Solving the Optimal Experiment Design Problem with Mixed-Integer Convex Methods

Deborah Hendrych
Zuse Institute Berlin, Germany
Technische Universität Berlin, Germany

Mathieu Besançon
Université Grenoble Alpes, Inria, LIG, Grenoble, France
Zuse Institute Berlin, Germany

Sebastian Pokutta
Technische Universität Berlin, Germany
Zuse Institute Berlin, Germany

Abstract

We tackle the Optimal Experiment Design Problem, which consists of choosing experiments to run or observations to select from a finite set to estimate the parameters of a system. The objective is to maximize some measure of information gained about the system from the observations, leading to a convex integer optimization problem. We leverage `Boscia.jl`, a recent algorithmic framework, which is based on a nonlinear branch-and-bound algorithm with node relaxations solved to approximate optimality using Frank-Wolfe algorithms. One particular advantage of the method is its efficient utilization of the polytope formed by the original constraints which is preserved by the method, unlike alternative methods relying on epigraph-based formulations. We assess our method against both generic and specialized convex mixed-integer approaches. Computational results highlight the performance of our proposed method, especially on large and challenging instances.

2012 ACM Subject Classification Theory of computation → Branch-and-bound; Theory of computation → Integer programming; Theory of computation → Convex optimization

Keywords and phrases Mixed-Integer Non-Linear Optimization, Optimal Experiment Design, Frank-Wolfe, Boscia

Digital Object Identifier 10.4230/LIPIcs.SEA.2024.16

Related Version *Extended Version*: `https://arxiv.org/abs/2312.11200`

Supplementary Material *Software (Source Code)*: `https://github.com/ZIB-IOL/OptimalDesignWithBoscia`, archived at `swh:1:dir:165e9afd84b473294aabe56781360b9d527c96b9`

Funding Research reported in this paper was partially supported through the Research Campus Modal funded by the German Federal Ministry of Education and Research (fund numbers 05M14ZAM,05M20ZBM) and the Deutsche Forschungsgemeinschaft (DFG) through the DFG Cluster of Excellence MATH+.

1 Introduction

The *Optimal Experiment Design Problem (OEDP)* arises in statistical estimation and empirical studies in many applications areas from Engineering to Chemistry. For OEDP, we assume we have a matrix A consisting of the rows $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^n$ where each row represents an experiment. The ultimate aim is to fit a regression model:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \|A\boldsymbol{\theta} - \mathbf{y}\|, \tag{1}$$

where $\mathbf{y}$ encodes the responses of the experiments and $\boldsymbol{\theta}$ are the parameters to be estimated. The set of parameters with size n is assumed to be (significantly) smaller than the number of distinct experiments m. Furthermore, we assume that A has full column rank, i.e. the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ span $\mathbb{R}^n$.

The problem is running all experiments, potentially even multiple times to account for errors, is often not realistic because of time and cost constraints. Thus, OEDP deals with finding a subset of size N of the experiments providing the "most information" about the experiment space [23, 12]. The *number of allowed experiments* N has to be larger than n to allow the regression model in Equation (1) to be solved. It is, however, assumed to be less than m.

In Section 2, we investigate the necessary conditions for a function to be a valid and useful information measure. Every information function leads to a different criterion. In this paper, we focus on two popular criteria, namely the A-criterion and D-criterion, see Section 2.1.

In general, OEDP leads to a *Mixed-Integer Non-Linear Problem (MINLP)*. There has been a lot of development in the last years in solving MINLP [17]. Nevertheless, the capabilities of current MINLP solvers are far away from their linear counterparts, the *Mixed-Integer Problem (MIP)* solvers [4], especially concerning the magnitude of the problems that can be handled. Therefore, instead of solving the actual MINLP, a continuous version of OEDP, called the *Limit Problem*, is often solved and the integer solution is created from the continuous solution by some form of rounding [23]. This does not necessarily lead to optimal solutions, though, and the procedure is not always applicable to a given continuous solution either.

The goal of this paper is to compare the performance of different MINLP approaches for OEDP problems. A special focus is put on our newly proposed framework Boscia [14] which can solve larger instances and significantly outperforms the other examined approaches. Since it leverages a formulation and solution method that differs from other approaches, we establish the convergence of the Frank-Wolfe algorithm used for the continuous relaxations on the considered OEDP problems in Section 3. The different solution methods are detailed in Section 4 and the evaluation of the computational experiments can be found in Section 5.

1.1 Related Work

As mentioned, one established method of solution is the reduction to a simpler problem by removing the integer constraints and employing heuristics to generate an integer solution from the continuous solution. Recently, there have been more publications tackling the MINLP formulation of the Optimal Experiment Design Problem. These, however, concentrate on specific information measures, in particular the A-criterion, see [21, 1], and the D-criterion, [27, 22, 19]. The most general solution approach known to the authors was introduced in [2]. It considers OEDP under matrix means which, in particular, includes the A-criterion and D-criterion. While the matrix means covers many information measures of interest, it still yields a restricting class of information functions. For example, the G-criterion and V-criterion are not included in this class of functions [12]. Our newly proposed framework Boscia only requires the information measures to be either L-smooth, i.e. the gradient is Lipschitz continuous, or generalized self-concordant, thereby covering a larger group of information functions. In addition, Boscia does not suppose any prior knowledge about the structure of the problem, being thus more flexible in terms of problem formulations. On the other hand, it is highly customizable, giving the user the ability to exploit the properties of their problems to speed up the solving process. An in-depth and unified theory for the Optimal Experiment Design Problem can be found in [23].

1.2 Contribution

Our contribution can be summarized as follows.

Unified view on experiment design formulations

First, we propose a unified view of multiple experiment design formulations as the optimization of a nonlinear (not necessarily Lipschitz-smooth) information function over a truncated scaled probability simplex intersected with the set of integers. Unlike most other formulations that replace the nonlinear objective with nonlinear and/or conic constraints, we preserve the original structure of the problem. Additionally, we can easily handle special cases of OEDP without any reformulations since we do not suppose a specific problem structure, unlike the approach in [2].

Superior solution via the Boscia framework

We use the recently proposed Boscia framework [14] solving MINLPs with a Frank-Wolfe method for the node relaxations of a branch-and-bound tree, and show the effectiveness of our method on instances generated with various degrees of correlation between the parameters.

1.3 Notation

In the following let $\lambda_i(A)$ denote the i-th eigenvalue of matrix A; we assume that these are sorted in increasing order. Moreover, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the minimum and maximum eigenvalue of A, respectively. Further, let $\log\det(\cdot)$ be the log-determinant of a positive definite matrix. Given matrices A and B of the same dimensions, $A \circ B$ denotes their Hadamard product. Given a vector $\mathbf{x}$, $\operatorname{diag}(\mathbf{x})$ denotes the diagonal matrix with $\mathbf{x}$ on its diagonal. The cones of positive definite and positive semi-definite matrices in $\mathbb{R}^{n\times n}$ will be denoted by $\mathbb{S}^n_{++}$ and $\mathbb{S}^n_{+}$, respectively. We will refer to them as PD and PSD cones. For $m \in \mathbb{Z}_{>0}$, let $[m] = \{1, 2, \ldots, m\}$. Lastly, we denote matrices with capital letters, e.g. X, vectors with bold small letters, e.g. $\mathbf{x}$, and scalars as simple small letters, e.g. λ.

2 Optimal Experiment Design

As explained in the introduction of the paper, the matrix A encodes the experiments. We are allowed to perform N experiments and are interested in finding the subset with the "most" information gain. Consequently, an important question to answer is how to quantify information. To that end, we introduce the information matrix

$$X(\mathbf{x}) = \sum_{i=1}^{m} x_i \mathbf{v}_i \mathbf{v}_i^\intercal = A^\intercal \operatorname{diag}(\mathbf{x}) A$$

where $x_i \in \mathbb{Z}_{\geq 0}$ denotes the number of times experiment i is to be performed. Throughout this paper, we will use both ways of expressing $X(\mathbf{x})$ but will favor the second representation. The inverse of the information matrix is the dispersion matrix:

$$D(\mathbf{x}) = \left(\sum_{i=1}^{m} x_i \mathbf{v}_i \mathbf{v}_i^\intercal \right)^{-1}.$$

It is a measure of the variance of the experiment parameters [1]. Maximizing over the information matrix is equivalent to minimizing over the dispersion matrix [23].

Note the following properties of the information matrix. The matrix $A^\intercal A$ has full rank and is positive definite. Because of the non-negativity of $\mathbf{x}$, the matrix $X(\mathbf{x})$ is in the PSD cone. In particular, $X(\mathbf{x})$ is positive definite for $\mathbf{x} \in \mathbb{Z}^m_{\geq 0}$ if the non-zero entries of $\mathbf{x}$ correspond to at least n linearly independent columns of A. Observe that the dispersion matrix only exists if $X(\mathbf{x})$ is positive definite, i.e. has full rank. To solve the regression problem (1), the information matrix $X(\mathbf{x}^*)$ corresponding to the optimal solution $\mathbf{x}^*$ of OEDP has to lie in the PD cone.

▶ Remark 1. Experiments can be run only once or be allowed to run multiple times to account for errors. In the latter case, we will suppose upper (and lower bounds) on the number of times a given experiment can be run. The sum of the upper bounds usually greatly exceeds N. If non-trivial lower bounds $\mathbf{l}$ are present, their sum may not exceed N otherwise there is no solution respecting the time and cost constraints.

