Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

GUILLAUME SAGNOL, RADOSLAV HARMAN

Computing exact D-optimal designs by mixed integer second order cone programming

ZIB-Report 13-39 (August 2013)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Computing exact D-optimal designs by mixed integer second order cone programming

Guillaume Sagnol, Radoslav Harman

August 6, 2013

Abstract

Let the design of an experiment be represented by an s-dimensional vector $\boldsymbol{w}$ of weights with non-negative components. Let the quality of $\boldsymbol{w}$ for the estimation of the parameters of the statistical model be measured by the criterion of D-optimality defined as the m-th root of the determinant of the information matrix $M(\boldsymbol{w}) = \sum_{i=1}^{s} w_i A_i A_i^T$, where A_i, $i = 1, ..., s$, are known matrices with m rows.

In the paper, we show that the criterion of D-optimality is second-order cone representable. As a result, the method of second order cone programming can be used to compute an approximate D-optimal design with any system of linear constraints on the vector of weights. More importantly, the proposed characterization allows us to compute an *exact* D-optimal design, which is possible thanks to high-quality branch-and-cut solvers specialized to solve mixed integer second order cone problems.

We prove that some other widely used criteria are also second order cone representable, for instance the criteria of A-, and G-optimality, as well as the criteria of D_K- and A_K-optimality, which are extensions of D-, and A-optimality used in the case when only a specific system of linear combinations of parameters is of interest.

We present several numerical examples demonstrating the efficiency and universality of the proposed method. We show that in many cases the mixed integer second order cone programming approach allows us to find a provably optimal exact design, while the standard heuristics systematically miss the optimum.

1 Introduction

Consider an optimal experimental design problem of the form

$$\max_{\boldsymbol{w} \in \mathcal{W}} \Phi \left(\sum_{i=1}^{s} w_i A_i A_i^T \right), \tag{1}$$

where Φ is a criterion mapping the space $\mathbb{S}_m^+$ of $m \times m$ positive semidefinite matrices over the set $\mathbb{R}_+ := [0, \infty)$. In (1), $A_i \in \mathbb{R}^{m \times l_i}$, $i = 1, ..., s$, are known matrices, and $\mathcal{W}$ is a compact subset of $\mathbb{R}_+^s$ representing the set of all permissible designs.

The problem (1) arises in linear regression models with a design space $\mathcal{X} \equiv [s] := \{1, ..., s\}$, independent trials, and a vector $\boldsymbol{\theta} \in \mathbb{R}^m$ of unknown parameters, provided that the trial in the i-th design point results in an l_i-dimensional response $\boldsymbol{y}_i$ satisfying $E(\boldsymbol{y}_i) = A_i^T \boldsymbol{\theta}$, and $\mathrm{Var}(\boldsymbol{y}_i) = \sigma^2 \boldsymbol{I}_{l_i}$, where $\boldsymbol{I_k}$ is the $k \times k-$identity matrix. In this case, the matrices $A_i A_i^T$ represent the information about the unknown parameters gained from a single trial in the i-th design point.

When the criterion Φ satisfies certain properties, Problem (1) can be interpreted as selecting the weights w_i that yield the *most accurate estimation* of $\boldsymbol{\theta}$. In this paper, we mainly focus on the $D-$optimal problem, where the criterion Φ is set to

$$\Phi_D : M \to (\det M)^{\frac{1}{m}}. \tag{2}$$

In the case of a Gaussian measurement error, this corresponds to the problem of minimizing the volume of the standard confidence ellipsoid for the best linear unbiased estimator (BLUE) $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$. More generally, if an the experimenter is interested in the estimation of the parameter subsystem $\boldsymbol{\vartheta} = K^T\boldsymbol{\theta}$, where K is a $m \times k$ matrix ($k \leq m$) of full column rank (rank $K = k$), a relevant criterion is D_K-optimality, which is defined by

$$\Phi_{D|K} : M \to (\det K^T M^- K)^{\frac{-1}{k}}. \tag{3}$$

Here M^- denotes a generalized inverse of M, i.e. a matrix satisfying $MM^-M = M$. Although the generalized inverse is not unique in general, the definition of $\Phi_{D|K}$ is consistent. Indeed the matrix $K^T M^- K$ does not depend on the choice of the generalized inverse M^- if the columns of K are included in the range of M, cf. Pukelsheim [Puk93]. The definition of $\Phi_{D|K}$ is extended to the set of all positive semidefinite matrices, by setting $\Phi_{D|K}(M) = 0$ whenever range $K \nsubseteq$ range M.

Other optimality criteria, such as A, A_K, G and $I-$optimality are also discussed in Section 5. For more details on the subject, we refer the reader to the monographs of Fedorov [Fed72], Pázman [Páz86] or Pukelsheim [Puk93].

In the standard form of the problem, $\mathcal{W}$ takes the form of the probability simplex

$$\mathcal{W}_\Delta := \{\boldsymbol{w} \in \mathbb{R}^s : \boldsymbol{w} \geq \boldsymbol{0}, \sum_{i=1}^{s} w_i = 1\},$$

and the design $\boldsymbol{w}$ is a weight vector indicating the percentage of trials in the individual design point. This problem, called *optimal approximate design problem* in the literature, is in fact the relaxation of a complicated combinatorial problem: the *optimal exact design problem of size* N, where $\mathcal{W}$ takes the form

$$\mathcal{W}_N := \{\frac{\boldsymbol{n}}{N} : \boldsymbol{n} \in \mathbb{N}_0^s : \sum_{i=1}^{s} n_i = N\}.$$

Here, the experiment consists of N trials, and $n_i = Nw_i$ indicates the number of trials in the design point i (in the above definition, $\mathbb{N}_0$ denotes the set of all nonnegative integers, i.e. $0 \in \mathbb{N}_0$). Note that the constraint $\boldsymbol{w} \in \mathcal{W}_\Delta$ is obtained from $\boldsymbol{w} \in \mathcal{W}_N$ by relaxing the integer constraints on Nw_i.

Many different approaches have been proposed to solve Problems of type (1). However, most methods are specialized and work only if the feasibility set $\mathcal{W}$ is the probability simplex $\mathcal{W}_\Delta$ or the standard discrete simplex $\mathcal{W}_N$. In the former case, the traditional methods are the Fedorov-Wynn type vertex-direction algorithms [Fed72, Wyn70], and the multiplicative algorithms [Tit76, STT78, Yu10]. In the latter case (exact optimal design, $\mathcal{W} = \mathcal{W}_N$), the classical methods are heuristics such as exchange algorithms [Fed72, Mit74, AD92], rounding methods [PR92], and metaheuristics like simulated annealing [Hai87] or genetic algorithms [HLCM$^+$03]. For some small to medium size models, branch-and-bound methods [Wel82] have been used to compute provably optimal solutions.

In many practical situations however, more complicated constraints are imposed on the design [CF95], and there is a need for more general algorithms. For example, assume that the experimental region can be partitioned as $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, and that no more than 40% (resp. 60%) of the trials can be chosen in $\mathcal{X}_1$ (resp.$\mathcal{X}_2$), i.e. the constraint $\boldsymbol{w} \in \mathcal{W}_\Delta$ is replaced by

$$\boldsymbol{w} \in \mathcal{W} := \{\boldsymbol{w} \in \mathbb{R}_+^s : \sum_{i \in \mathcal{X}_1} w_i \leq 0.4, \sum_{i \in \mathcal{X}_2} w_i \leq 0.6\}.$$

This is an example of *marginally constrained design*, as introduced by Cook and Thibodeau [CT80] (in general, marginally constrained designs rely on a partition $\mathcal{X} = \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_q$ of a two-dimensional design region $\mathcal{X}$, where each $\mathcal{X}_i$ is a *one-dimensional slice* of $\mathcal{X}$). Other examples of relevant design domains $\mathcal{W}$ defined by a set of linear inequalities are discussed in [VBW98]. For example, it is possible to consider the case where a total budget is allocated, and the design points are associated to possibly unequal costs $c_1, \dots, c_s$. It is also possible to consider decreasing costs when trials of specific design points are grouped, or to avoid designs that are concentrated on a small number of design points.

For some special linear constraints, the approximate D-optimal design problem can be solved by modifications of the vertex-direction algorithms and the multiplicative algorithms (see e.g. [CF95] and [MMTLF07]), but the convergence of these methods is usually slow. Recently, modern mathematical programming algorithms [VBW98, FL00, Har04, HJ08, Sag11, LP12, Pap12] have been gaining in popularity. The idea is to reformulate the optimal design problem under a canonical form that specialized solvers can handle, such as *maxdet programs* (MAXDET), *semidefinite programs* (SDP), or *second order cone programs* (SOCP). The great advantage of these methods is that using a mathematical programming reformulation of the problem, modern specialized solvers can be used to compute the approximate optimal designs, usually much more rapidly than the classical vertex exchange or multiplicative algorithms. Nevertheless, inclusion of general linear constraints to the mathematical programming characterizations is not completely straightforward. For instance, in a recent paper [Sag11], it has been proved that the D-optimal design problem can be solved by SOCP, but, as we show in Section 2, it is valid only for the classical approximate design problem, where $\boldsymbol{w}$ varies in the probability simplex $\mathcal{W}_\Delta$. In other words, the solution of the D-optimality SOCP of [Sag11], where the constraint $\boldsymbol{w} \in \mathcal{W}_\Delta$ is replaced by $\boldsymbol{w} \in \mathcal{W}$ for some arbitrary set $\mathcal{W}$, does not necessarily coincide with the design maximizing $\Phi_D\big(M(\boldsymbol{w})\big)$ over $\mathcal{W}$.

The main result of this paper is proved in Section 4 and states that the determinant criterion is *SOC-representable*. More precisely, it is possible to express that $(t, \boldsymbol{w})$ belongs to the hypograph of $\boldsymbol{w} \to \Phi_D\big(M(\boldsymbol{w})\big)$, i.e. $t^m \leq \det M(\boldsymbol{w})$, as a set of second order cone inequalities. (The necessary background related to the concept of SOC-representability will be introduced in Section 3). As a consequence, we obtain an alternative SOCP formulation for D-optimality, which remains valid for any weight domain $\mathcal{W}$. In particular, we can formulate a MISOCP to compute exact $D-$optimal designs. In Section 5 we prove that other widely used criteria, such as A, G, or $I-$optimality are also SOC-representable. The (MI)SOCPs of this paper are summarized in Table 1 (page 13).

Recently, much progress has been done in the development of solvers for second order cone programming, when some of the variables are constrained in the integral domain (MISOCP: Mixed Integer Second Order Cone Programming). Thus, the SOCP formulation of D-optimality presented in this article allows us to use those specialized codes to solve exact design problems. Compared to the raw branch-and-bound method to compute exact designs proposed by Welch [Wel82], the MISOCP approach is not only easier to implement, but it is also much more efficient. The reason behind is that specialized solvers such as CPLEX [CPL09] or MOSEK [AJJ+09] rely on branch-and-cut algorithms with sophisticated branching heuristics, and add *cut inequalities* during the solving process to separate non-integer solutions. Also, it may be necessary to point out that most solvers handling the former MAXDET formulation of D-optimality [VBW98] actually reformulate the problem using semidefinite programming, and there is currently no reliable solver to handle SDPs with integer variables.

We demonstrate the universality of the proposed approach in Section 6, with illustrative examples taken from many application areas of the theory of optimal experimental designs. The key aspects of the MISOCP approach will be emphasized:

1. The possibility to handle any system of linear constraints on the weights.
2. The possibility to compute exact-optimal designs with a proof of optimality.
3. For applications where the computing time must remain short, the MISOCP approach can find quickly a near exact-optimal design, and it gives a lower bound

on its efficiency (moreover this bound is usually much better than the standard bound obtained from the approximate optimal design).

In particular, our algorithm can compute constrained exact optimal designs, a feature out of reach of the standard computing methods, although some authors have proposed heuristics to handle some special cases such as cost constraints [TV04, WSB10]. A notable exception is the recent DQ-optimality approach of Harman and Filova [HF13], a heuristic based on Integer Quadratic Programming (IQP) able to handle the general case of linearly constrained exact designs. Our numerical results show that

(i) the SOCP approach is numerically more stable than the MAXDET programming approach for the case of approximate (i.e., continuous) optimal designs;

(ii) the MISOCP method can find a provably optimal design for many models where the KL-exchange algorithm [AD92] or the DQ-optimality IQP [HF13] misses the optimum;

(iii) the MISOCP approach finds exact optimal designs much faster than the raw branch-and-bound approach originally proposed by Welch [Wel82];

(iv) we can compute exact optimal designs with complicated constraints on the weights representing concrete restrictions that apply to design the experiment.

