Constructing two-level Q_B-optimal screening designs using mixed-integer programming and heuristic algorithms

Alan R. Vazquez[1,*], Weng Kee Wong[2], and Peter Goos[3,4]

[1]Department of Industrial Engineering, University of Arkansas, U.S.A.
[2]Department of Biostatistics, University of California, Los Angeles, U.S.A.
[3]Department of Biosystems, KU Leuven, Belgium
[4]Department of Engineering Management, University of Antwerp, Belgium
[*]Corresponding author. Email: alanv@uark.edu;
Contributing authors: wkwong@ucla.edu, peter.goos@kuleuven.be

January 13, 2023

Abstract

Two-level screening designs are widely applied in manufacturing industry to identify influential factors of a system. These designs have each factor at two levels and are traditionally constructed using standard algorithms, which rely on a pre-specified linear model. Since the assumed model may depart from the truth, two-level Q_B-optimal designs have been developed to provide efficient parameter estimates for several potential models. These designs also have an overarching goal that models that are more likely to be the best for explaining the data are estimated more efficiently than the rest. However, there is no effective algorithm for constructing them. This article proposes two methods: a mixed-integer programming algorithm that guarantees convergence to the two-level Q_B-optimal designs; and, a heuristic algorithm that employs a novel formula to find good designs in short computing times. Using numerical experiments, we show that our mixed-integer programming algorithm is attractive to find small optimal designs, and our heuristic algorithm is the most computationally-effective approach to construct both small and large designs, when compared to benchmark heuristic algorithms.

Keywords: Coordinate exchange, exact algorithm, iterated local search, model robust, orthogonal design, PBCE algorithm

Statements and Declarations

Not applicable.

1 Introduction

1.1 Background

Experimentation is crucial for product and process innovation. The initial stages of product prototyping involve screening experiments to study the joint impact of many input factors on an outcome or response. The goal is to identify the few factors that influence the response. Building prototypes is, however, time consuming and expensive. Therefore, screening experiments demand cost-efficient experimental designs that gather high-quality data from all factors using a limited number of runs.

Screening experiments are commonly conducted using a two-level design, in which every factor is studied at two levels. Traditional two-level designs are regular and nonregular fractional factorial designs (Wu and Hamada, 2009). In these designs, the two levels of every factor appear equally often and the main effects are not aliased with each other. Regular fractional factorial designs provide interactions that are either fully aliased or not aliased at all with main effects and other interactions. Nonregular fractional factorial designs provide interactions that may be partially aliased with each other and main effects. Regular and nonregular designs are available for run sizes that are multiples of four only. To overcome this limitation, algorithmically-generated optimal designs (Atkinson et al., 2007; Goos and Jones, 2011) have been developed to allow for more flexible run sizes. In this article, we develop computationally-effective algorithms to construct optimal screening designs to study main effects and two-factor interactions.

To construct an optimal screening design, we need an experimental domain containing the feasible test combinations of the m factors under study, a linear regression model that links these factors to the response variable, a given run size n, and an estimation-based criterion. Standard estimation-based criteria are D- and A-optimality criteria (Atkinson et al., 2007) that involve the determinant and trace, respectively, of the variance-covariance matrix of the ordinary least squares estimators of the model's coefficients. If the design has m factors, the experimental domain is a two-level full factorial design with 2^m test combinations or candidate points. A two-level optimal screening design is then a set of n (not necessarily different) candidate points that minimizes an estimation-based criterion.

Standard two-level optimal designs assume that the model specified a priori describes the true relationship between the factors and the response variable. However, screening designs are typically conducted when virtually nothing is known about the factors' effects and so, it is difficult to identify a model as the best beforehand. Assuming an incorrect model

leads to a sub-optimal design for the best model, thereby providing inefficient estimates of all its coefficients or not even allowing it to be fit.

1.2 Two-level model-robust designs

To address the dependence of standard optimal designs on a single assumed model, several authors have proposed two-level model-robust designs that provide efficient estimation for as many models as possible. To this end, these designs assume a *maximal* model of interest with a number of coefficients that may well exceed the run size available. The maximal model contains all main effects and, typically, all two-factor interactions too.

Model-robust designs can be broadly classified into frequentist and Bayesian. Frequentist model-robust designs include those of Lin (1993), Li and Nachtsheim (2000), Heredia-Langner et al. (2004), Jones et al. (2009) and Smucker et al. (2011, 2012). These designs account for model uncertainty through a pre-selected list of potential submodels of the maximal model. They optimize a compound criterion that incorporates the D-optimality criterion values for the submodels. However, constructing these designs is computationally demanding or infeasible when the list of submodels is large, which may because the maximal model contains both the main effects and two-factor interactions of a large number of factors. To address this limitation, Smucker and Drew (2015) recommend using a small surrogate list of submodels selected according to a balanced incomplete block design.

Bayesian model-robust designs include Bayesian D-optimal designs (DuMouchel and Jones, 1994; Jones et al., 2007) and generalized D- and A-optimal designs (Goos et al., 2005). These designs first split the effects in the maximal model into user-specified sets of primary and secondary effects. Next, they optimize modified versions of the D- or A-optimality criterion under a Bayesian framework for the submodel with the primary effects, and minimize the bias incurred by omitting the secondary effects. In addition, the generalized D- and A-optimal designs also maximize the power for testing the lack of fit for the submodel with the primary effects.

Bayesian model-robust designs also include two-level Q_B-optimal designs (Tsai et al., 2007; Tsai and Gilmour, 2010), whose construction is the main subject of this article. These designs optimize the Q_B criterion which assumes that a submodel of the maximal model provides the best fit to the data. The criterion incorporates model uncertainty through a prior probability for each submodel being the best fitting model. The prior probabilities can reflect our belief in *effect sparsity*, meaning that only a few effects will be active and included

in the best fitting model. They can also reflect our belief in *effect hierarchy*, meaning that main effects are more likely to be active than interactions, and in *effect heredity*, meaning that interactions are more likely to be active when the main effects of one or more of the factors involved are active too. Minimizing the Q_B criterion is equivalent to minimizing a weighted average of an approximated A-optimality criterion values of all submodels, the weights being their prior probabilities. Submodels with larger prior probabilities of being the best fitting models have more weight and are therefore estimated better.

Additional appealing features of Q_B-optimal designs are as follows. Compared to the designs of Li and Nachtsheim (2000), Heredia-Langner et al. (2004), Smucker et al. (2011, 2012), and Smucker and Drew (2015), Q_B-optimal designs can be efficient for estimating a larger number of submodels; see Section 3.3 for details. Further, compared to Bayesian D-optimal designs and generalized D- and A-optimal designs, Q_B-optimal designs consider submodels of different sizes which do not need to include a common set of primary effects. Tsai and Gilmour (2010) and Mee et al. (2017) also showed that, under specific maximal models and prior probabilities, the Q_B criterion reduces to traditional criteria for two-level screening designs, such as the $E(s^2)$ criterion (Booth and Cox, 1962), the G_2-aberration criterion (Tang and Deng, 1999), estimation and information capacities (Sun, 1993; Li and Nachtsheim, 2000), and projection estimation and information capacities (Loeppky et al., 2007). Therefore, two-level Q_B-optimal designs are also optimal in terms of these criteria.

1.3 Problem statement and Contributions

Despite the advantages of Q_B-optimal designs, algorithms to construct them from scratch have hitherto not been studied in detail in the literature. The only algorithm for constructing Q_B-optimal designs is the columnwise algorithm of Tsai et al. (2000). It is similar in spirit to the enumeration algorithms for orthogonal designs of Sun et al. (2008) and Schoen et al. (2010), and it shares their weaknesses. For instance, the enumeration algorithms are computationally very demanding for two-level designs with more than 20 runs or 19 factors. To overcome this limitation, the columnwise algorithm uses a partial instead of a complete enumeration of designs, which may result in missing the Q_B-optimal design.

The main purpose of this article is to develop effective algorithms for finding two-level Q_B-optimal designs. Our three main contributions are:

(a) An elegant mixed-integer quadratic programming (MIQP) algorithm to obtain the theoretical two-level Q_B-optimal designs.

(b) The Perturbation-Based Coordinate-Exchange (PBCE) algorithm for constructing large two-level designs that optimize the Q_B criterion.

(c) A novel formula to compute the Q_B-optimality criterion for two-level designs.

The MIQP algorithm uses novel MIQP problem formulations for finding two-level Q_B-optimal designs, which are solved to optimality using optimization solvers. Solvers such as Gurobi, CPLEX and SCIP implement state-of-the-art optimization methods from the literature on operations research, and are computationally more efficient than ever due to the recent advances in computer hardware (Bixby, 2012; Bertsimas et al., 2016). During the optimization routine, the solvers provide good designs and a lower bound on the Q_B-optimality criterion. This feature of the MIQP algorithm is not shared by the columnwise algorithm of Tsai et al. (2000). We show that, by leveraging computing power and modern optimization methods, our MIQP algorithm is successful for solving design problems with up to nine factors when the maximal model has main effects only, and up to six factors when it has both main effects and two-factor interactions.

The PBCE algorithm embeds a coordinate-exchange algorithm (Meyer and Nachtsheim, 1995) in an Iterated Local Search framework (Lourenço et al., 2019) for constructing larger designs that are good in terms of the Q_B criterion. The algorithm has two key ingredients that boost its computational performance. The first ingredient is the novel formula for computing the Q_B criterion of two-level designs, which uses the power moments of the differences between the design's runs (Butler, 2003b,a). The formula is computationally-cheaper for large two-level designs than the formulas available for the criterion in the literature, when interactions are included in the maximal model. The second ingredient of the PBCE algorithm is an operator that ranks the design runs in terms of their contribution to the Q_B criterion value. Runs with a small contribution are more desirable than runs with a large contribution. Thanks to the operator, the PBCE algorithm drives the search for Q_B-optimal designs towards promising regions of the experimental domain, comprised by design runs with a small contribution.

Through numerical experiments, we demonstrate that the PBCE algorithm is effective to construct two-level designs that optimize the Q_B criterion, when the maximal model has main effects only, or both main effects and two-factor interactions. We also show that, with one exception, it outperforms benchmark heuristic algorithms for design problems with nine or more factors, in terms of design quality or computing time.

1.4 Organization

The rest of the article is organized as follows. In Section 2, we introduce notation, definitions and preliminary concepts. In Section 3, we present the Q_B criterion using an alternative approximation to the A-optimality criterion in terms of an infinite power series of matrices. To the best of our knowledge, the development of the Q_B criterion in terms of infinite power series of matrices is new to the literature. In Section 4, we present the MIQP algorithm and, in Section 5, we show the computationally-cheap definition of the Q_B criterion for two-level designs and present the PBCE algorithm. In Section 6, we conduct numerical experiments and show the advantages of our proposed algorithms over alternative ones that cound be used to construct Q_B-optimal designs. We conclude in Section 7 with closing remarks and potential directions for future research. The online supplementary materials for this article include a Python implementation of the MIQP and PBCE algorithms.