Now we answer the question posed earlier: How can we measure information? We need a function ϕ receiving a positive definite matrix as input and returning a number, that is $\phi : \mathbb{S}^n_{++} \to \mathbb{R}$. We will lose information by compressing a matrix to a single number. Hence, the suitable choice of ϕ depends on the underlying problem. Nevertheless, there are some properties that any ϕ has to satisfy to qualify as an information measure.

▶ **Definition 2** (Information Function [23]). *An information function ϕ on $\mathbb{S}^n_{++}$ is a function $\phi : \mathbb{S}^n_{++} \to \mathbb{R}$ that is positively homogeneous, concave, nonnegative, non-constant, upper semi-continuous and respects the Loewner ordering.*

The Loewner Ordering is an ordering on the PSD cone. Let $D, B \in \mathbb{S}^n_+$, then:

$D \succcurlyeq B$ if and only if $D - B \in \mathbb{S}^n_+$.

Respecting the Loewner ordering together with concavity ensures that the intuition that running more experiments should not result in less information is upheld. Scaling should not change differences in information hence we require positive homogeneity. A constant function is not useful in this setting and non-negativity is a convention. Upper semi-continuity ensures that there are no sudden jumps around the optimal objective value of the optimization problem.
The *Optimal Experiment Design Problem* can then be defined as:

$$\begin{aligned} \max_{\mathbf{x}} \ & \log(\phi(X(\mathbf{x}))) \\ \text{s.t.} \ & \sum_{i=1}^{m} x_i = N \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \\ & \mathbf{x} \in \mathbb{Z}^m, \end{aligned} \tag{OEDP}$$

where $\mathbf{u}$ and $\mathbf{l}$ denote the upper and lower bounds, respectively.

▶ Remark 3. The log does not change the ordering and thus, is not strictly necessary. It can help reformulate the objective function, though.

A special case of non-trivial lower bounds is obtained if n linearly independent experiments have non-zero lower bounds. These experiments can be summarized in the matrix $C = A^\intercal \operatorname{diag}(\mathbf{l}) A$. Notice that C is positive definite. The information matrix then becomes:

$$X_C(\mathbf{x}) = C + A^\intercal \operatorname{diag}(\mathbf{x} - \mathbf{l}) A.$$

▶ **Definition 4** (Optimal Problem and Fusion Problem)**.** *In case the lower bounds are all zero, we call the resulting problem the* ***Optimal Problem*** *during this paper. Replacing the information matrix $X(\mathbf{x})$ with the fusion information matrix $X_C(\mathbf{x})$ in the objective in* (OEDP)*, yields the so-called* ***Fusion Problem****.*

The resulting optimization problem is an integer non-linear problem which, depending on the information function ϕ, can be $\mathcal{NP}$-hard. The two information measures we will focus on lead to $\mathcal{NP}$-hard problems [2]. For convenience in the latter section, we define

$$\mathcal{P} := [\mathbf{l}, \mathbf{u}] \cap \left\{ \mathbf{x} \in \mathbb{R}^m_{\geq 0} \;\middle|\; \sum_{i=1}^{m} x_i = N \right\} \tag{Convex Hull}$$

as the feasible region of the OEDPs without the integer constraints, i.e. the convex hull of all feasible integer points. Thus, the feasible region with integer constraints will be noted by $\mathcal{P} \cap \mathbb{Z}^m$.

2.1 The A-Optimal Problem and D-Optimal Problem

The most frequently used information functions arise from the matrix means ϕ_p [23, 1].

▶ **Definition 5** (Matrix Mean)**.** *Let $B \in \mathbb{S}^n_+$ and let $\lambda(B)$ denote its eigenvalues. The* matrix mean *ϕ_p of B is defined as*

$$\phi_p(B) = \begin{cases} \lambda_{\max}(B), & \text{for } p = \infty, \\ \left(\frac{1}{n}\operatorname{Tr}(B^p)\right)^{\frac{1}{p}}, & \text{for } p \neq 0, \pm\infty, \\ \det(B)^{\frac{1}{n}}, & \text{for } p = 0, \\ \lambda_{\min}(B), & \text{for } p = -\infty, \\ 0 & \text{for } p = [-\infty, 0] \text{ and } B \text{ singular.} \end{cases} \tag{2}$$

Note that the matrix means function ϕ_p satisfies the requirements of Definition 2 only for $p \leq 1$ [23]. The two most commonly used criteria arising from matrix means are the D-optimality and A-optimality criteria, corresponding to $p = 0$ and $p = -1$, respectively. We convert the problems to a minimization form from this point on for homogeneity with the convention of the used solution methods.

2.1.1 D-Criterion

Choosing $p = 0$ and noting that $\log\left(\det(X)^{\frac{1}{n}}\right) = \frac{1}{n}\log\det X$, yields

$$\begin{aligned} &\min_{\mathbf{x}} \; -\log\det(X(\mathbf{x})) \\ &\text{s.t.} \quad \mathbf{x} \in \mathcal{P} \cap \mathbb{Z}^m \end{aligned} \tag{D-Opt}$$

as the *D-Optimal Experiment Design Problem (D-Opt)*. Observe that the objective is equivalent to minimizing the determinant of the dispersion matrix. This determinant is also called *the generalized variance* of the parameter $\boldsymbol{\theta}$ [23]. A maximal value of $\det X$ corresponds to a minimal volume of standard ellipsoidal confidence region of $\boldsymbol{\theta}$ [22]. Additionally, the D-criterion is invariant under reparameterization, see [23].

2.1.2 A-Criterion

For parameters with a physical interpretation, the A-optimality criterion is a good choice as it amounts to minimizing the average of the variances of $\boldsymbol{\theta}$ [23]. Using the log rules for the objective, the *A-Optimal Experiment Design Problem* (A-Opt) can be stated as

$$\begin{aligned} &\min_{\mathbf{x}} \ \mathrm{Tr}\left((X(\mathbf{x}))^{-1}\right) \\ &\text{s.t.} \ \ \mathbf{x} \in \mathcal{P} \cap \mathbb{Z}^m. \end{aligned} \tag{A-Opt}$$

Note that the D-Optimal Problem is known to be $\mathcal{NP}$-hard since the eighties [27]. $\mathcal{NP}$-hardness of the A-Optimal Problem and the D-Fusion Problem was proved only recently, see[21] and [22], respectively. The hardness of the A-Fusion Problem is still open though the authors conjecture that it too is $\mathcal{NP}$-hard.

3 Convergence guarantees for the continuous subproblems

To apply the new framework Boscia, we need to guarantee that the Frank-Wolfe algorithm converges on the continuous subproblems. The conventional property guaranteeing convergence is L-smoothness of the objective, that is

$$\exists L \in \mathbb{R}_{>0} \text{ such that } \ \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\| \ \ \forall \mathbf{x}, \mathbf{y} \in \mathcal{P}.$$

In the case of both the D-Optimal Problem and the A-Optimal Problem, not every point $\mathbf{x} \in \mathcal{P}$ is domain-feasible for the objective functions due to the corresponding information matrix being singular. For such points $\mathbf{x}$, we define $-\log\det(X(\mathbf{x})) = \infty$ and $\mathrm{Tr}\left((X(\mathbf{x}))^{-1}\right) = \infty$ for the D-criterion and A-criterion, respectively. Thus, the objective functions of (A-Opt) and (D-Opt) are not L-smooth over the feasible region.

Lacking L-smoothness on the feasible region, we require a different property that guarantees convergence of the Frank-Wolfe algorithm for the Optimal Problems under both criteria. By [7], the Frank-Wolfe algorithm also converges (with similar convergence rates) if the objective is *generalized self-concordant*.

▶ **Definition 6** (Generalized Self-Concordance [26]). *A three-times differentiable, convex function $f : \mathbb{R}^n \to \mathbb{R}$ is (M_f, ν)-generalized self-concordant with order $\nu > 0$ and constant $M_f \geq 0$, if for all $x \in dom(f)$ and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we have*

$$\left|\left\langle \nabla^3 f(\mathbf{x})[\mathbf{u}]\mathbf{v}, \mathbf{v}\right\rangle\right| \leq M_f \|\mathbf{u}\|_x^2 \, \|\mathbf{v}\|_x^{\nu-2} \, \|\mathbf{v}\|_2^{3-\nu}$$

where

$$\|\mathbf{w}\|_x = \left\langle \nabla^2 f(\mathbf{x})\mathbf{w}, \mathbf{w}\right\rangle \ \text{ and } \nabla^3 f(\mathbf{x})[\mathbf{u}] = \lim_{\gamma \to 0} \gamma^{-1} \left(\nabla^2 f(\mathbf{x} + \gamma\mathbf{u}) - \nabla^2 f(\mathbf{x})\right).$$

Self-concordance is a special case of generalized self-concordance where $\nu = 3$ and $\mathbf{u} = \mathbf{v}$. This yields the condition

$$\left|\left\langle \nabla^3 f(\mathbf{x})[\mathbf{u}]\mathbf{u}, \mathbf{u}\right\rangle\right| \leq M_f \|\mathbf{u}\|_x^3.$$

For a univariate, three times differentiable function $f : \mathbb{R} \to \mathbb{R}$, (M_f, ν)-generalized self-concordance[1] *condition is*

$$|f'''(x)| \leq M_f f''(x)^{\frac{\nu}{2}}.$$

[1] For self-concordance, $\nu = 3$.

