased estimator (BLUE) of the response over a design region $\mathbf{X} \subseteq \mathbb{R}^{n_x}$ of interest, with respect to a given discrete probability measure, $\mu(x)$. Let us consider that for each $x \in \mathbf{X}$

there is a variance matrix $V(x) \in \mathbb{S}_{+}^{n_p}$, such that $\mathrm{tr}\left[\mathcal{M}(\xi|\mathbf{Z})^{-1}\, V(x)\right]$ measures the total variance of the linear estimator at x, given the FIM for the set of parameters of interest [66]. The variance matrix becomes $V(x) = \mathbf{f}(\mathbf{x})\, \mathbf{f}^{\intercal}(\mathbf{x})$. $\mathbf{X}$ is the continuous design space enclosing the Cartesian box containing the complete set of treatments, and the probability measure $\mu(x)$ is uniformly distributed in $\mathbf{X}$: this leads to a generalized variance matrix (GVM):

$$\mathcal{V}(\mathbf{X}, \mu) = \int_{\mathbf{X}} V(x)\, \mathrm{d}(\mu(x)) = \int_{\mathbf{X}} \mathbf{f}(\mathbf{x})\, \mathbf{f}^{\intercal}(\mathbf{x})\, \mathrm{d}(\mu(x)),$$

which is required to be non-singular. Consequently, the design of the I–optimal experimental plans can be formulated as

$$\xi_N^e = \arg \min_{\xi \in \Xi_N^e} \Big\{ \mathrm{tr}[\mathcal{M}(\xi|\mathbf{Z})^{-1}\, \mathcal{V}(\mathbf{X}, \mu)] \Big\}. \tag{12}$$

In its turn, the MISDP problem for I-optimal designs is obtained from (9), replacing the objective function (9a) by

$$\min_{\mathbf{n}, \mathcal{F}} \ \mathrm{tr}[\mathcal{F}\, \mathcal{V}(\mathbf{X}, \mu)], \tag{13}$$

where $\mathcal{V}(\mathbf{X}, \mu)$ is computed a priori to minimize the model predictions.

Table 8 compares A- and I-optimal designs for Model 3 (in Table 1) obtained with the formulations (9), and (13), (9b)–(9d), respectively. The GVM is, in this case, diagonal, i.e., $\mathcal{V}(\mathbf{X}, \mu) = \mathrm{diag}(\mathbf{v})$, with $\mathbf{v} = (2/3,\ 2/3,\ 2/3,\ 2/3,\ 2/9,\ 2/9,\ 2/9,\ 2/9,\ 2/9,\ 2/9)^{\intercal}$. The approximate designs for A- and I-optimality are uniform, with the star point requiring no replicates. As expected, the A-optimal designs for both values of N are different from those obtained for the I-optimality criterion.

Table 8. A- and I-optimal designs for Model 3, where $k = 17$ and $N = \{21, 24, 34\}$.

						Exact Design					
				Approximate Design		$N = 21$		$N = 24$		$N = 34$	
Treatment				**A-opt**	**I-opt**	**A-opt**	**I-opt**	**A-opt**	**I-opt**	**A-opt**	**I-opt**
−1	−1	−1	−1	1/16	1/16	1	2	1	1	2	2
−1	−1	−1	1	1/16	1/16	2	1	1	2	2	2
−1	−1	1	−1	1/16	1/16	2	1	2	2	2	2
−1	−1	1	1	1/16	1/16	1	1	2	1	2	2
−1	1	−1	−1	1/16	1/16	2	1	2	1	3	2
−1	1	−1	1	1/16	1/16	1	1	1	1	2	3
−1	1	1	−1	1/16	1/16	1	2	1	2	2	2
−1	1	1	1	1/16	1/16	1	1	2	2	2	2
1	−1	−1	−1	1/16	1/16	1	1	2	2	2	2
1	−1	−1	1	1/16	1/16	1	2	2	1	2	2
1	−1	1	−1	1/16	1/16	1	2	1	2	2	3
1	−1	1	1	1/16	1/16	2	1	1	1	2	2
1	1	−1	−1	1/16	1/16	1	1	2	2	2	2
1	1	−1	1	1/16	1/16	2	1	1	1	2	2
1	1	1	−1	1/16	1/16	1	1	1	2	2	2
1	1	1	1	1/16	1/16	1	2	2	1	3	2
0	0	0	0	0	0	0	0	0	0	0	0

5. Conclusions and Discussion

This paper is the first to apply MISDP to find D-, A- and I-optimal factorial exact designs for linear models, and locally optimal designs for nonlinear models. We developed new MISDP formulations to systematically handle the experimental design problem. The formulations proposed enable the handling of: (i) different optimality criteria; (ii) different models and covariates with multiple levels; (iii) unconstrained and constrained setups. We solved the MISDP by employing a branch-and-bound algorithm designed to handle the

integer variables, coupled with an interior point-based solver that solved the relaxed SDP problems. We compared the exact designs obtained for various N's with the corresponding approximate designs obtained with SDP. As an alternative to discrete optimization-based techniques, exact designs can be obtained with rounding procedures, by using approximate optimal designs: however, this technique may lead to slightly inefficient exact designs, especially when the number of experiments of the plan is small. Practically, it can easily handle constrained designs, and in this context is advantageous over rounding, the application of which may lead to inefficient or infeasible solutions.