2 Notation, definitions and preliminary concepts

We formally define the design space, the maximal models and the model space under study. We also introduce the generalized word counts (Tang and Deng, 1999; Tsai and Gilmour, 2010) to define the Q_B criterion as Tsai and Gilmour (2010) and Mee et al. (2017).

2.1 Design space

The candidate set $\mathcal{D}$ has all 2^m points from an m-factor two-level full factorial design with coded levels -1 and $+1$. We consider the $1 \times m$ vector $\mathbf{d}_u \in \mathcal{D}$ as the u-th candidate point and $N = 2^m$ as the size of the candidate set.

We denote a two-level m-factor n-run design in two equivalent ways, either by an $n \times m$ matrix $\mathbf{D}$ formed by stacking n points of $\mathcal{D}$, or by an $N \times 1$ vector $\mathbf{z}$, the elements of which are integers and indicate the number of times the candidate points appear in the design. More specifically, z_u is larger than zero if and only if $\mathbf{d}_u$ appears in the design z_u times. We have that $\mathbf{1}_N^T \mathbf{z} = n$, where $\mathbf{1}_N$ is the $N \times 1$ vector of ones. Our MIQP algorithm constructs designs using vector $\mathbf{z}$, while our PBCE algorithm uses matrix $\mathbf{D}$.

2.2 Model space

We consider the following maximal models:

- The main effects model which contains the intercept and all m main effects.
- The two-factor interaction model which contains the intercept, all m main effects and all $m(m-1)/2$ two-factor interactions.

We thereby assume that three-factor and higher-order interactions are negligible; although, in principle, it is possible to construct designs for studying these high-order effects.

We denote a maximal model by $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\mathbf{y}$ is an $n \times 1$ vector of responses, $\boldsymbol{\beta}$ is a $p \times 1$ vector of p unknown model coefficients, $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of independent random errors with elements $\epsilon_i \sim N(0, \sigma^2)$, and $\mathbf{X}$ is an $n \times p$ model matrix. Without loss of generality, we assume that $\sigma^2 = 1$. The matrix $\mathbf{X}$ includes the intercept column and the contrast vectors associated with the effects in the maximal model. We denote the i-th column of $\mathbf{X}$ by $\mathbf{x}_i$ and the information matrix of this model by $\mathbf{M} = \mathbf{X}^T\mathbf{X}$.

We consider a collection $\mathcal{S}$ of submodels of the maximal model with a number of elements denoted by $|\mathcal{S}|$. We assume that each submodel $S \in \mathcal{S}$ includes the intercept and denote its number of coefficients by p_s. For the main effects model, $\mathcal{S}$ has all possible submodels. For the two-factor interaction model, $\mathcal{S}$ has submodels satisfying *functional marginality* (McCullagh and Nelder, 1989, ch 3), meaning that, if a two-factor interaction is in the submodel, both main effects of the factors involved are in the submodel too. Functional marginality is different from effect heredity because the former imposes a specific structure on submodels involving interactions, while the latter a belief on associations between these and the main effects. Functional marginality is sensible because it discards submodels that include interactions only and that heavily depart from effect hierarchy. In doing so, it also reduces the number of submodels to consider.

For a submodel S, we consider a $p_s \times p$ matrix $\mathbf{W}_s$ whose elements are zero or one. Matrix $\mathbf{W}_s$ has a column associated to each of the p coefficients in the maximal model and a row associated to each of the p_s coefficients in S. Without loss of generality, we order the rows of $\mathbf{W}_s$ according to the hierarchy of the effects, from low to high order. That is, the first row corresponds to the intercept, the next rows to the main effects, and the last rows to the two-factor interactions, if any, in S. Each row in $\mathbf{W}_s$ has at most one entry equal to one, indicating the coefficient of the maximal model that is included in S. Note that $\mathbf{W}_s\,\mathbf{W}_s^T = \mathbf{I}_{p_s}$, where $\mathbf{I}_{p_s}$ is the $p_s \times p_s$ identity matrix. The model matrix of S is $\mathbf{X}_s = \mathbf{X}\mathbf{W}_s^T$ and its information matrix is $\mathbf{M}_s = \mathbf{W}_s\mathbf{X}^T\mathbf{X}\mathbf{W}_s^T$.

For a given run size n, we refer to a submodel $S \in \mathcal{S}$ as *eligible* if $p_s \leq n$. An eligible submodel is estimable if all its coefficients can be estimated simultaneously using ordinary

least squares (OLS). To this end, $\mathbf{M}_s^{-1}$ must exist.

2.3 Generalized word counts

Let $\mathbf{C}_k$ be the $N \times c_k$ matrix with the k-factor interaction contrast vectors of the m factors formed using the full candidate set $\mathcal{D}$, where $c_k = \binom{m}{k}$. For a two-level design encoded by the vector $\mathbf{z}$, the generalized word count of length k (Tang and Deng, 1999; Tsai and Gilmour, 2010) is

$$B_k = \frac{\mathbf{z}^T \mathbf{C}_k \mathbf{C}_k^T \mathbf{z}}{n^2},$$

which measures the aliasing between the intercept and the k-factor interactions. If the two-level design is encoded by matrix $\mathbf{D}$, the generalized word counts are computed as $B_k = (\mathbf{1}_n^T \mathbf{V}_k \mathbf{V}_k^T \mathbf{1}_n)/n^2$, with an $n \times c_k$ matrix $\mathbf{V}_k$ involving the k-factor interaction contrast vectors generated using $\mathbf{D}$, instead of the full candidate set. In either case, if $B_1 = B_2 = 0$, the two-level design is called orthogonal.

For a two-level orthogonal design $\mathbf{D}$, Butler (2003a) showed that the generalized word counts can be computed as linear combinations of the power moments of the $n \times n$ matrix $\mathbf{T} = \mathbf{D}\mathbf{D}^T$, each element T_{ij} of which measures the similarity between runs i and j in the design. The larger the T_{ij} value, the more similar the runs. Butler (2003a) defined the k-th power moment of $\mathbf{T}$ as $E_k = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij}^k$. By a counting argument, we generalize the link between the first four power moments and generalized word counts to any two-level design, orthogonal or non-orthogonal.

Lemma 1. *For an m-factor two-level design with coded levels -1 and $+1$,*

$$B_1 = E_1, \quad B_2 = \frac{E_2 - m}{2}, \quad B_3 = \frac{E_3 - (3m-2)E_1}{6}, \text{ and}$$
$$B_4 = \frac{E_4 - 2(3m-4)E_2 + 3m(m-2)}{24}.$$

The proof of Lemma 1 and the proofs of all other theoretical results in the article are included in Appendix A.

Lemma 1 benefits the computation of the generalized word counts of two-level designs. This is because, for a design encoded by matrix $\mathbf{D}$, the computational complexity of the original B_k is $O(\binom{m}{k} n)$ while that of E_k is $O(n^2 m)$ for any value of k. Although calculating B_1 and B_2 using Lemma 1 has a comparable complexity with their original definition, computing B_3 and B_4 is much cheaper than using all 3- and 4-factor interaction contrast vectors, especially when the number of factors is large. Lemma 1 is key to developing our novel formula for computing the Q_B criterion.

3 The Q_B criterion in full

The Q_B criterion has two building blocks: an approximation to the A-optimality criterion and a framework to specify the prior probabilities for the submodels of the maximal model. We now introduce each of these blocks separately before the criterion.

3.1 Prior probabilities for the submodels

To determine the prior probability that a submodel will provide the best fit to the data, we adopt the probability framework of Mee et al. (2017). This framework uses the following priors of Chipman (1996):

- π_1 is the prior probability that a main effect is active.
- π_2 is the prior conditional probability that a two-factor interaction is active if both of the main effects of the factors involved are active.
- π_3 is the prior conditional probability that a two-factor interaction is active if only one of the main effects is active.

The probabilities π_2 and π_3 reflect two versions of effect heredity called strong and weak, respectively (Wu and Hamada, 2009, ch. 9). Interactions that do not obey effect heredity have a probability of zero. We assume that the events that main effects are active are independent. Conditional on these events, we assume that the events that interactions are active are independent too (Chipman, 1996).

If the maximal model contains main effects only, the prior probability that a submodel S with a main effects is best is $P(S) = \pi_1^a(1-\pi_1)^{m-a}$. If the maximal model includes interactions, the probability that a submodel S with a main effects and b two-factor interactions (obeying functional marginality) is best then is

$$P(S) = \pi^a(1-\pi)^{m-a}\pi_2^b(1-\pi_2)^{a(a-1)/2-b},$$

where $\pi = \pi_1 + (1-\pi_1)[1-(1-\pi_1\pi_3)^{m-1}]$ is the probability that a main effect is included in S; see Tsai et al. (2007). The probability $P(S)$ is the weight of S in the Q_B criterion.

The choice of prior probabilities for the submodels is independent of the run size of the design. Therefore, submodels that are not eligible (i.e., for which a or $a+b$ are larger than n) may have prior probabilities larger than zero. For this reason, we recommend choosing the values of π_1, π_2 and π_3 such that the probabilities for these submodels are small.

Following Chipman et al. (1997), we also recommend $\pi_3 \leq \pi_2 \leq \pi_1 \leq 0.5$ to reflect effect sparsity, hierarchy and heredity. The meta-analysis of Li et al. (2006) provides empirical support for our recommendation. In the absence of prior knowledge, we suggest $\pi_1 = 0.41$ for main effects and, if interactions are considered, $\pi_2 = 0.33$ and $\pi_3 = 0.045$, based on the actual proportions of active effects found by Li et al. (2006).

3.2 Approximation to the A criterion

The A-optimality criterion (Atkinson et al., 2007, ch 10) for estimating an eligible submodel S is defined as $\text{Tr}(\mathbf{M}_s^{-1})$. Minimizing the criterion is equivalent to minimizing the average variance of the OLS estimates of the coefficients in S. To reduce the computational cost associated with $\mathbf{M}_s^{-1}$, we approximate this matrix using an infinite power series of matrices. Our approximation reduces to that of Tsai et al. (2000), but it is more attractive since it reveals a new property of the Q_B criterion.