2 Former SOCP formulation of D-optimality

We first recall the result of [Sag11] about D-optimality, rewritten with the notation of the present article. Note that $\|M\|_F := \sqrt{\operatorname{trace} MM^T}$ denotes the Frobenius norm of the matrix M, which also corresponds to the Euclidean norm of the vectorization of M: $\|M\|_F = \|\operatorname{vec}(M)\|$.

Proposition 2.1 (Former SOCP for D-optimality [Sag11])**.** *Let $(Z_1, \ldots, Z_s, L, \boldsymbol{w})$ be optimal for the following SOCP:*

$$
\begin{aligned}
\max_{\substack{Z_i \in \mathbb{R}^{l_i \times m} \\ L \in \mathbb{R}^{m \times m} \\ \boldsymbol{w} \in \mathbb{R}^s_+}} \quad & \left(\prod_{k=1}^{m} L_{k,k} \right)^{\frac{1}{m}} \\
\text{s.t.} \quad & \sum_{i=1}^{s} A_i Z_i = L \\
& L \textit{ is lower triangular} \\
& \|Z_i\|_F \leq \sqrt{m}\, w_i \qquad \forall i \in [s], \\
& \boldsymbol{w} \in \mathcal{W}_\Delta.
\end{aligned}
\tag{4}
$$

Then $\Phi_D\big(M(\boldsymbol{w})\big) = \det^{1/m} M(\boldsymbol{w}) = \Big(\big(\prod L_{k,k}\big)^{1/m}\Big)^2$ and $\boldsymbol{w} \in \mathcal{W}_\Delta$ is optimal for the standard approximate $D-$optimal design problem.

If we want to solve a $D-$optimal design problem over another design region $\mathcal{W}$, it is very tempting to replace the last constraint in Problem (4) by $\boldsymbol{w} \in \mathcal{W}$. However, we show with a small example that this approach does not work. Consider for example the following experimental design problem with three regression vectors in a two-dimensional space: $A_1 = [1, 0]^T$, $A_2 = [-\frac{1}{2}, \frac{\sqrt{3}}{2}]^T$, $A_3 = [-\frac{1}{2}, -\frac{\sqrt{3}}{2}]^T$. For symmetry reasons it is clear that the approximate $D-$optimal design (over $\mathcal{W}_\Delta$) is $w_1 = w_2 = w_3 = \frac{1}{3}$, and this is the vector $\boldsymbol{w}$ returned by Problem (4) indeed. Define now $\mathcal{W} := \{\boldsymbol{w} \in \mathbb{R}^3_+ : \sum_{i=1}^3 w_i = 1,\ w_1 \geq w_2 + 0.25\}$. The optimal design over $\mathcal{W}$ is $\boldsymbol{w}^* =$ `[0.4583, 0.2083, 0.3333]`, but solving Problem (4) with the additional constraint $w_1 \geq w_2 + 0.25$ yields the design $\boldsymbol{w} =$ `[0.4482,0.1982,0.3536]`, which is suboptimal.

We point out that a similar behaviour occurs for the problem of $\boldsymbol{c}-$optimality, where the optimality criterion is $\Phi_{\boldsymbol{c}} : M \to 1/\boldsymbol{c}^T M^{-1} \boldsymbol{c}$. The SOCP for $\boldsymbol{c}-$optimality (Theorem 3.3 in [Sag11]), which has a geometric interpretation related to Elfving's theorem, is only valid on the standard simplex domain $\mathcal{W}_{\Delta}$. However, an alternative SOCP is provided in Theorem 4.3 of the same paper, in which an arbitrary polyhedral domain $\mathcal{W}$ can be used (see also the generalization to A_K-optimality in Section 5.1 and its MISOCP formulation in Table 1).

In this article we give an alternative SOCP formulation of the $D-$optimal problem, which remains valid for any compact design region $\mathcal{W}$. Moreover, our SOCP handles the more general case of D_K-optimality. To derive our result we use the notion of *SOC-representability*, which we next present.

3 SOC-representability

In this section, we briefly review some basic notions about second order cone representability. A *Second Order Cone Program* (SOCP) is an optimization problem where a linear function $\boldsymbol{f}^T\boldsymbol{x}$ must be maximized, among the vectors $\boldsymbol{x}$ belonging to a set S defined by second order cone Inequalities:

$$S = \{\boldsymbol{x} \in \mathbb{R}^n : \forall i = 1, \ldots, N_c,\ \|A_i \boldsymbol{x} + \boldsymbol{b_i}\| \leq \boldsymbol{c_i}^T \boldsymbol{x} + d_i\}.$$

We now recall the definition of a second order cone representable set, as introduced by Ben-Tal and Nemirovski [BTN87]:

Definition 3.1 (SOC representability)**.** A convex set $S \subseteq \mathbb{R}^n$ is said to be *second order cone representable*, abbreviated *SOC-representable*, if S is the projection of a set in a higher dimensional space which can be described by a set of second order cone inequalities. In other words, S is SOC-representable if and only if there exists $A_i \in \mathbb{R}^{n_i \times (n+m)}, \boldsymbol{b_i} \in \mathbb{R}^{n_i}, \boldsymbol{c_i} \in \mathbb{R}^{n+m}, d_i \in \mathbb{R}$ $(i = 1, \ldots, N_c)$, such that

$$\boldsymbol{x} \in S \Longleftrightarrow \exists \boldsymbol{y} \in \mathbb{R}^m : \forall i = 1, \ldots, N_c, \quad \left\| A_i \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} + \boldsymbol{b_i} \right\| \leq \boldsymbol{c_i}^T \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} + d_i.$$

Definition 3.2 (SOC representability of a function)**.** A convex (resp. concave) function $f : S \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ is said to be SOC-representable if and only if the epigraph of f, $\{(t, \boldsymbol{x}) : f(\boldsymbol{x}) \leq t\}$ (resp. the hypograph $\{(t, \boldsymbol{x}) : t \leq f(\boldsymbol{x})\}$), is SOC-representable.

It follows immediately from these two definitions that the problem of maximizing a concave SOC-representable function (or minimizing a convex one) over a SOC-representable set can be cast as a SOCP. It is also easy to verify that sets defined by linear equalities (i.e., polyhedrons) are SOC-representable, that intersections of SOC-representable sets are SOC-representable and that the (pointwise) maximum of convex SOC-representable functions is still convex and SOC-representable.

We next give two useful lemmas, which show that the geometric mean of m nonnegative variables is SOC-representable.

Lemma 3.3 (rotated second order cone inequalities)**.** *The set*

$$S = \{\boldsymbol{x} \in \mathbb{R}^n, t \in \mathbb{R}, u \in \mathbb{R} : \|\boldsymbol{x}\|^2 \leq tu,\ t \geq 0, u \geq 0\} \subseteq \mathbb{R}^{n+2}$$

is SOC-representable. In fact, it is easy to see that

$$S = \{\boldsymbol{x} \in \mathbb{R}^n, t \in \mathbb{R}, u \in \mathbb{R} : \left\| \begin{matrix} 2\boldsymbol{x} \\ t - u \end{matrix} \right\| \leq t + u\}.$$

Lemma 3.4 (SOC-representability of a geometric mean [BTN87])**.** *Let $n \geq 1$ be an integer. The function f mapping $\boldsymbol{x} \in \mathbb{R}_+^n$ to $\prod_{i=1}^n x_i^{1/n}$ is SOC-representable.*

For a construction of the SOC representation of f, see [LVBL98] or [AG03]. We give below an example in the case $n=5$: for all $\boldsymbol{x} \in \mathbb{R}^5_+$, we have:

$$
\begin{aligned}
t^5 \leq x_1x_2x_3x_4x_5 &\Longleftrightarrow t^8 \leq x_1x_2x_3x_4x_5t^3 \\
&\Longleftrightarrow \exists \boldsymbol{u} \in \mathbb{R}^5_+ : \left\{ \begin{array}{llll} u_1^2 & \leq x_1x_2, & u_4^2 & \leq u_1u_2, \\ u_2^2 & \leq x_3x_4, & u_5^2 & \leq u_3t, \\ u_3^2 & \leq x_5t, & t^2 & \leq u_4u_5, \end{array} \right.
\end{aligned}
$$

and each of these inequalities can be transformed to a standard second order cone inequality by Lemma 3.3.

4 SOC-representability of the D-criterion

We shall first concentrate on the case of $D-$optimality. The more general result, which concerns D_K-optimality (see (3)), requires an additional intermediate result which is proved in appendix. We start our proof by a lemma which gives a SOCP flavour to the Cholesky decomposition of a Grammian matrix:

Lemma 4.1. *Let H be a $m \times n$ matrix ($m \leq n$) of full rank ($\operatorname{rank} H = m$). Let $f_1, \ldots, f_m$ be increasing functions, mapping $(0, +\infty)$ onto $\mathbb{R}$. If (Q, L) is a solution of the optimization problem*

$$
\begin{aligned}
\max_{\substack{Q \in \mathbb{R}^{n \times m} \\ L \in \mathbb{R}^{m \times m}}} \quad & \sum_{k=1}^{m} f_k(L_{k,k}) \\
\text{s.t.} \quad & HQ = L \\
& L \textit{ is lower triangular} \\
& \|Q\boldsymbol{e_k}\| \leq 1 \qquad (k = 1, \ldots, m),
\end{aligned}
\tag{5}
$$

where $\boldsymbol{e_k}$ is the k^{th} unit vector of $\mathbb{R}^m$, then $H^T = QL^T$ is the QR decomposition of H^T and LL^T is the Cholesky factorization of HH^T.

Proof. We first note that Problem (5) is separable. Let $\boldsymbol{q_1}, \ldots, \boldsymbol{q_m}$ be the columns of the matrix Q, and $\boldsymbol{h_1}^T, \ldots, \boldsymbol{h_m}^T$ denote the rows of H. Problem (5) may be rewritten as a sum of independent problems:

$$
\begin{aligned}
\sum_{k=1}^{m} \quad \max_{\boldsymbol{q_k} \in \mathbb{R}^n} \quad & f_k(\boldsymbol{h_k}^T \boldsymbol{q_k}) \\
\text{s.t.} \quad & \boldsymbol{h_i}^T \boldsymbol{q_k} = 0, \qquad (i = 1, \ldots, k-1) \\
& \|\boldsymbol{q_k}\| \leq 1
\end{aligned}
\tag{6}
$$

For each $k \in [m]$, we denote by H_k the matrix formed by the rows $\boldsymbol{h_1}^T, \ldots, \boldsymbol{h_{k-1}}^T$ (and we set H_1 to be the $n-$dimensional row vector of all zeros), so that

$$\boldsymbol{q_k} \in \operatorname{Ker} H_k.$$

Define $\boldsymbol{u_k}$ and $\boldsymbol{v_k}$ as the orthogonal projections of $\boldsymbol{h_k}$ over $\operatorname{Ker} H_k$ and range H_k^T respectively, so that

$$\boldsymbol{h_k}^T \boldsymbol{q_k} = (\boldsymbol{u_k} + \boldsymbol{v_k})^T \boldsymbol{q_k} = \boldsymbol{u_k}^T \boldsymbol{q_k}.$$

Note that $\boldsymbol{u_k} \neq \boldsymbol{0}$, thanks to our assumption that the rows of H are linearly independent. Now, we have from Cauchy Schwarz that $\boldsymbol{u_k}^T \boldsymbol{q_k} \leq \|\boldsymbol{u_k}\|$, because $\boldsymbol{q_k}$ lies in the unit ball of $\mathbb{R}^n$, and the upper bound is uniquely attained for $\boldsymbol{q_k} = \|\boldsymbol{u_k}\|^{-1} \boldsymbol{u_k}$, such that $\boldsymbol{h_k}^T \boldsymbol{q_k} = \|\boldsymbol{u_k}\| > 0$.

Since f_k is increasing, this shows that the maximum of Problem (6) is uniquely attained at $\boldsymbol{q_k} = \|\boldsymbol{u_k}\|^{-1} \boldsymbol{u_k}$, where $\boldsymbol{u_k}$ is the orthogonal projection of $\boldsymbol{h_k}$ over $\operatorname{Ker} H_k$. Clearly, this solution process is nothing but the Gram-Schmidt orthogonalization of the matrix H^T (with columns $\boldsymbol{h_1}, \dots, \boldsymbol{h_m}$), which is well known to provide a QR decomposition of the original matrix.

Hence, if (Q, L) is the solution of Problem (5), then $Q^TQ = \boldsymbol{I_m}$ and there exists an upper triangular matrix R such that $H^T = QR$. Besides, we have $L^T = Q^TH^T = Q^TQR = R$, which shows that L^T coincides with the right factor of the QR decomposition, i.e., L is the Cholesky factor of HH^T. □

The determinant of a triangular matrix can be simply obtained by multiplying its diagonal elements. In the next proposition, we show that Grammian determinants can be computed by SOCP (up to a positive exponent). This proposition is generalized in appendix, where it is shown that expressions of the form $\det\big(K^T(HH^T)^-K\big)^{-1}$, where the matrix K has full column rank, can be computed as the optimal value of a SOCP (up to a positive exponent).