By [26, Proposition 2], the composition of a generalized self-concordant function with a linear map is still generalized self-concordant. Hence, it suffices that we show that the functions $f(X) = -\log\det(X)$ and $g(X) = \operatorname{Tr}(X^{-p})$, $p > 0$, are generalized self-concordant for X in the PD cone. Note that f is the logarithmic barrier for the PSD cone which is known to be self-concordant [20]. Convergence of the Frank-Wolfe algorithm for the (D-Opt) Problem is therefore guaranteed.

For the (A-Opt) Problem, we can show that g is self-concordant on a part of the PD cone, namely one characterized by an upper bound on the maximum eigenvalue.

▶ **Theorem 7.** *The function $g(X) = \operatorname{Tr}(X^{-p})$, with $p > 0$, is $\left(3, \frac{(p+2)\sqrt[4]{a^{2p}n}}{\sqrt{p(p+1)}}\right)$-generalized self-concordant on $\operatorname{dom}(g) = \left\{X \in \mathbb{S}^n_{++} \mid 0 \prec X \preccurlyeq aI\right\}$ where $a \in \mathbb{R}_{>0}$ bounds the maximum eigenvalue of X.*

The proof of Theorem 7 and the proof that the maximum eigenvalue of the information matrix $X(\mathbf{x})$ has indeed an upper bound can be found in Appendix A. Thus, we have established convergence for both the D-Optimal Problem and the A-Optimal Problem. The objectives of the A-Fusion Problem and D-Fusion Problem are generalized self-concordant by the argument. Note that the objectives for the Fusion Problems are also L-smooth due to their information matrix always being positive definite.

4 Solution Methods

The main goal of this paper is to propose a new solution method for the Optimal and Fusion Problems under the A-criterion and the D-criterion based on the novel framework Boscia and assess its performance compared to several other convex MINLP approaches. In the following, we introduce the chosen MINLP solvers and state the necessary conditions and possible reformulations that are needed. We have chosen MINLP solution approaches which require relatively few changes to the formulations in (D-Opt) and (A-Opt). Hence, we are not investigating, for example, second-order cone formulations like in [24] in the scope of this paper.

Branch-and-Bound with Frank-Wolfe methods (Boscia)

The new framework introduced in [14] is implemented in the `Julia` package `Boscia.jl`. It is a Branch-and-Bound (BnB) framework that utilizes Frank-Wolfe methods to solve the relaxations at the node level. The Frank-Wolfe algorithm [13, 5], also called Conditional Gradient algorithm [18], and its variants are first-order methods solving problems of the type:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

where f is a convex, Lipschitz-smooth function and $\mathcal{X}$ is a compact convex set. These methods are especially useful if the linear minimization problem over $\mathcal{X}$ can be solved efficiently. The Frank-Wolfe methods used in Boscia are implemented in the `Julia` package `FrankWolfe.jl`, see [3].

At each iteration t, the Frank-Wolfe algorithm solves the linear minimization problem over $\mathcal{X}$ taking the current gradient as the linear objective, resulting in a vertex v_t of $\mathcal{X}$. The next iterate x_{t+1} is computed as a convex combination of the current iterate x_t and the vertex v_t. Many Frank-Wolfe variants explicitly store the vertex decomposition of the iterate, henceforth called the *active set*. We utilize the active set representation to facilitate warm starts in Boscia by splitting the active set when branching.

One novel aspect of Boscia is its use of a *Bounded Mixed-Integer Linear Minimization Oracle (BLMO)* as the *Linear Minimization Oracle (LMO)* in the Frank-Wolfe algorithm. The BLMO solves the mixed-integer linear problem over the feasible region $\mathcal{X}$ with additional bound constraints. Typically, the BLMO is a MIP solver but it can also be a combinatorial solver. This leads to more expensive node evaluations but has the benefit that feasible integer points are found from the root node and the feasible region is much tighter than the continuous relaxation for many problems. In addition, Frank-Wolfe methods can be lazified, i.e. calling the LMO at each iteration in the node evaluation can be avoided, see [6].

In the case of OEDP, strong lazification is not necessary since the corresponding BLMO is very simple. The feasible region $\mathcal{P}$ is just the scaled probability simplex intersected with integer bounds, which is amenable to efficient linear optimization. Given a linear objective d, we first assign $\mathbf{x} = \mathbf{l}$ to ensure that the lower bound constraints are met. Next, we traverse the objective entries in increasing order, adding to the corresponding variable the value of $\max\{u - l, N - \sum(\mathbf{x})\}$. This way, we ensure that both the upper-bound constraints and the knapsack constraint are satisfied. The BLMO over the feasible set can also be cast as a simple network flow problem with m input nodes connected to a single output node, which must receive a flow of N while the edges respect the lower and upper bounds.

Due to the convexity of the objective, the difference $\langle \nabla f(x_t), x_t - v_t \rangle$ is upper bounding the primal gap $f(x_t) - f(x^*)$ at each iteration. We call the quantity the dual gap (or the Frank-Wolfe gap). The dual gap can therefore be used as a stopping criterion. The error adaptiveness of the Frank-Wolfe algorithm can be exploited to a) solve nodes with smaller depth with a coarser precision and b) dynamically stop a node evaluation if the lower bound on the node solution exceeds the objective value at the incumbent.

Observe that in contrast to the epigraph-based formulation approaches that generate many hyperplanes, our method works with the equivalent of a single supporting hyperplane given by the current gradient and moves this hyperplane until it achieves optimality. It is known that once the optimal solution is found, a single supporting hyperplane can be sufficient to prove optimality (as described e.g. for generalized Benders decomposition in Ë[25]). Finding this final hyperplane, however, may require adding many cuts beforehand at suboptimal iterations. In the case of the problems discussed in this paper, the constraint polytope is uni-modular. Adding hyperplanes created from the gradient will not maintain this structure and in consequence, yields a numerically more challenging MIP. Our approach, on the other hand, keeps the polytope and thereby its uni-modularity intact.

Outer Approximation (SCIP+OA)

Outer Approximation schemes are a popular and well-established way of solving MINLPs [17]. This approach requires an epigraph formulation of (OEDP):

$$\begin{aligned} \min_{t,\mathbf{x}} \; & t \\ \text{s.t.} \; & t \geq -\log\left(\phi_p(\mathbf{x})\right) \\ & \mathbf{x} \in \mathcal{P} \\ & t \in \mathbb{R}, \mathbf{x} \in \mathbb{Z}^m. \end{aligned} \tag{E-OEDP}$$

This approach approximates the feasible region of (E-OEDP) with linear cuts derived from the gradient of the non-linear constraints, in our case ∇f. Note that this requires the information matrix $X(\mathbf{x})$ at the current iterate $\mathbf{x}$ to be positive definite, otherwise an evaluation of the gradient is not possible or rather it will evaluate to ∞. The implementation is done with the

`Julia` wrapper of `SCIP`, [4]. Observe that generating cuts that prohibit points leading to singular $X(\mathbf{x})$, we will refer to them as *domain cuts*, is non-trivial. Thus, this approach can only be used for the Fusion Problem where the corresponding information matrix is always positive definite.

LP/NLP Branch-and-Bound (Pajarito)

Another Outer Approximation approach, as implemented in `Pajarito.jl` [11], represents the non-linearities as conic constraints. This is particularly convenient in combination with the conic interior point solver `Hypatia.jl` [10] as it implements the log det cone (the epigraph of the perspective function of log det) directly:

$$\mathcal{K}_{\log\det} := \operatorname{cl}\left\{(u, v, W) \in \mathbb{R} \times \mathbb{R}_{>0} \times \mathbb{S}^n_{++} \mid u \leq v \log\det(W/v)\right\}.$$

The formulation of (D-Opt) then becomes

$$\begin{aligned}
\max_{t,\mathbf{x}}\ & t \\
\text{s.t. }\ & (t, 1, X(x)) \in \mathcal{K}_{\log\det} \\
& \mathbf{x} \in \mathcal{P} \\
& t \in \mathbb{R}, \mathbf{x} \in \mathbb{Z}^m.
\end{aligned}$$

For the representation of the trace inverse, we utilize the dual of the separable spectral function cone [9, Section 6]:

$$\mathcal{K}_{\text{sepspec}} := \operatorname{cl}\left\{(u, v, w) \in \mathbb{R} \times \mathbb{R}_{>0} \times \operatorname{int}(\mathcal{Q}) \mid u \geq v\varphi(w/v)\right\}.$$

For our purposes, $\mathcal{Q}$ is the PSD cone and the spectral function φ is the negative square root whose convex conjugate is precisely the trace inverse, see [8, Table 1][2]:

$$\mathcal{K}^*_{\text{sepspec}} := \operatorname{cl}\left\{(u, v, w) \in \mathbb{R} \times \mathbb{R}_{>0} \times \operatorname{int}(\mathbb{S}^n_+) \mid v \geq u/4 \operatorname{Tr}((w/u)^{-1})\right\}.$$

The conic formulation of (A-Opt) is therefore

$$\begin{aligned}
\min_{t,\mathbf{x}}\ & 4t \\
\text{s.t. }\ & (1, t, X(x)) \in \mathcal{K}^*_{\text{sepspec}} \\
& \mathbf{x} \in \mathcal{P} \\
& t \in \mathbb{R}, \mathbf{x} \in \mathbb{Z}^m.
\end{aligned}$$

Additionally, the conic formulation allows for the computation of domain cuts for $\mathbf{x}$. Hence, this solver can be used on all problems. Note that we use `HiGHS` [15] as a MIP solver within Pajarito.