First, we partition the information matrix of the maximal model as $\mathbf{M} = n\mathbf{I}_p + \mathbf{B}$, where $\mathbf{B} = (b_{ij})$ is a $p \times p$ *hollow* symmetric matrix with diagonal elements equal to zero and non-diagonal elements $b_{ij} = \mathbf{x}_i^T\mathbf{x}_j$, where $-n \leq b_{ij} \leq n$. The non-diagonal elements of $\mathbf{B}$ are the inner products of two columns of the maximal's model matrix. The information matrix of S then is

$$\mathbf{M}_s = \mathbf{W}_s\,\mathbf{X}^T\mathbf{X}\mathbf{W}_s^T = n\mathbf{W}_s\,\mathbf{W}_s^T + \mathbf{W}_s\,\mathbf{B}\mathbf{W}_s^T = n\mathbf{I}_{p_s} + \mathbf{W}_s\,\mathbf{B}\mathbf{W}_s^T.$$

Following Harville (2011, ch. 18), we compute the inverse of $\mathbf{M}_s$ using the following infinite power series:

$$\mathbf{M}_s^{-1} = \frac{1}{n}\left(\mathbf{I}_{p_s} - \mathbf{Z}_s\right)^{-1} = \frac{1}{n}\sum_{i=0}^{\infty}\mathbf{Z}_s^i, \tag{1}$$

where $\mathbf{Z}_s = -\frac{1}{n}\mathbf{W}_s\,\mathbf{B}\mathbf{W}_s^T$ is a $p_s \times p_s$ hollow symmetric matrix, $\mathbf{Z}_s^0 = \mathbf{I}_{p_s}$, $\mathbf{Z}_s^1 = \mathbf{Z}_s$, $\mathbf{Z}_s^2 = \mathbf{Z}_s\mathbf{Z}_s$, $\mathbf{Z}_s^3 = \mathbf{Z}_s(\mathbf{Z}_s\mathbf{Z}_s)$ and so on. Note that the non-diagonal elements of $\mathbf{Z}_s$ involving two effects can be interpreted as the pairwise correlations between these effects' contrast vectors. The elements of $\mathbf{Z}_s$ involving the intercept and an effect can be seen as a measure of how balanced the effect's contrast vector is in terms of the number of -1s and $+1$s. Clearly, the minimum value of the A criterion is obtained when all non-diagonal elements of $\mathbf{Z}_s$ are zero, which holds when the submodel's model matrix $\mathbf{X}_s$ is orthogonal.

The approximation of Tsai et al. (2000) to $\mathbf{M}_s^{-1}$ involves the sum of the first three elements of the series in Equation (1). More specifically, $\mathbf{M}_s^{-1}$ is approximated by

$$\mathbf{Q}_s = \frac{1}{n}\left(\mathbf{I}_{p_s} + \mathbf{Z}_s + \mathbf{Z}_s^2\right).$$

The diagonal elements of $\mathbf{Q}_s$ are thus approximate variances of the OLS estimates of the coefficients in S. Therefore, the approximation to the A-optimality criterion for estimating S is

$$\operatorname{Tr}(\mathbf{Q}_s) = \frac{p_s}{n} + \frac{1}{n}\operatorname{Tr}(\mathbf{Z}_s) + \frac{1}{n}\operatorname{Tr}(\mathbf{Z}_s^2) = \frac{p_s}{n} + \frac{1}{n}\|\mathbf{Z}_s\|_F^2, \tag{2}$$

since $\operatorname{Tr}(\mathbf{Z}_s) = 0$ because $\mathbf{Z}_s$ is hollow symmetric, and where $\|\mathbf{Z}_s\|_F = \left(\sum_{i=1}^n \sum_{j=1}^n z_{ij,s}^2\right)^{1/2}$ is the Frobenius norm.

This derivation of the approximation to $\mathbf{M}_s^{-1}$ provides useful information. For instance, $\mathbf{Q}_s$ is a valid approximation to $\mathbf{M}_s^{-1}$ only if the infinite series in Equation (1) converges. Harville (2011, ch. 18) shows that the series converges if and only if $\mathbf{M}_s^{-1}$ exists. The following result shows a particular situation in which the series converges:

Proposition 1. *Consider an $n \times n$ matrix $\mathbf{Z}$. If $\|\mathbf{Z}\|_F < 1$, then $\sum_{i=0}^{\infty} \mathbf{Z}^i$ converges.*

The result is a consequence of Theorems 18.2.16 and 18.2.19 in Harville (2011, ch. 18).

The condition in Proposition 1 implies that, for S to be estimable, the contrast vectors of its effects should be (nearly) entry-balanced and have a sum of squared correlations smaller than 1. If the vectors are entry-balanced and this sum is zero, then S is estimated with full efficiency.

Even if the submodel S is not estimable or eligible, the approximation to the A-optimality criterion in Equation (2) is useful to evaluate a design for that model. For fixed values of n and p_s, the smaller the value of $\operatorname{Tr}(\mathbf{Q}_s)$, the more balanced the contrast vectors of the effects in S and the smaller their sum of squared correlations. In other words, the smaller the aliasing among the effects in S.

3.3 The $\mathbf{Q_B}$ criterion

The Q_B criterion (Tsai et al., 2007) for two-level designs is

$$\begin{aligned} Q_B &= \sum_{S \in \mathcal{S}} P(S) \operatorname{Tr}(\mathbf{Q}_s) \\ &= \frac{1}{n}\left\{\sum_{S \in \mathcal{S}} p_s P(S) + \sum_{S \in \mathcal{S}} P(S)\|\mathbf{Z}_s\|_F^2\right\}, \\ &= \frac{1}{n}\left\{E(p_s) + \sum_{S \in \mathcal{S}} P(S)\|\mathbf{Z}_s\|_F^2\right\}. \end{aligned} \tag{3}$$

Since $E(p_s)$ is constant, minimizing the Q_B criterion is equivalent to minimizing a weighted average of the squared Frobenius norm values of the matrices $\mathbf{Z}_s$ for all submodels $S \in \mathcal{S}$

of the maximal model, the weights being the prior probabilities for these submodels. For the submodels with a larger prior probability, the elements $\|\mathbf{Z}_s\|_F^2$ get a larger weight in the criterion and so, a higher priority to be minimized than those of submodels with a smaller probability. From Proposition 1, we then have that submodels with a larger prior probability have a larger stimulus to become estimable when minimizing the Q_B criterion. Therefore, a Q_B-optimal design implicitly maximizes the number of estimable submodels. This property of the criterion was not evident from the criterion's original definition in Tsai et al. (2007).

The minimum value of the Q_B criterion is achieved when all the submodel's model matrices are orthogonal and so, all $\mathbf{Z}_s$'s are zero matrices. In this case, the Q_B criterion value is $E(p_s)/n$.

Tsai and Gilmour (2010) showed that, if the maximal model contains the main effects only, the second term in Equation (3) reduces to

$$\big\{\xi_1 B_1 + 2\xi_2 B_2\big\}/n, \tag{4}$$

where $\xi_i = \pi_1^i$ is the sum of the prior probabilities over all submodels that include i main effects. If the maximal model also contains the two-factor interactions, this term equals

$$\big\{\,[\xi_{10} + 2(m-1)\xi_{21}]\,B_1 + [2\xi_{20} + \xi_{21} + 2(m-2)\xi_{32}]\,B_2 + 6\xi_{31}B_3 + 6\xi_{42}B_4\big\}/n, \tag{5}$$

where ξ_{ij} is the sum of the prior probabilities over all submodels that include any i main effects and any j interactions that obey the functional marginality assumption. Mee et al. (2017) provides formulas to calculate ξ_{ij}. In particular, if we assume strong heredity ($\pi_3 = 0$), we have that $\xi_{ij} = \pi_1^i \pi_2^j$.

The Q_B criterion includes the approximated A-optimality criterion values of all eligible submodels of the maximal model. This is in contrast with the compound criteria used by Li and Nachtsheim (2000), Heredia-Langner et al. (2004), Jones et al. (2009), and Smucker et al. (2011, 2012), for constructing model-robust designs. These criteria necessitate the explicit calculation of all eligible submodel's D-optimality criterion values, which is computationally demanding or infeasible when the maximal model has many effects. The Q_B criterion does not have this issue since it uses the approximated A-optimality criterion and Equations (4) and (5) avoid the explicit evaluation of all submodels in Equation (3), and are computationally inexpensive. Therefore, Q_B-optimal designs can be efficient for estimating a larger number of submodels than the benchmark designs.

In the next section, we introduce a mixed-integer quadratic programming approach to minimize Equations (4) and (5).

4 A mixed-integer programming approach to find Q_B-optimal designs

Mixed-integer programming (MIP) is an optimization method to determine the values of a set of discrete and continuous decision variables so as to maximize or minimize an objective function, while satisfying a set of linear constraints (Bertsimas and Weismantel, 2005; Wolsey, 2020). Optimization solvers such as Gurobi, CPLEX and SCIP implement the most recent developments in the components of MIP, such as cutting plane theory, disjunctive programming for branching rules, primal heuristics, linear optimization methods, pre-processing techniques and symmetry breaking methods (Jünger et al., 2010). The solvers provide feasible solutions and a bound for the objective function's optimal value. During the optimization routine, the gap between this bound and the best solution's objective value gets smaller. The smaller this gap, the closer the best solution is to the optimum. A gap of zero means that the solution is optimal. In contrast with MIP, heuristic algorithms do not provide such bound and gap.

The availability of state-of-the-art solvers combined with the recent advances in computer hardware render MIP as an attractive tool to solve complex optimization problems in statistics. For example, MIP has previously been used to find D-optimal designs for linear regression models (Harman and Filová, 2014), enumerate two- and mixed-level orthogonal designs (Bulutoglu and Margot, 2008; Grömping and Fontana, 2019; Núñez-Ares and Goos, 2020), and search for optimal arrangements of orthogonal designs that involve randomization restrictions (Sartono et al., 2015a,b; Vo-Thanh et al., 2018).

If the objective function of a MIP problem is quadratic in the decision variables, the problem is called a mixed-integer quadratic programming (MIQP) problem. In the rest of this section, we present the first MIQP problem formulations to find two-level Q_B-optimal designs for the main effects model and the two-factor interaction model. We also provide their implementation details.

4.1 The formulation for the main effects model

Our MIQP problem formulation for finding an n-run two-level Q_B-optimal designs when the maximal model contains the main effects only is:

$$\min_{\mathbf{z},\,\mathbf{y}_1,\,\mathbf{y}_2} \xi_1 \mathbf{y}_1^T \mathbf{y}_1 + 2\xi_2 \mathbf{y}_2^T \mathbf{y}_2 \tag{6a}$$

subject to

$$\mathbf{y}_k = \frac{1}{N} \mathbf{C}_k^T \mathbf{z}, \quad k = 1, 2, \tag{6b}$$

$$n = \mathbf{1}_N^T \mathbf{z}, \tag{6c}$$

$$z_u \in \{0, \ldots, n_r\}, \quad u = 0, \ldots, N. \tag{6d}$$

In this problem formulation, $y_i^{(k)}$ is the i-th element of the $c_k \times 1$ vector $\mathbf{y}_k$, n_r is the maximum number of occurrences allowed for a candidate point in the Q_B-optimal design, and $\mathbf{C}_k$ and $\mathbf{z}$ are defined in Section 2. This problem formulation has $m + \binom{m}{2}$ continuous decision variables contained within $\mathbf{y}_1$ and $\mathbf{y}_2$, N integer decision variables contained within $\mathbf{z}$, and $m + \binom{m}{2} + 1$ linear constraints contained within Equations (6b) and (6c).

Up to the proportional constant n, the objective function in Equation (6a) is identical to Equation (4). The objective function is quadratic and expressed in terms of the vectors $\mathbf{y}_k$, which involve the normalized inner products between the intercept column and the k-factor interaction contrast vectors constructed from the full candidate set $\mathcal{D}$.