The numerical experiments reported in this paper show that uniform exact designs are easily computed, and they arise especially for the D-optimality criterion. Contrarily, for non-uniform designs, the CPU time is larger, and most of it is needed to assure the global optimality of the solution. On the other hand, it is common for these problems to have multiple optima, which also increases their numerical complexity. All the exact designs found are in good agreement with the equivalent approximate designs, which is a good indication that the inefficiency induced by using rounding procedures in unconstrained setups is small. Herein, we handled problems including various factors with a small number of levels, but the formulation proposed is general, so it can handle problems where the number of levels per factor is high, typical of other application areas.

Integer conic programming deserves increasing effort, as some problems can be conveniently formulated within this framework—one example being the optimal design of small-size experiments. Our paper is expected to contribute to the dissemination of this class of problems, and to providing an additional application that can urge the development of specific solvers able to handle it efficiently.

Funding: This research received no external funding.

Data Availability Statement: Data availability is not applicable to this article as no new data were created or analysed in this study.

Conflicts of Interest: The author declares no conflict of interest.

Acronyms

Acronym	Designation
FIM	Fisher Information Matrix
LMI	linear matrix inequality
MISDP	mixed-integer semidefinite programming
MINLP	mixed-integer nonlinear programming
NLP	nonlinear programming
ODoE	optimal design of experiments
SDP	semidefinite programming
SDr	semidefinite representable

References

1. Fedorov, V.V.; Leonov, S.L. *Optimal Design for Nonlinear Response Models*; Chapman and Hall/CRC Press: Boca Raton, FL, USA, 2014.
2. Kiefer, J.C. Optimum experimental designs. *J. R. Stat. Soc. Ser. B* **1959**, *21*, 272–319. [CrossRef]
3. Kiefer, J. General equivalence theory for optimum designs (approximate theory). *Ann. Stat.* **1974**, *2*, 849–879. [CrossRef]
4. Pukelsheim, F. *Optimal Design of Experiments*; SIAM: Philadelphia, PA, USA, 1993.
5. Vandenberghe, L.; Boyd, S. Applications of Semidefinite Programming. *Appl. Numer. Math.* **1999**, *29*, 283–299. [CrossRef]
6. Boer, E.P.J.; Hendrix, E.M.T. Global optimization problems in optimal design of experiments in regression models. *J. Glob. Optim.* **2000**, *18*, 385–398. [CrossRef]
7. Gribik, P.R.; Kortanek, K.O. Equivalence theorems and cutting plane algorithms for a class of experimental design problems. *SIAM J. Appl. Math.* **1977**, *32*, 232–259. [CrossRef]
8. Imhof, L.; López-Fidalgo, J.; Wong, W.K. Efficiencies of rounded optimal approximate designs for small samples. *Stat. Neerl.* **2001**, *55*, 301–318. [CrossRef]
9. Wynn, H.P. The sequential generation of D–optimum experimental designs. *Ann. Math. Statist.* **1970**, *41*, 1655–1664. [CrossRef]
10. Fedorov, V.V. *Theory of Optimal Experiments*; Academic Press: Cambridge, MA, USA, 1972.
11. Cook, R.; Nachtsheim, C. Comparison of algorithms for constructing D–optimal design. *Technometrics* **1980**, *22*, 315–324. [CrossRef]