Proposition 4.2. *Let H be a $m \times n$ matrix ($m \leq n$), and let L be optimal for the following problem:*

$$\begin{aligned} \max_{\substack{Q \in \mathbb{R}^{n\times m}\\ L \in \mathbb{R}^{m \times m}}} \quad & \left(\prod_{k=1}^m L_{k,k} \right)^{1/m} \\ \text{s.t.} \quad & HQ = L \\ & L \textit{ is lower triangular} \\ & \|Q\boldsymbol{e_k}\| \leq 1 \qquad (k = 1, \dots, m) \end{aligned} \tag{7}$$

Then, we have:

$$\det HH^T = \left(\prod_{k=1}^m L_{k,k} \right)^2.$$

Proof. First note that the set of all feasible solutions of Problem (7) is nonempty (consider $Q = \boldsymbol{0}$, $L = \boldsymbol{0}$) and compact, so that this optimization problem has some solutions indeed. If H is of full rank, we apply the result of Lemma 4.1 with $f_k = \log$ for each $k \in [m]$. If L is optimal for Problem (7), then L is the Cholesky factor of HH^T, so that $\log\det HH^T = 2\log\det L = 2\log\prod_{k=1}^m L_{k,k}$.

For the case where H is rank deficient, there exists a row $\boldsymbol{h_j}^T$ of H that can be expressed as a linear combination of the previous rows: $\boldsymbol{h_j} = \sum_{k=1}^{j-1} \alpha_k \boldsymbol{h_k}$. Denote the j^{th} column of Q by $\boldsymbol{q_j}$, so that the constraints "$HQ = L$, L *lower triangular*" imply

$$L_{j,j} = \boldsymbol{h_j}^T\boldsymbol{q_j} = \sum_{k=1}^{j-1} \alpha_k \boldsymbol{h_k}^T \boldsymbol{q_j} = \sum_{k=1}^{m-1} \alpha_k \underbrace{L_{k,j}}_{=0} = 0.$$

Hence, for any feasible (Q, L), the value of the product in the objective function is 0, and the equality of the proposition holds ($\det HH^T = 0$). □

By Lemma 3.4, geometric means are SOC-representable. Hence Proposition 4.2 yields a SOCP to compute the determinant (up to a positive exponent) of any positive definite matrix M for which a decomposition of the form $M = HH^T$ is known. Now let $\boldsymbol{w} \in \mathcal{W}$ be a design. We could apply Proposition 4.2 to the matrix $H = [\sqrt{w_1}A_1, \dots, \sqrt{w_s}A_s]$ in order to express the determinant criterion $\Phi_D\big(M(\boldsymbol{w})\big)$ as the optimal value of a second order cone program. (Or more generally, apply Proposition A.1 of the appendix to express $\Phi_{D|K}\big(M(\boldsymbol{w})\big)$ as the optimal value of a SOCP.) This is what we do in the next proposition. Besides, we make a change of variables which transforms the optimization problem into a SOCP where $\boldsymbol{w}$ may play the role of a variable.

Theorem 4.3. *Let K be a $m \times k$ matrix ($k \leq m$) of full column rank. For all non-negative weight vectors $\boldsymbol{w} \in \mathbb{R}^s_+$, denote by $OPT(\boldsymbol{w})$ the optimal value of the following* *optimization problem, where the optimization variables are* $t_{ij} \in \mathbb{R}_+$ $(\forall i \in [s], \forall j \in [k])$, $Z_i \in \mathbb{R}^{l_i \times k}$ $(\forall i \in [s])$, *and* $J \in \mathbb{R}^{k \times k}$.

$$\max_{Z_i, t_{ij}, J} \quad \left(\prod_{j=1}^{k} J_{j,j} \right)^{1/k} \tag{8a}$$

$$\text{s.t.} \quad \sum_{i=1}^{s} A_i Z_i = KJ \tag{8b}$$

$$J \textit{ is lower triangular} \tag{8c}$$

$$\|Z_i \boldsymbol{e_j}\|^2 \leq t_{ij} w_i \ \ (i \in [s],\ j \in [k]) \tag{8d}$$

$$\sum_{i=1}^{s} t_{ij} \leq J_{j,j} \quad (j \in [k]) \tag{8e}$$

Then, we have

$$\Phi_{D|K}\ \big(M(\boldsymbol{w})\big) = \big(\det K^T M(\boldsymbol{w})^- K\big)^{-1/k} = OPT(\boldsymbol{w}).$$

Proof. Let $\boldsymbol{w}$ be an arbitrary nonnegative vector of $\mathbb{R}^s$, and define $H := [\sqrt{w_1} A_1, \ldots, \sqrt{w_s} A_s]$. We are going to show that every feasible solution of Problem (8) yields a feasible solution for Problem (12) in which $J_{j,j} = L_{j,j}^2$ for all $j \in [k]$, and vice versa. Hence the optimal value of Problem (8) is the square of the optimal value of Problem (12), from which the conclusion follows:

$$\begin{aligned} OPT(\boldsymbol{w}) = \left(\prod_{j=1}^{k} J_{j,j} \right)^{1/k} &= \left(\prod_{j=1}^{k} L_{j,j} \right)^{2/k} \\ &= \big(\det K^T (HH^T)^- K\big)^{-1/k} \\ &= \Big(\det K^T \big(\sum_i w_i A_i A_i^T\big)^- K\Big)^{-1/k} = \Phi_{D|K}\big(M(\boldsymbol{w})\big). \end{aligned}$$

Consider a feasible solution (Z_i, t_{ij}, J) of Problem (8). We denote by $\boldsymbol{z}_{ij}$ the the j^{th} column of Z_i: $\boldsymbol{z}_{ij} := Z_i \boldsymbol{e_j}$. We now make the following change of variable: denote by Q_i the matrix whose j^{th} column is $\boldsymbol{q_{ij}}$, where

$$\boldsymbol{q_{ij}} = \begin{cases} \frac{\boldsymbol{z_{ij}}}{\sqrt{w_i}\sqrt{J_{j,j}}} & \text{if } w_i > 0 \text{ and } J_{j,j} > 0; \\ \boldsymbol{0} & \text{otherwise,} \end{cases}$$

and define Q as the vertical concatenation of the Q_i: $Q = [Q_1^T, \ldots, Q_s^T]^T$. Let j be an index in $[k]$. We first handle the case $J_{j,j} = 0$, where every $\boldsymbol{q_{ij}} = \boldsymbol{0}$, so that $\|Q\boldsymbol{e_j}\|^2 = \sum_i \|\boldsymbol{q_{ij}}\|^2 = 0 \leq 1$. Otherwise ($J_{j,j} > 0$), Constraint (8d) together with the nonnegativity of t_{ij} implies $\|\boldsymbol{q_{ij}}\|^2 \leq \frac{t_{ij}}{J_{j,j}}$, and by Constraint (8e) we must have

$$\|Q\boldsymbol{e_j}\|^2 = \sum_i \|\boldsymbol{q_{ij}}\|^2 \leq \sum_i \frac{t_{ij}}{J_{j,j}} \leq 1.$$

Observe that constraints (8d)-(8e) also imply that $\boldsymbol{z_{ij}} = \boldsymbol{0}$ whenever $w_i = 0$ or $J_{j,j} = 0$, so that for all $i \in [s]$, $j \in [k]$, we can write $\boldsymbol{z_{ij}} = \sqrt{w_i}\sqrt{J_{j,j}}\boldsymbol{q_{ij}}$. Now, we define the matrix L columnwise as follows:

$$\forall j \in [k],\ L\boldsymbol{e_j} := \begin{cases} \frac{J\boldsymbol{e_j}}{\sqrt{J_{j,j}}} & \text{if } J_{j,j} > 0; \\ \boldsymbol{0} & \text{otherwise.} \end{cases}$$

We can now prove that $HQ = KL$, where H has been set to $[\sqrt{w_1} A_1, \ldots, \sqrt{w_s} A_s]$, which we do columnwise. If $J_{j,j} = 0$, then we know that $Q\boldsymbol{e_j} = \boldsymbol{0}$, so the jth columns of HQ and KL are zero. If $J_{j,j} > 0$, then we have

$$KL\boldsymbol{e_j} = \frac{KJ\boldsymbol{e_j}}{\sqrt{J_{j,j}}} = \frac{\sum_i A_i \boldsymbol{z_{ij}}}{\sqrt{J_{j,j}}} = \sum_i \sqrt{w_i} A_i \boldsymbol{q_{ij}} = HQ\boldsymbol{e_j}.$$

Hence, the proposed change of variables transforms a feasible solution (Z, t_{ij}, J) of Problem (8) into a feasible pair (Q, L) for Problem (12), with the property $J_{j,j} = L_{j,j}^2$ for all $j \in [k]$.

Conversely, let (Q, L) be feasible for Problem (12), where H has been set to $[\sqrt{w_1}A_1, \ldots, \sqrt{w_s}A_s]$. For $i \in [s]$, define Z_i as the matrix of size $l_i \times k$ whose j^{th} column is $\boldsymbol{z_{ij}} = \sqrt{w_i}L_{j,j}\boldsymbol{q_{ij}}$, and J as the matrix whose jth column is $J\boldsymbol{e_j} = L_{j,j}L\boldsymbol{e_j}$. We have $\sum_i A_i Z_i = KJ$, which can be verified columnwise:

$$KJ\boldsymbol{e_j} = L_{j,j}KL\boldsymbol{e_j} = L_{j,j}HQ\boldsymbol{e_j} = L_{j,j}\sum_i \sqrt{w_i}A_i\boldsymbol{q_{ij}} = \sum_i A_i\boldsymbol{z_{ij}} = \sum_i A_i Z_i \boldsymbol{e_j}.$$

Define further $t_{ij} = L_{j,j}^2\|\boldsymbol{q_{ij}}\|^2$, so that Constraints (8d) and (8e) hold. This shows that (Z_i, t_{ij}, L) is feasible, with $J_{j,j} = L_{j,j}^2$ for all $j \in [k]$, and the proof is complete. □

Corollary 4.4 (SOC-representability of Φ_D and $\Phi_{D|K}$). *The function $\boldsymbol{w} \to \Phi_D\big(M(\boldsymbol{w})\big)$ is SOC-representable. For any $m \times k$ matrix K of rank k, $\boldsymbol{w} \to \Phi_{D|K}\big(M(\boldsymbol{w})\big)$ is SOC-representable.*

Proof. Problem (8) can be reformulated as a SOCP, since by Lemmas 3.4 and 3.3 the geometric mean in (8a) is SOC-representable, as well as inequalities of the form (8d). Hence the optimal value of (8), $\boldsymbol{w} \to OPT(\boldsymbol{w})$, is SOC-representable, and we know from Theorem 4.3 that $OPT(\boldsymbol{w}) = \Phi_{D|K}\big(M(\boldsymbol{w})\big)$. In particular, the case $K = \boldsymbol{I_m}$ yields a SOC-representation of $\boldsymbol{w} \to \Phi_D\big(M(\boldsymbol{w})\big)$. □

Corollary 4.5 ((MI)SOCP formulation of the $D-$optimal design problem). *If the set $\mathcal{W}$ is SOC-representable (in particular, if $\mathcal{W}$ is defined by a set of linear inequalities), then the constrained D-optimal (or more generally D_K-optimal) design problem* (1) *can be cast as a SOCP. If the set $\mathcal{W}$ is the intersection of a SOC-representable set with the integer lattice $\mathbb{Z}^s$, then the D (or D_K-) optimal design problem can be cast as a MISOCP.*

The (MI)SOCP formulations of Problem (1) for D_K-optimality ($\Phi = \Phi_{D|K}$), as well as for the other criteria presented in next section, are summarized in Table 1.

5 Other optimality criteria

5.1 A_K-optimality

Another widely used criterion in optimal design is $A-$optimality, which is defined through

$$\Phi_A : \; M \to \big(\text{trace } M^{-1}\big)^{-1}.$$

More generally, it is possible to use the criterion of A_K-optimality if the experimenter is interested in the estimation of the parameter subsystem $K^T\boldsymbol{\theta}$:

$$\Phi_{A|K} : \; M \to \big(\text{trace}\, K^T M^- K\big)^{-1}.$$

Here M^- denotes a generalized inverse of M, see the discussion following Eq. (3) in the introduction. As for $\Phi_{D|K}$, the criterion $\Phi_{A|K}$ is defined over the set of matrices whose range does not include the columns of K:

$$\text{range } K \not\subseteq \text{range}\, M \Longrightarrow \Phi_{A|K}(M) = 0,$$

and for consistency we adopt the convention trace $K^T M^- K := +\infty$ whenever the range inclusion condition is not satisfied. Note that $\Phi_{A|K}$ coincides with Φ_A if $K = \boldsymbol{I_m}$ and $\Phi_{A|K}$ reduces to the criterion of $\boldsymbol{c}-$optimality when $K = \boldsymbol{c}$ is a column vector.