The problem formulation has three types of constraints. The first type in Equation (6b) substitutes the vector $\mathbf{C}_k^T \mathbf{z}$ with $\mathbf{y}_k$ in the objective function. These constraints normalize the elements in $\mathbf{C}_k^T \mathbf{z}$ by N, the total number of candidate points. The benefit of this normalization is that it implicitly imposes bounds on the decision variables $y_i^{(k)}$. Specifically, the variables $y_i^{(k)}$ have an absolute value of at most one. The second type of constraints in Equation (6c) ensures that the final design has a total of n runs and the third type of constraints in Equation (6d) ensures that the variables z_u are integer-valued and have a value of at most n_r.

The solution to the problem in Equations (6a)–(6d) provides the vector $\mathbf{z}$ and its nonzero components indicate the number of occurrences of each candidate point in the two-level Q_B-optimal design for the main effects model.

4.2 The formulation for the two-factor interaction model

Our MIQP problem formulation for finding an n-run two-level Q_B-optimal design when the maximal model contains the main effects and the two-factor interactions is:

$$\min_{\mathbf{z},\, \mathbf{y}_1,\, \mathbf{y}_2\, \mathbf{y}_3,\, \mathbf{y}_4} \left[\xi_{10} + 2(m-1)\xi_{21}\right] \mathbf{y}_1^T\mathbf{y}_1 + \left[2\xi_{20} + \xi_{21} + 2(m-2)\xi_{32}\right] \mathbf{y}_2^T\mathbf{y}_2 + 6\xi_{31}\mathbf{y}_3^T\mathbf{y}_3 + 6\xi_{42}\mathbf{y}_4^T\mathbf{y}_4 \tag{7a}$$

subject to

$$\mathbf{y}_k = \frac{1}{N}\mathbf{C}_k^T\mathbf{z}, \quad k = 1, 2, 3, 4, \tag{7b}$$

$$n = \mathbf{1}_N^T\mathbf{z}, \tag{7c}$$

$$z_u \in \{0, \ldots, n_r\}, \quad u = 0, \ldots, N. \tag{7d}$$

This problem formulation has $\sum_{v=1}^{4}\binom{m}{v}$ continuous decision variables contained within $\mathbf{y}_1, \ldots, \mathbf{y}_4$, and $\sum_{v=1}^{4}\binom{m}{v} + 1$ linear constraints contained within Equations (7b) and (7c). Up to the proportional constant n, the objective function in Equation (7a) is identical to Equation (5). Equations (7b)–(7d) have a similar interpretation as Equations(6b)–(6d) in the problem formulation for the main effects model.

The solution to the problem formulation in Equations (7a)–(7d) is the vector $\mathbf{z}$ indicating the number of replicates of each candidate point in the two-level Q_B-optimal design for the two-factor interaction model.

4.3 Implementation

We implemented our MIQP formulations in Python v.3.7 and used the solver Gurobi v9.1 to find a solution. The Gurobi solver has many tuning parameters that control its performance (Gurobi Optimization, LLC, 2020). For our problems, preliminary tests of the solver (not discussed here) revealed that their default settings work well in practice.

The Gurobi solver reports information on the progress of the optimization. The most relevant part of the output is the relative gap. This gap equals $(o - b)/o$, where o is the objective function value of the current best solution and b the best lower bound of the objective function value found so far. If the solver is stopped prematurely, a positive relative gap indicates that the solver has not yet found the solution and relative gap of zero means that the solver has found the optimal solution.

5 A heuristic algorithm to find large designs

The MIQP problems in the previous section belong to the class of cardinality-constrained quadratic optimization problems (Bertsimas and Shioda, 2009), which are NP-hard (Bienstock, 1996). This renders our MIQP algorithm computationally demanding (or even infeasible) for finding two-level Q_B-optimal designs when we have many factors or runs. To construct efficient designs for these cases, we introduce an effective heuristic approach called Perturbation-Based Coordinate-Exchange (PBCE) algorithm. The objective function of our algorithm involves a novel computationally-cheap calculation of the Q_B criterion.

5.1 The objective function

For an n-run m-factor two-level design encoded by matrix $\mathbf{D}$, Lemma 1 provides alternative expressions to calculate the first four generalized word counts. Substituting these expressions into Equations (4) and (5) leads to the result below.

Theorem 1. *For two-level designs with n runs, m factors and coded levels -1 and $+1$, we have the following:*

Case I. Under the main effects model, the Q_B-optimal design minimizes $\frac{1}{n}\{-m\xi_2 + \xi_1 E_1 + \xi_2 E_2\}$.

Case II. Under the two-factor interaction model, the Q_B-optimal design minimizes $\frac{1}{n}\left\{w_0 + \sum_{k=1}^{4} w_k E_k\right\}$, *where*

$$w_0 = m\left[\frac{3(m-2)\xi_{42}}{4} - \xi_{20} - \frac{\xi_{21}}{2} - (m-2)\xi_{32}\right],\ w_1 = \left[\xi_{10} + 2(m-1)\xi_{21} - (3m-2)\xi_{31}\right],$$
$$w_2 = \left[\xi_{20} + \frac{\xi_{21}}{2} + (m-2)\xi_{32} - \frac{(3m-4)}{2}\right],\ w_3 = \xi_{31},\ \textit{and}\ w_4 = \frac{\xi_{42}}{2}.$$

We can use Theorem 1 and Equation (4) to calculate the Q_B criterion value of design $\mathbf{D}$ for the main effects model with comparable computational cost. However, for the two-factor interaction model with a large number of factors, the Q_B criterion in Theorem 1 is computationally much cheaper than Equation (5). This is because the complexity of computing E_3 and E_4 is much lower than that for computing the original B_3 and B_4 for this case; see Section 2.3. Theorem 1 benefits the computational performance of our PBCE algorithm, which calculates the Q_B criterion several times during its optimization routine.

Theorem 1 allows us to compute the contribution of each row in the design matrix to the Q_B criterion value.

Corollary 1.1. *The contribution of the j-th row of a two-level design to its Q_B criterion value is $\frac{1}{n^3}\sum_{k=1}^{2}\xi_k\left(m^k+2\sum_{i\neq j}T_{ij}^k\right)$ and $\frac{1}{n^3}\sum_{k=1}^{4}w_k\left(m^k+2\sum_{i\neq j}T_{ij}^k\right)$ for Case I and II, respectively.*

Given design $\mathbf{D}$ and $\mathbf{T}=\mathbf{D}\mathbf{D}^T$, calculating the row contributions of $\mathbf{T}$ is computationally inexpensive. Ideally, all of them are small because this means that the Q_B criterion value of the design is also small. The row contributions in Corollary 1.1 play an important role in our PBCE algorithm, which we describe next.

5.2 The PBCE algorithm

Our PBCE algorithm falls within the framework of Iterated Local Search (ILS; Lourenço et al., 2019). ILS is a metaheuristic that enhances the performance of local search algorithms for combinatorial optimization. Local search algorithms perform local changes to an existing solution to attempt to find a better one (Michalewicz and Fogel, 2004). A weakness of these algorithms is that they tend to get stuck in a locally optimal solution instead of the global optimum, since they do not examine all possible changes to the existing solution. To overcome this deficiency, ILS perturbs the solution obtained from a local search algorithm by making small modifications to it. The perturbed solution is then used as an initial solution for a subsequent execution of the algorithm. The size of the perturbation should be large enough so that the algorithm can escape from a locally optimal solution, but it should not be too large because this would imply that the search essentially restarts from scratch at each iteration. The premise of ILS is that a solution obtained from a local search algorithm is generally quite good, so that it is sensible to retain some part of that solution when attempting to find a better solution.

ILS has been successfully applied to a wide variety of problems such as learning Bayesian networks (De Campos et al., 2003), the traveling salesman problem (Merz and Huhse, 2008), the vehicle routing problem with backhauls (Palhazi-Cuervo et al., 2014), and the design of water distribution networks (De Corte and Sörensen, 2016). In the literature on experimental design, ILS has been used to construct maximin Latin hypercube designs (Grosso et al., 2009) and D-optimal designs for linear regression models (Palhazi-Cuervo et al., 2016).

5.2.1 Local search algorithm

Our local search method involves the coordinate-exchange (CE) algorithm (Meyer and Nachtsheim, 1995). The input of our CE algorithm is an initial two-level n-run m-factor design with coded levels -1 and $+1$. The algorithm attempts to improve the initial design by systematically switching the signs of each of its coordinates. More specifically, if the current coordinate is a -1, the algorithm switches it to $+1$ and vice versa. Our algorithm proceeds column by column, starting at the leftmost column and ending at the rightmost column. The coordinates in each column are explored from top to bottom. As soon as a sign switch in a coordinate improves the Q_B criterion, the algorithm updates the current best design and continues its operations on the newly obtained best design. The algorithm stops when no better design is found after switching the signs of all nm coordinates.

5.2.2 Perturbation operator

To escape from the locally optimal solution of the CE algorithm, we apply a perturbation operator after its completion. The operator we use to perturb a design exchanges some of its coordinates as follows. First, the operator ranks the design rows in descending order of their contributions to the Q_B criterion value of the design; see Corollary 1.1. Next, the first $\lceil n\alpha \rceil$ rows in the ranking are selected, where $0 < \alpha < 1$ is a user-selected tuning parameter. When there is a tie, a random ordering is used for the tied rows. Finally, the operator perturbs each of the selected rows by switching the signs of $\lceil m\alpha \rceil$ randomly chosen elements, where $\lceil x \rceil$ is the smallest integer greater than or equal to x.

The motivation for our perturbation operator is that rows with a small contribution to the Q_B criterion value are more desirable than rows with a large contribution. So, our operator tends to modify problematic rows, while keeping the best rows unchanged. The tuning parameter α controls the perturbation size, since the total number of coordinates that are exchanged is $\lceil n\alpha \rceil \lceil m\alpha \rceil$. The larger the value of α, the larger the number of coordinates that are affected.

5.2.3 The main algorithm

The PBCE algorithm is outlined in Algorithm 1. The inputs to the algorithm are the numbers of runs and factors in the design, the perturbation size α and the maximum number of perturbations without improvement, M. The algorithm begins by generating a starting two-level design at random and supplying it as input to the CE algorithm. The output

Algorithm 1: Pseudocode of the Perturbation-Based Coordinate-Exchange (PBCE) algorithm.

Input: Number of factors, number of runs, perturbation size α, and maximum number of perturbations without improvement M.