A Custom Branch-and-Bound for OEDP (Co-BnB)

The most general solver strategy for (OEDP) with matrix means criteria was introduced in [2]. Like Boscia , it is a Branch-and-Bound-based approach with a first-order method to solve the node problems. The first-order method in question is a coordinate-descent-like

[2] For further details, see this discussion [16].

algorithm. We refer to this approach as Co-BnB in the rest of this paper. As the termination criterion, this method exploits the fact that the objective function is a matrix mean and shows the connection of the resulting optimization problem:

$$\begin{aligned} \max_{\mathbf{w}} \ & \log\left(\phi_p(X(\mathbf{w}))\right) \\ \text{s.t.} \ & \sum_{i=1}^{m} w_i = 1 \\ & \mathbf{w} \geq 0 \\ & w_i N \in \mathbb{Z} \, \forall i \in [m] \end{aligned} \tag{M-OEDP}$$

to the generalization of the Minimum Volume Enclosing Ellipsoid Problem (MVEP) [2]. The variables w can be interpreted as a probability distribution and the number of times the experiments are to be run is denoted by wN. Concerning (OEDP), one can say $x = wN$. In the case of the Fusion Problem, the already completed experiments have to be explicitly added as variables with fixed values.

Note that we have improved and adapted the step size rules within the first-order method, see the arvix version linked on the front page. Further, note that the solver was developed for instances with a plethora of experiments and very few parameters. The solver employs the simplest Branch-and-Bound strategy, i.e. utilizing the most fractional branching rule and traversing the tree using the minimum lower bound. In the next section, we will see that the method works well in cases where n is small but struggles if n increases.

5 Computational Experiments

In this section, we present the computational experiments for the Optimal Problem and Fusion Problem, both under the A- and D-criterion, respectively. The resulting problems will be referred to as the *A-Fusion Problem (AF)*, *D-Fusion Problem (DF)*, *A-Optimal Problem (AO)*, and *D-Optimal Problem (DO)*.

Experimental Setup

For the instance generation, we choose the number of experiments $m \in \{50, 60, 80, 100, 120\}$, the number of parameters $n \in \{\lfloor m/4 \rfloor, \lfloor m/10 \rfloor\}$, and the number of allowed experiments $N = \lfloor 1.5n \rfloor$ for the Optimal Problems and $N \in [m/20, m/3]$ for the Fusion Problems. The lower bounds are zero. Note that for the Fusion Problems, the fixed experiments are encoded in a separate matrix. The upper bounds are randomly sampled between 1 and $N/3$ for (A-Opt) and (D-Opt). In the Fusion case, they are sampled between 1 and $m/10$. We generate both independent and correlated data. Also, note that the matrices generated are dense. The number of random seeds is 5. In total, there are 50 instances for each combination of problem and data set.

Experiments were run on a cluster equipped with Intel Xeon Gold 6338 CPUs running at 2 GHz and a one-hour time limit.

Start Solution

Note that both the objectives (D-Opt) and (A-Opt) are only well defined if the information matrix $X(\mathbf{x})$ has full rank. This is the case for the Fusion Problem, not necessarily for the Optimal Problem. Both Boscia and Co-BnB require a feasible starting point $\mathbf{z}_0$. For its construction, we find a set $S \subset [m]$ of n linearly independent experiments, i.e. n linearly

independent rows of A. Assign those experiments their upper bound. If the sum $\langle 1, z_0 \rangle$ exceeds N, remove 1 from the experiment with the largest entry. If the sum is less than N, pick an experiment in $[m] \backslash S$ at random and assign it as many runs as possible. Repeat until the sum $\langle 1, z_0 \rangle$ is equal to N. Note that due to the monotonic progress of both first-order methods, the current iterate will never become domain infeasible, i.e. $X(\mathbf{x})$ will not become singular. If the iterate is domain infeasible after branching, we discard that node.

Results

An overview of the results of the computational experiments is given in Table 1. The new framework Boscia solves the most instances by far. In comparison to the Outer Approximation methods, it terminates nearly twice as often. The Co-BnB fares better. Nonetheless, it solves fewer instances to optimality than Boscia. The notable exception is the A-Fusion Problem with correlated data. Note that in general, it fares well for the instances where $n = \lfloor m/10 \rfloor$ as it was designed for such problems. It struggles for the instances where $n = \lfloor m/4 \rfloor$.

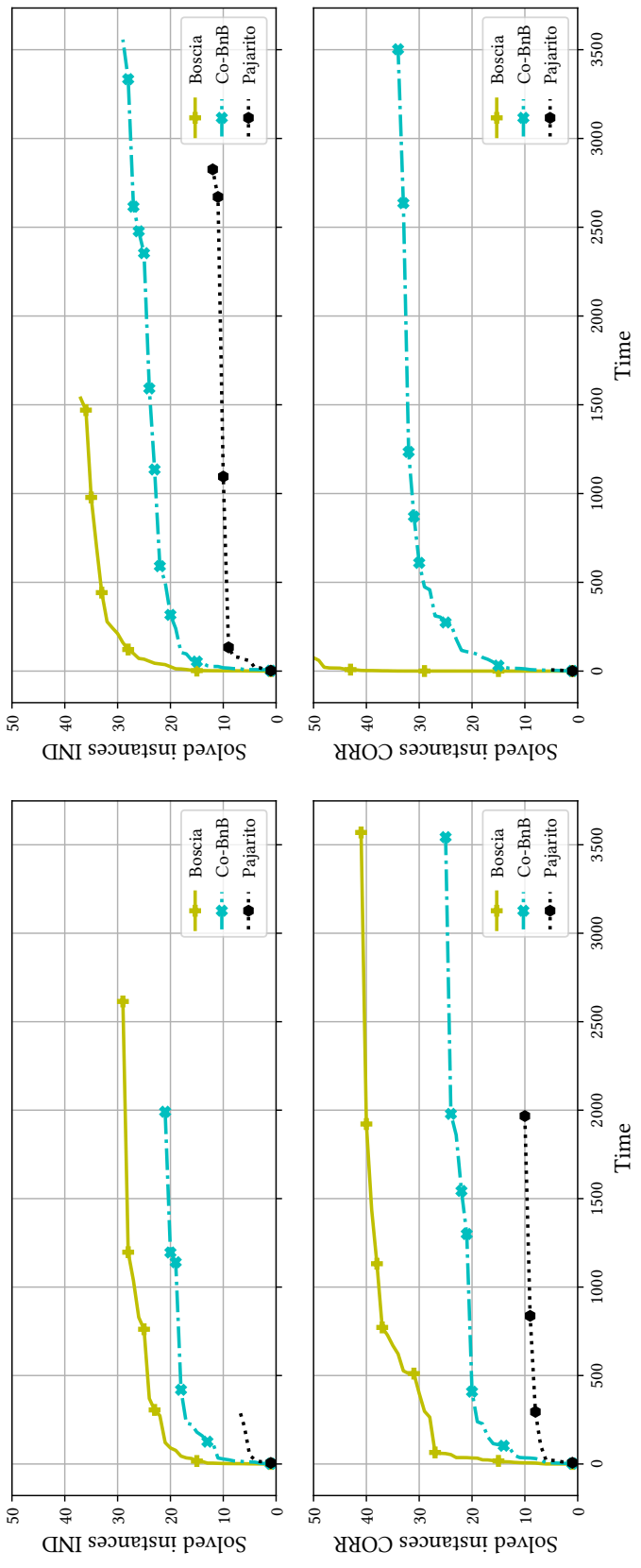
In terms of time, Boscia also shows promising results, especially for instances of larger scale. For small-scale instances, the Outer Approximation approaches are fast, in some cases faster than Boscia. The graphical view of the number of solved instances over time is shown in Figure 1 for the Optimal Problems and in Figure 2 for the Fusion Problems.

Notice that there is a sharp increase in the beginning, especially for Co-BnB and Boscia. In contrast to Boscia, the curves for Co-BnB flatten more after the first 10 seconds. The notable exception is the A-Fusion Problem with correlated data for which the Co-BnB method works consistently better than boscia, see also Table 1. The better performance of Co-BnB can be explained by the different objectives used. For the A-criterion, Co-BnB uses $\log\left(\mathrm{Tr}(X^{-1})\right)$ instead of $\mathrm{Tr}(X^{-1})$ which both Boscia and SCIP utilize. Computing the Optimal Problems and Fusion Problems using the logarithm formulation with both Boscia and SCIP yields twice as good a performance on the A-Fusion problem with correlated data, see Table 2. Observe that the performance of Boscia on the Optimal Problems decreases when $\log\left(\mathrm{Tr}(X^{-1})\right)$ is used as the objective. It should be noted that whether generalized self-concordance holds for the logarithm of the trace inverse (potentially limited to a subset of its domain) is still an open question.

As previously stated, the Co-BnB framework was developed for the case where the number of parameters n, and consequently the number of allowed experiment N, is significantly smaller than the number of experiments m. There are no advanced Branch-and-Bound algorithmic strategies proposed, such as improved traverse strategies or branching decisions. A greater value of N naturally increases the size of the tree and the number of nodes to be processed. Boscia has the advantage here since it finds many integer feasible points while solving the relaxations which have the potential to improve the incumbent. A better incumbent, in turn, lets us prune non-improving nodes early on.

