An Alternating Method for Cardinality-Constrained Optimization: A Computational Study for the Best Subset Selection and Sparse Portfolio Problems

Carina Moreira Costa, Dennis Kreber, Martin Schmidt

Abstract. Cardinality-constrained optimization problems are notoriously hard to solve both in theory and practice. However, as famous examples such as the sparse portfolio optimization and best subset selection problems show, this class is extremely important in real-world applications. In this paper, we apply a penalty alternating direction method to these problems. The key idea is to split the problem along its discrete-continuous structure to obtain two subproblems that are much easier to solve than the original problem. In addition, the coupling between these subproblems is achieved via a classic penalty framework. The method can be seen as a primal heuristic for which convergence results are readily available from the literature. In our extensive computational study, we first show that the method is competitive to a commercial MIP solver for the portfolio optimization problem. On these instances, we also test a variant of our approach that uses a perspective reformulation of the problem. Regarding the best subset selection problem, it turns out that our method significantly outperforms commercial solvers and that it is at least competitive to state-of-the-art methods from the literature.

1. Introduction

In many cases, optimization models require a cardinality constraint to filter underlying data. Such a mechanism is required if, for instance, the number of utilized variables should be small such that the solution is easier to interpret or such that variables are not used, which allegedly contain no valuable statistical information. On the other hand, it might happen that the usage of a variable is bound to some cost and a given budget cannot be exceeded. Moreover, a cardinality constraint can be interpreted as an optimization version of Occam's razor, i.e., one wants to compute a good solution while only using a small proportion of the available assumptions.

In this paper, we consider optimization problems of the general form

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{C}, \\ & |\operatorname{supp}(x)| \leq k \end{aligned} \tag{CC}$$

with $f : \mathbb{R}^n \to \mathbb{R}$ being a convex function, $\mathcal{C} \subseteq \mathbb{R}^n$ being a convex set, $\operatorname{supp}(x) := \{i \in [n] \colon x_i \neq 0\}$, $[n] := \{1, \ldots, n\}$, and $k \in \mathbb{N}$ being a fixed cardinality ceiling. Even though Problem (CC) does not exhibit any integer variables explicitly, it has a discrete structure and hence can be formulated as a mixed-integer problem. This fact can easily be seen by introducing new variables $z \in \{0, 1\}^n$ and reformulating

2010 *Mathematics Subject Classification.* 90Cxx, 90C11, 90C90, 90-08.

Key words and phrases. Cardinality Constraints, Alternating Direction Methods, Penalty Methods, Best Subset Selection, Portfolio Optimization.

the constraint $|\mathrm{supp}(x)| \leq k$ as

$$-Mz \leq x \leq Mz, \quad \sum_{i=1}^{n} z_i \leq k, \quad z \in \{0,1\}^n \tag{1}$$

with M being an upper bound on $\|x\|_\infty$.

Applications like the best subset selection problem [13, 59] or the cardinality-constrained portfolio optimization problem [11, 16] exactly fit into the formulation (CC). Moreover, extensions using nonconvex objective functions are also considered in the literature, e.g., for sparse principal component analysis [29]. However, let us note that there is also research on reformulating the sparse principal component analysis problem as a mixed-integer semidefinite program [10, 55], and hence as a mixed integer convex problem. The combinatorial structure combined with frequently occurring nonlinear objective functions makes this problem class difficult to solve in theory and practice. Hence, major research interests have been sparked in recent years. A wide range of results includes the perspective reformulation [35, 36, 47], which enables a tight formulation of Problem (CC), studies concerning bound tightening [5], or the application of disjunctive programming [19]. Additionally, local solvers are subject of interest as well; see, e.g., [21]. Interestingly, some methods from the low-rank optimization literature are similar in their spirit to the method that we propose in this paper; see, e.g., the nonlinear factorization method of [22] or the alternating least squares implementation of softimpute [48]. Although these methods are concerned with low-rank optimization problems, a rank constraint on a matrix is in fact a cardinality constraint on its singular values. Both in the cited papers as well as in this paper, the way to handle the present nonconvex constraints is to make use of alternating direction optimization schemes.

1.1. **Contribution.** Due to the hardness of (CC), fast local methods or primal heuristics are desired to deliver feasible points of good quality for warmstarting a global solver or—on the other hand—they can even act as standalone approaches for large-scale instances. Our contribution is to derive such a primal heuristic that, additionally, can be analyzed theoretically. To this end, we consider the special combinatorial structure of Problem (CC). On the one hand, the considered problems can be solved efficiently if the cardinality constraint is omitted and, on the other hand, the cardinality constraint, if decoupled from the rest, is easy to handle as well since its set of feasible indicator vectors is the uniform matroid [61]. Thus, we consider these two parts of the problem separately, each with its own set of variables. In other words, we consider these parts as separate directions of optimization in a higher dimensional space. However, only optimizing into each direction individually would very likely not yield a feasible point for the complete problem as we would only consider the feasible region for each direction independently. As a remedy, one can add a coupling condition that penalizes the deviation between the solutions of the two directions forcing them to be equal if the penalization is sufficiently large. We later show evidence that the proposed approach is very fast and reliably provides feasible points of very good quality. In particular, by using the classic theory of alternating direction methods, it can be shown that a partial minimum, i.e., a solution which cannot be improved in any of the decomposed parts, is returned. Thus, unlike many other heuristics, our proposed method guarantees certain qualitative characteristics. Our numerical results show that this method is competitive to commercial software packages for the cardinality-constrained portfolio optimization problem and that it significantly outperforms them for the case of the best subset selection problem. For the latter, our method also outperforms or is competitive to other heuristics that are state-of-the-art in the current literature. In particular, we compared our method with the MaxMin method as proposed in [14] as well as with the L0Learn method

as discussed in [28, 50]. For the portfolio optimization problem, we also tested a variant of our method using the perspective reformulation of the problem; see, e.g., [35]. However, although this can yield an improvement in terms of objective function values, it leads to a significantly slower approach compared to the originally proposed method.

1.2. **Structure of the Paper.** As just mentioned, we decided to focus in our computational analysis on important instances of the general framework (CC) instead of analyzing the general problem class calibrated with, e.g., random data. Hence, we apply the method to the best subset selection problem and the sparse portfolio optimization problem, which we introduce in detail in Section 2. Afterward, in Section 3, we review PADMs in general and show how to apply them to the best subset selection and the cardinality-constrained portfolio optimization problems. In Section 4, we recall how the perspective reformulation techniques are applied to the aforementioned problems for obtaining strong relaxations of the underlying MIQPs and we also discuss a possible combination of the perspective reformulation techniques with the PADM proposed in Section 3. In Section 5, we then present the numerical results of the PADM applied to both problems studied and the paper closes with some concluding remarks in Section 6.

2. Applications

In this section, we discuss two famous instances of the general problem class in (CC): the cardinality-constrained portfolio optimization problem (Section 2.1) and the best subset selection problem (Section 2.2).

2.1. **The Cardinality-Constrained Portfolio Optimization Problem.** During the last decades, there has been a lot of research in the optimization and investment communities regarding the famous mean-variance Markowitz optimization model [56]. This attention is mainly due to the inclusion of real-world financial constraints, which makes the traditional model more realistic but also more complex. For a survey on different constraints proposed in the literature; see, e.g., [52, 58]. One popular constraint is the cardinality constraint, which limits the number of assets to be included in the portfolio. In what follows, we describe the resulting cardinality-constrained portfolio optimization problem.

We are given n possibly risky assets with mean return vector $r \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$ as well as the level of expected portfolio return $R > 0$. Then, the traditional Markowitz optimization framework computes the optimal proportion of each asset such that the risk is minimized for the given desired return or the return is maximized for a given level of risk. Let $0 < k \leq n$ be the maximum number of assets that can be included in the portfolio. Thus, the cardinality-constrained portfolio optimization problem is given by

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & x^\top \Sigma\, x \\ \text{s.t.} \quad & r^\top x \geq R, \\ & e^\top x = 1, \\ & x \geq 0, \\ & \|x\|_0 \leq k, \end{aligned} \tag{CCPO}$$

where e is the vector of all ones in suitable dimension. By introducing binary variables $z \in \{0,1\}^n$, this problem can be reformulated as

$$\begin{aligned} \min_{x,z} \quad & x^\top \Sigma\, x \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \\ & 0 \leq x_i \leq z_i \quad \text{for all} \quad i = 1, \dots, n, \\ & \sum_{i=1}^{n} z_i \leq k, \quad z_i \in \{0,1\} \quad \text{for all} \quad i = 1, \dots, n. \end{aligned} \tag{2}$$

This is a mixed-integer quadratic program (MIQP). In general, this is an NP-hard problem [8, 40] and there are a couple of branch-and-bound algorithms [15, 16, 18, 27, 36, 37, 40, 68, 71] for solving Problem (CCPO) to global optimality. However, it is not always suitable to use such global methods to find optimal solutions, especially for large-scale instances. To the best of our knowledge, the exact method that scales best even for large instances has been recently proposed in [8], where instances are solved for which the number of assets is around 1000 or larger. Heuristics and local search methods have also been studied. Earlier heuristics rely on genetic algorithms, tabu search, and simulated annealing [26]. More recent heuristics are mostly based on first-order methods; see, e.g., [8, 64]. In particular, we highlight [64], because it uses a similar decomposition idea as we do—however, the resolution approach is rather different. In [8], a heuristic tailored to the sparse portfolio problem is used to compute high-quality feasible solutions, which are then used to warm-start an exact method. This heuristic has been initially proposed in [13] for realizing warm-starts for the best subset selection problem, which we discuss next.