Generate two-level design $\mathbf{D}_0$ at random
$\mathbf{D} \leftarrow$ Coordinate_Exchange($\mathbf{D}_0$)
Set $i \leftarrow 0$
while $i < M$ **do**
    $i \leftarrow i + 1$
    $\mathbf{D}_p \leftarrow$ Perturb($\mathbf{D}$, α)
    $\mathbf{D}'_p \leftarrow$ Coordinate_Exchange($\mathbf{D}_p$)
    if $\mathbf{D}'_p$ *is better than* $\mathbf{D}$ **then**
        $\mathbf{D} \leftarrow \mathbf{D}'_p$
        $i \leftarrow 0$

Output: Q_B-efficient design $\mathbf{D}$.

of this process is design $\mathbf{D}$, which is the current best design so far. The PBCE algorithm then applies the perturbation operator to create a modified design $\mathbf{D}_p$ by sign-switching $\lceil n\alpha \rceil \lceil m\alpha \rceil$ coordinates in the $\lceil m\alpha \rceil$ selected rows of $\mathbf{D}$. After that, the algorithm executes the CE algorithm with $\mathbf{D}_p$ as its input to find an improved design $\mathbf{D}'_p$. If this design is better than the current best design, $\mathbf{D}$ is replaced by $\mathbf{D}'_p$ and the PBCE algorithm continues its operations on the newly obtained best design. Otherwise, the algorithm creates another modified design $\mathbf{D}_p$ by perturbing again $\mathbf{D}$; then, it improves $\mathbf{D}_p$ using the CE algorithm. The resulting improved design $\mathbf{D}'_p$ is compared with $\mathbf{D}$ to test if a better design was found. This process is repeated until M perturbations to the current best design $\mathbf{D}$ have been performed without finding a better $\mathbf{D}'_p$ than $\mathbf{D}$.

To increase the likelihood of finding a two-level Q_B-optimal design, the whole procedure is repeated multiple times, each time starting from another randomly generated starting design. The output of the PBCE algorithm is the overall best design among all its restarts. A statistical analysis of the impact of the tuning parameters on the performance of the PBCE algorithm (not shown here) suggests that $\alpha = 0.1$, $M = 100$ and 5 restarts are generally satisfactory to find high-quality designs. A Python implementation of our PBCE algorithm is included in the supplementary materials.

6 Numerical experiments

In this section, we conduct numerical experiments to assess the performance of our MIQP and PBCE algorithms to construct two-level Q_B-optimal designs. First, we consider the main effects model and design problems with four to 17 factors and five to 18 runs. Next, we consider the two-factor interaction model and design problems with four to 11 factors and 11 to 48 runs. Our computer hardware involves a standard CPU with an Intel(R) Core(TM) i7 processor with 2.6Ghz and 16 GB of RAM.

In our experiments, we include other heuristic algorithms in the literature that may be used to search for two-level Q_B-optimal designs. Specifically, they are the:

- Coordinate-exchange (CE) algorithm of Meyer and Nachtsheim (1995).
- Point-exchange (PE) algorithm of Cook and Nachtsheim (1980).
- Restricted columnwise-pairwise (RCP) algorithm of Li (2006, sec. 5.1).

In practice, these algorithms are commonly executed a large number of times. We therefore execute each algorithm 1000 times and report its best design among all executions, except for the PE algorithm. For constructing designs with more than nine factors, we execute this algorithm 10 times only because of its computational cost relative to other heuristic algorithms. Appendix B provides further details using additional experiments to evaluate the computing time required by the heuristic algorithms. There, we also show that the PBCE algorithm is computationally cheaper than the other algorithms under our setup.

For the MIQP algorithm, we set a maximum search time of 20 minutes for the Gurobi solver for each combination of number of runs and number of factors. We chose this time limit to keep all our numerical experiments within computational reach. We also set $n_r = 1$, thereby restricting the search to designs without replicated runs, but other values for n_r are possible.

Our specific goals in this section are to investigate whether the PBCE and MIQP algorithms can (i) find the few two-level Q_B-optimal designs reported in the literature; (ii) find new two-level Q_B-optimal designs; and, (iii) outperform traditional algorithms in optimal experimental design.

6.1 Design problems involving the main effects model

We consider two types of design problems. The first type involves problems in which specific optimal designs are given by Tsai and Gilmour (2016). The second type involves problems

in which the run size is an odd number.

6.1.1 Comparisons with optimal designs from the literature

Tsai and Gilmour (2016) construct two-level n-run $(n-1)$-factor Q_B-optimal designs under the main effects model for $n \equiv 2 \pmod 4$. Their construction method uses conference matrices (Elster and Neumaier, 1995) and produces optimal designs for any prior probability for the main effects, π_1. For illustrative purposes, our design problems involve 5-, 9-, 13- and 17-factor designs with 6, 10, 14 and 18 runs, respectively. We consider prior probabilities π_1 of 0.104, 0.188, 0.410 and 0.625, to reflect screening experiments in which the expected number of active main effects is very small, small, moderate and large, respectively. Tsai and Gilmour (2016) show that the Q_B-optimal designs for these cases are not orthogonal and not necessarily level-balanced.

Table 1 displays the Q_B-efficiencies of the designs found by the CE, RCP, PE, PBCE and MIQP algorithms, relative to the optimal designs of Tsai and Gilmour (2016). To simplify the calculations, the Q_B criterion value of a design involves the second term in Equation (3) only, which is calculated using Equation (4). The Q_B efficiency of a design in the table then is the ratio between the values of Equation (4) for the design of Tsai and Gilmour (2016) and this design.

For five and nine factors, Table 1 shows that the CE, PE, MIQP and PBCE algorithms found the optimal designs for all values of π_1. Regarding the MIQP algorithm, the Gurobi solver proved the optimality of the 6-run 5-factor Q_B-optimal designs within seconds. For the 10-run 9-factor designs, however, the solver did not finish the search for the optimal designs within 20 min. For π_1 equal to 0.104, 0.188, 0.410, and 0.625, the relative gaps between the best solutions and lower bounds obtained by the solver were 58.5%, 53.9%, 49.97%, and 46.7%, respectively. Recall that the relative gap indicates how far the solver was from certifying the optimality of the design obtained, with a relative gap of 0% meaning that an optimal design was found; see Section 4.3. In contrast with all these algorithms, the RCP algorithm found 5-factor 6-run optimal designs for two values of π_1 and did not find any 9-factor 10-run optimal design.

For 13 factors, the PBCE algorithm found the 14-run optimal design for three values of π_1. In contrast, the CE and PE algorithms obtained optimal designs for two values only, while the RCP algorithm did not find any optimal design. For $\pi_1 = 0.104$, none of the heuristic algorithms found the 13-factor 14-run optimal design. In this case, the PBCE and PE algorithms obtained designs with a Q_B-efficiency of 98.7%. In contrast, the CE and

RCP algorithms found designs with an efficiency of 98.1% and 78.5%, respectively. The MIQP algorithm was computationally infeasible for 13-factor 14-run designs.

For 17 factors and 18 runs, the PBCE algorithm found designs with higher Q_B-efficiencies than the CE and RCP algorithms for three values of π_1. More specifically, for π_1 equal to 0.104, 0.188 and 0.410, the Q_B-efficiencies of our algorithm range from 95.8% to 100%, while the efficiencies for the CE and RCP algorithms are not higher than 90.2% and 68.6%, respectively. For $\pi_1 = 0.625$, the only algorithm that found the optimal design was the CE algorithm. The PBCE and RCP algorithms obtained designs with an efficiency of 82.6% and 48.9%, respectively. The MIQP and PE algorithms were computationally infeasible for 17-factor 18-run designs.

In summary, the PBCE and CE algorithms perform best for all the cases in Table 1. There are five cases in which these algorithms produce a different design. In four of these cases, the PCBE algorithm is best. For problems with six runs and five factors, the MIQP algorithm can certify the optimality of the generated designs.

6.1.2 Designs with an odd number of runs

The method of Tsai and Gilmour (2016) cannot be used to construct two-level n-run designs with fewer than $n-1$ factors or an odd value of n. So, Q_B-optimal designs are unknown for these cases. To demonstrate that our algorithm is capable to generate such designs, we consider design problems with four to seven factors. For each number of factors m, the problems have run sizes equal to the three smallest odd integers that are larger than m. As the prior probabilities for the main effects, we use π_1 equal to 0.41 and 0.82 which represent scenarios with a moderate and large expected number of active effects.

Table 2 shows the Q_B criterion values (calculated using Equation 4) of the designs found by the CE, PE, PBCE and MIQP algorithms. For each design problem, these algorithms found two-level designs with the same Q_B criterion value.

The MIQP algorithm certified that the Q_B criterion values in Table 2 are optimal, within the class of two-level unreplicated designs, except for the 13-run 7-factor designs. For these problems, the relative gap was 34.77% and 41.82% for π_1 equal to 0.41 and 0.82, respectively. In any case, the capability of the MIQP algorithm to certify optimality for the other problems is not a feature shared by the CE, PE, RCP and PBCE algorithms, since they are heuristic. Therefore, the MIQP algorithm allows us to claim that we fill the gaps in the catalog of available two-level Q_B-optimal designs under the main effects model for up to seven factors and up to 11 runs.

Table 1: Q_B-efficiencies of two-level designs under a main effects model obtained by the CE, RCP, PE, PBCE and MIQP algorithms, relative to the optimal designs of Tsai and Gilmour (2016). For 13 and 17 factors, the results of the PE or MIQP algorithm are not reported because they did not converge under our setup. *: The MIQP algorithm proved the optimality of the designs.

Factors	Runs	π_1	CE	RCP	PBCE	PE	MIQP
5	6	0.104	1.000	1.000	1.000	1.000	1.000*
		0.188	1.000	0.866	1.000	1.000	1.000*
		0.410	1.000	0.644	1.000	1.000	1.000*
		0.625	1.000	0.560	1.000	1.000	1.000*
9	10	0.104	1.000	0.878	1.000	1.000	1.000
		0.188	1.000	0.722	1.000	1.000	1.000
		0.410	1.000	0.580	1.000	1.000	1.000
		0.625	1.000	0.533	1.000	1.000	1.000
13	14	0.104	0.981	0.785	0.987	0.987	
		0.188	1.000	0.658	1.000	1.000	NA
		0.410	1.000	0.555	1.000	1.000	
		0.625	1.000	0.523	1.000	1.000	
17	18	0.104	0.889	0.686	0.958		
		0.188	0.841	0.588	1.000	NA	NA
		0.410	0.902	0.512	1.000		
		0.625	1.000	0.489	0.826		

Table 2: Q_B criterion values of two-level designs with odd run sizes under the main effects model obtained by the CE, PE, PBCE and MIQP algorithms. The MIQP algorithm proved that all designs are optimal except for the 7-factor 13-run designs.

		π_1	
Factors	Runs	0.41	0.82
4	5	0.0293	0.0908
	7	0.0107	0.0331
	9	0.0050	0.0156
5	7	0.0158	0.0512
	9	0.0074	0.0241
	11	0.0041	0.0132
6	7	0.0219	0.0732
	9	0.0103	0.0344
	11	0.0056	0.0189
7	9	0.0136	0.0466
	11	0.0075	0.0255
	13	0.0045	0.0155

6.2 Design problems involving the two-factor interaction model

As with the main effects model, we use two types of design problems for the two-factor interaction model. The first type involves designs with up to six factors, while the second type involves designs with seven and 11 factors. For the second type of problems, benchmark designs are included in Mee et al. (2017).