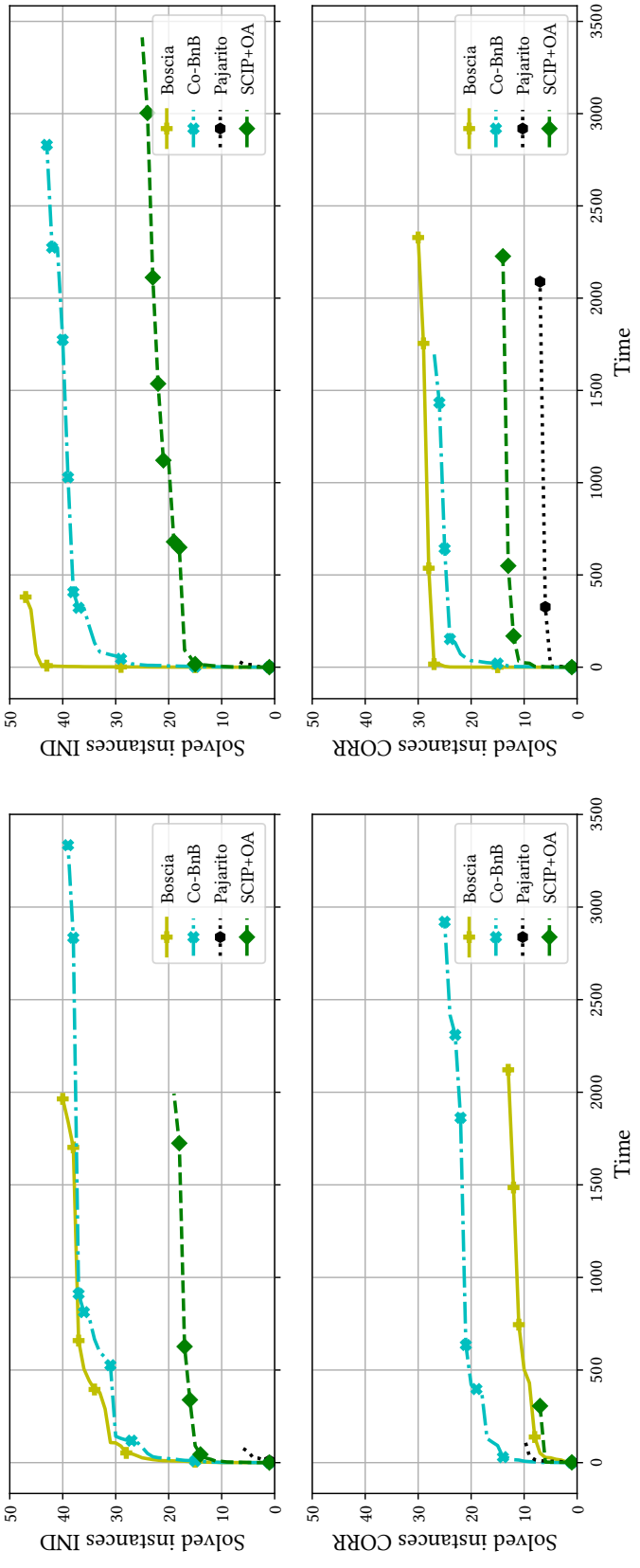
Observe that the curves for the two Outer Approximation approaches also flatten out. In addition, their increase at the start is not as sharp as for the two Branch-and-Bound approaches. Keep in mind that while the relaxations of the two Branch-and-Bound approaches keep the simple feasible region intact, the Outer Approximation methods add many additional constraints, i.e. the cuts. This results in larger MIPs to be solved. Furthermore, these cuts are dense leading to further difficulty for the MIP solvers.

Aside from the performance comparison of the solvers, we investigate how the problems themselves compare to each other and if the difficulty of the instances is solver-dependent or if there are clear trends.



(a) A-Optimal Problem. **(b)** D-Optimal Problem.

Figure 1 The accumulated number of instances solved to optimality over time for the A-Optimal Problem and D-Optimal Problem with both data sets. The upper plot shows the number of instances solved on the problem with independent data and the bottom the number of solved instances on the correlated data.



(a) A-Fusion Problem.

(b) D-Fusion Problem.

Figure 2 The accumulated number of instances solved to optimality over time for the A-Fusion Problem and D-Fusion Problem with both data sets. The upper plot shows the number of instances solved on the problem with independent data and the bottom the number of solved instances on the correlated data.

Table 1 Comparing the performance of Boscia, Pajarito, SCIP and Co-BnB on the different problems and the different data sets, i.e. A-Fusion (AF), D-Fusion (DF), A-Optimal (AO) and D-Optimal (DO). One data set contains independent data, the other has correlated data.
The instances for each problem are split into increasingly smaller subsets depending on their minimum solve time, i.e. the minimum time any of the solvers took to solve it. The cut-offs are at 0 seconds (all problems), took at least 10 seconds to solve, 100 s, 1000 s and lastly 2000 s. Note that if none of the solvers terminates on any instance of a subset, the corresponding row is omitted from the table. The average time is taken using the geometric mean shifted by 1 second. Also, note that this is the average time over all instances in that group, i.e. it includes the time outs.

				Boscia		Co-BnB		Pajarito		SCIP+OA	
Type	Corr.	Solved after (s)	# inst.	% solved	Time (s)	% solved	Time (s)	% solved	Time (s)	% solved	Time (s)
AO	no	0	50	**58 %**	**208.53**	42 %	640.68	14 %	1901.7		
		10	37	**43 %**	**973.13**	22 %	2284.39	0 %	3600.05		
		100	30	**30 %**	**2084.72**	3 %	3531.73	0 %	3600.07		
		1000	24	**12 %**	**3221.62**	0 %	3602.71	0 %	3600.07		
		2000	22	**5 %**	**3548.1**	0 %	3602.94	0 %	3600.07		
AO	yes	0	50	**82 %**	**98.5**	50 %	541.22	20 %	1591.74		
		10	37	**76 %**	**330.67**	41 %	1145.77	11 %	2777.96		
		100	23	**61 %**	**1359.18**	9 %	3441.78	0 %	3600.05		
		1000	13	**31 %**	**2922.48**	0 %	3600.16	0 %	3600.02		
		2000	10	**10 %**	**3596.95**	0 %	3600.19	0 %	3600.02		
AF	no	0	50	**80 %**	**54.82**	78 %	82.96	12 %	2006.81	38 %	464.82
		10	31	**68 %**	**365.69**	65 %	543.73	0 %	3600.03	6 %	3153.28
		100	21	**52 %**	**1277.13**	48 %	1596.99	0 %	3600.04	0 %	3600.07
		1000	13	**23 %**	**3081.0**	15 %	3514.57	0 %	3600.05	0 %	3600.1
		2000	10	0 %	3600.13	**10 %**	**3573.82**	0 %	3600.06	0 %	3600.12
AF	yes	0	50	26 %	1359.35	**50 %**	**370.57**	20 %	1132.66	14 %	1471.59
		10	39	13 %	2684.24	**36 %**	**1421.25**	0 %	3600.06	3 %	3379.67
		100	35	9 %	3123.03	**29 %**	**2225.95**	0 %	3600.07	0 %	3600.03
		1000	29	0 %	3600.06	**14 %**	**3393.85**	0 %	3600.08	0 %	3600.04
		2000	28	0 %	3600.06	**11 %**	**3467.38**	0 %	3600.08	0 %	3600.04
DF	no	0	50	**94 %**	**3.32**	86 %	38.28	14 %	1576.5	50 %	333.25
		10	6	**50 %**	**858.17**	17 %	3336.0	0 %	3613.96	0 %	3600.21
		100	5	**40 %**	**1408.69**	0 %	3602.04	0 %	3614.84	0 %	3600.23
DF	yes	0	50	**60 %**	**50.68**	54 %	185.07	14 %	1761.18	28 %	753.56
		10	24	**17 %**	**2534.26**	8 %	2944.52	0 %	3605.93	0 %	3600.05
		100	23	**13 %**	**3151.98**	4 %	3458.68	0 %	3606.19	0 %	3600.05
		1000	22	**9 %**	**3416.05**	5 %	3452.39	0 %	3606.45	0 %	3600.05
D	no	0	50	**74 %**	**81.07**	58 %	442.28	24 %	732.57		
		10	33	**61 %**	**479.16**	36 %	2199.09	6 %	1182.22		
		100	21	**43 %**	**1471.23**	10 %	3525.97	5 %	1083.49		
		1000	6	**33 %**	**2691.04**	0 %	3603.88	0 %	2735.7		
D	yes	0	50	**100 %**	**1.26**	68 %	223.34	10 %	755.88		
		10	7	**100 %**	**24.66**	0 %	3600.22	0 %	492.01		

It can be observed in Table 1 that most solvers solve fewer instances under the A-criterion. The notable exception is Pajarito on the A-Optimal Problem where it solves more instances to optimality compared to the D-criterion. It should be noted, however, that Pajarito encountered slow progress for many instances under the A-criterion because too many cuts were added or the cuts were too close, i.e. the normal vectors of the hyperplane were too parallel to each other.