The function $\boldsymbol{w} \to \text{trace}\, K^T M(\boldsymbol{w})^- K$ was shown to be SOC-representable in [Sag11]. This fact has not been stated in those terms in the latter article, but can be obtained by following the first steps of the proof of Theorem 4.3 of [Sag11]. We give below a short proof of this result for completeness.

Proposition 5.1. *Let K be a $(m \times k)-$matrix, and let $\boldsymbol{w} \in \mathbb{R}_+^s$ be a vector of design weights. Then,*

$$\begin{aligned} \text{trace } K^T M(\boldsymbol{w})^- K = \min_{\boldsymbol{\mu} \in \mathbb{R}_+^s, Y_i \in \mathbb{R}^{l_i \times k}} \quad & \sum_{i \in [s]} \mu_i \\ \text{s. t.} \quad & \sum_i A_i Y_i = K \\ & \|Y_i\|_F^2 \leq w_i \mu_i. \end{aligned}$$

Proof. We first handle the case where the columns of K are included in the range of $M(\boldsymbol{w})$, so that trace $K^T M(\boldsymbol{w})^- K < \infty$. A well known consequence of the Gauss-Markov theorem is that the variance of the best linear unbiased estimator of $K^T \boldsymbol{\theta}$ is proportional to $K^T M(\boldsymbol{w})^- K$. More precisely, if $I \subseteq [s]$ denotes the subset of indices i such that $w_i > 0$, we have:

$$\begin{aligned} K^T M(\boldsymbol{w})^- K = \min_{(Y_i)_{i \in I}} {}_{\preceq} \quad & \sum_{i \in I} \frac{Y_i^T Y_i}{w_i} \\ \text{s. t.} \quad & \sum_{i \in I} A_i Y_i = K, \end{aligned}$$

where the variables Y_i $(i \in I)$ are of size $l_i \times k$, and the minimum is taken with respect to the Löwner ordering of $(k \times k)-$symmetric matrices (see e.g. Pukelsheim [Puk93]). The equality of the lemma is then simply obtained by taking the trace, and by introducing auxiliary variables $\boldsymbol{\mu} \in \mathbb{R}_+^s$ and $Y_i \in \mathbb{R}^{l_i \times k}$ for all $i \in [s] \setminus I$, that satisfy

$$\forall i \in [s], \begin{cases} \mu_i \geq \frac{\text{trace } Y_i^T Y_i}{w_i} = \frac{1}{w_i} \|Y_i\|_F^2 & \text{if } w_i > 0; \\ \mu_i = 0 & \text{otherwise,} \end{cases}$$

and $Y_i = \mathbf{0} \in \mathbb{R}^{l_i \times k}$ for $i \notin I$.

Assume now that a column of K does not lie in the range of the singular information matrix $M(\boldsymbol{w})$, which is also the range of the matrix $[A_{i_1}, \ldots, A_{i_q}]$, where $I = \{i \in [s] : w_i > 0\} = \{i_1, i_2, \ldots, i_q\}$. Then, the equation $\sum_{i \in I} A_i Y_i = K$ has no solution $(Y_{i_1}, \ldots, Y_{i_q})$, and so the SOCP of the proposition has no feasible solution, which implies that his optimal value is $+\infty$. ☐

Corollary 5.2. *Let K be a $m \times k$ matrix. The convex function $f : \boldsymbol{w} \to$ trace $K^T M(\boldsymbol{w})^- K$, which maps $\mathbb{R}_+^s$ onto $\mathbb{R} \cup \{+\infty\}$, is SOC-representable.*

The reformulation of Problem (1) for the criterion $\Phi = \Phi_{A|K}$ as a (MI)SOCP is indicated in Table 1.

Remark 5.3 (The case of $\boldsymbol{c}-$optimality). The case of $\boldsymbol{c}-$optimality arises as a special case of both A_K and D_K-optimality when the matrix $K = \boldsymbol{c} \neq \mathbf{0}$ is a column vector $(k = 1)$. In this situation, the reader can verify that the two SOCP formulations (for $\Phi_{A|\boldsymbol{c}}$ and $\Phi_{D|\boldsymbol{c}}-$ in Table 1) are equivalent, which can be verified by the change of variables: $Y_i = J_{1,1}^{-1} Z_i$, $\mu_i = J_{1,1}^{-2} t_{i1}$ (note that here the matrix J is of size 1, i.e. a scalar).

We next show how Proposition 5.1 can be used to obtain a SOC-representation of several other criteria, namely for G and $I-$optimality.

5.2 G-optimality

A criterion closely related to $D-$optimality is the criterion of $G-$optimality,

$$\Phi_G : M \to -\max_{i \in [s]} \text{ trace } A_i^T M^- A_i.$$

In the common case of single-response experiments for linear models, the matrices A_i are column vectors, and the scalar $A_i^T M(\boldsymbol{w})^- A_i$ represents the variance of the prediction $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = A_i^T \hat{\boldsymbol{\theta}}$. Hence $G-$optimality seeks at minimizing the maximum variance of the predicted values $\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{s}}$.

The G and $D-$optimality criteria are related to each other by the celebrated equivalence theorem of Kiefer and Wolfowitz [KW60], which has been generalized to the case of multivariate regression ($l_i > 1$) by Fedorov in 1972. We give below a version of this theorem for the case of a finite design space $\mathcal{X} \equiv [s]$:

Theorem 5.4 (Equivalence Theorem [Fed72]). *Assume that the matrix* $\mathcal{A} = [A_1, \ldots, A_s] \in \mathbb{R}^{m \times l}$ *contains* m *independent vectors among its columns. Then the following statements are equivalent:*

(i) *The design* $\boldsymbol{w}$ *maximizes* $\Phi_D\big(M(\boldsymbol{w})\big)$ *over* $\mathcal{W}_\Delta$.

(ii) *The design* $\boldsymbol{w}$ *maximizes* $\Phi_G\big(M(\boldsymbol{w})\big)$ *over* $\mathcal{W}_\Delta$.

(iii) *For all* $i \in [s]$, $\operatorname{trace} A_i^T M(\boldsymbol{w})^- A_i \leq m$.

Moreover, if the design $\boldsymbol{w}^* \in \mathcal{W}_\Delta$ *is* $D-$*optimal, then the bound of* (iii) *is attained at the support points of* $\boldsymbol{w}^*$*:*

$$w_i^* > 0 \Longrightarrow \operatorname{trace} A_i^T M(\boldsymbol{w}^*)^- A_i = m.$$

In other words, the concepts of $D-$optimality and $G-$optimality coincide when the weight domain $\mathcal{W}$ is the probability simplex $\mathcal{W}_\Delta$. However, exact $G-$optimal designs do not necessarily coincide with their $D-$optimal counterpart. In a recent article [RJBM10], the Brent's minimization algorithm has been proposed to compute near exact G-optimal factorial designs. But in general, we do not know any standard algorithm for the computation of $G-$optimal designs over arbitrary weight domains $\mathcal{W}$ that are defined by a set of linear inequalities.

We know from Corollary 5.2 that the convex functions $f_i : \boldsymbol{w} \to \operatorname{trace} A_i^T M(\boldsymbol{w})^- A_i$ are SOC-representable, and hence their maximum is also convex and SOC-representable. A (MI)SOCP formulation of Problem (1) for the criterion $\Phi = \Phi_G$ is indicated in Table 1. For the case where the weight domain $\mathcal{W}$ is the probability simplex $\mathcal{W}_\Delta$, it gives a new alternative SOCP formulation for $D-$optimality. Note however that in this situation, the SOCP formulation (4) for $D-$optimality from [Sag11] is usually more compact (i.e., it involves less variables and less constraints) than the $G-$optimality SOCP of Table 1.

5.3 I-optimality

Another widely used criterion is the one of $I-$ optimality (or $V-$optimality). Here, the criterion is the average of the variances of the predicted values $\hat{\boldsymbol{y}}_{\boldsymbol{1}}, \ldots, \hat{\boldsymbol{y}}_{\boldsymbol{s}}$:

$$\Phi_I : M \to -\frac{1}{s} \sum_{i \in [s]} \operatorname{trace} A_i^T M^- A_i.$$

In fact, this criterion coincides with the $\Phi_{A|K}$ criterion, by setting K to any matrix satisfying $KK^T = \frac{1}{s}\sum_{i=1}^s A_i A_i^T$ (see e.g. §9.8 in [Puk93]). Hence Φ_I-optimal designs can be computed by SOCP. Note that there is also a weighted version of $I-$optimality, which can be reduced to an A_K-optimal design problem in the same manner.

6 Experimental Results

In this section we will present numerical results for several examples taken from various application areas of the theory of optimal designs. With these examples we aim to demonstrate the universality of the (MI)SOCP technique for the computation of exact or approximate $D-$ (and $A-$) optimal designs.

Our computations were worked out on a PC with 4 cores at 3GHz. We have used MOSEK [AJJ+09] to solve the approximate optimal design problems, and CPLEX [CPL09] for the exact optimal design problems (with integer constraints). The solvers were interfaced through the Python package PICOS [Sag12], which allows the users to pass (MI)SOCP models such as those of Table 1 to different solvers in a painless fashion. We refer the reader to the example section of the PICOS documentation for a practical implementation of the (MI)SOCPs for optimal design problems.

It is common to compare several designs against each other by using the metric of $D-$efficiency. The latter is defined as

$$\mathrm{eff}_D(\boldsymbol{w}) = \frac{\Phi_D\big(M(\boldsymbol{w})\big)}{\Phi_D\big(M(\boldsymbol{w}^*)\big)} = \left(\frac{\det M(\boldsymbol{w})}{\det M(\boldsymbol{w}^*)}\right)^{1/m},$$

where $\boldsymbol{w}^*$ is a reference design. Unless stated otherwise, we always give $D-$efficiencies relatively to the optimal design, i.e. $\boldsymbol{w}^*$ is a solution of Problem (1).

Constrained designs for the quadratic model with two factors Consider the quadratic regression model on a (18×3)-grid in the plane:

$$\begin{aligned} y(\boldsymbol{x}) =& \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_1 x_2 + \epsilon(\boldsymbol{x}) \\ &\boldsymbol{x} = [x_1, x_2]^T \in \mathcal{X} = \{94.9, 95.1, 95.2, \ldots, 96.6, 96.7\} \times \{0, 10, 20\}. \end{aligned} \tag{9}$$

This model was used in [MMTLF07] for the sintering of uranium pellets, and it served as an example for constrained DQ-optimality in [HF13]. The explanatory variables represent the "initial density" (x_1) and the "percentage of additive U_3O_8" (x_2). The nature of the experiment requires marginal constraints on the variable x_1. More precisely, the numbers of trials under the 18 levels of the variable x_1 are restricted to 1, 3, 14, 59, 52, 29, 25, 32, 36, 29, 36, 38, 12, 10, 8, 2, 3, 3, which amounts to the total of $N = 392$ trials. If we denote the levels of the factor x_1 by $L_1, \ldots, L_{18}$, and the required marginal sums by $a_1, \ldots, a_{18}$, the constraints have the form

$$\boldsymbol{w} \in \mathcal{W} := \{\boldsymbol{w} \in \mathbb{N}_0^{\mathcal{X}} : \forall j \in 1, \ldots, 18, \sum_{x_2 \in \{0,10,20\}} w(L_j, x_2) = a_j\}.$$

We first say a word about the computation of an approximate D-optimal design for Model (9), i.e. when the integer constraint on $w(x_1, x_2)$ is relaxed. Martín-Martín et. al. [MMTLF07] have adapted the multiplicative algorithm to compute marginally constrained approximate optimal designs. Their algorithm finds an optimal design in 4.8 seconds (it is written 0.08 minute in the article). In comparison, MOSEK solved the SOCP of Table 1 in 0.04 second. The SOCP approach has also been compared to the widespread MAXDET programming approach [VBW98]. In a first attempt, we have tried to solve the MAXDET SDP with SeDuMi [Stu99] interfaced through YALMIP [Lö04], by using the natural observation matrices $A(x_1, x_2) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2]^T$. The solver ran into numerical problems and was not able to find a solution. However, it is well known that the $D-$optimal design problem is invariant to linear transformations of the parameter, hence we can shift the regression domain $\mathcal{X}$ to the regular 18×3 grid $\mathcal{X}'$ over $[-1, 1] \times [-1, 1]$. This has the effect to scale the $D-$criterion Φ_D by a constant multiplicative factor, so the $D-$optimal design remains the same. The transformed problem (over $\mathcal{X}'$) has better numerical properties, and SeDuMi was able to find a solution in 0.3s. A similar behaviour was observed with the CVXOPT solver [DV06], interfaced through PICOS: CVXOPT failed to solve the original problem over $\mathcal{X}$, but found a solution of the transformed problem after 0.14s. To our mind, this example shows that the SOCP approach is not only faster than the MAXDET approach for the computation of approximate optimal designs, but also numerically more stable.