2.2. **The Best Subset Selection Problem.** Consider the linear regression model $y = X\beta^0 + \varepsilon$ with response vector $y \in \mathbb{R}^n$, observed data $X \in \mathbb{R}^{n\times p}$, true predictors $\beta^0 \in \mathbb{R}^p$, and errors $\varepsilon \in \mathbb{R}^n$. Assuming that β^0 and ε are unknown, we try to estimate the true predictors β^0 with a linear regression. However, it is often not known which recorded variables are actually relevant, i.e., the regressors β^0 are sparse. More formally, this means that $|\operatorname{supp}(\beta^0)| < p$ holds. The best subset selection problem [59] aims to address this aspect by allowing to have only k many non-zero entries in β. The objective is then to select k out of p features, which best describe the relation between X and y. Usually, a proper k is then chosen via cross-validation. Accordingly, the best subset selection problem is defined as the optimization problem

$$\begin{aligned} \min_{\beta\in\mathbb{R}^p} \quad & \|X\beta - y\|_2^2 \\ \text{s.t.} \quad & |\operatorname{supp}(\beta)| \leq k. \end{aligned} \tag{BSS}$$

Clearly, the best subset selection fits the framework of (CC) and, hence, is an appropriate candidate for studying our proposed methods. Problem (BSS) is NP-hard [60] and it is folklore that it is indeed notoriously difficult to solve in practice as well [13, 57]. Hence, the problem gained a lot of research interest in recent years [3, 13, 14, 32]. Since (BSS) is considered computationally hard and since it is common to consider thousands of variables in practice, some research has concentrated on finding viable heuristics [57, 72] for the best subset selection problem. The most prominent approach is the Lasso method [20, 65], which replaces $|\operatorname{supp}(\beta)|$ with $\|\beta\|_1$. Often the modified cardinality constraint is then put into the objective function as a regularization term:

$$\min_{\beta\in\mathbb{R}^p} \quad \|X\beta - y\|_2^2 + \lambda\|\beta\|_1. \tag{LASSO}$$

The Lasso method enjoys major scientific interest and thus many results exist proving beneficial characteristics of Lasso under various assumptions concerning the

underlying data [20]. These assumptions include the compatibility condition [42] or the restricted isometry property [24]. Unfortunately, if these conditions are not satisfied, the resulting statistical properties can cease to hold [13, 57, 70]. Moreover, testing the conditions is NP-hard [30, 67].

In contrast, empirical evidence indicates that the best subset selection can produce solutions that have superior predictive quality compared to Lasso [13, 57]. Since the best subset selection is still computationally challenging and because Lasso can have disadvantages concerning the statistical performance, it is of high interest to develop better heuristics for (BSS).

Recently, more efforts have been made to leverage techniques from discrete optimization to find good heuristic solutions for (BSS). The authors of [13] propose a first-order method combined with the hard-thresholding operator. Their efforts are focused on solving the best subset selection exactly and, hence, they use their heuristic as a warmstart. Another approach [14] takes the dual of the continuous part of the problem and transforms it to a max-min problem, which can be solved efficiently as a second-order cone problem. This MaxMin method provides very good results and will be one of the benchmarks that we compare our method with. One of the very recent and most promising approaches is the L0Learn method [28, 50]. Similar to the algorithms used to find a solution for Lasso, L0Learn utilizes a coordinate-descent approach. L0Learn also serves as a comparison for our proposed method.

Additionally, screening and excluding variables to decrease the dimension can also be helpful in solving (BSS). In [2], the authors propose rules, which can be efficiently checked and which provide guarantees for variables to be excluded. Relaxing the condition that a decision variable must be either 0 or 1 in every optimal solution to be excluded leads to the backbone method [12]. Here, it is enough that a variable is set to 1 in at least one near-optimal solution in order to fix it.

3. Decomposition and Penalty Alternating Direction Methods

Next, we review penalty alternating direction methods (PADMs) in general (Section 3.1) and then discuss in detail on how to efficiently apply these methods to the cardinality-constrained portfolio optimization problem (Section 3.2) and to the best subset selection problem (Section 3.3).

3.1. General Description of PADMs.

To describe the general ideas behind PADMs, we consider the general problem

$$\min_{u,v} \quad f(u,v) \tag{3a}$$

$$\text{s.t.} \quad g(u,v) = 0, \quad h(u,v) \geq 0, \tag{3b}$$

$$u \in U \subseteq \mathbb{R}^{n_u}, \quad v \in V \subseteq \mathbb{R}^{n_v}, \tag{3c}$$

for which we make the following assumption.

Assumption 1. The objective function $f : \mathbb{R}^{n_u+n_v} \to \mathbb{R}$ and the constraint functions $g : \mathbb{R}^{n_u+n_v} \to \mathbb{R}^m$, $h : \mathbb{R}^{n_u+n_v} \to \mathbb{R}^q$ are continuous and the sets U and V are non-empty and compact.

Alternating direction methods are iterative procedures that solve Problem (3) by alternatingly solving two simpler subproblems. Given an iterate (u^l, v^l), Problem (3) with v fixed to v^l is solved into the direction of u, yielding a new u-iterate u^{l+1}. Subsequently, u is fixed to u^{l+1} and Problem (3) is solved into the direction of v, yielding a new v-iterate v^{l+1}. The method is formally stated in Algorithm 1. The for-loop is repeated until a termination criterion is reached. To present the general

convergence result of Algorithm 1, we need the following definition that uses the abbreviation

$$\Omega = \{(u, v) \in U \times V \colon g(u, v) = 0,\ h(u, v) \geq 0\} \subseteq U \times V$$

for the feasible set of Problem (3).

Algorithm 1 A standard ADM.

1: Choose initial values $(u^0, v^0) \in U \times V$.
2: **for** $l = 0, 1, \ldots$ **do**
3: Compute
$$u^{l+1} \in \arg\min_u \left\{ f(u, v^l) \colon g(u, v^l) = 0,\ h(u, v^l) \geq 0,\ u \in U \right\}.$$
4: Compute
$$v^{l+1} \in \arg\min_v \left\{ f(u^{l+1}, v) \colon g(u^{l+1}, v) = 0,\ h(u^{l+1}, v) \geq 0,\ v \in V \right\}.$$
5: Set $l \leftarrow l + 1$.
6: **end for**

Definition 2. Let $(u^*, v^*) \in \Omega$ be a feasible point of Problem (3). Then, (u^*, v^*) is called a *partial minimum* of Problem (3) whenever it satisfies

$$f(u^*, v^*) \leq f(u, v^*) \quad \text{for all} \quad (u, v^*) \in \Omega,$$
$$f(u^*, v^*) \leq f(u^*, v) \quad \text{for all} \quad (u^*, v) \in \Omega.$$

Consider now the set of possible future iterates starting from the current iterate (u^l, v^l), i.e.,

$$\Theta(u^l, v^l) = \left\{ (u^*, v^*) \colon f(u^*, v^l) \leq f(u, v^l)\, \forall u \in U;\ f(u^*, v^*) \leq f(u^*, v)\, \forall v \in V \right\}.$$

The general convergence result of Algorithm 1 is stated in the following theorem; see Theorem 4.9 in [46] for the proof. For further details on the convergence theory of classic ADMs; see also [69].

Theorem 3. Let $\{(u^l, v^l)\}_{l=0}^{\infty}$ be a sequence generated by Algorithm 1 with $(u^{l+1}, v^{l+1}) \in \Theta(u^l, v^l)$. Suppose that the solution of the first optimization problem (in Line 3) is always unique. Then, every convergent subsequence of $\{(u^l, v^l)\}_{l=0}^{\infty}$ converges to a partial minimum. In addition, if w and w' are two limit points of such subsequences, then $f(w) = f(w')$ holds.

Usually, the feasible set of Problem (3) is not completely decomposable into the disjoint sets U and V since the constraints in (3b) couple the two subproblems. These coupling constraints may lead to poor convergence rates in practice. Thus, an extension of the ADM called penalty alternating direction method (PADM) has been proposed in [43], which has been already used successfully to solve many problems in practical applications; see, e.g., [23, 44, 45, 53, 62]. The main idea is to move the coupling constraints g and h to the objective function by means of an ℓ_1 penalty function, yielding

$$\phi_1(u, v; \mu, \rho) := f(u, v) + \sum_{i=1}^{m} \mu_i |g_i(u, v)| + \sum_{i=1}^{q} \rho_i [h_i(u, v)]^-,$$

with $[\alpha]^- = \max\{0, -\alpha\}$ and $\mu = (\mu_i)_{i=1}^m$, $\rho = (\rho_i)_{i=1}^q \geq 0$ being the penalty parameters of the equality and inequality constraints, respectively.

The penalty ADM works as follows. Given a starting point and initial values for all penalty parameters, the ADM of Algorithm 1 computes a partial minimum of the penalty problem

$$\min_{u,v} \quad \phi_1(u, v; \mu, \rho) \quad \text{s.t.} \quad u \in U,\ v \in V. \tag{4}$$

With this partial minimum at hand, we verify if the coupling constraints are satisfied. If they are, we stop with a feasible solution of Problem (3). If not, the penalty parameters are updated and the next penalty problem is solved to partial minimality. In this way, the method is composed out of an inner and an outer loop. In the inner loop, a partial minimum of the current penalty problem is computed with a classic ADM, while in the outer loop, the penalty parameters are updated producing a new penalty problem. Thus, the PADM generates a sequence of partial minima of a sequence of penalty problems of type (4). The method is formally stated in Algorithm 2.

Algorithm 2 The ℓ_1 penalty ADM.

Choose initial values $(u^{0,0}, v^{0,0}) \in U \times V$ and penalty parameters μ^0, $\rho^0 \geq 0$.
for $t = 0, 1, \ldots$ **do**
  Set $l = 0$.
  while $(u^{t,l}, v^{t,l})$ is not a partial minimum of (4) with $\mu = \mu^t$ and $\rho = \rho^t$ **do**
    Compute

$$u^{t,l+1} \in \arg\min_u \left\{\phi_1(u, v^{t,l}; \mu^t, \rho^t) \colon u \in U\right\}.$$

    Compute

$$v^{t,l+1} \in \arg\min_v \left\{\phi_1(u^{t,l+1}, v; \mu^t, \rho^t) \colon v \in V\right\}.$$

    Set $l \leftarrow l + 1$.
  end while
  if $(u^{t,l}, v^{t,l})$ satisfies the coupling constraints **then**
    Stop with $(u^{t,l}, v^{t,l})$ being a feasible point of (3).
  else
    Choose new penalty parameters $\mu^{t+1} \geq \mu^t$ and $\rho^{t+1} \geq \rho^t$.
  end if
end for

In the following, we state the general convergence result of Algorithm 2; see [43] for a proof (Theorem 8) and for further details about the method.

Theorem 4. Suppose that Assumption 1 holds and that $\mu_i^t \nearrow \infty$ for all $i = 1, \ldots, m$ and $\rho_i^t \nearrow \infty$ for all $i = 1, \ldots, q$. Moreover, let (u^t, v^t) be a sequence of partial minima of (4) (for $\mu = \mu^t$ and $\rho = \rho^t$) generated by Algorithm 2 with $(u^t, v^t) \to (u^*, v^*)$. Then, there exist $\bar{\mu}, \bar{\rho} \geq 0$ such that (u^*, v^*) is a partial minimizer of the weighted ℓ_1 feasibility measure

$$\chi_{\bar{\mu},\bar{\rho}}(u, v) := \sum_{i=1}^{m} \bar{\mu}_i |g_i(u, v)| + \sum_{i=1}^{q} \bar{\rho}_i [h_i(u, v)]^-.$$

If, in addition, (u^*, v^*) is feasible for the original problem (3), the following holds:

(a) If f is continuous, then (u^*, v^*) is a partial minimum of (3).
(b) If f is continuously differentiable, then (u^*, v^*) is a stationary point of (3).
(c) If f is continuously differentiable and f and Ω are convex, then (u^*, v^*) is a global optimum of (3).