6.2.1 Designs with four, five and six factors

The design problems we consider in this section have four, five and six factors with run sizes that range from 11 to 13, 16 to 18, and 22 to 24, respectively. We chose these problems to reflect situations in which the number of runs is larger than or equal to the number of coefficients in the two-factor interaction model. Because of these run sizes, we use large prior probabilities for the main effects and interactions. More specifically, we set $\pi_1 = 0.82$,

$\pi_2 = 0.66$ and $\pi_3 = 0.09$, which are twice the prior probabilities obtained from Li et al. (2006). The design problems here thus also reflect scenarios with a large expected number of active effects.

Table 3 shows the Q_B criterion values (calculated using Equation 5) obtained by the algorithms. For all problems, the CE, PBCE, PE and MIQP algorithms constructed designs with the same values of the Q_B criterion. For problems with four factors and those with five factors and up to 16 runs, the RCP algorithm matched the best Q_B criterion values in Table 3. This algorithm, however, constructed a 5-factor 18-run design and 6-factor designs with 22 to 24 runs with a larger criterion value than the other algorithms.

The MIQP algorithm was of particular use for the 4- and 5-factor designs in Table 3, because it certified that the designs are optimal in terms of the Q_B criterion. The same is true for the 6-factor designs with 22 and 23 runs. The algorithm, however, could not prove the optimality of the 6-factor 24-run design within 20 min. The relative gap between the best solution and lower bound obtained by the solver was 20.47%.

Table 3: Q_B criterion values of two-level designs under the two-factor interaction model obtained by the CE, RCP, PE, PBCE and MIQP algorithms. The prior probabilities are $\pi_1 = 0.82$, $\pi_2 = 0.66$ and $\pi_3 = 0.09$. *: The MIQP algorithm proved the optimality of the designs.

Factors	Runs	CE	RCP	PBCE	PE	MIQP
4	11	0.0892	0.0892	0.0892	0.0892	0.0892*
	12	0.0659	0.0659	0.0659	0.0659	0.0659*
	13	0.0455	0.0455	0.0455	0.0455	0.0455*
5	16	0.0000	0.0000	0.0000	0.0000	0.0000*
	17	0.0178	0.0178	0.0178	0.0178	0.0178*
	18	0.0280	0.0282	0.0280	0.0280	0.0280*
6	22	0.0598	0.0788	0.0598	0.0598	0.0598*
	23	0.0577	0.0666	0.0577	0.0577	0.0577*
	24	0.0526	0.0612	0.0526	0.0526	0.0526

Table 4: Q_B-efficiencies of 7- and 11-factor designs obtained by the PBCE, CE, RCP, PE and MIQP algorithms, relative to the best designs in Tables 9 and 10 of Mee et al. (2017) in terms of the Q_B criterion.

Factors	Runs	CE	RCP	PBCE	PE	MIQP
7	16	1.000	1.000	1.000	1.000	1.000
	20	1.020	1.020	1.020	1.020	1.020
	24	1.000	0.949	1.000	1.000	1.000
	28	1.025	1.025	1.025	1.025	1.025
	32	1.507	1.507	1.507	1.507	1.507
11	20	1.000	0.919	1.000	1.000	0.801
	24	1.000	0.771	1.000	0.804	0.651
	32	0.900	0.854	1.000	0.890	0.720
	40	0.993	0.893	1.060	0.990	0.765
	48	0.679	0.610	0.708	0.663	0.485

6.2.2 Comparisons with available benchmark designs with seven and 11 factors

Mee et al. (2017) report 7- and 11-factor designs with 16 to 48 runs that are good in terms of the Q_B criterion. For the 7-factor designs, these authors consider prior probabilities for main effects (π_1) and two-factor interactions satisfying strong effect heredity (π_2) equal to 0.5 and 0.8, respectively. For the 11-factor designs, their prior probabilities are $\pi_1 = 0.5$ and $\pi_2 = 0.4$. The prior probability for interactions satisfying weak effect heredity is $\pi_3 = 0$.

Table 4 shows the Q_B-efficiencies of 7-factor designs found by the CE, RCP, PE, PBCE and MIQP algorithms relative to the best designs of Mee et al. (2017) in terms of the Q_B criterion. These designs have 16, 20, 24, 28 and 32 runs. In Table 4, we calculate the relative Q_B-efficiency of a design as the ratio between the values of Equation (5) for the design of Mee et al. (2017) and this design.

The MIQP, PBCE, PE and CE algorithms found 7-factor designs which are at least as good as the best designs of Mee et al. (2017) in terms of the Q_B criterion. In particular, the 20-, 28- and 32-run designs produced by the algorithms are 2%, 2.5% and 50.7% more efficient, respectively, than the benchmark designs.

Table 4 also shows the Q_B-efficiencies of 11-factor designs obtained by all algorithms

relative to the best designs of Mee et al. (2017) in terms of the Q_B criterion. These designs have 20, 24, 32, 40 and 48 runs. The table shows that the PBCE algorithm is the best among the algorithms, since it found 11-factor designs that are at least as good as the designs of Mee et al. (2017) for four of the five run sizes in the table. More specifically, our algorithm obtained designs with 20, 24 and 32 runs that have the same relative efficiency as the benchmark designs, while, for 40 runs, it found a design that is 6% more efficient. However, the relative Q_B-efficiency of the 11-factor 48-run design obtained by the PBCE algorithm is only 70.8%.

Regarding the MIQP algorithm, the Gurobi solver could not finish the search for the 7- and 11-factor Q_B-optimal designs within 20 min. For the 7-factor designs with 16, 20, 24, 28 and 32 runs, the relative gaps between the best solutions and lower bounds obtained by the solver were 36.33%, 56.13%, 56.41%, 60.29%, and 37.29%, respectively. For the 11-factor designs, the relative gaps were between 93.65% and 97.18% for all run sizes. These large relative gaps mean that certifying the optimality of 7- and 11-factor designs under the two-factor interaction model is difficult.

6.3 Discussion

Overall, our PBCE algorithm is the most effective approach to construct efficient two-level designs in terms of the Q_B criterion. Under the main effects model, it finds designs which are either optimal or at least as good as those obtained by all the other algorithms, except for a 17-factor 18-run design with $\pi_1 = 0.625$. The fact that the algorithm did not find this optimal design may be because the number of restarts from different initial designs was only five. As a proof of concept, we increased the number of restarts of the algorithm from five to 10. In this case, the PBCE algorithm does find the optimal design. In Appendix B, we demonstrate that the PBCE algorithm is computationally cheaper than the benchmark heuristic algorithms.

The CE algorithm matches the results of the PBCE algorithm for the design problems in Tables 2 and 3, and those with up to nine factors and up to 10 runs in Table 1 and with seven factors and up to 32 runs in Table 4. Regarding the 13 remaining problems in Tables 1 and 4, our PBCE algorithm finds better designs than the CE algorithm for seven of them. Moreover, it does that using less computing time; see Appendix B. This shows that a plain coordinate-exchange algorithm is generally not enough to obtain good large designs in terms of the Q_B criterion. It also demonstrates the added value of our

perturbation operator and the ILS framework.

Under the two-factor interaction model, our PBCE algorithm provides 7- and 11-factor designs that match or even improve upon the best designs of Mee et al. (2017) in terms of the Q_B criterion, except for the 11-factor 48-run design. This is because the 48-run 11-factor design of these authors is an orthogonal design with $B_1 = B_2 = B_3 = 0$. Two-level orthogonal designs with this property are called two-level strength-3 designs (Mee, 2009). It is well-known that constructing large strength-3 designs using coordinate-exchange-based algorithms is challenging (Eendebak and Schoen, 2017).

For specific combinations of numbers of runs and numbers of factors in Table 4, the best designs of Mee et al. (2017) in terms of the Q_B criterion are two-level orthogonal designs. These designs were obtained by the enumeration algorithm of Schoen et al. (2010). Since the columnwise algorithm of Tsai et al. (2000) resembles this algorithm, our experiments implicitly compare their performance with our MIQP and PBCE algorithms. The 48-run 11-factor strength-3 design was actually obtained by the algorithm of Schoen et al. (2010) after 51 days of computing!

The MIQP algorithm finds the same designs as the PBCE algorithm for the problems in Tables 2 and 3, and those with five and nine factors in Table 1 and seven factors in Table 4. In contrast with the PBCE algorithm and all the other heuristic algorithms, the MIQP algorithm (through the Gurobi solver) outputs two-level Q_B-optimal designs with up to seven factors and up to 11 runs under the main effects model, and with up to six factors and up to 23 runs under the two-factor interaction model. So, for small- and moderately-sized design problems, the MIQP algorithm stands out as the only general approach that guarantees the Q_B-optimality of the two-level designs. This algorithm is therefore attractive to users who wish to ensure that they are using the best possible design in their applications.

Within 20 min, however, the Gurobi solver could not prove the optimality of the 9-factor designs in Table 1, the 7-factor 13-run designs in Table 2, the 6-factor 24-run design in Table 3, and the 7-factor designs in Table 4. Using the cluster of the Department of Statistics at UCLA, we ran additional numerical experiments to study the computing time needed to prove the optimality of selected designs. More specifically, we studied the 9-factor 10-run design with a prior probability of 0.104 and 0.625 in Table 1, and the 7-factor designs with 16 and 32 runs in Table 4. Figure 1 shows the computational performance of the Gurobi solver for these cases. For a 9-factor 10-run design with a prior of 0.104, the solver finished the search within four hours; see Figure 1a. For a similar design with a

prior probability of 0.625, the solver finished the search within two hours; see Figure 1b. For the 7-factor designs with 16 and 32 runs, the solver finished the search within 30 and one hours, respectively; see Figures 1c and 1d.

7 Concluding remarks

We introduced the MIQP and PBCE algorithms to construct two-level designs that optimize the Q_B criterion. Our MIQP algorithm uses an existing expression of the criterion and is guaranteed to find the Q_B-optimal designs. The PBCE algorithm features a novel expression of the Q_B criterion and showed that it has potential to construct optimal or highly-efficient designs requiring short computing time. In practice, we recommend the MIQP algorithm for finding the two-level Q_B-optimal designs for the main effects model up to eight factors, and for the two-factor interaction model up to six factors. For all other situations, we recommend the PBCE algorithm.

A byproduct of this research is that we obtained new two-level Q_B-optimal designs of specific sizes. A comprehensive comparison of these designs with two-level fractional factorial designs, standard two-level optimal designs, and the model-robust designs in Section 1.2 is an interesting topic for future research. Hitherto, this was impossible due to the lack of computationally-effective methodology to construct Q_B-optimal designs.