12. Meyer, R.K.; Nachtsheim, C.J. The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics* **1995**, *37*, 60–69. [CrossRef]
13. Titterington, D.M. Algorithms for computing D–optimal design on finite design spaces. In Proceedings of the 1976 Conference on Information Science and Systems, Baltimore, MD, USA, 31 March–2 April 1976; pp. 213–216.
14. Dette, H.; Pepelyshev, A.; Zhigljavsky, A.A. Improving updating rules in multiplicative algorithms for computing D–optimal designs. *Comput. Stat. Data Anal.* **2008**, *53*, 312–320. [CrossRef]
15. Yu, Y. D–optimal designs via a cocktail algorithm. *Stat. Comput.* **2010**, *21*, 475–481. [CrossRef]
16. Yang, M.; Biedermann, S.; Tang, E. On optimal designs for nonlinear models: A general and efficient algorithm. *J. Am. Stat. Assoc.* **2013**, *108*, 1411–1420. [CrossRef]
17. Harman, R.; Jurík, T. Computing c–optimal experimental designs using the Simplex method of linear programming. *Comput. Stat. Data Anal.* **2008**, *53*, 247–254. [CrossRef]
18. Sagnol, G. Computing optimal designs of multiresponse experiments reduces to second-order cone programming. *J. Stat. Plan. Inference* **2011**, *141*, 1684–1708. [CrossRef]
19. Duarte, B.P.M.; Wong, W.K. Finding Bayesian optimal designs for nonlinear models: A semidefinite programming-based approach. *Int. Stat. Rev.* **2015**, *83*, 239–262. [CrossRef]
20. Duarte, B.P.M.; Wong, W.K.; Atkinson, A.C. A Semi-Infinite Programming based algorithm for determining T–optimum designs for model discrimination. *J. Multivar. Anal.* **2015**, *135*, 11–24. [CrossRef]
21. Molchanov, I.; Zuyev, S. Steepest descent algorithms in a space of measures. *Stat. Comput.* **2002**, *12*, 115–123. [CrossRef]
22. Pukelsheim, F.; Rieder, S. Efficient rounding of approximate designs. *Biometrika* **1992**, *79*, 763–770. [CrossRef]
23. de la Calle-Arroyo, C.; López-Fidalgo, J.; Rodríguez-Aragón, L.J. *Optedr: Calculating Optimal and D-Augmented Designs*; R Package Version 2.7.3; R Foundation for Statistical Computing: Vienna, Austria, 2022.
24. Mitchell, T.J.; Miller, F.L., Jr. *Use of Design Repair to Construct Designs for Special Linear Models*; Technical Report 130-131; Oak Ridge National Laboratory: Oak Ridge, TN, USA, 1970.
25. Mitchell, T.J. An algorithm for the construction of D–optimal experimental designs. *Technometrics* **1974**, *16*, 203–210. [CrossRef]
26. Atkinson, A.C.; Donev, A.N. The construction of exact D–optimum experimental designs with application to blocking response surface designs. *Biometrika* **1989**, *76*, 515–526. [CrossRef]
27. Goos, P.; Donev, A. Blocking response surface designs. *Comput. Stat. Data Anal.* **2006**, *51*, 1075–1088. [CrossRef]
28. Goos, P.; Vandebroek, M. D–optimal split-plot designs with given numbers and sizes of whole plots. *Technometrics* **2003**, *45*, 235–245. [CrossRef]
29. Donev, A.N. Crossover designs with correlated observations. *J. Biopharm. Stat.* **1998**, *8*, 249–262. [CrossRef] [PubMed]
30. Sartono, B.; Schoen, E.; Peter, G. Blocking orthogonal designs with Mixed Integer Linear Programming. *Technometrics* **2015**, *57*, 428–439. [CrossRef]
31. Sartono, B.; Goos, P.; Schoen, E. Constructing general orthogonal fractional factorial split-plot designs. *Technometrics* **2015**, *57*, 488–502. [CrossRef]
32. Ares, J.N.; Goos, P. An integer linear programming approach to find trend-robust run orders of experimental designs. *J. Qual. Technol.* **2019**, *51*, 37–50. [CrossRef]
33. Vo-Thanh, N.; Jans, R.; Schoen, E.D.; Goos, P. Symmetry breaking in mixed integer linear programming formulations for blocking two-level orthogonal experimental designs. *Comput. Oper. Res.* **2018**, *97*, 96–110. [CrossRef]
34. Welch, W.J. Branch-and-bound search for experimental designs based on D–optimality and other criteria. *Technometrics* **1982**, *24*, 41–48. [CrossRef]
35. Harman, R.; Filová, L. Computing efficient exact designs of experiments using integer quadratic programming. *Comput. Stat. Data Anal.* **2014**, *71*, 1159–1167. [CrossRef]
36. Sagnol, G.; Harman, R. Computing exact D–optimal designs by mixed integer second order cone programming. *Ann. Stat.* **2015**, *43*, 2198–2224. [CrossRef]
37. Duarte, B.P.M.; Sagnol, G. Approximate and exact D–optimal designs for 2^k factorial experiments for Generalized Linear Models via Second Order Cone Programming. *Stat. Pap.* **2020**, *61*, 2737–2767. [CrossRef]
38. Andersen, E.; Jensen, B.; Jensen, J.; Sandvik, R.; Worsøe, U. *MOSEK Version 6*; Technical Report TR–2009–3; MOSEK: Copenhagen, Denmark, 2009.
39. Sagnol, G.; Stahlberg, M. PICOS: A Python interface to conic optimization solvers. *J. Open Source Softw.* **2022**, *7*, 3915. [CrossRef]
40. Esteban-Bravo, M.; Leszkiewicz, A.; Vidal-Sanz, J.M. Exact optimal experimental designs with constraints. *Stat. Comput.* **2017**, *27*, 845–863. [CrossRef]
41. Duarte, B.P.; Granjo, J.F.; Wong, W.K. Optimal exact designs of experiments via Mixed Integer Nonlinear Programming. *Stat. Comput.* **2020**, *30*, 93–112. [CrossRef]
42. Pronzato, L.; Pázman, A. *Design of Experiments in Nonlinear Models*; Springer: New York, NY, USA, 2013.
43. Lasserre, J.B. Convex sets with semidefinite representation. *Math. Program.* **2009**, *120*, 457–477. [CrossRef]
44. Helton, J.W.; Vinnikov, V. Linear matrix inequality representation of sets. *Comm. Pure Appl. Math.* **2007**, *60*, 654–674. [CrossRef]
45. Ben-Tal, A.; Nemirovski, A.S. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2001.
46. Boyd, S.; Vandenberghe, L. *Convex Optimization*; University Press: Cambridge, UK, 2004.
47. Sagnol, G. On the semidefinite representation of real functions applied to symmetric matrices. *Linear Algebra Appl.* **2013**, *439*, 2829–2843. [CrossRef]