Let us note that the above result does not ensure the convergence of iterates or the feasibility of limit points but states properties of the limit points in case of convergence. However, our numerical results later show that we neither face convergence nor infeasibility issues for the two problems that we apply our methods to.

3.2. **Application to the Cardinality-Constrained Portfolio Optimization Problem.** We now turn to the application of Algorithm 2 to Problem (CCPO). First, we duplicate the continuous variables $x \in \mathbb{R}^n$ via the introduction of new variables $w \in \mathbb{R}^n$ and rewrite Problem (CCPO) as

$$\begin{aligned} \min_{x,w} \quad & x^\top \Sigma\, x \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \quad x \geq 0, \\ & x = w, \quad \|w\|_0 \leq k. \end{aligned}$$

The constraint $x = w$ is the so-called copy constraint that couples the variables x and w. It allows us to transfer the hard cardinality constraint previously on x to w. Copy constraints are commonly used in decomposition methods to split the genuine problem into parts. Now, to alleviate this constraint, we move it to the objective function by means of an ℓ_1 penalty term with penalty parameter $\mu \geq 0$, obtaining the following problem which is in the form of (4)

$$\begin{aligned} \min_{x,w} \quad & x^\top \Sigma\, x + \mu \|x - w\|_1 \\ \text{s.t.} \quad & x \in \tilde{U} := \{\tilde{x} \in \mathbb{R}^n : r^\top \tilde{x} \geq R,\, e^\top \tilde{x} = 1,\, \tilde{x} \geq 0\}, \\ & w \in \tilde{V} := \{\tilde{w} \in \mathbb{R}^n : e^\top \tilde{w} = 1,\, \|\tilde{w}\|_0 \leq k\}. \end{aligned} \tag{5}$$

Note that the constraint $e^\top \tilde{w} = 1$ does not necessarily need to be included in the set $\tilde{V}$ since the same constraint is considered for x. However, previous computational experiments revealed that having this constraint in both subproblems significantly improves the solution process of the PADM.

Now, we can fully decompose the variable space of Problem (5) into two blocks: one for x and the other one for w. Thus, given the outer iteration t, in each inner iteration l of Algorithm 2 the two subproblems that need to be solved are given by

$$x^{t,l+1} \in \operatorname*{arg\,min}_{x \in \tilde{U}} \; x^\top \Sigma\, x + \mu^t \|x - w^{t,l}\|_1, \tag{6}$$

and

$$w^{t,l+1} \in \operatorname*{arg\,min}_{w \in \tilde{V}} \; \|x^{t,l+1} - w\|_1. \tag{7}$$

Here, we already discarded the constant term $(x^{t,l+1})^\top \Sigma\, x^{t,l+1}$ and the penalty parameter μ^t in the objective function of Problem (7) since they do not have an effect on the solution of the optimization problem.

These problems can be solved efficiently. Problem (6) is the traditional portfolio optimization problem with an ℓ_1 regularization term. Therefore, it is a convex optimization problem which can be solved by any state-of-the-art quadratic programming solver. However, it might not have a unique solution. On the other hand, a solution of Problem (7) can be stated in closed form, which is what we show in the next two propositions. In order to simplify the presentation, we denote the sub-vector of a vector $x \in \mathbb{R}^n$ corresponding to the index set $S = \{i_1, \ldots, i_m\} \subseteq [n]$ by $x_S \in \mathbb{R}^m$, i.e., $x_S := (x_{i_1}, \ldots, x_{i_m})^\top$.

Proposition 5. Suppose that $\bar{x}$ is an optimal solution of Problem (6). For every optimal solution $\tilde{w}$ of Problem (7) with $\operatorname{supp}(\tilde{w}) = S$, there exists an optimal

solution w^* of Problem (7) such that $\operatorname{supp}(w^*) = S$ and

$$w_i^* = \begin{cases} \bar{x}_i/(e^\top \bar{x}_S), & \text{if } i \in S, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

holds.

Proposition 6. Suppose that $\bar{x}$ is an optimal solution of Problem (6). Let $\bar{x}_{i_1} \geq \bar{x}_{i_2} \geq \cdots \geq \bar{x}_{i_n}$ be the sorted entries of $\bar{x}$. Then, there exists an optimal solution w^* of Problem (7) such that $S = \{i_1, \ldots, i_k\} = \operatorname{supp}(w^*)$ holds.

The proofs are given in Sections A.1 and A.2 in the appendix. From Proposition 5, we can directly compute a solution of Problem (7) using Formula (8) and the set S characterized in Proposition 6. Although both the first and the second subproblem do not necessarily have to possess unique solutions, Formula (8) also serves as a "tie-breaking" rule that stays the same over all iterations. Thus, the general PADM theory applies.

3.3. **Application to the Best Subset Selection Problem.** In this section, we propose a tailored version of Algorithm 2 to compute partial minima of Problem (BSS).

For applying the PADM to the best subset selection problem we first duplicate the continuous variables of Problem (BSS) using the new variables $\gamma, \delta \in \mathbb{R}^p$ and then rewrite the problem equivalently as

$$\begin{aligned} \min_{\gamma,\delta} \quad & \|X\gamma - y\|_2^2 \\ \text{s.t.} \quad & \gamma = \delta, \\ & |\operatorname{supp}(\delta)| \leq k. \end{aligned}$$

Now, we move the coupling condition $\gamma = \delta$ to the objective function by using an ℓ_1 penalty term with penalty parameter $\mu \geq 0$, yielding

$$\begin{aligned} \min_{\gamma,\delta} \quad & \|X\gamma - y\|_2^2 + \mu\|\gamma - \delta\|_1 \\ \text{s.t.} \quad & \gamma \in \bar{U} := \mathbb{R}^p, \\ & \delta \in \bar{V} := \{\delta' \in \mathbb{R}^p \colon \|\delta'\|_0 \leq k\}. \end{aligned} \tag{9}$$

Note that this problem is of the form of Problem (4). Thus, we can apply Algorithm 2 to Problem (9) with the search space decomposed into two blocks: one for γ and the other one for δ.

In each inner iteration, the two subproblems to be solved read

$$\gamma^{t,l+1} \in \operatorname*{arg\,min}_{\gamma \in \bar{U}} \|X\gamma - y\|_2^2 + \mu^t \|\gamma - \delta^{t,l}\|_1, \tag{10}$$

and

$$\delta^{t,l+1} \in \operatorname*{arg\,min}_{\delta \in \bar{V}} \|\gamma^{t,l+1} - \delta\|_1. \tag{11}$$

With these two subproblems at hand we have to ensure two properties. First, considering that we want to avoid the computational hardness of (BSS) both subproblems (10) and (11) should be easier to handle. Second, we have to make sure that Theorem 3 can be applied.

We first show that both directions are efficiently solvable. For Problem (10) it is easy to see that the optimization problem is convex and thus, can be solved in polynomial time. Moreover, Problem (10) rather closely resembles (LASSO).

Indeed, this first subproblem can be solved via the Lasso method by substituting $\tilde{\gamma} := \gamma - \delta^{t,l}$. Consequently, we can solve the equivalent optimization problem

$$\min_{\tilde{\gamma} \in \mathbb{R}^p} \ \left\| X\tilde{\gamma} - \left(y - X\delta^{t,l}\right)\right\|_2^2 + \mu^t \left\|\tilde{\gamma}\right\| \tag{12}$$

by using tailored solvers such as the coordinate descent solver in [39] or a LARS solver [33], which exploit the sparsity structure of the solution. These specialized solvers are known to be exceptionally fast.

For Problem (11), a simple Greedy algorithm finds a global optimal solution. This is shown in the next proposition. The proof is given in Section A.3 in the appendix.

Proposition 7. Given the outer iteration t, the inner iteration l, and a solution $\gamma^{t,l+1}$ of Problem (10). Then, the optimal solution of Problem (11) is given by

$$\delta^{t,l+1} = (\gamma_1^{t,l+1}, \dots, \gamma_k^{t,l+1}, 0, \dots, 0),$$

with $|\gamma_1^{t,l+1}| \geq \cdots \geq |\gamma_k^{t,l+1}| \geq \cdots \geq |\gamma_p^{t,l+1}|$.

Hence, both subproblems can be solved very fast. It remains to be clarified if the assumptions of Theorem 3 hold. Clearly, Assumption 1 is satisfied. Thus, we only need to check if one of the two subproblems always yields a unique solution. For Problem (11) this is not the case. It is possible that the coefficients $\gamma^{t,l+1}$ do not have a unique order if there are multiple k-largest elements, i.e.,

$$|\gamma_1^{t,l+1}| \geq \cdots \geq |\gamma_k^{t,l+1}| = |\gamma_{k+1}^{t,l+1}| = \cdots = |\gamma_{k+q}^{t,l+1}| \geq \cdots \geq |\gamma_p^{t,l+1}|$$

for some $q \geq 1$. Fortunately, (10) has a unique solution under some mild assumptions. That is, the following result, taken from [66], holds for the Lasso problem and consequently for (10) as well.

Proposition 8. If the entries of $X \in \mathbb{R}^{n \times p}$ are drawn from a continuous probability distribution on $\mathbb{R}^{n \times p}$, then for any y and $\lambda > 0$, the Lasso solution is unique with probability one.

The assumption that X is drawn from a continuous probability distribution is not a major restriction since it usually holds in real-world applications. Note that no requirements for y are needed. Thus, we can safely apply the transformation (12).

Furthermore, we could also force the second subproblem (11) to be unique by adding a regularization term $\pi \sum_{i=1}^{p} i\, \chi_{\delta_i \neq 0}(\delta_i)$ to (9) with $\chi_{\delta_i \neq 0}$ being the indicator function that returns 1 if δ_i is non-zero and 0 otherwise. If π is chosen sufficiently small, the optimal selection will not be changed but nevertheless it causes Problem (11) to always have a unique solution. While this is a mathematical technicality, in the algorithm it would simply mean to first pick the element with the smallest index if there is no unique choice during the greedy selection.