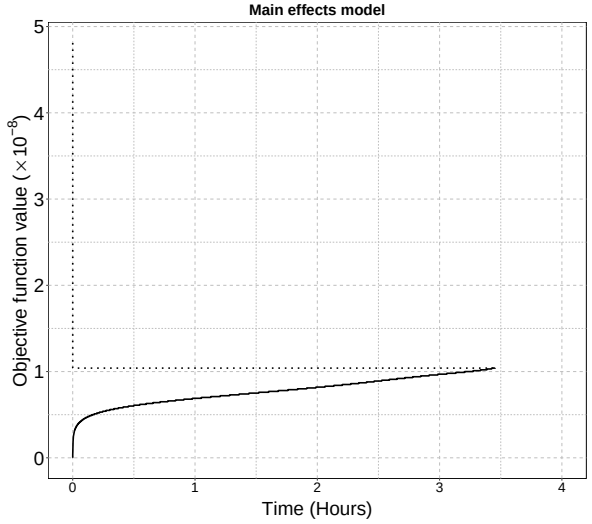
Throughout the article, we assumed a maximal model with main effects and, if required, two-factor interactions. For maximal models with third- and higher-order interactions, the expressions for computing the Q_B criterion in Sections 3.3 and 5.1 would have to include generalized word counts and power moments of order at least five. Finding computationally-efficient expressions of the criterion for these models is also interesting future research. Still, another avenue for future research is to adapt our algorithms to construct three-level Q_B-optimal designs for a maximal model with quadratic effects.

Appendix A: Proofs

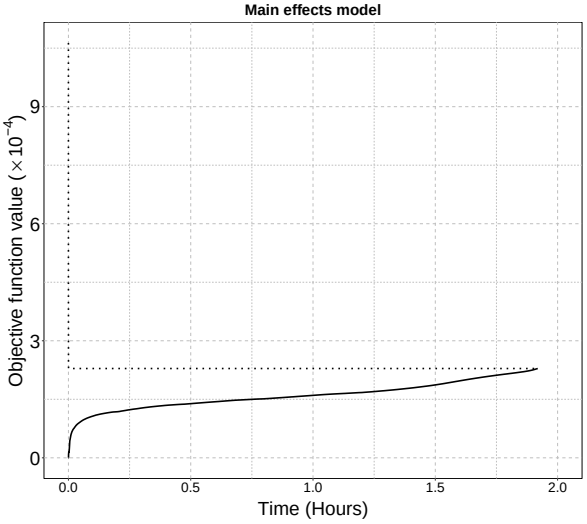
Proof of Lemma 1. Let $\mathbf{D} = (D_{ij})$ where $D_{ij} = \pm 1$. We define

$$s_{i_1 i_2 \ldots i_k} = \sum_{l=1}^{n} D_{li_1} D_{li_2} \cdots D_{li_k},$$

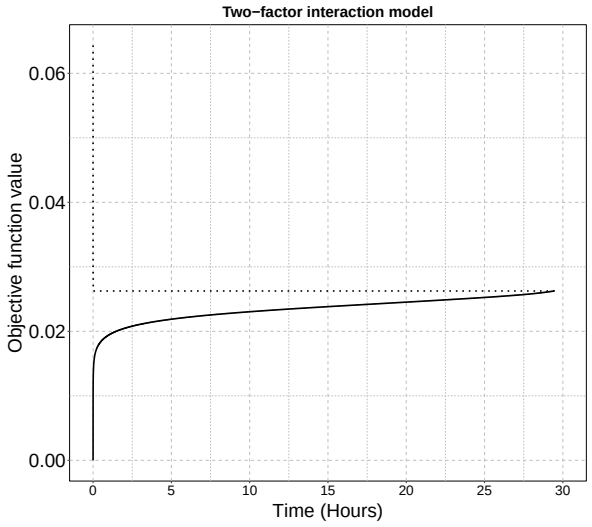
where $i_r \in \{1, \ldots, m\}$. Note that when all elements in $(i_1, i_2, \ldots, i_k)$ are different, $s_{i_1 i_2 \ldots i_k}$ is a J_k-characteristic of $\mathbf{D}$ (Tang, 2001). The generalized word count of length k of $\mathbf{D}$ can



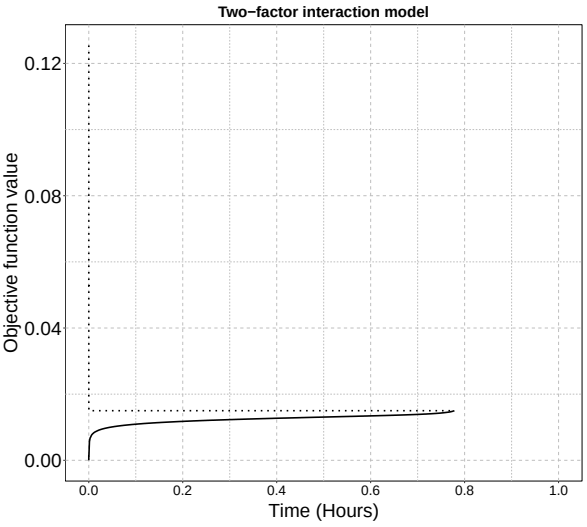
(a) 9-factor 10-run design with $\pi_1 = 0.104$



(b) 9-factor 10-run design with $\pi_1 = 0.625$



(c) 7-factor 16-run design with $\pi_1 = 0.5$ and $\pi_2 = 0.8$



(d) 7-factor 32-run design with $\pi_1 = 0.5$ and $\pi_2 = 0.8$

Figure 1: Performance of the Gurobi solver for finding selected two-level Q_B-optimal designs when the maximum search time is larger than 20 min. The dotted and solid lines show the evolution of the current best Q_B criterion value and lower bound, respectively.

be calculated as follows:

$$B_k = \frac{1}{n^2} \sum_{i_1 < \cdots < i_k} s^2_{i_1 \ldots i_k}.$$

From Lemma 1 of Butler (2003b), we have that

$$E_k = \frac{1}{n^2} \sum_{i_1=1}^{m} \sum_{i_2=1}^{m} \cdots \sum_{i_k=1}^{m} s^2_{i_1 i_2 \ldots i_k}. \tag{8}$$

We will show that the power moments E_1, E_2, E_3 and E_4, can be expressed as linear combinations of the generalized word counts B_1, B_2, B_3 and B_4. The first power moment is obvious since $E_1 = B_1$. For the second power moment, we have that

$$E_2 = \frac{1}{n^2} \sum_{i_1=1}^{m} s^2_{i_1 i_1} + \frac{2}{n^2} \sum_{i_1=1}^{m-1} \sum_{i_2=i_1+1}^{m} s^2_{i_1 i_2} = m + 2B_2.$$

The third power moment E_3 in Equation (8) can expressed as a sum of $n^{-2} s^2_{i_1,i_2,i_3}$ over three types of triplets (i_1, i_2, i_3). The first type involves triplets in which all elements are equal. The sum of all $n^{-2} s^2_{i_1,i_2,i_3}$ over this type equals B_1. The second type involves triplets in which all elements are distinct. The sum of all $n^{-2} s^2_{i_1,i_2,i_3}$ over the second type equals $6B_3$, where the factor 6 is due to the fact that there are 6 permutations of the elements in (i_1, i_2, i_3). The third type of triplets is such that exactly two elements are equal. The sum of all $n^{-2} s^2_{i_1,i_2,i_3}$ over this type is $3(m-1)B_1$. Therefore, we have that $E_3 = (3m-2)B_1 + 6B_3$.

The fourth power moment E_4 in Equation (8) can expressed as a sum of $n^{-2} s^2_{i_1,i_2,i_3,i_4}$ over five types of quadruplets (i_1, i_2, i_3, i_4). These types are summarized as follows: (1) All elements are equal; (2) all elements are distinct; (3) exactly two elements are equal; (4) exactly three elements are equal; and, (5) each element is equal to exactly one other element. By a counting argument, we can show that the sum of the $n^{-2} s^2_{i_1,i_2,i_3,i_4}$ over the first, second, third, fourth and fifth type of quadruplets equals m, $24B_4$, $12(m-2)B_2$, $8B_2$ and $3m(m-1)$, respectively. The sum of these totals shows that $E_4 = 24B_4 + 4(3m-4)B_2 + m(3m-2)$.

Proof of Corollary 1.1. Let $\mathbf{d}_j$ denote the $m \times 1$ vector involving the j-th row of $\mathbf{D}$. Without loss of generality, $\mathbf{D} = [\mathbf{d}_j, \mathbf{D}_{-j}^T]^T$ where $\mathbf{D}_{-j}$ is the $(n-1) \times m$ design matrix excluding the j-th row. We then have that

$$\mathbf{T} = \begin{pmatrix} m & \mathbf{d}_j^T \mathbf{D}_{-j}^T \\ \mathbf{D}_{-j} \mathbf{d}_j & \mathbf{D}_{-j} \mathbf{D}_{-j}^T \end{pmatrix}.$$

Let $\mathbf{r} = \mathbf{D}_{-j} \mathbf{d}_j$ and $\mathbf{S} = \mathbf{D}_{-j} \mathbf{D}_{-j}^T$. The k-th power moment of $\mathbf{T}$ is

$$n^2 E_k = m^k + 2 \sum_{u=1}^{n-1} r_u^k + \sum_{u=1}^{n-1} \sum_{v=1}^{n-1} S_{uv}^k, \tag{9}$$

where r_u is the u-th element of $\mathbf{r}$ and S_{uv} is the element in the u-th row and the v-th column of $\mathbf{E}$. For Case II, substituting the first four power moments given by Equation (9) in the Q_B criterion in Theorem 1 gives

$$\frac{1}{n}\left\{w_0 + \sum_{k=1}^{4}\frac{w_k}{n^2}\left(m^k + 2\sum_{u=1}^{n-1} r_u^k\right) + \sum_{k=1}^{4}\frac{w_k}{n^2}\left(\sum_{u=1}^{n-1}\sum_{v=1}^{n-1} S_{uv}^k\right)\right\}.$$

The contribution of the j-th row of $\mathbf{D}$ to its Q_B criterion value is the second term in the expression above, where we have that $\sum_{u=1}^{n-1} r_u^k = \sum_{i\neq j} T_{ij}^k$. The expression for Case I is found similarly.

Appendix B: Computing times

Our goal here is to compare the computing times required for a completed optimization by the heuristic algorithms. An optimization of the PBCE algorithm involves its standard settings, that is, a perturbation size (α) of 0.1, a maximum number of perturbation without improvement (M) equal to 100, and 5 restarts of the whole procedure. An optimization of the CE, RCP and PE algorithms involves 1000 iterations. In this way, we mimic executions of the algorithms conducted by a standard user.

We consider three sets of design problems. The first set has all combinations of numbers of runs and numbers of factors in Table 1 with $\pi_1 = 0.625$, while the second set has the 7-factor designs in Table 4. The third set of problems involves the 11-factor designs with 20 and 24 runs in Table 4.

Table 5 shows the computing times required for a completed optimization by the heuristic algorithms. For each design problem, the table gives the averages and standard deviations of 10 optimizations. Preliminary experiments revealed that a single optimization of the PE algorithm with 1000 iterations is computationally more demanding than the other algorithms, especially for designs with more than nine factors. For this reason, Table 5 gives the computing times required for optimizations of the PE algorithm with 10 iterations only. Using these computing times, we estimate the average time required by an optimization of the PE algorithm with 1000 iterations.