Concerning the exact design problem, we could solve the MISOCP of Table 1 in 1.41s. In [HF13] another illustrative example with an additional cost constraints is considered. Here it is assumed that 1% of the additive costs one unit, and a total budget of 1965 price units is allowed:

$$
\begin{aligned}
\max_{\boldsymbol{w}\in\mathcal{W}} \ \Phi_{D|K}\big(M(\boldsymbol{w})\big) \ &= \max_{\boldsymbol{w},Z_i,t_{ij},J} \ \prod_{j=1}^{k}(J_{j,j})^{\frac{1}{k}} \\
&\qquad \text{s.t.} \ \sum_{i\in[s]} A_i Z_i = KJ \\
&\qquad\qquad J \textit{ is lower triangular} \\
&\qquad\qquad \|Z_i \boldsymbol{e_j}\|^2 \leq t_{ij} w_i \ (i\in[s], \ j\in[k]) \\
&\qquad\qquad \sum_{i=1}^{s} t_{ij} \leq J_{j,j} \quad (j\in[k]) \\
&\qquad\qquad t_{ij} \geq 0 \ \ (i\in[s], \ j\in[k]) \\
&\qquad\qquad \boldsymbol{w}\in\mathcal{W}
\end{aligned}
$$

$$
\begin{aligned}
\left(\max_{\boldsymbol{w}\in\mathcal{W}} \ \Phi_{A|K}\big(M(\boldsymbol{w})\big)\right)^{-1} \ &= \min_{\boldsymbol{w},Y_i,\mu_i} \ \sum_{i\in[s]} \mu_i \\
&\qquad \text{s.t.} \ \sum_{i\in[s]} A_i Y_i = K \\
&\qquad\qquad \|Y_i\|_F^2 \leq \mu_i w_i \ (i\in[s]) \\
&\qquad\qquad \mu_i \geq 0 \ (i\in[s]) \\
&\qquad\qquad \boldsymbol{w}\in\mathcal{W}
\end{aligned}
$$

$$
\begin{aligned}
-\left(\max_{\boldsymbol{w}\in\mathcal{W}} \ \Phi_{G}\big(M(\boldsymbol{w})\big)\right) \ &= \min_{\boldsymbol{w},H_i^j,u_i^j,\rho} \ \rho \\
&\qquad \text{s.t.} \ \sum_{j\in[s]} A_j H_i^j = A_i \ (i\in[s]) \\
&\qquad\qquad \|H_i^j\|_F^2 \leq w_j u_i^j \ (i\in[s], j\in[s]) \\
&\qquad\qquad u_i^j \geq 0 \ (i\in[s], j\in[s]) \\
&\qquad\qquad \sum_{j\in[s]} u_i^j \leq \rho \ (i\in[s]) \\
&\qquad\qquad \boldsymbol{w}\in\mathcal{W}.
\end{aligned}
$$

Table 1: SOCP formulations of the A_K,D_K,and $G-$ optimal design problems over an arbitrary weight region $\mathcal{W}$. In the above, K represents a given $m\times k$ matrix of full column rank. The particular case $k=1$ (where $\boldsymbol{c}=K$ is a column vector) gives SOCP formulations for the $\boldsymbol{c}-$optimal design problem, and the case $K=\boldsymbol{I_m}$ yields the standard A and $D-$optimality problems. In these SOCPs, the variables Z_i and Y_i $(i\in[s])$ are of size $l_i\times k$, the variables H_i^j $(i\in[s], j\in[s])$ are of size $l_j\times l_i$, J is of size $k\times k$, the weight vector is $\boldsymbol{w}\in\mathcal{W}\subseteq\mathbb{R}_+^s$, and the variables t_{ij} $(i\in[s], j\in[k])$, u_i^j $(i\in[s], j\in[s])$, μ_i $(i\in[s])$ and ρ are scalar.

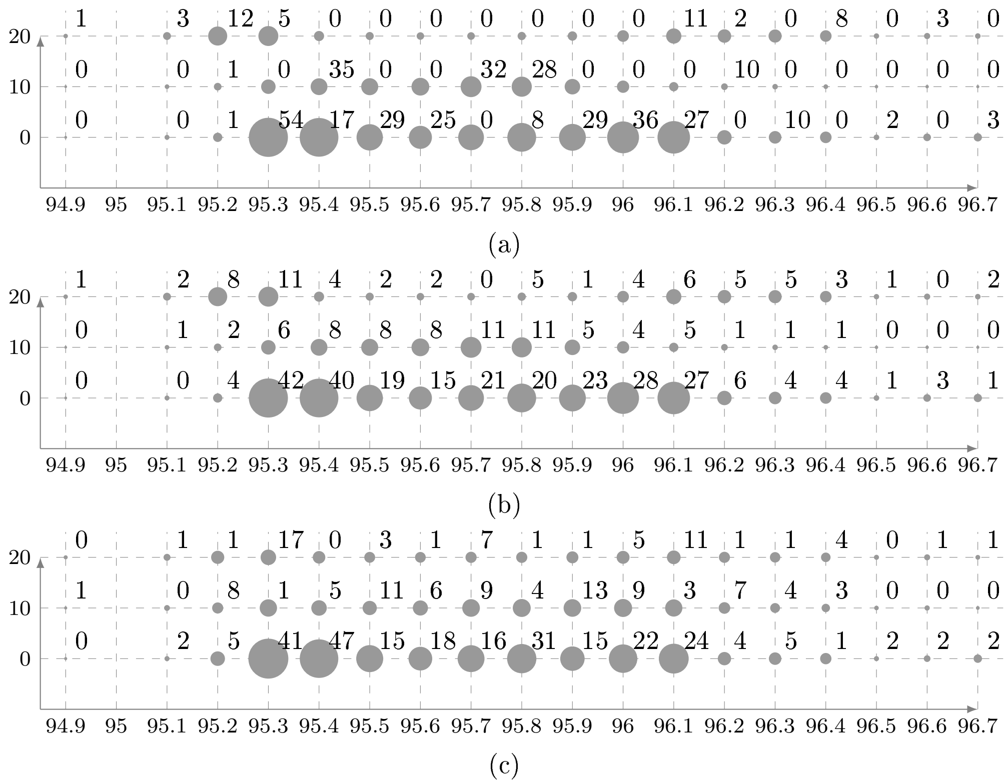


Figure 1: Exact and approximate optimal designs for the design problem with both marginal and cost constraints (cf. Eq. (10)). The areas of the gray discs are proportional to the weights of the continuous optimum (approximate $D-$optimal design in (a) and (b), approximate $A-$optimal design in (c)), while the integers near the discs indicate the exact designs, i.e. the number of times that each individual trial should be performed: (a) DQ-optimal design $\boldsymbol{w}_{DQ}$ computed in [HF13] ($\mathrm{eff}_D(\boldsymbol{w}_{DQ}) = 96.75\%$); (b) $D-$optimal design $\boldsymbol{w}_D$ computed by MISOCP ($\mathrm{eff}_D(\boldsymbol{w}_D) = 100\%$); (c) $A-$optimal design $\boldsymbol{w}_A$ computed by MISOCP ($\mathrm{eff}_D(\boldsymbol{w}_A) = 98.24\%$).

$$\boldsymbol{w} \in \mathcal{W} := \left\{ \boldsymbol{w} \in \mathbb{N}_0^{\mathcal{X}} : \begin{array}{c} \forall j \in \{1, \ldots, 18\}, \sum_{x_2 \in \{0,10,20\}} w(L_j, x_2) = a_j, \\ \sum_{j=1}^{18} 10\ w(L_j, 10) + 20\ w(L_j, 20) \leq 1965. \end{array} \right\} \tag{10}$$

The authors of [HF13] have used the DQ-optimality method to compute an efficient design. Their design $\boldsymbol{w}_{DQ}$ is plotted on Figure 1(a) and has a $D-$efficiency of 96.68% relatively to the continuous optimum. We were able to compute an exact optimal design by using the MISOCP approach (Figure 1(b)). This took 1.55s with CPLEX. The exact optimal design achieves almost the same criterion value as the continuous optimum, and hence the $D-$efficiency of the DQ-optimal design relatively to the *true* optimum is just a bit higher than 96.68%: $\mathrm{eff}_D(\boldsymbol{w}_{DQ}) = 96.75\%$. To illustrate the universality of the MISOCP method, we have also computed an exact $A-$optimal design for this problem. The solution is depicted on Figure 1(c) and was found after 10.84s with CPLEX. We point out that the exact $A-$optimal design has a $D-$efficiency of 98.24%.

Two-block designs An important category of models studied in the experimental design literature is the class of *block designs*. Here the effect of t treatments should be compared, but their effects can only be measured inside a number b of *blocks*, each of whom induces a block effect on the measurements. The optimal design problem consists in choosing which treatments should be tested together in each block. We refer the reader to Bailey and Cameron [BC09] for a comprehensive review on the combinatorics of block designs.

In the case where the blocks are of size two, i.e. the treatments can be tested pairwise against each other, a design can be represented by a vector $\boldsymbol{w} = [w_{1,2}, w_{1,3}, \ldots, w_{1,t}, \ldots, w_{t-1,t}]$ of size $s = \binom{t}{2}$. For $i < j$, $w_{i,j}$ indicates the number of blocks where treatments i and j are tested simultaneously. The observation matrix associated with the block (i, j) is the column vector of dimension $m = (t-1)$:

$$A_{i,j} = P(\boldsymbol{e_i} - \boldsymbol{e_j}),$$

where $\boldsymbol{e_i}$ denotes the i^{th} unit vector in the canonical basis of $\mathbb{R}^t$ and P is the matrix of projection that transforms a $t-$dimensional vector $\boldsymbol{v}$ to the vector obtained by keeping the first $(t-1)$ coordinates of $\boldsymbol{v}$.

The problem of D-optimality has a nice graph theoretic interpretation: let $G(\boldsymbol{w})$ be the graph with t vertices and an edge of multiplicity $w_{i,j}$ for every pair of nodes (i, j). (If $w_{i,j} = 0$, then there is no edge from i to j). This graph is called the *concurrence graph* of the design. It is not difficult to see that $M(\boldsymbol{w})$ is the submatrix of the Laplacian of $G(\boldsymbol{w})$, obtained by removing its last row and last column. So by Kirchhoff's theorem the cofactor $\det M(\boldsymbol{w})$ coincides with the number of spanning trees of $G(\boldsymbol{w})$. In other words, the exact $D-$optimal designs of size N coincide with the graphs with t nodes and N vertices that have a maximum number of spanning trees.

We have computed some 2-block designs for different values of t and N, with three different algorithms: The MISOCP approach proposed in this paper, the KL-exchange algorithm of Atkinson and Donev [AD92], and the DQ-optimality integer quadratic program of Harman and Filova [HF13]. Results are reported in Table 2. For every algorithm we have indicated the number of spanning trees in the concurrence graph of the obtained design, as well as its $D-$efficiency. Note that the D-efficiency is often very high, even when the number of spanning trees is far from the optimum, because of the exponent $\frac{1}{m}$ in the criterion Φ_D (2) which shifts the ratio toward 1. We next explain the settings used by each of these algorithms for the computation of the results in Table 2.

We used CPLEX to solve the MISOCPs. To achieve a faster convergence, linear equalities were added in the SOCP formulation to restrain the search on the space $\mathcal{W}_{\text{eq}}$ of equireplicate designs. The latter are designs where the numbers of times that each treatment is tested (the replication numbers) are as similar as possible. In other words, the designs $\boldsymbol{w} \in \mathcal{W}_{\text{eq}}$ are those designs whose concurrence graph is almost regular, i.e. the difference of degrees between any two nodes is at most 1. The equireplicatedness of designs is a natural property wished by many practitioners, and it has been conjectured that every optimal 2-block-designs is equireplicate for $t-1 \leq N \leq \binom{t}{2}$. The conjecture is known to hold for $t \leq 11$ [CE05]. In order to demonstrate the flexibility of the MISOCP approach, we have also computed designs of $N = 15$ blocks on $t = 10$ treatments by imposing other kind of constraints on the replication numbers. The concurrence graphs of these constrained optimal designs are displayed on Figure 2.