48. Gally, T.; Pfetsch, M.E.; Ulbrich, S. A framework for solving mixed-integer semidefinite programs. *Optim. Methods Softw.* **2018**, *33*, 594–632. [CrossRef]
49. Gally, T.; Gehb, C.M.; Kolvenbach, P.; Kuttich, A.; Pfetsch, M.E.; Ulbrich, S. Robust Truss Topology Design with Beam Elements via Mixed Integer Nonlinear Semidefinite Programming. *Appl. Mech. Mater.* **2015**, *807*, 229–238. [CrossRef]
50. Xygkis, T.C.; Korres, G.N. Optimized measurement allocation for power distribution systems using Mixed Integer SDP. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2967–2976. [CrossRef]
51. Schäfer, C. Optimization Approaches for Actuator and Sensor Placement and Its Application to Model Predictive Control of Dynamical Systems. Ph.D. Thesis, Fachbereich Mathematik, Technische Universität Darmstadt, Darmstadt, Germany, 2015.
52. Vandenberghe, L.; Boyd, S.; Wu, S.P. Determinant maximization with linear matrix inequality constraints. *SIAM J. Matrix Anal. Appl.* **1998**, *19*, 499–533. [CrossRef]
53. Alizadeh, F.; Goldfarb, D. Second-order cone programming. *Math. Program.* **2003**, *95*, 3–51. [CrossRef]
54. Grant, M.; Boyd, S.; Ye, Y. *cvx Users Guide for cvx, Version 1.22*; CVX Research, Inc.: Austin, TX, USA, 2012.
55. Fedorov, V.; Lee, J. Design of experiments in Statistics. In *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*; Springer: Dordrecht, The Netherlands, 2000; pp. 511–532. [CrossRef]
56. Löfberg, J. YALMIP : A Toolbox for Modeling and Optimization in MatLab. In Proceedings of the IEEE International Symposium on Computed Aided Control Systems Design, Taipei, Taiwan, 2–4 September 2004; pp. 284–289. [CrossRef]
57. Sturm, J. Using `SeDuMi` 1.02, a Matlab toolbox for optimization oversymmetric cones. *Optim. Methods Softw.* **1999**, *11*, 625–653. [CrossRef]
58. The Mathworks, Inc.. *Optimization Toolbox^{TM}—User's Guide*; The MathWorks, Inc.: Natick, MA, USA, 2018.
59. Cheng, C.S. *Theory of Factorial Design*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2016.
60. Atkinson, A.C.; Donev, A.N.; Tobias, R.D. *Optimum Experimental Designs, with SAS*; Oxford University Press: Oxford, UK, 2007.
61. Antony, J. *Design of Experiments for Engineers and Scientists*; Elsevier: Amsterdam, The Netherlands, 2014.
62. Goos, P.; Jones, B. *Optimal Design of Experiments: A Case Study Approach*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
63. Box, G. *Statistics for Experimenters: Design, Innovation, and Discovery, Second Edition + JMP Version 6 Software Set*; Wiley: Hoboken, NJ, USA, 2009.
64. Atkinson, A.C. Optimal Design. In *Wiley StatsRef: Statistics Reference Online*; Chapter 1; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2015; pp. 1–17. [CrossRef]
65. Goos, P.; Jones, B.; Syafitri, U. I–optimal design of mixture experiments. *J. Am. Stat. Assoc.* **2016**, *111*, 899–911. [CrossRef]
66. Filová, L.; Harman, R. Ascent with quadratic assistance for the construction of exact experimental designs. *Comput. Stat.* **2020**, *35*, 775–801. [CrossRef]