Clearly, the PBCE algorithm outperforms the CE and RCP algorithms in terms of the computing time. For the 9-, 11-, 13- and 17-factor designs in Table 5, the PBCE algorithm even takes less computing time than the PE algorithm with 10 iterations. For the 17-factor 18-run design, the PE algorithm was computationally infeasible because its 10 iterations

Table 5: Average computing times in seconds and their standard deviations for 10 optimizations performed by the heuristic algorithms. Each optimization of the PBCE algorithm involved its standard settings, while each optimization of the CE and RCP algorithm involved 1000 iterations. An optimization of the PE algorithm involved 10 iterations. NA: not available because the algorithm took more than an hour to complete a single optimization.

Factors	Runs	CE	RCP	PBCE	PE
5	6	1.158 ± 0.097	0.924 ± 0.028	0.689 ± 0.009	0.093 ± 0.006
9	10	5.258 ± 0.039	3.052 ± 0.037	2.004 ± 0.060	4.055 ± 0.483
13	14	16.687 ± 0.492	10.340 ± 0.086	6.021 ± 0.554	140.719 ± 9.266
17	18	37.381 ± 0.262	24.140 ± 0.087	15.050 ± 1.878	NA
7	16	11.406 ± 0.093	8.565 ± 0.107	4.174 ± 0.135	2.279 ± 0.126
	20	15.915 ± 0.125	11.366 ± 0.091	7.136 ± 0.554	2.850 ± 0.147
	24	21.086 ± 0.196	15.790 ± 0.157	10.632 ± 1.262	3.496 ± 0.218
	28	26.664 ± 0.117	20.742 ± 0.305	11.476 ± 1.049	4.478 ± 0.285
	32	32.307 ± 0.122	26.658 ± 0.235	10.531 ± 0.550	5.329 ± 0.380
11	20	37.210 ± 0.850	23.569 ± 0.160	11.509 ± 0.603	81.571 ± 6.788
	24	50.503 ± 0.538	32.240 ± 0.392	10.968 ± 0.277	101.599 ± 6.834

took longer than an hour. In contrast, the PBCE algorithm required 15 seconds to solve the optimization problem in this case.

For the 6- and 7-factor designs, the average computing times of the PE algorithm in Table 5 should be multiplied by 100, so as to obtain the times required by optimizations with 1000 iterations. Therefore, the computing times for the PE algorithm are around 9.3 seconds for six factors, and between 227.9 and 532.9 seconds for seven factors. Clearly, these computing times are much larger than those of the PBCE algorithm in the table.

In conclusion, a completed optimization of the PBCE algorithm is computationally less expensive than one of the benchmark heuristic algorithms.

Supplementary files

- **Supplementary files.zip** A Python implementation of all algorithms discussed in the main text.

Acknowledgments

The research of the first author was financially supported by the Flemish Fund for Scientific Research FWO through a Junior Postdoctoral Fellowship. The first author thanks José Nũnez Ares for the discussions that shifted the initial focus of this research towards mixed-integer programming.

References

Atkinson, A., Donev, A., and Tobias, R. (2007). *Optimum Experimental Designs, With SAS*. OUP Oxford.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via modern optimization lens. *Annals of Statistics*, 44:813–852.

Bertsimas, D. and Shioda, R. (2009). Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43:1–22.

Bertsimas, D. and Weismantel, R. (2005). *Optimization Over Integers*. Dynamic Ideas Press.

Bienstock, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, 74:121–140.

Bixby, R. (2012). A brief history of linear and mixed-integer programming computation. *Doumenta Mathematica. Extra Volume: Optimization Stories*, pages 107–121.

Booth, K. H. V. and Cox, D. R. (1962). Some systematic supersaturated designs. *Technometrics*, 4:489–495.

Bulutoglu, D. A. and Margot, F. (2008). Classification of orthogonal arrays by integer programming. *Journal of Statistical Planning and Inference*, 138:654–666.

Butler, N. A. (2003a). Minimum aberration construction results for nonregular two-level fractional factorial designs. *Biometrika*, 90:891–898.

Butler, N. A. (2003b). Some theory for constructing minimum aberration fractional factorial designs. *Biometrika*, 90:233–238.

Chipman, H. (1996). Bayesian variable selection with related predictors. *Canadian Journal of Statistics*, 24:17–36.

Chipman, H., Hamada, M., and Wu, C. F. J. (1997). A Bayesian variable-selection approach for analyzing designed experiments with complex aliasing. *Technometrics*, 39:372–381.

Cook, R. D. and Nachtsheim, C. J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22:315–324.

De Campos, L. M., Fernández-Luna, J. M., and Puerta, J. M. (2003). An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests. *International Journal of Intelligent Systems*, 18(2):221–235.

De Corte, A. and Sörensen, K. (2016). An iterated local search algorithm for water distribution network design optimization. *Networks*, 67(3):187–198.

DuMouchel, W. and Jones, B. (1994). A simple Bayesian modification of D-optimal designs to reduce dependence on an assumed model. *Technometrics*, 36(1):37–47.

Eendebak, P. and Schoen, E. D. (2017). Two-level designs to estimate all main effects and two-factor interactions. *Technometrics*, 59:69–79.

Elster, C. and Neumaier, A. (1995). Screening by conference designs. *Biometrika*, 82:589–602.

Goos, P. and Jones, B. (2011). *Design of Experiments: A Case Study Approach*. New York: Wiley.

Goos, P., Kobilinsky, A., O'Brien, T. E., and Vandebroek, M. (2005). Model-robust and model-sensitive designs. *Computational Statistics and Data Analysis*, 49:201–216.

Grömping, U. and Fontana, R. (2019). An algorithm for generating good mixed level factorial designs. *Computational Statistics and Data Analysis*, 137:101–114.

Grosso, A., Jamali, A., and Locatelli, M. (2009). Finding maximin Latin hypercube designs by iterated local search heuristics. *European Journal of Operational Research*, 197(2):541–547.

Gurobi Optimization, LLC (2020). Gurobi Optimizer Reference Manual. Available at http://www.gurobi.com.

Harman, R. and Filová, L. (2014). Computing efficient exact designs of experiments using integer quadratic programming. *Computational Statistics and Data Analysis*.

Harville, D. A. (2011). *Matrix Algebra From a Statistician's Perspective*. Springer New York.

Heredia-Langner, A., Montgomery, D. C., Carlyle, W. M., and Borror, C. M. (2004). Model-robust optimal designs: A genetic algorithm approach. *Journal of Quality Technology*, 36:263–279.

Jones, B., Lin, D. K. L., and Nachtsheim, C. J. (2007). Bayesian D-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, 138:86–92.

Jones, B. A., Li, W., Nachtsheim, C. J., and Ye, K. Q. (2009). Model-robust supersaturated and partially supersaturated designs. *Journal of Statistical Planning and Inference*, 139:45–53.

Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., and Wolsey, L. A., editors (2010). *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*. Springer.

Li, W. (2006). Screening designs for model selection. In A., D. and S., L., editors, *Screening*, pages 207–234. Springer, New York, NY.

Li, W. and Nachtsheim, C. J. (2000). Model-robust factorial designs. *Technometrics*, 42:345–352.

Li, X., Sudarsanam, N., and Frey, D. D. (2006). Regularities in data from factorial experiments. *Complexity*, 11:32–45.

Lin, D. K. J. (1993). Another look at first-order saturated design: The p-efficient designs. *Technometrics*, 35:284–292.

Loeppky, J. L., Sitter, R. R., and Tang, B. (2007). Nonregular designs with desirable projection properties. *Technometrics*, 49:454–467.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2019). Iterated local search: Framework and applications. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, pages 129–168. Springer International Publishing.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman and Hall.

Mee, R. (2009). *A Comprehensive Guide to Factorial Two-Level Experimentation*. Mathematics and Statistics. Springer.

Mee, R. W., Schoen, E. D., and Edwards, D. E. (2017). Selecting an orthogonal or non-orthogonal two-level design for screening. *Technometrics*, 59:305–318.

Merz, P. and Huhse, J. (2008). An iterated local search approach for finding provably good solutions for very large tsp instances. In Rudolph, G., Jansen, T., Beume, N., Lucas, S., and Poloni, C., editors, *Parallel Problem Solving from Nature – PPSN X*, pages 929–939, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

Michalewicz, Z. and Fogel, D. (2004). *How to Solve It: Modern Heuristics*. Springer.

Núñez-Ares, J. and Goos, P. (2020). Enumeration and multicriteria selection of orthogonal minimally aliased response surface designs. *Technometrics*, 62:21–36.

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

Palhazi-Cuervo, D., Goos, P., Sörensen, K., and Arráiz, E. (2014). An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, 237(2):454–464.

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

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

Schoen, E. D., Eendebak, P. T., and Nguyen, M. V. M. (2010). Complete enumeration of pure-level and mixed-level orthogonal arrays. *Journal of Combinatorial Designs*, 18:123–140.

Smucker, B. and Drew, N. M. (2015). Approximate model spaces for model-robust experiment design. *Technometrics*, 57:54–63.

Smucker, B. J., del Castillo, E., and Rosenberg, J. L. (2011). Exchange algorithms for constructing model-robust experimental designs. *Journal of Quality Technology*, 43:28–42.

Smucker, B. J., del Castillo, E., and Rosenberg, J. L. (2012). Model-robust two-level designs using coordinate exchange algorithms and a maximin criterion. *Tehnometrics*, 54:367–275.

Sun, D. X. (1993). *Estimation Capacity and Related Topics in Experimental Design*. PhD thesis, University of Waterloo, Department of Statistics and Actuarial Science, Waterloo ON, Canada.

Sun, D. X., Li, W., and Ye, K. Q. (2008). Algorithmic construction of catalogs of non-isomorphic two-level orthogonal designs for economic run sizes. *Statistics and Applications*, 6:141–155.

Tang, B. (2001). Theory of J-characteristics for fractional factorial designs and projection justification of minimum G_2-aberration. *Biometrika*, 88:401–407.

Tang, B. and Deng, L. Y. (1999). Minimum G_2-aberration for non-regular fractional factorial designs. *The Annals of Statistics*, 27:1914–1926.

Tsai, P.-W. and Gilmour, S. G. (2010). A general criterion for factorial designs under model uncertainty. *Technometrics*, 52:231–242.

Tsai, P.-W. and Gilmour, S. G. (2016). New families of Q_B-optimal saturated two-level main effects screening designs. *Statistica Sinica*, 26:605–617.

Tsai, P.-W., Gilmour, S. G., and Mead, R. (2000). Projective three-level main effects designs robust to model uncertainty. *Biometrika*, 87:467–475.

Tsai, P.-W., Gilmour, S. G., and Mead, R. (2007). Three-level main-effects designs exploiting prior information about model uncertainty. *Journal of Statistical Planning and Inference*, 137:619–627.

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

Wolsey, L. (2020). *Integer Programming*. Wiley, 2nd edition.

Wu, C. F. J. and Hamada, M. S. (2009). *Experiments: Planning, Analysis and Optimization*. Wiley, 2nd edition.