For the KL-exchange algorithm we have used the procedure described in [AD92]: an initial design with $N^{(2)}$ blocks is first chosen at random, where $N^{(2)}$ itself is randomly taken in the interval $0 \leq N^{(2)} \leq \lfloor m/2 \rfloor$. Then, this design is completed to form a design with N blocks, by using a greedy, forward sequential procedure. Finally, the KL-exchange procedure takes place *per se*: design points are replaced by other candidate points in a greedy manner until no improvement occurs. The authors of [AD92] suggest to repeat the above procedure several times, and to keep the best design obtained after N^{R} runs. Two parameters (K and L) are used to specify the size of the pools of candidate points for addition and deletion from the current design. For our experiments we have indicated the best design after $N^{\text{R}} = 20$ runs of the KL-exchange algorithm, with K and L chosen at random in their admissible range for each run of the algorithm. In the table we have also displayed the frequency at which the global optimum was found, out of $N^{\text{R}} = 1000$ runs of the exchange algorithm. This table shows that the KL-exchange algorithm is able to find a very efficient design for all the considered values of t and N. However, there are many examples where the exchange algorithm systematically misses the optimum, and cases where the probability to find the optimum in one run is very low.

The Integer Quadratic Programs (IQP) for DQ-optimality were also solved with CPLEX. The authors of [HF13] mention the case of block designs as a pathological example, where bad designs can be returned. This happened here for $t = 8$ treatments and $N = 12$ blocks, where a disconnected design (no spanning tree, i.e. the information matrix is singular) is DQ-optimal.

	MISOCP				KL-exchange			DQ-opt	
(t, N)	sp.tr.$^{(1)}$	$\mathrm{eff}_D{}^{(2)}$	$t_{\mathrm{sol}}{}^{(3)}$	$t_{\mathrm{proof}}{}^{(4)}$	sp.tr.$^{(1)}$	$\mathrm{eff}_D{}^{(2)}$	success$^{(5)}$	sp.tr.$^{(1)}$	$\mathrm{eff}_D{}^{(2)}$
(8,12)	392	100.00%	0.76	4.68	392	100.00%	93.5%	0	0.00%
(8,14)	1280	100.00%	0.80	28.50	1272	99.91%	0.0%	1280	100.00%
(8,16)	4096	100.00%	0.55	2.23	3840	99.08%	0.0%	4096	100.00%
(9,11)	96	100.00%	4.88	156.57	96	100.00%	5.5%	72	96.47%
(9,13)	560	100.00%	1.23	77.36	553	99.84%	1.6%	320	93.24%
(9,14)	1200	100.00%	7.71	33.55	1168	99.66%	0.8%	1047	98.31%
(9,15)	2223	100.00%	1.33	360.49	2176	99.73%	0.9%	2007	98.73%
(9,16)	4032	100.00%	48.25	390.15	3968	99.80%	0.0%	2871	95.84%
(10,12)	128	100.00%	23.11	1857.60	128	100.00%	11.6%	57	91.40%
(10,15)	2000	100.00%	3.52	745.35	1881	99.32%	0.0%	1815	98.93%
(10,20)	40960	100.00%	31.44	2545.57	39040	99.47%	0.0%	36900	98.85%

(1) Number of spanning trees in the concurrence graph of the design; for the KL exchange algorithm, the value is based on the best design found in $N^{\mathrm{R}} = 20$ independent runs.

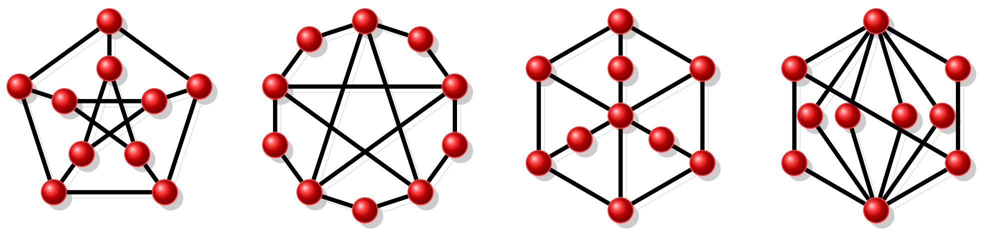
(2) $D-$efficiency of the design (relatively to the exact optimal design of size N); for the KL exchange algorithm, the value is based on the best design found in $N^{\mathrm{R}} = 20$ independent runs.

(3) CPU time (sec.) until CPLEX found the optimal solution

(4) CPU time (sec.) until CPLEX closed the gap (proof of optimality)

(5) Frequency of success of the KL-exchange algorithm, calculated on 1000 runs

Table 2: Comparison of three algorithms for the computation of exact two-blocks designs, with N blocks on t treatments.



(a) (b) (c) (d)

Figure 2: Concurrence graphs of the $D-$optimal designs of $N = 15$ blocks on $t = 10$ treatments, among the class of 2-block designs that (a) are equireplicate; (b) have half of the treatments replicated 2 times, and the other half replicated 4 times; (c) have one treatment replicated at least 6 times; (d) have two treatments replicated at least 6 times.

We have also compared the running time of the MISOCP approach proposed in this paper with that of the original branch and bound approach of Welch [Wel82]. The algorithm proposed by Welch is, to the best of our knowledge, the only other method which has been proposed in the literature to compute provably exact optimal designs (if we except the brute force enumeration techniques that have been used in certain simple cases). In this algorithm, continuous relaxations of the design problem with bounds on the weights must be solved at each node of a binary search tree. This is done by a coordinate exchange algorithm. In addition, the algorithm of Welch uses

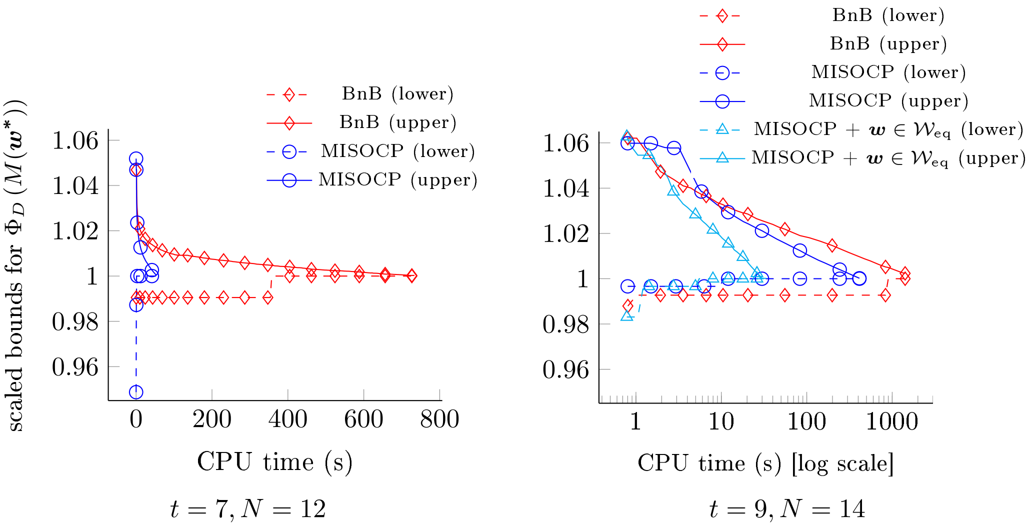


Figure 3: Evolution of the lower and upper bounds with time for two instances of optimal block design, for the branch and bound and MISOCP approached

another upper bound based on the Hadamard inequality for positive definite matrices, and it is also possible to take into account the spectral bound of Ko, Lee, and Wayne [KLW98]. In our experiments, the use of this spectral bound did not seem to improve our computation times. Moreover, we point out that it would be possible to use both the Hadamard and the spectral bounds in conjunction with the proposed MISOCP formulation, by using solver callbacks which allow the user to interact with the branch-and-cut process.

The MISOCP approach also relies on a branch-and-bound procedure, but additional cut inequalities are automatically added by the solver to separate non-integer solutions. Another important difference between the two approaches resides in the branching decisions: high-quality integer programming solvers implement sophisticated heuristics to choose the next variable to branch on, which can considerably reduce the computation time.

The graphics of Figure 3 show the evolution of the best lower and upper bounds with the CPU time in seconds, for $t = 7, N = 12$ (left) and $t = 9, N = 14$ (right). For the latter case we have used a log scale for the time, and we have included the bounds for the MISOCP with additional constraints forcing the design to be equireplicate. The y-axis is scaled relatively to the optimum $\Phi_D\big(M(\boldsymbol{w}^*)\big)$, so that at a given point in time, the ratio between the lower and the upper bounds can be interpreted as a guarantee of $D-$efficiency for the best design found so far. For both instances, the MISOCP is much faster (by two orders of magnitude) to find the optimal solution: for $(t, N) = (7, 12)$ the lower bounds reach the optimum after 1.26s (MISOCP) vs. 357s (B&B), and for $(t, N) = (9, 14)$ after 10.44s (MISOCP) vs. 841s (B&B). The MISOCP was also much quicker to provide a proof of optimality: for $(t, N) = (7, 12)$, the lower and upper bounds met after 53s (MISOCP) vs. 734s (B&B), and for $(t, N) = (9, 14)$ the optimality was proved after 414s by the MISOCP while the branch and bound algorithm of Welch still had a gap of 0.2% after 1500s. Note that the addition of constraints to force the design to be equireplicate drastically reduces the computation time (optimality was proved in the class of equireplicate designs after 33s).

For completeness, we have also tried to use the MAXDET programming approach in conjunction with the branch and bound algorithm implemented in YALMIP [Lö04]. For the case of $t = 7$ treatments and $N = 12$ blocks, it took more than 2 hours to close the gap (vs. 53s for the MISOCP).

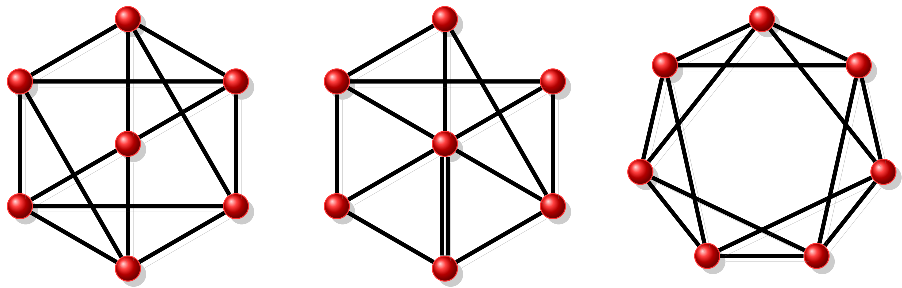


Figure 4: Concurrence Graphs of the $D-$, $A-$ and $G-$ optimal designs for $t = 7$ treatments and $N = 14$ blocks.

Block designs with other optimality criteria Another meaningful criterion for block designs is $A-$optimality. For a block design $\boldsymbol{w} \in \mathbb{N}_0^s$, the quantity $R_{i,j} := A_{i,j} M(\boldsymbol{w})^- A_{i,j}$ is proportional to the variance of the best estimator (BLUE) for the difference of values of treatments i and j. The $A-$optimal design is known to minimize the average pairwise variance $s^{-1} \sum_{i \neq j} R_{i,j}$ of the estimators of differences in treatment effects (see e.g. [BC09]). In other words, in block designs the concept of $A-$optimality coincides with that of $I-$optimality (see § 5.3).

The $A-$optimal also has a nice graph theoretic interpretation: consider the concurrence graph $G(\boldsymbol{w})$ as an electric circuit whose each edge represents a unit resistor. Then, the quantity $R_{i,j}$ is the effective resistance between the poles i and j of the circuit. An $A-$optimal design hence connects the vertices so as to minimize the average pairwise resistance of the electrical network. Similarly, a $G-$optimal design maximizes the smallest effective resistance between 2 nodes if the network.

We have solved the MISOCPs of Table 1, to find the D, A, and $G-$optimal designs for $t = 7$ treatments and $N = 14$ blocks. The concurrence graphs of these designs are displayed on Figure 4. Note that these three designs are distinct (the concurrence graphs are not isomorphic), and the concurrence graph of the $A-$optimal design *is not* regular. These designs were found after a CPU time of respectively 0.79s, 1.37s and 4.02s for the D,A, and $G-$optimality criteria. It took a much longer time to close the MISOCP gap, and hence obtain a proof of optimality, especially for the $G-$criterion (respective CPU times: 70.0s, 73.8s, and 13300s).