Taking a look at some example contour plots shown in Figure 3, we observe that the contour lines for the A-criterion are steeper than those of the D-criterion for both the Optimal Problem and the Fusion Problem, respectively. This points to the condition number increasing.

Table 2 Comparing the performance of Boscia, Pajarito, SCIP and Co-BnB on A-Optimal and A-Fusion problem with both data sets, i.e. independent and correlated data.
In contrast to Table 1, Boscia and SCIP also have $\log\left(\mathrm{Tr}(X^{-1})\right)$ as objective function.
The instances for each problem are split into increasingly smaller subsets depending on their minimum solve time, i.e. the minimum time any of the solvers took to solve it. The cut-offs are at 0 seconds (all problems), took at least 10 seconds to solve, 100 s, 1000 s and lastly 2000 s. Note that if none of the solvers terminates on any instance of a subset, the corresponding row is omitted from the table. The average time is taken using the geometric mean shifted by 1 second. Also, note that this is the average time over all instances in that group, i.e. it includes the time outs.

				Boscia		Co-BnB		Pajarito		SCIP+OA	
Type	Corr.	Solved after (s)	# inst.	% solved	Time (s)	% solved	Time (s)	% solved	Time (s)	% solved	Time (s)
AO	no	0	50	**56 %**	**325.09**	42 %	640.68	14 %	1901.7		
		10	40	**45 %**	**952.21**	28 %	1789.44	0 %	3600.05		
		100	33	**33 %**	**2127.44**	12 %	2986.04	0 %	3600.06		
		1000	26	**15 %**	**3202.8**	0 %	3602.5	0 %	3600.06		
		2000	24	**8 %**	**3472.93**	0 %	3602.7	0 %	3600.07		
AO	yes	0	50	**64 %**	**185.28**	50 %	541.22	20 %	1591.74		
		10	38	**53 %**	**613.98**	34 %	1669.08	0 %	3600.05		
		100	29	**38 %**	**1641.98**	14 %	3346.09	0 %	3600.05		
		1000	19	**5 %**	**3533.58**	0 %	3600.11	0 %	3600.04		
AF	no	0	50	**84 %**	**31.27**	78 %	82.96	12 %	2006.81	38 %	477.72
		10	25	**68 %**	**270.78**	56 %	1074.55	0 %	3600.03	0 %	3600.06
		100	14	**43 %**	**1457.98**	21 %	2474.08	0 %	3600.05	0 %	3600.08
		1000	9	**11 %**	**3553.67**	11 %	3570.97	0 %	3600.06	0 %	3600.12
AF	yes	0	50	**52 %**	**207.34**	50 %	370.57	20 %	1132.66	26 %	934.9
		10	33	**27 %**	**1780.79**	24 %	2097.49	0 %	3600.07	3 %	3278.79
		100	29	17 %	3075.17	**21 %**	**2656.55**	0 %	3600.07	0 %	3600.04
		1000	26	8 %	3461.56	**12 %**	**3457.38**	0 %	3600.07	0 %	3600.05
		2000	25	4 %	3545.06	**8 %**	**3513.62**	0 %	3600.07	0 %	3600.05

In terms of the data, one could assume that all problems would be easier to solve with independent data. Noticeably, this is not the case. Rather, it differs for the two problem types. The Fusion Problems are easier with independent data, the Optimal Problems are more often solved with the correlated data. Figures 4a–4d depict the progress of the incumbent, lower bound, and dual gap within Boscia for selected instances of each combination of problem and data type.

Interestingly, the independent data leads to proof of optimality, i.e. the optimal solution is found early on and the lower bound has to close the gap, regardless of the problem, see Figures 4a and 4c. The difference in problems has, however, an impact on how fast the lower bound can catch up with the incumbent. Comparing Figures 4a and 4c, the lower bound curve in Figure 4c gets closer to the incumbent initially, i.e. in the first 2 seconds, and flattens more compared to the lower bound curve in Figure 4a. The reason for this difference is likely that in the case of the Optimal Problem there is a larger region around the optimal solution where the corresponding points/designs $\mathbf{x}$ provide roughly the same information. These other candidates have to be checked to ensure the optimality of the incumbent and thus the solving process slows down. If we can prove either strong convexity or sharpness for the objectives, this can potentially be used to speed up the progression of the lower bound. Currently, Boscia can only utilize strong convexity but the adaptation for sharpness is in development.

On the other hand, the correlated data leads to solution processes that are very incumbent-driven, i.e. most improvement on the dual gap stems from the improving incumbent, not from the lower bound, as seen in Figures 4b and 4d. Incumbent-driven solution processes

can be identified by the dual gap making sudden jumps and the absence of (a lot of) progress between these jumps. As before, the solution process speed depends on the Problem. In Figure 4b, the dual gap makes big jumps throughout most of the solving process, in contrast to the dual gap Figure 4d. For both problems, this indicates that the optimal solution is in the interior. In the case of the Fusion Problem, the lower bound is increasing slower compared to the Optimal Problem in Figure 4b. This points to the existence of many points yielding a similar objective value. The key ingredients for improvement will be the incorporation of more sophisticated primal heuristics and further investigation on how to improve the lower bound progress.

6 Conclusion

We proposed a new approach for the Optimal Experiment Design Problem based on the Boscia framework, and show its performance compared to other MINLP approaches. In addition, it also performs better compared to the approach specifically developed for the OEDP, in particular for large-scale instances and a larger number of parameters. This superiority can be explained by the fact that Boscia keeps the structure of the problem intact and that it utilizes a combinatorial solver to find integer feasible points at each node.

References

1 Selin Damla Ahipaşaoğlu. A first-order algorithm for the A-optimal experimental design problem: a mathematical programming approach. *Statistics and Computing*, 25(6):1113–1127, 2015.

2 Selin Damla Ahipaşaoğlu. A branch-and-bound algorithm for the exact optimal experimental design problem. *Statistics and Computing*, 31(5):65, 2021.

3 Mathieu Besançon, Alejandro Carderera, and Sebastian Pokutta. FrankWolfe.jl: A high-performance and flexible toolbox for Frank–Wolfe algorithms and conditional gradients. *INFORMS Journal on Computing*, 2022.

4 Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, et al. Enabling research through the SCIP Optimization Suite 8.0. *ACM Transactions on Mathematical Software*, 49(2):1–21, 2023.

5 Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. Conditional gradient methods. *arXiv preprint arXiv:2211.14103*, 2022.

6 Gábor Braun, Sebastian Pokutta, and Daniel Zink. Lazifying conditional gradient algorithms. In *International conference on machine learning*, pages 566–575. PMLR, 2017.

7 Alejandro Carderera, Mathieu Besançon, and Sebastian Pokutta. Scalable Frank–Wolfe on generalized self-concordant functions via simple steps. *SIAM Journal on Optimization*, 2024. To appear.

8 Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. Conic optimization with spectral functions on Euclidean Jordan algebras. *Mathematics of Operations Research*, 2022.

9 Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. Performance enhancements for a generic conic interior point algorithm. *Mathematical Programming Computation*, 2022. doi:10.1007/s12532-022-00226-0.

10 Chris Coey, Lea Kapelevich, and Juan Pablo Vielma. Solving natural conic formulations with Hypatia.jl. *INFORMS Journal on Computing*, 34(5):2686–2699, 2022. doi:10.1287/ijoc.2022.1202.

11 Chris Coey, Miles Lubin, and Juan Pablo Vielma. Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation*, 12(2):249–293, 2020.

12 P Fernandes de Aguiar, B Bourguignon, MS Khots, DL Massart, and R Phan-Than-Luu. D-optimal designs. *Chemometrics and intelligent laboratory systems*, 30(2):199–210, 1995.

13 Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

14 Deborah Hendrych, Hannah Troppens, Mathieu Besançon, and Sebastian Pokutta. Convex mixed-integer optimization with Frank–Wolfe methods, 2023. `arXiv:2208.11010`.

15 Qi Huangfu and Julian Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018.

16 Lea Kapelevich. How to optimize trace of matrix inverse with JuMP or Convex? `https://discourse.julialang.org/t/how-to-optimize-trace-of-matrix-inverse-with-jump-or-convex/94167/6`, accessed 4th December 2023, 2023.

17 Jan Kronqvist, David E Bernal, Andreas Lundell, and Ignacio E Grossmann. A review and comparison of solvers for convex MINLP. *Optimization and Engineering*, 20:397–455, 2019.

18 Evgeny S Levitin and Boris T Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

19 Yongchun Li, Marcia Fampa, Jon Lee, Feng Qiu, Weijun Xie, and Rui Yao. D-optimal data fusion: Exact and approximation algorithms. *INFORMS Journal on Computing*, 36(1):97–120, 2024.

20 Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

21 Aleksandar Nikolov, Mohit Singh, and Uthaipon Tantipongpipat. Proportional volume sampling and approximation algorithms for A-optimal design. *Mathematics of Operations Research*, 47(2):847–877, 2022.

22 Gabriel Ponte, Marcia Fampa, and Jon Lee. Branch-and-bound for D-optimality with fast local search and variable-bound tightening. *arXiv preprint arXiv:2302.07386*, 2023.

23 Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

24 Guillaume Sagnol and Radoslav Harman. Computing exact D-optimal designs by mixed integer second-order cone programming. *The Annals of Statistics*, 43(5):2198–2224, 2015. `doi:10.1214/15-AOS1339`.