Thus, the PADM fits well for being applied to the best subset selection problem. For the relation between the PADM and the trimmed Lasso method, we refer to Section A.4 in the appendix, where we also discuss some robustness aspects of the mentioned methods.

4. Perspective Reformulations

Exact methods for solving problems of the form of (CC) are usually based on branch-and-bound techniques. Thus, a good continuous relaxation of the problem is essential. The standard big-M formulation obtained by reformulating the cardinality

constraint via (1) usually provides only weak continuous relaxations, which slows down branch-and-bound. For a more detailed discussion, see [17].

Fortunately, problems of the form of (CC) are MIQPs with semi-continuous variables x and binary variables z. For such problems, a better MIQP reformulation is proposed in [35] using so-called perspective cuts. For doing so, the authors explore the convex envelope of the objective function and show that it is the perspective function of the continuous part of the objective function. This insight can be used to provide a reformulation with a tighter continuous relaxation—called the perspective reformulation. However, this technique is only applicable for problems with convex and separable objective functions but, in the same work [35], the authors present a trick for problems such as (CCPO) with non-separable cost matrix Σ so that the technique can be applied again. The idea is to extract a positive semidefinite diagonal matrix D such that $\Sigma - D$ is positive semidefinite. This gives rise to a new separable part containing D. Different ways exist for computing a proper diagonal matrix; see, e.g., [38]. The matrix D, which gives the tightest continuous relaxation is presented in [71], but requires solving a structured semidefinite program.

We now discuss how the perspective reformulation can be applied to Problem (CCPO). First, a positive semidefinite diagonal matrix D with entries $D_{ii} \geq 0$, $i \in [n]$, is extracted from Σ such that $\Sigma - D$ is still positive semidefinite. With this at hand, we re-write the problem as follows:

$$\begin{aligned} \min_{x,z} \quad & x^\top (\Sigma - D)\, x + x^\top D\, x \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \\ & 0 \leq x \leq z, \quad e^\top z \leq k, \quad z \in \{0,1\}^n. \end{aligned}$$

Now, the perspective reformulation is applied to the convex and separable part $x^\top D x$, i.e., this term is replaced by its convex envelope, which is exactly its perspective function, yielding

$$\begin{aligned} \min_{x,z} \quad & x^\top (\Sigma - D)\, x + \sum_{i\in[n]} D_{i,i} \frac{x_i^2}{z_i} \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \\ & 0 \leq x \leq z, \quad e^\top z \leq k, \quad z \in \{0,1\}^n. \end{aligned}$$

Here, we assume that $x_i^2/z_i = 0$ if $z_i = 0$ for all $i \in [n]$. Note that the fractional term renders this problem intractable at a first glance. To resolve this issue, another reformulation is discussed in [1, 34, 47, 63], which further allows to impose the cardinality constraint via second-order cone (SOC) constraints and drop the (considerably weaker) big-M constraints in (1), resulting in the mixed-integer second-order cone problem (MISOCP)

$$\begin{aligned} \min_{x,z,\theta} \quad & x^\top (\Sigma - D)\, x + \sum_{i\in[n]} D_{i,i}\, \theta_i \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \\ & e^\top z \leq k, \quad z \in \{0,1\}^n, \\ & x_i^2 \leq \theta_i z_i \quad \text{for all} \quad i = 1, \dots, n, \\ & x, \theta \geq 0. \end{aligned} \tag{13}$$

This reformulation has been shown to outperform the big-M formulation in several works; see, e.g., [8, 9, 71].

4.1. **Combining Perspective Reformulations with PADM.** We now show how we can use perspective reformulations in combination with the PADM. We discuss the main ideas for the case of Problem (CCPO). The ideas can, however, be applied to Problem (BSS) as well. We start with the problem

$$\begin{aligned} \min_{x,w} \quad & x^\top(\Sigma - D)\, x + w^\top D\, w \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \quad x \geq 0, \\ & x = w, \quad \|w\|_0 \leq k, \end{aligned}$$

which is equivalent to (CCPO). As before, we penalize the coupling constraints via an ℓ_1 penalty term

$$\begin{aligned} \min_{x,w} \quad & x^\top(\Sigma - D)\, x + w^\top D\, w + \mu\|x - w\|_1 \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \quad x \geq 0, \quad \|w\|_0 \leq k. \end{aligned}$$

Thus, given the outer iteration t, in each inner iteration l of Algorithm 2, the two subproblems that need to be solved are

$$\begin{aligned} x^{t,l+1} \in \arg\min_{x\in\mathbb{R}^n} \quad & x^\top(\Sigma - D)\, x + \mu^t\|x - w^{t,l}\|_1 \\ \text{s.t.} \quad & r^\top x \geq R, \quad e^\top x = 1, \quad x \geq 0, \end{aligned}$$

and

$$\begin{aligned} w^{t,l+1} \in \arg\min_{w\in\mathbb{R}^n} \quad & w^\top D\, w + \mu^t\|x^{t,l+1} - w\|_1 \\ \text{s.t.} \quad & e^\top w = 1, \quad \|w\|_0 \leq k. \end{aligned} \tag{14}$$

Note that, when compared to the subproblem (7), we are now adding more information from the original problem to the subproblem in the direction of w. The underlying rationale is that this may improve the performance of PADM especially in cases for which the covariance matrix Σ is diagonally dominant. On the other hand, this comes with additional difficulties. First, a closed form solution for (14) seems to be out of reach. Nonetheless, we can again introduce binary variables z, apply the perspective reformulation to the term $w^\top D\, w$, and reformulate it as an MISOCP similarly to as it is done in the previous section. Second, the first subproblem is now "less convex" since we are extracting the matrix D from Σ. Consequently, we may loose uniqueness properties for the first subproblem. These difficulties may harm the solution process of PADM. Therefore, we explore this in more depth in our numerical studies.

5. Computational Study

In this section, we present and discuss the numerical results for the cardinality-constrained portfolio optimization problem in Section 5.2 and the best subset selection problem in Section 5.3. Before, we briefly discuss the software and hardware setup in the next section.

5.1. **Software and Hardware Setup.** All computations were conducted on a computer with two Intel Xeon CPU E5-2699 v4 at 2.20 GHz (2×44 threads) and 756 GB RAM.

The PADM approach for the best subset selection problem has been implemented in C++. For solving the Lasso subproblems, we use the PICASSO library [41]. For computing the global solutions and for solving second-order cone programs, we use CPLEX 12.8. The PADM for portfolio optimization has been coded in Python and GUROBI 9.03 is used for solving the first subproblem (6) as well as the big-M and

the MISOCP formulations. The PADM implementation and the setup of the computational study is available under an open-source license at https://gitlab.com/Dnis/computational-study-padm-for-cardinality-constrained-problems.

5.2. **Numerical Results for the Cardinality-Constrained Portfolio Optimization Problem.** In this section, we present the computational results for real-world as well as large-scale and randomly generated cardinality-constrained portfolio optimization instances. First, we comment on some implementation details in the next section. The test set that we consider is described in Section 5.2.2. Finally, in Section 5.2.3, we evaluate the performance of the algorithm proposed in Section 3.2 in terms of solution quality and running time by comparing it with the commercial solver Gurobi applied to the big-M based MIQP formulation (2). In cases for which a diagonal matrix D (see Section 4) is available, we also compare against the MISOCP formulation (13). Moreover, we also present and discuss some results obtained by combining the PADM with the perspective reformulation.

5.2.1. *Implementation Issues.* We set the initial penalty parameter μ^0 based on the magnitude of usual objective function values of the problem. This criterion turned out to give a good balancing between feasibility and optimality. Thus, for the real-world instances, the initial penalty parameter is set to 10^{-4}, while for the random instances, it is set to 1. In both cases the penalty parameter is updated using the factor 10. Each inner loop is stopped when a partial minimum is obtained, i.e., whenever $\|(x^{t,l}, w^{t,l}) - (x^{t,l-1}, w^{t,l-1})\|_\infty < 10^{-5}$. Finally, the PADM terminates with a feasible partial minimum if the coupling constraint is satisfied, i.e., if $\|x - w\|_1 < 10^{-5}$ holds.

5.2.2. *Description of the Test Set.* We first consider the five mean-variance portfolio optimization problems described in [26]. These instances are publicly available in the OR-Library test set [4]. They are built from weekly price data using real-world capital market indices and have been widely used to test algorithms for the sparse portfolio optimization problem. Since a required level of return R is not given, we define it similarly to [8, 71], i.e., first we compute

$$R_{\min} = r^\top x^*, \qquad x^* = \arg\min_x \left\{ x^\top \Sigma\, x \colon e^\top x = 1,\ x \geq 0 \right\},$$
$$R_{\max} = r^\top \bar{x}, \qquad \bar{x} = \arg\max_x \left\{ r^\top x \colon e^\top x = 1,\ x \geq 0 \right\},$$

and then set $R = R_{\min} + 0.3(R_{\max} - R_{\min})$.

Another test set of mean-variance portfolio optimization problems that is widely used in the literature is described in [38].[1] In total, there are 90 instances, 30 for each value of $n \in \{200, 300, 400\}$. Moreover, for each n, there are three subsets of 10 instances identified as n^+, n^0, and n^-, where the superscript represents the degree of diagonal dominance, i.e., strongly, weakly, and not diagonally dominant, respectively. A proper matrix D, as discussed in Section 4, is given as well. Specifically, we use those in the folder "s" which are independent of the choice of the sparsity k and are obtained by solving the SDP discussed in [38].

The largest of all those instances contains 400 assets. To test our algorithm on large-scale instances, we also consider further randomly generated cardinality-constrained portfolio problems. To this end, we follow the procedure given in [51], and use the Python implementation made available by [25].[2] This procedure generates random covariance matrices in a way so that they mimic covariance matrices of real-world data. In total, we use 60 randomly generated instances with the number

[1] The data is available at http://groups.di.unipi.it/optimize/Data/MV.html.

[2] The implementation is available at http://sertalpbilal.github.io/randomportfolio/.

TABLE 1. Comparison of running times (in seconds) and objective function values obtained with PADM and Gurobi on five real-world cardinality-constrained portfolio instances.