Block designs with larger blocks Of course, it is also possible to consider blocks of size $k \geq 2$. If the blocks contain each k distinct elements, then the graph theoretic interpretation presented above remains valid, but the concurrence graphs have to be constructed in a different way. In the case of two-block designs, a block (i_1, i_2) was associated with an edge between the nodes i_1 and i_2. The counterpart for larger blocks consists in associating the block $(i_1, i_2, \ldots, i_k)$ with the complete subgraph joining the nodes $i_1, \ldots, i_k$. The observation matrix $A_{(i_1,i_2,\ldots,i_k)}$ can thus be chosen as any matrix A satisfying

$$AA^T = L_{(i_1,i_2,\ldots,i_k)}^{[-1,-1]},$$

where $L_{(i_1,i_2,\ldots,i_k)}^{[-1,-1]}$ is the $(t-1) \times (t-1)-$submatrix obtained by removing the last row and the last column of the Laplacian of the complete subgraph with vertices $i_1, \ldots, i_k$. Note that the Laplacian $L_{(i_1,i_2,\ldots,i_k)}$ has rank $k-1$, so it is possible to choose $A_{(i_1,i_2,\ldots,i_k)}$ of size $(t-1) \times (k-1)$. We have used our MISOCP approach to compute an exact $D-$optimal design with $N = 5$ blocks of $k = 4$ treatments, when the total number of treatments is $t = 10$. The optimum has been found in 16s, and a proof of optimality was provided after 182s. The optimal design has a concurrence graph with 2.048.000 spanning trees and can be represented in the following table (with the block in columns):

0	1	4	0	1
2	3	6	5	2
3	5	8	7	7
4	6	9	8	9

In comparison, the best design returned by the exchange algorithm (best design out of $N^{\mathrm{R}} = 100$ runs) has a concurrence graph with 1.720.320 spanning trees, and the design we found with the DQ-optimality IQP has only 184.320 spanning trees.

Locally $D-$optimal design in a study of chemical kinetics Another classical field of application of the theory of optimal experimental designs is the study of chemical kinetics. The goal is to select the points in time at which a chemical reaction should be observed, in order to estimate the kinetic parameters $\boldsymbol{\theta} \in \mathbb{R}^m$ of the reaction (rates, orders,...). The measurements at time t are of the form $\boldsymbol{y}_t = \boldsymbol{\eta_t}(\boldsymbol{\theta}) + \boldsymbol{\epsilon_t}$, where $\boldsymbol{\eta_t}(\boldsymbol{\theta}) = [\eta_t^1, \ldots, \eta_t^k]^T$ is the vector of the concentrations of k reactants at time t. The kinetic models are usually given as a set of differential equations, which can be solved numerically to find the concentrations $\boldsymbol{\eta_t}(\boldsymbol{\theta})$ over time. Unlike the linear model described in the introduction of this paper, in chemical kinetics the expected measurements $\mathbb{E}[\boldsymbol{y}_t] = \boldsymbol{\eta_t}(\boldsymbol{\theta})$ at time t depend nonlinearly on the vector $\boldsymbol{\theta}$ of unknown parameters of the reaction. So a classical approach is to search for a *locally optimal design* using a prior estimate $\boldsymbol{\theta}_0$, i.e. a design which would be optimum is the true value of the parameters was $\boldsymbol{\theta}_0$. To do this, the observation equations are linearized around $\boldsymbol{\theta}_0$, so in practice we replace the observation matrix A_t of each individual trial at time t by its *sensitivity* at $\boldsymbol{\theta_0}$, which is defined as:

$$F_t := \left.\frac{\partial \boldsymbol{\eta}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \left.\begin{pmatrix} \frac{\partial \eta_t^1}{\partial \theta_1} & \cdots & \frac{\partial \eta_t^k}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \eta_t^1}{\partial \theta_m} & \cdots & \frac{\partial \eta_t^k}{\partial \theta_m} \end{pmatrix}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \in \mathbb{R}^{m\times k}.$$

A classical example is presented in [AD92]: the study of two consecutive reactions

$$A \xrightarrow{\theta_1} B \xrightarrow{\theta_2} C.$$

The chemical reactions are assumed to be of order θ_3 and θ_4 respectively, so the concentrations of the reactants are determined by the differential equations

$$\begin{aligned} \frac{d[A]}{dt} &= -\,\theta_1[A]^{\theta_3} \\ \frac{d[B]}{dt} &= \theta_1[A]^{\theta_3} - \theta_2[B]^{\theta_4} \\ \frac{d[C]}{dt} &= \theta_2[B]^{\theta_4}, \end{aligned} \tag{11}$$

together with the initial condition $([A],[B],[C])|_{t=0} = (1,0,0)$. These equations can be differentiated with respect to $\theta_1, \ldots, \theta_4$, which gives another set of differential equations that determines the elements $\frac{\partial \eta_t^j}{\partial \theta_i}$ of the sensitivity matrices.

We now assume that measurements can be performed at each $t \in \mathcal{X} = \{0.2, 0.4, \ldots, 19.8, 20\}$, and that the observed quantities are the concentrations of the reactants A and C, i.e. $k = 2$ and $\boldsymbol{\eta_t}^T = ([A](t), [C](t))$. We have solved numerically the differential equations governing the entries of $(F_t)_{t\in\mathcal{X}}$ for the prior $\boldsymbol{\theta_0} := [1, 0.5, 1, 2]^T$. These sensitivities are plotted in Figure 5.

We have used the MISOCP method to compute the exact $D-$optimal design of size $N = 5$ for this problem (for the prior $\boldsymbol{\theta_0}$). The optimum consists in taking 1 measurement at $t = 0.8$, 3 measurements at $t = 2.8$, and 1 measurement at $t = 16.6$. In comparison, the exchange algorithm (using the same settings as described for the block designs, with $N^{\mathrm{R}} = 100$) found a design with 1 measurement for each $t \in \{0.8, 3.4, 17.4\}$, and 2 measurements at $t = 2.6$. This design is of course very close

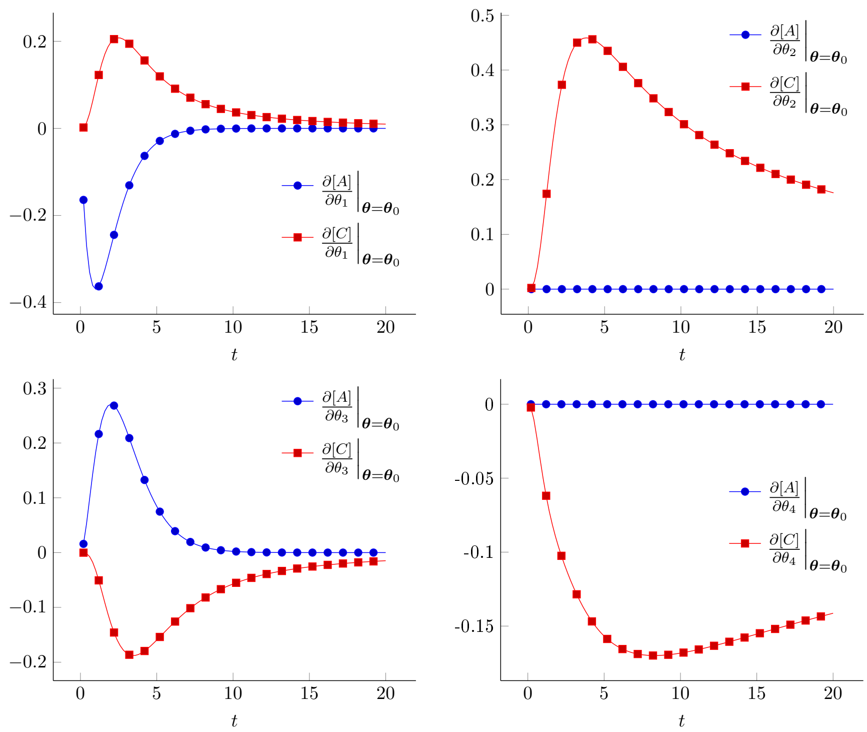


Figure 5: Sensitivities of the measurements (entries of F_t) plotted against time for the prior $\boldsymbol{\theta_0} = [1, 0.5, 1, 2]^T$.

to the optimum (its $D-$efficiency is 98.42%), but we point out that the *true* optimum could not be found by the exchange algorithm, even with a very large number of tries. We have run the exchange procedure $N^R = 5000$ times which took 100s and returned a design of $D-$efficiency 99.42%, while the MISOCP found a provable optimal design after $25s$.

We have plotted these designs on Figure 6 together with the concentrations of the reactants over time when we assume $\boldsymbol{\theta} = \boldsymbol{\theta_0}$. On the figure, we have also plotted other designs which can be of interest for the practitioners. For example, it might be natural to search designs where at most 1 measurement is taken at a given point in time. The exchange algorithm can also be adapted to the case of binary designs (by rejecting candidate points that are already in the support of the design during the exchange procedure). It returned a design of $D-$efficiency 98.97%. The last case we have considered is the following, assume that the experimenter must wait at least one second after a measurement before performing another measurement. This constraint can be modelled as a set of inequalities that can be added in the MISOCP formulation:

$$\{w_{0.2} + w_{0.4} + w_{0.6} + w_{0.8} + w_{1.0} \leq 1,\ \ w_{0.4} + w_{0.6} + w_{0.8} + w_{1.0} + w_{1.2} \leq 1,$$
$$\ldots,\ \ w_{19.2} + w_{19.4} + w_{19.6} + w_{19.8} + w_{20.0} \leq 1\}.$$

This model was solved in 42s with CPLEX, and the corresponding optimal design is depicted on the last row of Figure 6. We do not know any other algorithm which can handle this kind of exact design problem with several linear constraints.

References

[AD92] A.C. Atkinson and A.N. Donev. *Optimum Experimental Designs*, volume 8. Oxford Statistical Science Series, 1992.

[AG03] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, 2003.

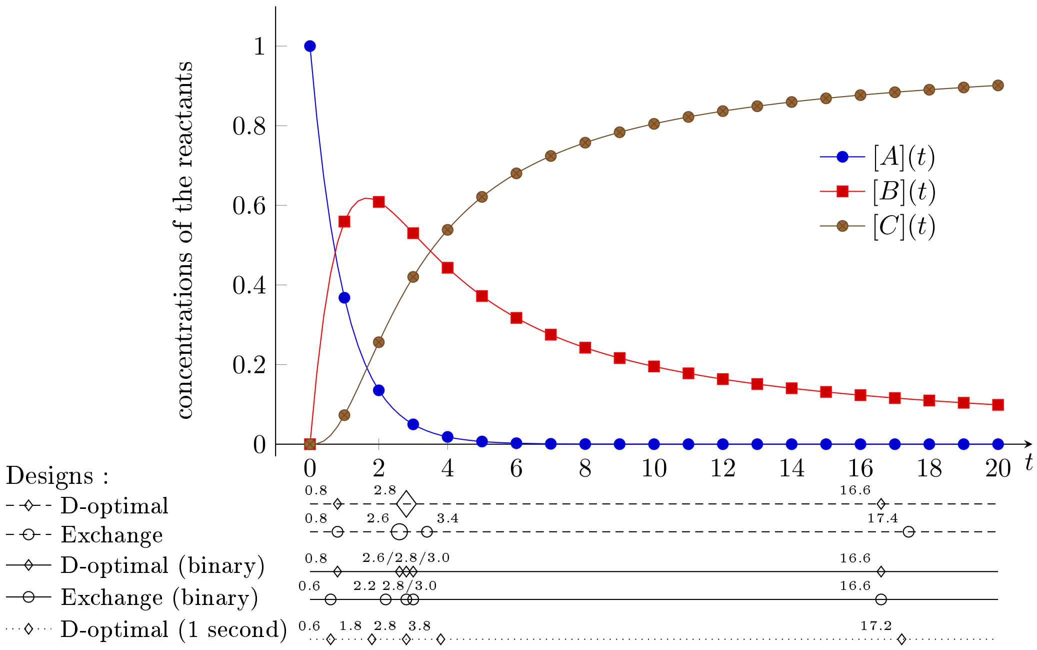


Figure 6: Concentration of the reactants against time (determined by solving Eq. (11), assuming $\boldsymbol{\theta} = \boldsymbol{\theta_0} = [1, 0.5, 1, 2]^T$). Several designs are represented below the graph. The marks indicate the time at which the measurements should be performed, and the size of the marks indicate the number of measurements at a given point in time. *Binary* means that the design space is restricted to designs having at most one measurement for each $t \in \mathcal{X}$, and *1 second* means that at least 1 second must separate 2 measurements.

[AJJ+09] E.D. Andersen, B. Jensen, J. Jensen, R. Sandvik, and U. Worsøe. Mosek version 6. Technical report, Technical Report TR–2009–3, MOSEK, 2009.

[BC09] RA Bailey and Peter J Cameron. Combinatorics of optimal designs. *Surveys in Combinatorics*, 365:19–73, 2009.

[BTN87] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. Society For Industrial Mathematics, 1987.

[CE05] A. Chen and A.H. Esfahanian. A demography of t-optimal (n, m) graphs where n <= 12. In *AMCS*, pages 121–127, 2005.

[CF95] D. Cook and V. Fedorov. Constrained optimization of experimental design. *Statistics*, 26:129–178, 1995.