25 NV Sahinidis and Ignacio E Grossmann. Convergence properties of generalized Benders decomposition. *Computers & Chemical Engineering*, 15(7):481–491, 1991.

26 Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 178(1-2):145–213, 2019.

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

A Proof of Generalized Self-Concordance for the A-criterion

As stated in Section 3, we need to prove that the objective function resulting from the A-criterion is generalized self-concordant to establish convergence of Frank-Wolfe. In fact, we can prove the statement for the *General-Trace-Inverse (GTI)* $g(x) = \operatorname{Tr}(X^{-p})$ for $p > 0$.

First, we need to compute the first three partial derivatives of g. To help us in this endeavor, we prove the following lemma.

▶ **Lemma 8.** *Let $r \in \mathbb{R}$ and let the matrix X be diagonalizable. Then, we can define the function $f(X) = \operatorname{Tr}(X^r)$. The gradient of f is then*

$$\nabla f(x) = rX^{r-1}$$

Proof. We use the definition of X^r to prove the result.

$$\operatorname{Tr}(X^r) = \operatorname{Tr}\left(\sum_{k=0}^{\infty} \frac{r^k}{k!} \log(X)^k\right) = \sum_{k=0}^{\infty} \frac{r^k}{k!} \operatorname{Tr}\left(\log(X)^k\right)$$

Thus, using the sum rule, $(\operatorname{Tr}(X^n))' = nX^{n-1}$ for any positive integer n and $\nabla \log(X) = \nabla X X^{-1}$, we have

$$\begin{aligned}
\nabla \operatorname{Tr}(X^r) &= \sum_{k=1}^{\infty} \frac{r^k}{k!} k \log(X)^{k-1} I X^{-1} \\
&= r\left(\sum_{k=0}^{\infty} \frac{r^k}{k!} \log(X)^k\right) X^{-1} \\
&= r X^r X^{-1} = r X^{r-1}
\end{aligned}$$

This concludes the proof. ◀

Let $A \in \mathbb{S}^n_{++}, B \in \mathbb{S}^n_+$ and $t \in \mathbb{R}$ such that $A + tB \in \mathbb{S}^n_{++}$. We define $h(t) = g(A + tB)$. For the proof of self-concordance, we need the first three derivatives of $h(t)$.

$$h(t) = \operatorname{Tr}((A+tB)^{-p}) = \operatorname{Tr}\left(\left((A+tB)^{-1}\right)^p\right)$$

Using Lemma 8 for the first derivative, yields:

$$\begin{aligned}
h'(t) &= \operatorname{Tr}\left(p\left((A+tB)^{-1}\right)^{p-1}(-1)(A+tB)^{-1}B(A+tB)^{-1}\right) \\
&= -p \operatorname{Tr}\left(\left((A+tB)^{-1}\right)^p B(A+tB)^{-1}\right).
\end{aligned}$$

For the second derivative, we find

$$\begin{aligned}
h''(t) &= -p \operatorname{Tr}\Big(p\left((A+tB)^{-1}\right)^{p-1}(-1)(A+tB)^{-1}B(A+tB)^{-1}B(A+tB)^{-1} \\
&\quad + \left((A+tB)^{-1}\right)^p B(-1)(A+tB)^{-1}B(A+tB)^{-1}\Big) \\
&= p \operatorname{Tr}\left((p+1)\left((A+tB)^{-1}\right)^p B(A+tB)^{-1}B(A+tB)^{-1}\right).
\end{aligned}$$

Restricting h'' to $t = 0$, yields:

$$h''(t)_{\restriction t=0} = p(p+1)\operatorname{Tr}\left(A^{-p}BA^{-1}BA^{-1}\right).$$

Lastly, for the third derivative, we have:

$$h'''(t) = -p(p+1)(p+2)\operatorname{Tr}\left((A+tB)^{-p}B(A+tB)^{-1}B(A+tB)^{-1}B(A+tB)^{-1}\right).$$

And restricted to $t = 0$, yields:

$$h'''(t)_{\restriction t=0} = -p(p+1)(p+2)\operatorname{Tr}\left(A^{-p}BA^{-1}BA^{-1}BA^{-1}\right).$$

Proof of Theorem 7. We define $M(t) = V + tU$ where $V \in \operatorname{dom}(g)$, $U \in \mathbb{S}^n_+$ and $t \in \mathbb{R}$ such that $V + tU \in \mathbb{S}^n_{++}$. Then:

$$h(t) = g(M(t)) = \operatorname{Tr}\left(M(t)^{-p}\right).$$

From the derivative computation above, we have

$$h''(t)_{\restriction t=0} = p(p+1)\operatorname{Tr}\left(V^{-p}UV^{-1}UV^{-1}\right)$$
$$h'''(t)_{\restriction t=0} = -p(p+1)(p+2)\operatorname{Tr}\left(V^{-p}UV^{-1}UV^{-1}UV^{-1}\right).$$

Thus, the condition we want to satisfy is:

$$p(p+1)(p+2)\operatorname{Tr}\left(V^{-p}UV^{-1}UV^{-1}UV^{-1}\right) \leq M_f p(p+1)Tr\left(V^{-p}UV^{-1}UV^{-1}\right)^{\nu/2}.$$

Note that

$$\operatorname{Tr}\left(V^{-p}UV^{-1}UV^{-1}UV^{-1}\right) = \operatorname{Tr}\left(V^{-\frac{p+1}{2}}UV^{-1}UV^{-1}UV^{-\frac{p+1}{2}}\right) \geq 0$$

and

$$\operatorname{Tr}\left(V^{-p}UV^{-1}UV^{-1}\right) = \operatorname{Tr}\left(V^{-\frac{p+1}{2}}UV^{-1}UV^{-\frac{p+1}{2}}\right) \geq 0$$

by positive definiteness of V and semi-definiteness U. If the trace is equal to 0, the statement holds directly. Otherwise, we divide both sides by the LHS and first try to lower bound the following fraction:

$$\frac{\operatorname{Tr}\left(V^{-\frac{p+1}{2}}UV^{-1}UV^{-\frac{p+1}{2}}\right)^{\nu/2}}{\operatorname{Tr}\left(V^{-\frac{p+1}{2}}UV^{-1}UV^{-1}UV^{-\frac{p+1}{2}}\right)}.$$

We want to lower-bound this fraction. To that end, we define $H = V^{-1/2}UV^{-1/2}$ and $L = V^{-p/2}H$, then:

$$\frac{\operatorname{Tr}\left(V^{-\frac{p+1}{2}}UV^{-1}UV^{-\frac{p+1}{2}}\right)^{\nu/2}}{\operatorname{Tr}\left(V^{-\frac{p+1}{2}}UV^{-1}UV^{-1}UV^{-\frac{p+1}{2}}\right)} = \frac{\operatorname{Tr}\left(LL^\intercal\right)^{\nu/2}}{\operatorname{Tr}\left(LHL^\intercal\right)}.$$

Using Cauchy-Schwartz, we find:

$$\begin{aligned}
\operatorname{Tr}\left(LHL^\intercal\right) &= \langle L^\intercal L, H\rangle \\
&\leq \operatorname{Tr}\left(LL^\intercal\right)\sqrt{\operatorname{Tr}\left(H^2\right)} \\
\frac{\operatorname{Tr}\left(LL^\intercal\right)^{\nu/2}}{\operatorname{Tr}\left(LHL^\intercal\right)} &\geq \frac{\operatorname{Tr}\left(LL^\intercal\right)^{\nu/2}}{\operatorname{Tr}\left(LL^\intercal\right)\sqrt{\operatorname{Tr}\left(H^2\right)}} \\
\operatorname{Tr}\left(HH\right) &= \operatorname{Tr}\left(V^{p/2}V^{-p/2}HHV^{-p/2}V^{p/2}\right) \\
&= \operatorname{Tr}\left(V^{p/2}LL^\intercal V^{p/2}\right) \\
&\leq \operatorname{Tr}\left(LL^\intercal\right)\sqrt{\operatorname{Tr}\left(V^{2p}\right)} \\
\frac{\operatorname{Tr}\left(LL^\intercal\right)^{\nu/2}}{\operatorname{Tr}\left(LL^\intercal\right)\sqrt{\operatorname{Tr}\left(H^2\right)}} &\geq \frac{\operatorname{Tr}\left(LL^\intercal\right)^{\nu/2}}{\operatorname{Tr}\left(LL^\intercal\right)\sqrt{\operatorname{Tr}\left(LL^\intercal\right)\sqrt{\operatorname{Tr}\left(V^{2p}\right)}}} \\
&= \operatorname{Tr}\left(LL^\intercal\right)^{\nu/2-3/2}\frac{1}{\sqrt{\|V^p\|_F}} \\
&\geq \operatorname{Tr}\left(LL^\intercal\right)^{\nu/2-3/2}\frac{1}{\sqrt{\|V^p\|_2}\sqrt[4]{n}}.
\end{aligned}$$

By the assumption that V is a feasible point, we have $\|V\|_2 = \lambda_{\max}(V) \leq a$, thus:

$$\frac{\operatorname{Tr}(LL^\intercal)^{\nu/2}}{\operatorname{Tr}(LL^\intercal)\sqrt{\operatorname{Tr}(H^2)}} \geq \operatorname{Tr}(LL^\intercal)^{\nu/2-3/2}\frac{1}{\sqrt{a^p}\sqrt[4]{n}}.$$