Instance	n	k	PADM		Big-M	
			Time	Obj. Value	Time	Obj. Value
port1	31	5	0.06	0.000 76	**0.03**	0.000 76
		10	0.02	0.000 75	**0.01**	0.000 75
		20	0.02	0.000 75	**0.01**	0.000 75
port2	85	5	0.34	0.000 25	**0.28**	0.000 24
		10	0.81	0.000 19	**0.12**	0.000 19
		20	0.17	0.000 18	**0.07**	0.000 18
port3	89	5	**0.10**	0.000 31	0.49	0.000 28
		10	**0.82**	0.000 25	0.88	0.000 25
		20	0.18	0.000 24	**0.08**	0.000 24
port4	98	5	**0.94**	0.000 25	8.38	0.000 25
		10	**4.36**	0.000 20	5.63	0.000 20
		20	**0.05**	0.000 18	0.37	0.000 18
port5	225	5	12.83	0.000 36	**0.40**	0.000 36
		10	0.96	0.000 34	**0.35**	0.000 34
		20	0.23	0.000 34	**0.15**	0.000 34

of assets varying from 400 to 5000. Again, we compute the level of expected return R as described above.

5.2.3. *Discussion of the Results.* We start with the discussion of the results for the five real-world instances. We first compute partial minima using the proposed PADM and globally optimal solutions with the big-M formulation (2) using Gurobi without a time limit. In Table 1, we present the instance ID, the size n of the problem, the given cardinality bound k, the solution time (in seconds) required to find a solution, and the objective function value reported by both PADM and Gurobi. For each problem instance, the smaller running time is printed in bold. Regarding the running times, there is no clear trend in general. Both approaches are able to compute a solution very quickly. For some instances, Gurobi is the faster approach while for other instances, PADM is faster. In terms of solution quality, PADM always finds a partial minimum that is very close to the optimal solution or it even finds the optimal solution. In fact, only two instances are not solved to global optimality by PADM. Therefore, we conclude that PADM is competitive to the commercial solver Gurobi on these instances.

We now turn to the discussion of the results for the instances generated in [38]. Since the required diagonal matrices are available for these instances, the perspective reformulation techniques from Section 4 will also be employed. First, we compare our plain PADM with the big-M as well as the MISOCP formulations (the latter two both solved with Gurobi) in terms of running times and objective function values. The results are presented in Table 2, where we display the instance ID (see Section 5.2.2 for the details), the cardinality ceiling k, the average running times (in seconds), and the average objective function values over the 10 instances for each method. The column "Gap" shows the relative gap computed as $(f^{\text{PADM}} - f^{\text{LB}})/|f^{\text{LB}}|$, where f^{PADM} denotes the average objective function value of PADM and f^{LB} denotes the best average objective function value, i.e., the best lower bound obtained either

TABLE 2. Results on the mean-variance portfolio optimization problems generated by [38]. The running times are shown in seconds, and the time limit is 600 s. The column "Gap" shows the relative gap between the average objective function value of PADM and the best lower bound.

		PADM			MISOCP		Big-M	
ID	k	Time	Obj.	Gap	Time	Obj.	Time	Obj.
200^+	5	0.51	422.04	0.02	13.81	413.65	527.75	413.65
200^0		0.72	150.70	0.04	70.16	145.26	600.03	145.26
200^-		0.59	110.74	0.05	131.53	105.94	520.07	105.98
200^+	10	0.58	219.50	0.03	23.41	212.87	600.04	212.92
200^0		0.58	78.61	0.03	287.88	76.03	600.04	76.03
200^-		0.54	59.18	0.05	491.74	56.38	600.04	56.38
200^+	15	0.59	151.56	0.04	31.01	146.31	600.05	146.35
200^0		0.51	54.84	0.03	384.01	53.04	600.04	53.07
200^-		0.56	41.72	0.04	547.66	39.99	600.04	39.99
300^+	5	1.03	619.04	0.02	61.61	609.17	600.06	609.17
300^0		1.09	223.20	0.03	349.67	217.77	600.05	217.73
300^-		1.04	176.65	0.05	382.18	168.66	432.43	168.55
300^+	10	1.04	316.45	0.02	242.60	309.79	600.05	309.78
300^0		1.28	116.68	0.04	570.89	112.80	600.05	112.60
300^-		1.49	93.70	0.06	505.59	88.78	544.30	88.64
300^+	15	1.05	215.61	0.03	235.08	209.97	600.05	210.04
300^0		0.89	80.58	0.04	579.31	77.68	600.05	77.62
300^-		0.91	66.11	0.07	527.58	62.15	545.99	62.03
400^+	5	1.36	848.43	0.03	332.55	820.23	542.89	820.20
400^0		1.33	296.64	0.03	576.75	288.00	600.07	287.74
400^-		1.37	215.27	0.05	602.26	206.32	600.07	205.97
400^+	10	1.81	435.12	0.04	534.34	418.24	600.07	417.67
400^0		1.67	153.95	0.04	602.42	148.36	600.07	147.77
400^-		1.75	113.41	0.06	602.16	107.21	600.07	106.58
400^+	15	1.73	297.04	0.04	488.91	284.76	600.08	284.66
400^0		1.71	106.19	0.05	601.92	101.64	600.06	101.29
400^-		1.75	78.22	0.06	601.92	73.79	600.07	73.54

from the MISOCP or the big-M formulation. Taking a closer look at the gap values, we can see that the PADM computes feasible points with objective function values very close to the optimal solution or to the best known lower bound—the largest gap is 7 %. Thus, the solutions of PADM are almost as good as the solutions of the exact methods. Regarding running times, we can directly see that PADM always finds a feasible solution very quickly, never requiring more than 2 seconds on average. The MISOCP formulation solved with Gurobi is able to find the optimal solution in a reasonable amount of time, especially for the diagonally dominant instances, but as the size of the instances grows, the required solution time also increases. In contrast, Gurobi applied to the big-M formulation almost always reaches the time limit and, thus, is outperformed by the MISOCP formulation as expected. We conclude from this experiment that the PADM performs very well across all variants of diagonal dominance properties of the covariance matrices.

As discussed in Section 4.1, the PADM may be improved by exploiting perspective reformulations. Our expectation is that this particularly might lead to improvements in terms of the objective function values for those cases for which the covariance matrices are diagonally dominant. Therefore we also test the Perspective-PADM variant on the same instances as discussed in Table 2. However, as shown in Table 3, the solution process of the PADM is significantly harmed by using these ideas: Although some of the obtained objective function values are better when incorporating the perspective formulation, the running times slow down significantly. What happens is that the two subproblems in each iteration are much harder to solve now. As a consequence, we are no longer able to compute a feasible solution for all of the instances. The percentage of instances (among the 10 for each scenario) for which the PADM could find a feasible solution is shown in the column "Feasible" of Table 3. Note that only for the diagonally dominant instances with 200 assets and $k = 5$ the PADM was able to find feasible solutions for all of the 10 instances. Nevertheless, for the same scenario, the relative gap (last column) is worse (compared to Table 2, the relative gap is 4 % vs. 2 %). Thus, the numerical results show that the Perspective-PADM variant does not seem to be beneficial. We tried to further improve on this in several ways. First, we observed that the constraint $r^\top x \geq R$ is very hard to be satisfied in the w-space, leading to very many inner ADM iterations. We tried to overcome this issue by including this constraint also in the subproblem (14). However, this was only beneficial for a few instances. Second, we scaled the diagonal matrices by a factor < 1 so that its effect on the subproblems is smaller. However, again, this was only beneficial for a few instances. Another idea for an improvement that could be tried is to calibrate a parameter that controls the balancing between feasibility and optimality in Subproblem (14). However, this would require many experiments to find a parameter that works for all of the instances—if this is possible at all. For this reason and because we are already satisfied with the results shown in Table 2, we refrained from executing additional adjustments.

Let us now discuss the results for the large-scale and randomly generated instances. Here, we set the time limit to 1 h. The results are shown in Table 4, where we report averages since we randomly generated 10 instances for each value of n. Again, we indicate the faster running time in bold. First, it can be seen that by using the big-M formulation, Gurobi is not able to compute an optimal solution within the time limit for all instances with $k \in \{5, 10\}$. Thus, smaller values of k seem to make the problem much harder from the point of view of solving it to global optimality. In contrast, PADM never reaches the time limit and always returns a feasible point of good quality very fast for these instances. For the instances with $k = 50$, PADM is much faster than Gurobi and the reported objective function values are the same. Thus, PADM is able to compute globally optimal solutions for $k = 50$. Larger values of k, i.e., $k \in \{100, 200\}$, also makes the problem easier to solve for PADM. For these k, Gurobi is the faster method, but also PADM always finds the globally optimal solution within reasonable time. Possibly, an MISOCP formulation would perform better than the big-M formulation if the covariance matrices of these instances are diagonally dominant. However, in this work, we refrain from testing an MISOCP formulation on these instances, since this would require computing proper diagonal matrices D, which is not at the core of this paper.

Our further computational analysis has shown that Gurobi often needs a lot of time to prove global optimality of an already found feasible solution. Thus, we also run Gurobi on the large-scale and randomly generated instances setting the time limit to the time that PADM required to compute a partial minimum. Afterward, we

TABLE 3. Results of Perspective-PADM on the portfolio optimization problems generated by [38]. The column "Feasible" shows the percentage of instances for which Perspective-PADM was able to find a feasible solution. For those instances, we present the average running times (in seconds) and average objective function values. The time limit is 600 s. The column "Gap" shows the relative gap between the average objective function value of Perspective-PADM and the best lower bound (the latter shown in Table 2).

		Perspective-PADM			
ID	k	Feasible (%)	Time	Obj.	Gap
200^+	5	100	61.84	431.97	0.04
200^0		80	53.59	148.70	0.02
200^-		90	117.73	109.75	0.04
200^+	10	80	95.13	222.24	0.04
200^0		90	126.93	77.13	0.01
200^-		80	151.76	58.09	0.03
200^+	15	80	23.89	147.39	0.01
200^0		90	47.19	53.64	0.01
200^-		80	12.76	40.69	0.02
300^+	5	70	123.73	613.86	0.01
300^0		70	125.70	228.35	0.05
300^-		50	89.28	171.56	0.02
300^+	10	50	149.34	309.86	0.00
300^0		80	228.24	117.40	0.04
300^-		60	140.11	92.82	0.05
300^+	15	60	187.56	210.66	0.00
300^0		50	34.29	80.84	0.04
300^-		40	97.74	62.56	0.01
400^+	5	70	54.88	867.89	0.06
400^0		70	94.68	304.51	0.06
400^-		70	52.64	218.95	0.06
400^+	10	50	260.40	457.56	0.1
400^0		70	272.18	160.22	0.08
400^-		60	252.50	109.24	0.03
400^+	15	30	107.03	313.83	0.1
400^0		10	33.79	102.71	0.01
400^-		30	392.40	76.00	0.03

compare the best solution found by Gurobi within this time limit with the solution computed by PADM. This setting provides a reasonable comparison between our primal heuristic and primal heuristics implemented in Gurobi. The results are shown in the last column of Table 4, where we report the ratio between the objective function value at the partial minimum obtained with PADM and the objective function value of the best feasible point that Gurobi found within this time limit. We observe that when k is small, Gurobi is also able to give a feasible solution and it is slightly better than the feasible solution given by PADM. However, the deviation of these ratios from 1 is almost in the range of numerical tolerances. For the remaining instances, both give the same solution.