[CPL09] IBM ILOG CPLEX. V12. 1 user's manual for cplex. Technical report, International Business Machines Corporation, 2009.

[CT80] R. D. Cook and L.A. Thibodeau. Marginally restricted d-optimal designs. *Journal of the American Statistical Association*, 75(370):366–371, 1980.

[DV06] J. Dahl and L. Vandenberghe. Cvxopt: A python package for convex optimization. In *Proc. Eur. Conf. Op. Res*, 2006.

[Fed72] V.V. Fedorov. *Theory of optimal experiments*. New York : Academic Press, 1972. Translated and edited by W. J. Studden and E. M. Klimko.

[FL00] V. Fedorov and J. Lee. Design of experiments in statistics. In H.Wolkowicz, R.Saigal, and L.Vandenberghe, editors, *Handbook of semidefinite programming*, chapter 17. Kluwer, 2000.

[Hai87] L.M. Haines. The application of the annealing algorithm to the construction of exact optimal designs for linear–regression models. *Technometrics*, 29(4):439–447, 1987.

[Har04] R. Harman. Minimal efficiency of designs under the class of orthogonally invariant information criteria. *Metrika*, 60(2):137–153, 2004.

[HF13] R. Harman and L. Filová. Computing efficient exact designs of experiments using integer quadratic programming. *Computational Statistics & Data Analysis*, 2013.

[HJ08] R. Harman and T. Jurík. Computing c-optimal experimental designs using the simplex method of linear programming. *Computational Statistics and data analysis*, 53:247–254, 2008.

[HLCM+03] A. Heredia-Langner, W.M. Carlyle, D.C. Montgomery, C.M. Borror, and G.C. Runger. Genetic algorithms for the construction of D-optimal designs. *Journal of quality technology*, 35(1):28–46, 2003.

[KLW98] C. W. Ko, J. Lee, and K. Wayne. Comparison of spectral and hadamard bounds for d-optimality. In *MODA 5—Advances in Model-Oriented Data Analysis and Experimental Design*, pages 21–29. Springer, 1998.

[KW60] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[LP12] Z. Lu and T.K. Pong. Computing optimal experimental designs via interior point method. Submitted. Preprint arXiv:1009.1909v3, 2012.

[LVBL98] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[Lö04] J. Löfberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[Mit74] T.J. Mitchell. An algorithm for the construction of "D-optimal" experimental designs. *Technometrics. A Journal of Statistics for the Physical, Chemical and Engineering Sciences*, 16:203–210, 1974.

[MMTLF07] R. Martín-Martín, B. Torsney, and J. López-Fidalgo. Construction of marginally and conditionally restricted designs using multiplicative algorithms. *Computational Statistics & Data Analysis*, 51(12):5547–5561, 2007.

[Pap12] D. Papp. Optimal designs for rational function regression. *Journal of the American Statistical Association*, 107(497):400–411, 2012.

[Páz86] A. Pázman. *Foundations of optimum experimental design*. D. Reidel Dordrecht, 1986.

[PR92] F. Pukelsheim and S. Rieder. Efficient rounding of approximate designs. *Biometrika*, pages 763–770, 1992.

[Puk93] F. Pukelsheim. *Optimal Design of Experiments*. Wiley, 1993.

[RJBM10] M. Rodriguez, B. Jones, C.M. Borror, and D.C. Montgomery. Generating and assessing exact g-optimal designs. *Journal of quality technology*, 42(1):3–20, 2010.

[Sag11] G. Sagnol. Computing optimal designs of multiresponse experiments reduces to second-order cone programming. *Journal of Statistical Planning and Inference*, 141(5):1684 – 1708, 2011.

[Sag12] G. Sagnol. Picos, a python interface to conic optimization solvers. Technical Report 12-48, ZIB, 2012. `http://picos.zib.de`.

[STT78] S.D. Silvey, D.M. Titterington, and B. Torsney. An algorithm for optimal designs on a finite design space. *Communications in Statistics - Theory and Methods*, 7(14):1379–1389, 1978.

[Stu99] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[Tit76] D.M. Titterington. Algorithms for computing D-optimal design on finite design spaces. In *Proceedings of the 1976 Conf. on Information Science and Systems*, pages 213–216, Baltimore, USA, 1976. Dept. of Electronic Engineering, John Hopkins University.

[TV04] L. Tack and M. Vandebroek. Budget constrained run orders in optimum design. *Journal of statistical planning and inference*, 124(1):231–249, 2004.

[VBW98] L. Vandenberghe, S. Boyd, and S. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19:499–533, 1998.

[Wel82] W.J. Welch. Branch-and-bound search for experimental designs based on d optimality and other criteria. *Technometrics*, 24(1):41–48, 1982.

[WSB10] S.E. Wright, B.M. Sigal, and A.J. Bailer. Workweek optimization of experimental designs: exact designs for variable sampling costs. *Journal of agricultural, biological, and environmental statistics*, 15(4):491–509, 2010.

[Wyn70] H.P. Wynn. The sequential generation of D-optimum experimental designs. *Annals of Mathematical Statistics*, 41:1655–1664, 1970.

[Yu10] Y. Yu. Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics*, 38(3):1593–1606, 2010.

A Appendix

We present here a generalization of Proposition 4.2, which will be useful to prove Theorem 4.3 in the case of D_K-optimality. Let K be a $m \times k$ matrix of full column rank. Recall that the definition of $\Phi_{D|K}$ involves a generalized inverse (see (3)); we refer to our discussion after Eq. (3) for properties of generalized inverses, and we recall that $\Phi_{D|K}(M) = 0$ if some column of K does not lie in the range of M.

Proposition A.1. *Let H be a $m \times n$ matrix ($m \leq n$), K be a $m \times k$ matrix of full column rank, and let L be optimal for the following problem:*

$$\begin{aligned} \max_{\substack{Q \in \mathbb{R}^{n \times k} \\ L \in \mathbb{R}^{k \times k}}} \quad & \left(\prod_{j=1}^{k} L_{j,j} \right)^{1/k} \\ \text{s.t.} \quad & HQ = KL \\ & L \textit{ is lower triangular} \\ & \|Q\boldsymbol{e_j}\| \leq 1 \qquad (j = 1, \ldots, k) \end{aligned} \tag{12}$$

Then, we have:

$$\det \left(K^T (HH^T)^- K \right)^{-1} = \left(\prod_{k=1}^{m} L_{k,k} \right)^2. \tag{13}$$

Proof. We first handle the singular case, where $\operatorname{range} K \not\subseteq \operatorname{range} HH^T$. Here the matrix $K^T(HH^T)^-K$ is not defined, but by convention $\det\left(K^T(HH^T)^-K\right)^{-1} = \Phi_{D|K}(HH^T)^k = 0$. Note that $Q = \mathbf{0}$, $L = \mathbf{0}$ is a feasible solution to Problem (12), so the optimal value of this problem is at least 0. We then show by contradiction that in every feasible solution, at least one diagonal element of L must be zero, so (13) holds. Otherwise, the triangular matrix L would be invertible, so that range $K =$ range KL, and the equality $HQ = KL$ implies range $K \subseteq$ range $H = \operatorname{range} HH^T$.

So we focus henceforth on the regular case, where $\operatorname{range} K \subseteq \operatorname{range} H$ and the matrix $K^T(HH^T)^-K$ is invertible. We know that the set of feasible solutions of Problem (12) is nonempty (consider $Q = \mathbf{0}$, $L = \mathbf{0}$), and the full rank assumption on K guarantees that this set is compact. Hence there exists an optimal solution for Problem (12), and in what follows Q and L denote a pair of optimal variables. We start with a simple observation which allows us to concentrate on the case where $K = K_0 := [\mathbf{0}, \boldsymbol{I_k}]^T$, i.e. K is the $m \times k$ matrix whose $(m-k)$ first rows are zeros, and the last k rows form an identity block. We call (P_0) the proposition obtained by fixing $K = K_0$ in Proposition A.1. Assume that (P_0) holds, and let K be an arbitrary $m \times k$ matrix of full column rank. Denote by U a matrix obtained by appending $(m-k)$ column vectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_{m-k}}$ to the left of K, in such a way that $U = [\boldsymbol{u_1}, \ldots, \boldsymbol{u_{m-k}}, K]$ is invertible (this can be done since K has full column rank), and observe that $UK_0 = K$. Since U is invertible, we have $HQ = KL \Leftrightarrow U^{-1}HQ = K_0L$, so replacing K by K_0 and H by $H' := U^{-1}H$ leaves the solutions of Problem (12) unchanged. By our assumption that (P_0) holds, we obtain:

$$\begin{aligned} \left(\prod_{k=1}^{m} L_{k,k} \right)^2 = \det \left(K_0^T (H'H'^T)^- K_0 \right)^{-1} &= \det \left(K_0^T U^T (HH^T)^- U K_0 \right)^{-1} \\ &= \det \left(K^T (HH^T)^- K \right)^{-1}. \end{aligned}$$

This shows that Proposition A.1 holds if (P_0) does, and so we assume without loss of generality that $K = [\mathbf{0}, \boldsymbol{I_k}]^T$. If we partition the matrix $M = HH^T$ as

$$M = \begin{pmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{pmatrix},$$

where the blocks M_{11} and M_{22} are of respective size $(m-k, m-k)$ and (k, k), the Schur complement lemma gives $(K^T(HH^T)^-K)^{-1} = M_{22} - M_{12}^T M_{11}^- M_{12}$ (see §3.11

in [Puk93]). Define $L' := KL = [\mathbf{0}, L^T]^T$, and observe similarly as in the proof of Proposition 4.1 that Problem (12) can be solved independently for each pair of columns ($\boldsymbol{q_j} = Q\boldsymbol{e_j}$, $\boldsymbol{l_j} = L'\boldsymbol{e_j}$), for $j = 1\ldots, k$. This process is nothing but the k last steps of the Gram-Schmidt orthogonalization of the rows of H. In particular, one can find $(m-k)$ column vectors $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_{m-k}}$ such that the matrix $\tilde{L} = [\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_{m-k}}, L']$ is a Cholesky factor of M (i.e., $\tilde{L}$ is lower triangular and $M = \tilde{L}\tilde{L}^T$). Recall that the Cholesky decomposition of M is uniquely defined if M is positive definite, i.e. if the m rows of H are mutually independent. However, if at some point of the orthogonalization process, the j^{th} row of H belong to the space spanned by the row vectors $\boldsymbol{h_1}^T, \ldots, \boldsymbol{h_{j-1}}^T$, then $\boldsymbol{v_j}$ can be any vector of the form $\boldsymbol{v_j} = H\boldsymbol{q'_j}$, where $\boldsymbol{q'_j}$ lies in the nullspace of the matrix formed by the rows $\boldsymbol{h_1}^T, \ldots, \boldsymbol{h_{j-1}}^T$. In this particular case, we assume $\boldsymbol{q'_j} = \mathbf{0}$, $\boldsymbol{v_j} = \mathbf{0}$. Decompose $\tilde{L}$ as

$$\tilde{L} = \begin{pmatrix} \tilde{L}_{11} & \mathbf{0} \\ \tilde{L}_{12}^T & \tilde{L}_{22} \end{pmatrix},$$

so that $L = \tilde{L}_{22}$. Denote by I the set of indices such that $(\tilde{L}_{11})_{i,i} = 0$. According to our previous discussion, I also coincides with the set of columns full of zeros in $\tilde{L}_{11}$ and $\tilde{L}_{12}^T$. So we have range $\tilde{L}_{12} \subseteq \{\boldsymbol{u} : \forall i \in I, u_i = 0\} =$ range $\tilde{L}_{11}^T$. This implies the existence of a matrix X such that $\tilde{L}_{11}^T X = \tilde{L}_{12}$. The equation $M = \tilde{L}\tilde{L}^T$ can be written blockwise, which yields $\tilde{L}_{11}\tilde{L}_{11}^T = M_{11}$, $\tilde{L}_{11}\tilde{L}_{12} = M_{12}$, and $\tilde{L}_{12}^T\tilde{L}_{12} + LL^T = M_{22}$. The two former equations imply $M_{11}X = M_{12}$, and the latter can be rewritten as:

$$\begin{aligned} LL^T = \ & M_{22} - \tilde{L}_{12}^T\tilde{L}_{12} = M_{22} - X^T\tilde{L}_{11}\tilde{L}_{11}^T X \\ &= M_{22} - X^T M_{11} X \\ &= M_{22} - X^T M_{11} M_{11}^- M_{11} X \\ &= M_{22} - M_{12}^T M_{11}^- M_{12} \\ &= (K^T(HH^T)^- K)^{-1}. \end{aligned}$$

So we can conclude by taking the determinant:

$$\det\ (K^T(HH^T)^- K)^{-1} = \det\ LL^T = (\det\ L)^2.$$

□