Then,

$$\frac{M_f(p(p+1))^{\nu/2}}{p(p+1)(p+2)}\frac{\operatorname{Tr}\left(V^{-p}UV^{-1}UV^{-1}\right)^{\nu-2}}{\operatorname{Tr}(V^{-p}UV^{-1}UV^{-1}UV^{-1})} \geq \frac{M_f(p(p+1))^{\nu/2}}{p(p+1)(p+2)}\operatorname{Tr}(LL^\intercal)^{\nu/2-3/2}\frac{1}{\sqrt{a^p}\sqrt[4]{n}}.$$

We can select $\nu = 3$ and re-express the GSC condition as

$$1 \leq \frac{M_f(p(p+1))^{1/2}}{(p+2)}\frac{1}{\sqrt[4]{a^{2p}n}}.$$
$$M_f \geq \frac{(p+2)\sqrt[4]{a^{2p}n}}{\sqrt{p(p+1)}}.$$

Thus, g is $\left(3, \frac{(p+2)\sqrt[4]{a^{2p}n}}{\sqrt{p(p+1)}}\right)$-generalized self-concordant on the set $\left\{X \in \mathbb{S}^n_{++} \mid 0 \prec X \preccurlyeq aI\right\}$. ◀

▶ **Corollary 9.** *The function $g(X) = \operatorname{Tr}\left(X^{-1}\right)$ is $\left(3, \frac{3\sqrt[4]{a^2n}}{\sqrt{2}}\right)$-generalized self-concordant on $\operatorname{dom}(g) = \left\{X \in \mathbb{S}^n_{++} \mid 0 \prec X \preccurlyeq aI\right\}$.*

To ensure that Theorem 7 is applicable for the objective of (A-Opt), we show that the maximum eigenvalue of the information matrix is bounded.

▶ **Lemma 10.** *Let $x \in \mathcal{P}$. Then,*

$$\lambda_{\max}(X(\mathbf{x})) \leq m \max_{j\in[m]} u_j \max_{i\in[m]} \|\mathbf{v}_i\|_2^2.$$

where $\mathbf{u}$ denotes the upper bounds and $\mathbf{v}_i$ are the rows of the experiment matrix A.

Proof.

$$\lambda_{\max}(X(\mathbf{x})) = \lambda_{\max}\left(\sum_{i=1}^m x_i\mathbf{v}_i\mathbf{v}_i^\intercal\right).$$

By the Courant Fischer Min-Max Theorem, we have:

$$\lambda_{\max}(X(\mathbf{x})) \leq \sum_{i=1}^m \lambda_{\max}\left(x_i\mathbf{v}_i\mathbf{v}_i^\intercal\right).$$

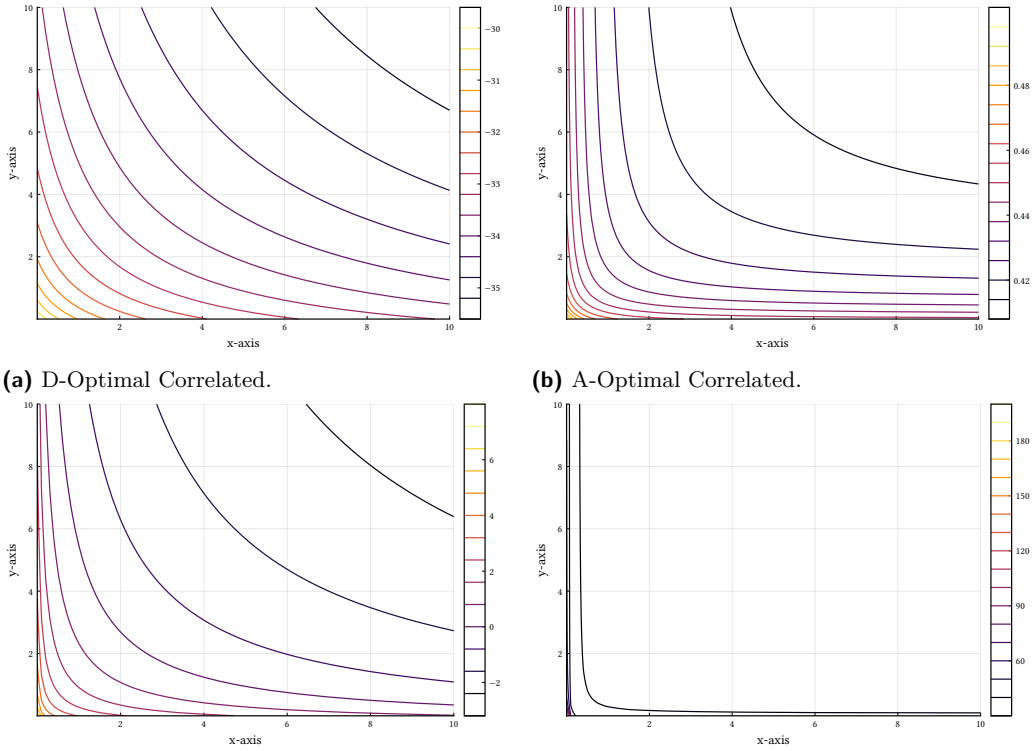
Using $\mathbf{x} \leq \mathbf{u}$, yields:

$$\lambda_{\max}(X(\mathbf{x})) \leq m \max_{j\in[m]} u_j \max_{i\in[m]} \|\mathbf{v}_i\|_2^2.$$ ◀

Combing Theorem 7 and Lemma 10, yields:

▶ **Corollary 11.** *The function $g(\mathbf{x}) = \operatorname{Tr}\left(X(\mathbf{x})^{-p}\right)$ is self-concordant on the convex feasible region $\mathcal{P}$.*

B Level sets of the different criteria



(a) D-Optimal Correlated.

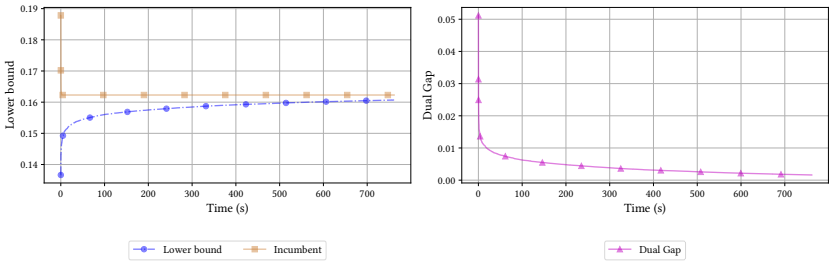
(b) A-Optimal Correlated.

(c) D-Fusion Independent.

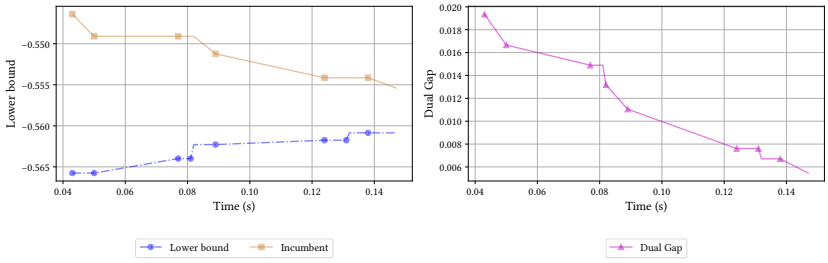
(d) A-Fusion Independent.

Figure 3 Example contour plots in two dimensions for objectives of both Optimal Problems with correlated data and both Fusion Problems with independent.

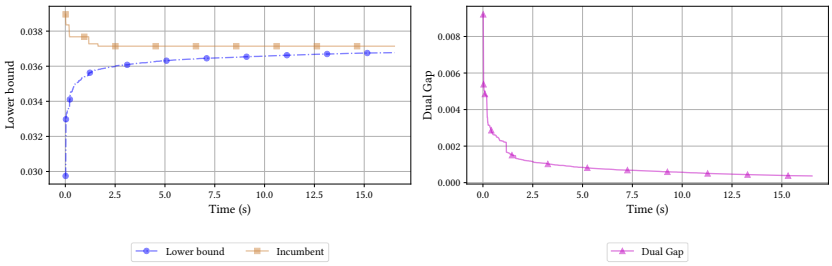
C Progress Plots of Boscia



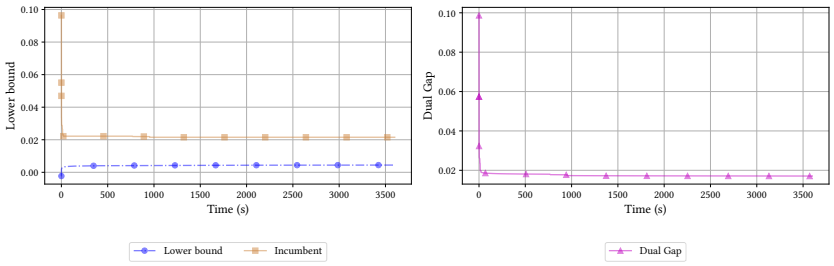
(a) $m = 60$ and $n = 15$.



(b) $m = 100$ and $n = 10$.



(c) $m = 100$ and $n = 10$.



(d) $m = 80$ and $n = 20$.

Figure 4 Progress of the incumbent and the lower bound on the left and progress of the dual gap for (a) the A-Optimal Problem with independent data, (b) the D-Optimal with correlated data and the A-Fusion Problem with (c) independent and (d) correlated data.