TABLE 4. Performance of PADM and big-M formulation solved with Gurobi on large-scale randomly generated cardinality-constrained portfolio instances. The time limit is 1 h for the results in columns 3–7. The last column compares the objective function values obtained by Gurobi within the time limit set to the time required by PADM to compute a partial minimum.

n	k	PADM		Big-M			PADM/Big-M
		Time	Obj.	Time	Obj.	Gap (%)	
400	5	**1.37**	20.63	> 3600	20.54	4.84	1.0045
	10	**6.47**	19.74	> 3600	19.71	1.85	1.0016
	50	**0.68**	19.25	8.46	19.25	0.01	1.0000
	100	0.28	19.25	**0.25**	19.25	0.00	1.0000
	200	0.28	19.25	**0.26**	19.25	0.00	1.0000
600	5	**1.96**	20.52	> 3600	20.45	5.49	1.0033
	10	**14.78**	19.64	> 3600	19.61	2.06	1.0017
	50	**1.70**	19.14	390.94	19.14	0.01	1.0000
	100	0.52	19.13	**0.47**	19.13	0.00	1.0000
	200	0.50	19.13	**0.49**	19.13	0.00	1.0000
1000	5	**10.11**	20.50	> 3600	20.40	6.16	1.0047
	10	**39.14**	19.57	> 3600	19.54	2.30	1.0017
	50	**3.40**	19.05	753.16	19.05	0.01	1.0000
	100	1.44	19.04	**1.11**	19.04	0.00	1.0000
	200	1.42	19.04	**1.24**	19.04	0.00	1.0000
2000	5	**63.64**	20.33	> 3600	20.25	6.68	1.0041
	10	**212.16**	19.41	> 3600	19.39	2.58	1.0014
	50	**14.44**	18.88	2654.62	18.88	0.02	1.0000
	100	8.07	18.88	**3.94**	18.88	0.00	1.0000
	200	8.13	18.88	**3.88**	18.88	0.00	1.0000
3000	5	**115.194**	20.20	> 3600	20.16	6.74	1.0017
	10	**915.89**	19.35	> 3600	19.31	2.63	1.0022
	50	**96.96**	18.80	2828.98	18.80	0.02	1.0000
	100	21.53	18.79	**7.96**	18.79	0.00	1.0000
	200	21.49	18.79	**8.14**	18.79	0.00	1.0000
5000	5	**378.63**	20.15	> 3600	20.05	6.78	1.0050
	10	**1679.16**	19.22	> 3600	19.20	2.68	1.0015
	50	**126.31**	18.69	> 3600	18.69	0.03	1.0000
	100	42.58	18.68	**13.41**	18.68	0.00	1.0000
	200	42.63	18.68	**13.32**	18.68	0.00	1.0000

From this experiment, we conclude that PADM is competitive to the commercial solver Gurobi on the tested cardinality-constrained portfolio optimization problems. For k not being too small, PADM always computed the global optimum and for k being not too large, it is even faster than Gurobi. However, one of course has to mention that PADM does not provide any guarantee that the solution is globally optimal.

5.3. **Numerical Results for the Best Subset Selection Problem.** In this section, we present and discuss the results for the best subset selection problem.

Similarly to Section 5.2, we first discuss the issue of selecting a good initial penalty parameter in Section 5.3.1, then explain the test data setup in Section 5.3.2, and finally discuss the results in the remaining part of this section.

5.3.1. *Selection of the Initial Penalty Parameter.* In the case of the best subset selection problem, the PADM resembles Lasso in the first part of its iterations. Since Lasso is known to produce sparse solutions, it can happen that $|\operatorname{supp}(\gamma^{0,1})| < k$ holds if the initial penalty parameter is chosen too large. It can easily be seen that each further iteration either decreases the number of non-zero entries or keeps it unchanged. Hence, once a solution $\gamma^{t,l}$ has less than k non-zero entries, the final result will have less than k non-zero values as well.

However, there always exists a global optimal solution to (BSS) that has exactly k non-zero values. Moreover, if a solution with cardinality less than k is globally optimal, then either $\operatorname{rank}(X) < k$ holds or a subset $S \subseteq [p]$ exists with $|S| < k$ such that $y \in \operatorname{span}(X_S)$ holds for the response vector y. Therefore, if these unusual cases are not given, having a cardinality less than k means that the result cannot be optimal.

That is why we want to avoid starting with a penalty parameter that is too large. Fortunately, many applications for which Lasso is conventionally used require the computation of the entire λ-path, i.e., all solutions for all $\lambda > 0$ or, at least, for a large sample of λ values. Consequently, most Lasso solvers are specialized to compute the λ-path. In particular, the LARS method computes the λ-path for $[\lambda', \infty)$ as a side product of just solving (LASSO) for λ'.

For computing the initial penalty parameter, we hence proceed as follows. We first compute the λ-path and search for the largest λ that yields a solution with k or more non-zero entries. Since the penalty parameter controls the balance between optimality and feasibility in the PADM, we want to avoid a penalty parameter that directly leads to a feasible solution. Instead, our preliminary numerical experiments revealed that setting $\mu = 0.3\,\lambda$ results in the overall best performance of the method. After that, we double the penalty parameter in each outer iteration.

5.3.2. *Description of the Test Set.* For the test data, we replicate the standard setup as seen in [13, 49] and create the test data for the best subset selection problem as follows. First, we synthetically generate the design matrix $X \in \mathbb{R}^{n\times p}$, true coefficients $\beta^0 \in \mathbb{R}^p$, and noise $\varepsilon \in \mathbb{R}^n$ as described in detail below. Then, the response $y \in \mathbb{R}^n$ is computed by $y = X\beta^0 + \varepsilon$. This has the advantage that we know the true predictors and that we can examine the selection accuracy of the methods, i.e., we can assess how many correct non-zero coefficients are chosen. We generate the following instance sizes for $n > p$:

- dim-small: $n = 180$, $p = 60$, $\|\beta^0\|_0 = 30$,
- dim-medium: $n = 300$, $p = 100$, $\|\beta^0\|_0 = 50$,
- dim-large-1: $n = 1500$, $p = 500$, $\|\beta^0\|_0 = 250$,
- dim-large-2: $n = 6000$, $p = 2000$, $\|\beta^0\|_0 = 600$,
- dim-large-3: $n = 24\,000$, $p = 8000$, $\|\beta^0\|_0 = 2500$,
- dim-huge: $n = 45\,000$, $p = 15\,000$, $\|\beta^0\|_0 = 5000$.

We deliberately choose $n = 3p$ for all these settings such that the predictive qualities are comparable between the setups. Furthermore, we study the following degenerated instances where $p > n$ holds:

- dim-deg-1: $n = 200$, $p = 1000$, $\|\beta^0\|_0 = 100$,
- dim-deg-2: $n = 200$, $p = 10\,000$, $\|\beta^0\|_0 = 100$.

According to the setting, we draw each row of X i. i. d. from $N_p(0, I)$, where N_p denotes the multivariate normal distribution. Note that the correlation matrix I can be a parameter as well. However, preliminary results showed that the correlation

does not seem to have any notable effect on the relative performance of the examined methods. Hence, we only consider the identity correlation matrix. We then draw $\beta^0 \in \mathbb{R}^p$ subject to the sparsity condition for the respective setting's dimension. To this end, if k is the number of desired non-zero values, a subset of size k is uniformly sampled from $[p]$ and the entries of β^0 are set to 1 if the respective index is an element of the sampled subset or 0 otherwise. In other words, we draw a subset $S \subseteq [p]$ with $|S| = k$ and create the entries of β^0 according to the rule

$$\beta_i^0 := \begin{cases} 1, & \text{if } i \in S, \\ 0, & \text{otherwise.} \end{cases}$$

After creating the coefficients, we generate the noise ε added to $X\beta^0$, which is drawn i. i. d. from the multivariate normal distribution $N_n(0, \sigma^2 I)$, with σ^2 chosen in accordance to the signal-to-noise ratio (SNR). Low SNR means that there is a high disturbance in the signal. Otherwise, if the SNR is high, noise is relatively low. The ratio is defined as

$$\text{SNR} := \frac{\text{Var}(x^0\beta^0)}{\text{Var}(\varepsilon)} = \frac{\|\beta^0\|_2^2}{\sigma^2}.$$

For dim-small and dim-medium, we consider the SNR values $\text{SNR} \in \{0.05, 0.3, 1, 6\}$ and for dim-large and dim-huge, we only assess $\text{SNR} = 1$.

Moreover, for each setting except for dim-huge, we generate 50 instances. Due to its size, we only examine 5 instances for dim-huge.

We compare the following four approaches:

- PADM, as proposed in this article,
- CPLEX applied to compute an exact solution,
- the MaxMin method as proposed in [14], and
- L0Learn as discussed in [28, 50].

For the global solution, we are using the big-M reformulation with M chosen as in [54]. The third method is a max-min approach, which can be stated as a second-order cone program. This heuristic is both known to be very efficient and to yield feasible points of good quality. We will refer to the heuristic proposed in [14] as the MaxMin method. Finally, the L0Learn method is a well-known state-of-the-art heuristic providing fast and good heuristic solutions. However, the method does not aim to solve the cardinality-constrained best subset selection problem as discussed in this article. Instead, it tries to find a solution to the regularized best subset selection problem

$$\min_{\beta \in \mathbb{R}^p} \|X\beta - y\|_2^2 + \kappa |\text{supp}(\beta)|,$$

where $\kappa > 0$ is a regularization parameter. While both the truncated and the regularized problem aim to induce sparsity in the coefficients, there is no one-to-one connection between the sparsity parameter k and the regularization parameter κ. Indeed, there exist cases for which certain cardinalities cannot be generated by the regularized best subset selection, no matter how κ is chosen [54]. Hence, L0Learn does not exactly fit our setting, and is, thus, inherently at a disadvantage. L0Learn provides a parameter for setting the maximum cardinality. We utilize this functionality to compute a κ-path and then pick the solution with the largest sparsity—ideally equal to k. In addition, we also evaluate L0Learn using a more sophisticated and finetuned setting. In this case, the maximum support is set to $2k$, we set the `scaleDownFactor` to 0.95, and the size of the λ grid to $\lfloor \frac{k}{2} \rfloor$. This finetuned variant yields solutions with support larger than k. Hence, from the solution path we manually select the largest subset smaller or equal to k. We will

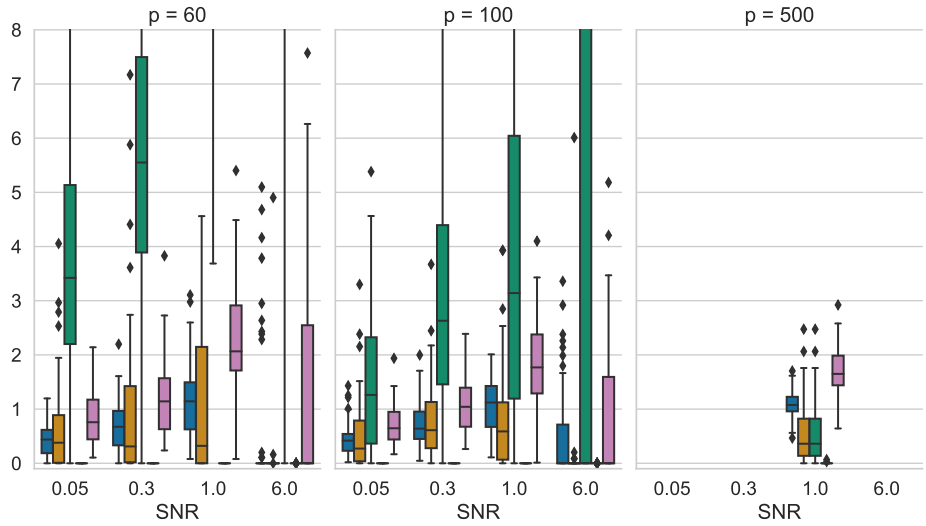

FIGURE 1. Relative gap (in %) of the methods PADM (blue), L0Learn (yellow), finetuned L0Learn (green), MIP (orange), and MaxMin (purple) for the settings dim-small, dim-medium, and dim-large-1.

later see that this finetuning of parameters is especially tailored for large values of p, whereas it might lead to an inferior behavior for smaller instances.

For all four approaches we set a time limit of 1 h and a termination tolerance of 1×10^{-8}. We always set k to the number of correct predictors.

5.3.3. *Solution Quality.* We now start by discussing the quality of the feasible solutions provided by PADM. To this end, we compare the objective function values of the heuristics with the exact solution.

Since the best subset selection problem is very hard to solve to global optimality, the dim-small instances are the only ones for which we can compute provably optimal solutions using the global solver. For the other cases we compare the relative gap between the different approaches. In particular, for each instance we determine the smallest residual sum of squares of all approaches for the particular instance. We denote this minimum value by $\widehat{\text{RSS}}$. Then, assuming RSS is the residual sum of squares for one of the approaches, the gap is defined by

$$\frac{\text{RSS} - \widehat{\text{RSS}}}{\text{RSS}} \cdot 100.$$

Clearly, a gap of 0 % means that the approach yields the best solution and, in the case of dim-small, provably the globally optimal point. A higher gap indicates that the method returned an inferior solution.

Looking at Figure 1 we find that the solutions produced by MaxMin have the highest gap for $p = 500$ whereas the finetuned version of L0Learn produces the highest gaps for $p \in \{60, 100\}$. Although we are not able to receive provably optimal solutions from the MIP approach for every setting, solving an MIP provides the best objective values. Comparing L0Learn and PADM we observe that, in average, PADM provides slightly worse solutions—however, with less variance compared to L0Learn. Furthermore, it seems that PADM performs better on noisy data and

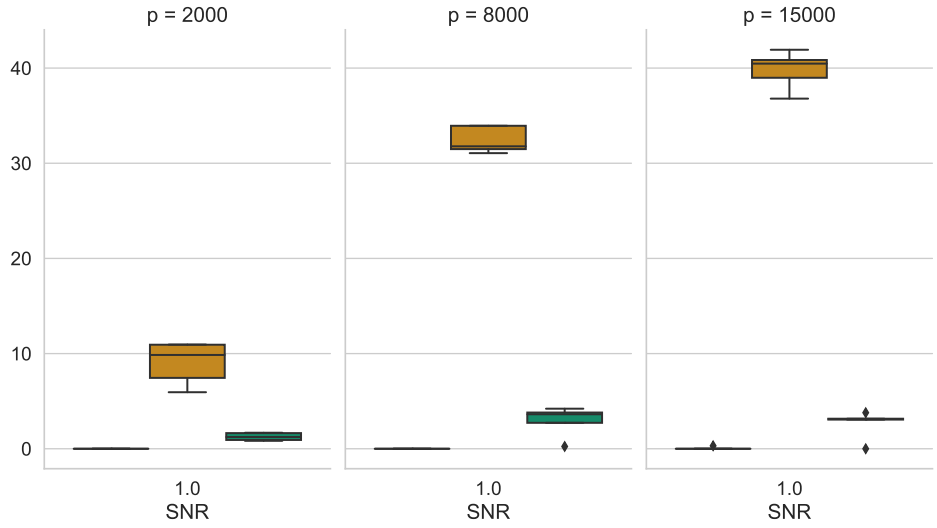

FIGURE 2. Relative gap (in %) of the methods PADM (blue), L0Learn (yellow), and finetuned L0Learn (green) for the settings dim-large-2, dim-large-3, and dim-huge.

looses some ground when noise is low. These differences are, however, in a range of $< 1\,\%$, which is rather small in absolute terms.

The picture changes when we consider larger instances; see Figure 2. The plot shows the relative gaps between PADM, L0Learn, and finetuned L0Learn for the settings dim-large-2, dim-large-3, and dim-huge. We see that with increasing dimension the solutions produced by L0Learn become significantly worse and can reach a gap of around $40\,\%$ in relation to PADM. On the other hand, the finetuned version of L0Learn now performs much better compared to the results for smaller p as shown in Figure 1. The finetuned version reaches gaps that are larger than those of PADM but is never worse than $5\,\%$ in relation to PADM.

5.3.4. *Running Times.* In summary, we conclude that PADM provides points very close to the global optimum and very competitive values compared to MaxMin and (finetuned) L0Learn. In the following we want to study the computational efficiency of the PADM. Since the subproblems in our proposed approach can be solved in a highly efficient way, we expect our proposed method to have competitive running times.

We compare the running times via performance profiles as introduced in [31]. That is, for an instance i we have the running times t_i^m for each method $m \in M = \{\text{PADM, L0Learn, finetuned L0Learn, MaxMin, MIP}\}$. We then compute the performance ratio

$$\frac{t_i^m}{\min\{t_i^a : a \in M\}}$$

for each $m \in M$ and plot the resulting data via an ECDF plot.

Figure 3 shows the performance profiles. We see that L0Learn dominates all other approaches as it is always the fastest approach. Our proposed approach comes second, closely followed by the finetuned L0Learn method. MaxMin comes fourth, and the exact approach fifth. About $50\,\%$ of all instances could not be solved within

TABLE 5. Average running times in seconds. Dashes indicate that the experiments were not conducted because they failed due to the time limit (MaxMin) or already reached the time limit in smaller experiments (MIP). For the degenerated cases only L0Learn and PADM are applicable.

Instance	MIP	MaxMin	L0Learn	finetuned L0Learn	PADM
dim-small	58.446	0.032	**0.007**	0.015	0.010
dim-medium	980.111	0.076	**0.013**	0.022	0.023
dim-large-1	3600.01	5.87	**0.39**	0.83	0.58
dim-large-2	–	219.70	**3.27**	11.03	10.77
dim-large-3	–	–	**30.67**	188.82	258.85
dim-huge	–	–	**114.63**	1212.35	994.37
dim-deg-1	–	–	**0.10**	**0.10**	0.23
dim-deg-2	–	–	0.45	**0.36**	0.80

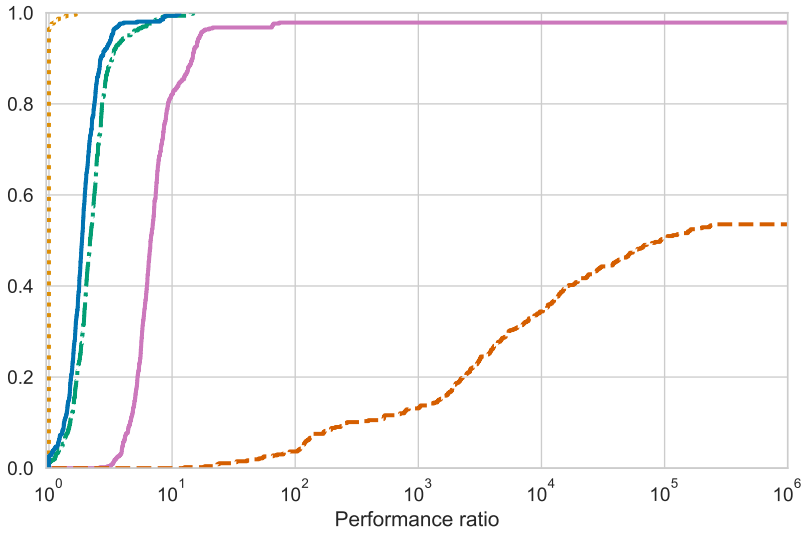

FIGURE 3. Performance profile for PADM (blue), L0Learn (yellow), finetuned L0Learn (green), MIP (orange), and MaxMin (purple).

the time limit by the exact approach and a small amount of instances could also not be solved within the time limit by MaxMin.

Additionally, Table 5 shows the average running times for the different instances. We can see a similar pattern: L0Learn consistently outperforms PADM, while PADM outperforms MaxMin and the exact solver by large margins. Moreover, the finetuned version of L0Learn and PADM have rather comparable running times. However, it seems that L0Learn's performance comes at a cost. The difference in running times becomes especially apparent in higher dimensions. In the last section, we have already seen that solutions become significantly worse for L0Learn in these settings and we will see that this effect also occurs when considering the statistical quality and the sparsity of the solutions. On the other hand, PADM and the finetuned version of L0Learn are both comparable w.r.t. running times and the

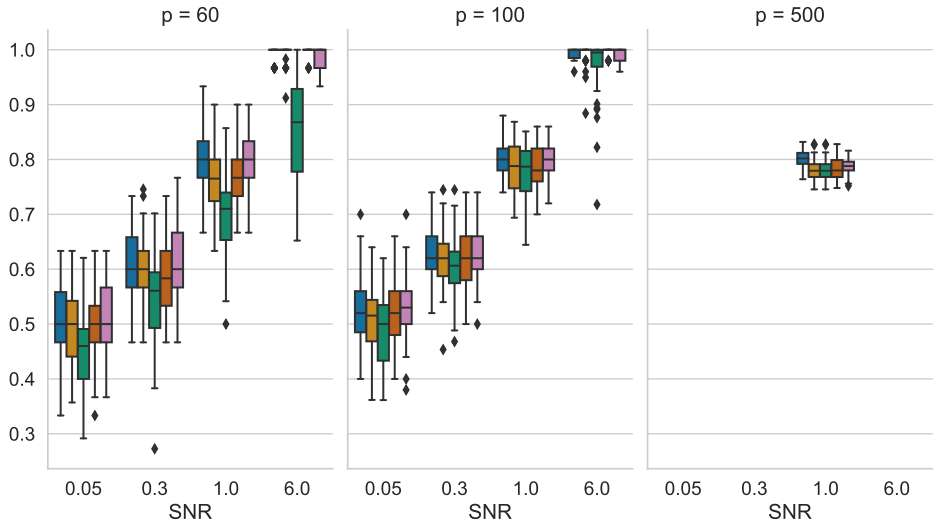

FIGURE 4. Selection accuracy values between PADM (blue), L0Learn (yellow), finetuned L0Learn (green), the exact solver (orange), and MaxMin (purple) for dim-small, dim-medium, and dim-large-1.

quality of the solutions. However, for both measures PADM performs slightly better than the finetuned version of L0Learn and has the additional advantage that no finetuning is required.

5.3.5. *Statistical Performance.* Finally, we discuss the selection accuracy of the three methods. The selection accuracy is the percentage of correctly selected predictors. To this end, suppose that $S \subseteq [p]$ is the set of selected indices and that $S_0 \subseteq [p]$ is the true set of indices, i.e., S_0 is given by $S_0 = \text{supp}(\beta^0)$. Then, the selection accuracy is defined by

$$1 - \frac{|S \,\Delta\, S_0|}{|S| + |S_0|},$$

where Δ is the symmetric difference. Although the objective function value and the selection accuracy could be correlated, they do not have to be. That is because the solution of the optimization problem depends on the given training data. However, it, of course, does not depend on new, i.e., unknown, data to which it later is applied to. It might select a subset that indeed is optimal, but only because the noise interference caused it to be optimal. Hence, it also makes sense to consider statistical performance, for example indicated by the selection accuracy. For all our tested methods we set k to the correct sparsity.

In Figure 4 we can observe that PADM is highly competitive and in some cases it even yields the most accurate selection. This is, e.g., the case for the dim-large-1 instances. Interestingly, we have previously seen that, in this setting, PADM provides solutions that are slightly worse than those of L0Learn w.r.t. the objective function value. Nevertheless, the selection accuracy is superior. One possible explanation might be that this is a side-effect of the connection between our method and (trimmed) Lasso. Specifically, it is shown in [6, 7] that both Lasso and trimmed Lasso have a robustification effect, which can positively affect the prediction quality. Since each inner loop of the PADM algorithm is aiming to solve a trimmed Lasso

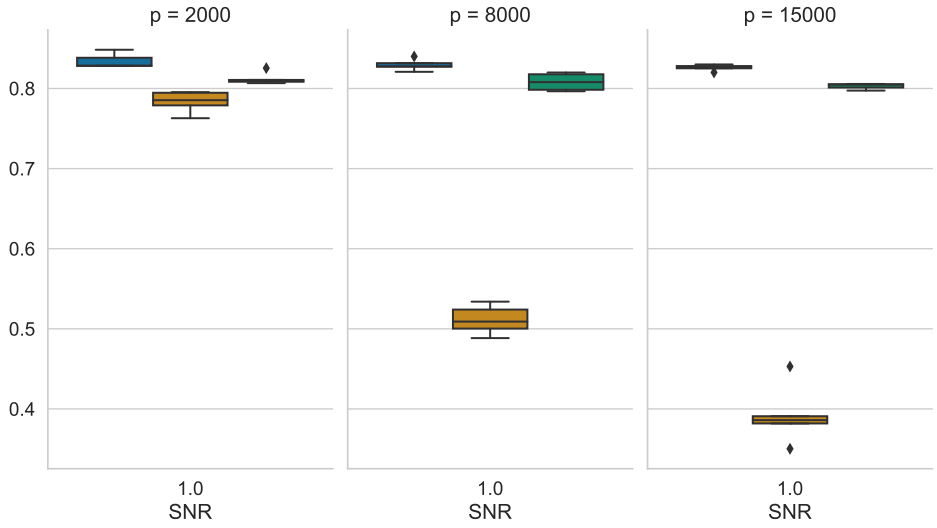

FIGURE 5. Selection accuracy values between PADM (blue), L0Learn (yellow), and finetuned L0Learn (green) for dim-large-2, dim-large-3, and dim-huge.

problem (or a Lasso problem in the very first iteration), it might happen that PADM returns a subset for which the selection is influenced by the robustification effects caused by Lasso and trimmed Lasso. A detailed analysis of this hypothesis could be further examined in future research but is out of scope of this work.

Even more interesting is the selection behavior for large scale instances. In Figure 5 we compare the selection accuracy of PADM and (finetuned) L0Learn for the instances dim-large-2, dim-large-3, and dim-huge. While PADM keeps a steady selection accuracy of about 80 %, the accuracy of L0Learn falls off sharply with increasing dimension. The accuracy of the finetuned version of L0Learn is also around 80 % but slightly worse than the one of PADM. The observation is consistent with the results for the relative gap, which also becomes worse for L0Learn in higher dimensions. We have noted that the sparsity of L0Learn cannot be controlled directly. Hence, we suspected that the decrease in selection accuracy and objective quality could be linked to a decrease of the support size. Indeed, in Figure 6 we see the relative support size $|\text{supp}(\beta)|/k$ for the high-dimensional settings. Clearly, a value of 1 means that the method yields a support of correct size and a value smaller than 1 means that the method only yields a proportion of the aimed sparsity. We can see that with increasing dimension the relative support size produced by L0Learn decreases while it is almost perfect for its finetuned version.

Finally, we consider the degenerated cases dim-deg-1 and dim-deg-2. Here, we cannot observe that L0Learn suffers from a significantly smaller support. Nevertheless, we notice a different behavior for the PADM and L0Learn for these instances.

Figure 7 shows the plots for the gaps and the selection accuracy for the degenerated cases. We see that PADM has a significantly worse residual sum of squares, i.e., much worse objective values. The same is true for the finetuned version of L0Learn for $p = 10\,000$. Yet, PADM selects predictors with much higher accuracy than (finetuned) L0Learn. Since our overall objective with the best subset selection is to find a well-suited prediction model, the selection accuracy is the more important metric for the end result. Our guess is that this behavior is, once again, due to

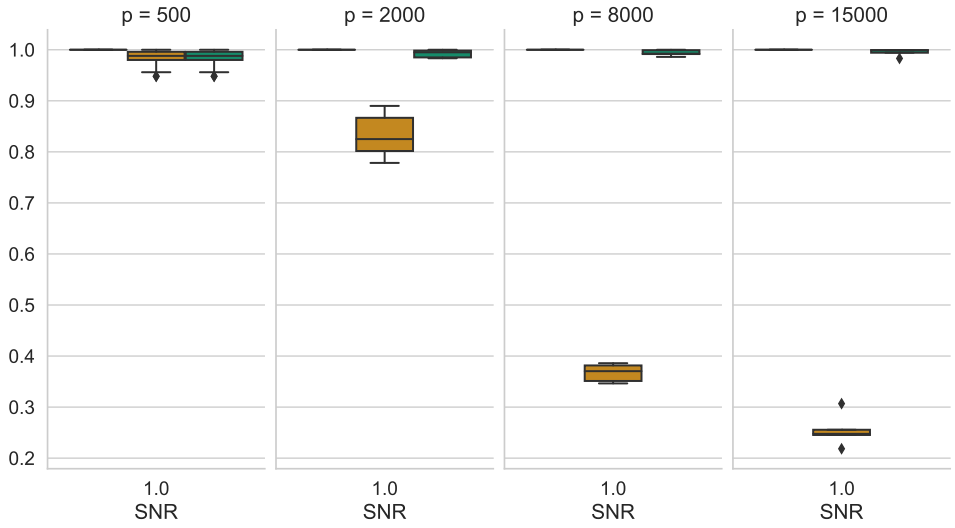

FIGURE 6. Relative support size $|\text{supp}(\beta)|/k$ between PADM (blue), L0Learn (yellow), and finetuned L0Learn (green) for dim-large-1, dim-large-2, dim-large-3, and dim-huge.

the Lasso and trimmed Lasso method. After all, it is comparable to having a regularization. With a regularization we pay the price of a worse objective value but gain improved predictive performance.

Our proposed method delivers very good results. The PADM provides feasible points that are very close to a global optimum in a very fast manner. That is, we have measured running times magnitudes faster on average than the MaxMin approach and while solving the problem to global optimality can take between hundreds and thousands of seconds, PADM stays under a second for instances up to dim-large-1. However, compared to the prominent L0Learn approach, our proposed methods can be up to 30 times slower. Nevertheless, we have seen that this performance advantage comes at a cost, i.e., for higher dimensions L0Learn declines immensely in solution quality (or needs to be finetuned, which requires detailed knowledge about the method itself) whereas PADM stays consistent. Moreover, the relative gap for PADM compared to the MIP approach is in the range of a single digit and oftentimes less than 1 %.

Furthermore, we have shown that our approach is able to solve instances with 15 000 variables in about 15 min with consistently good results and without compromises in selection accuracy.

6. Conclusion

In this paper, we applied a penalty alternating direction method to two famous instantiations of cardinality-constrained optimization problems: (i) the cardinality-constrained portfolio optimization problem and (ii) the best subset selection problem. The decomposition of these problems along their discrete-continuous structure allows to alternatingly solve two subproblems that are much easier to solve, while the convergence of the iterates is ensured by a classic penalty framework. The numerical results are very convincing. For cardinality-constrained portfolio optimization, our method is competitive with highly evolved commercial solvers. Interestingly,

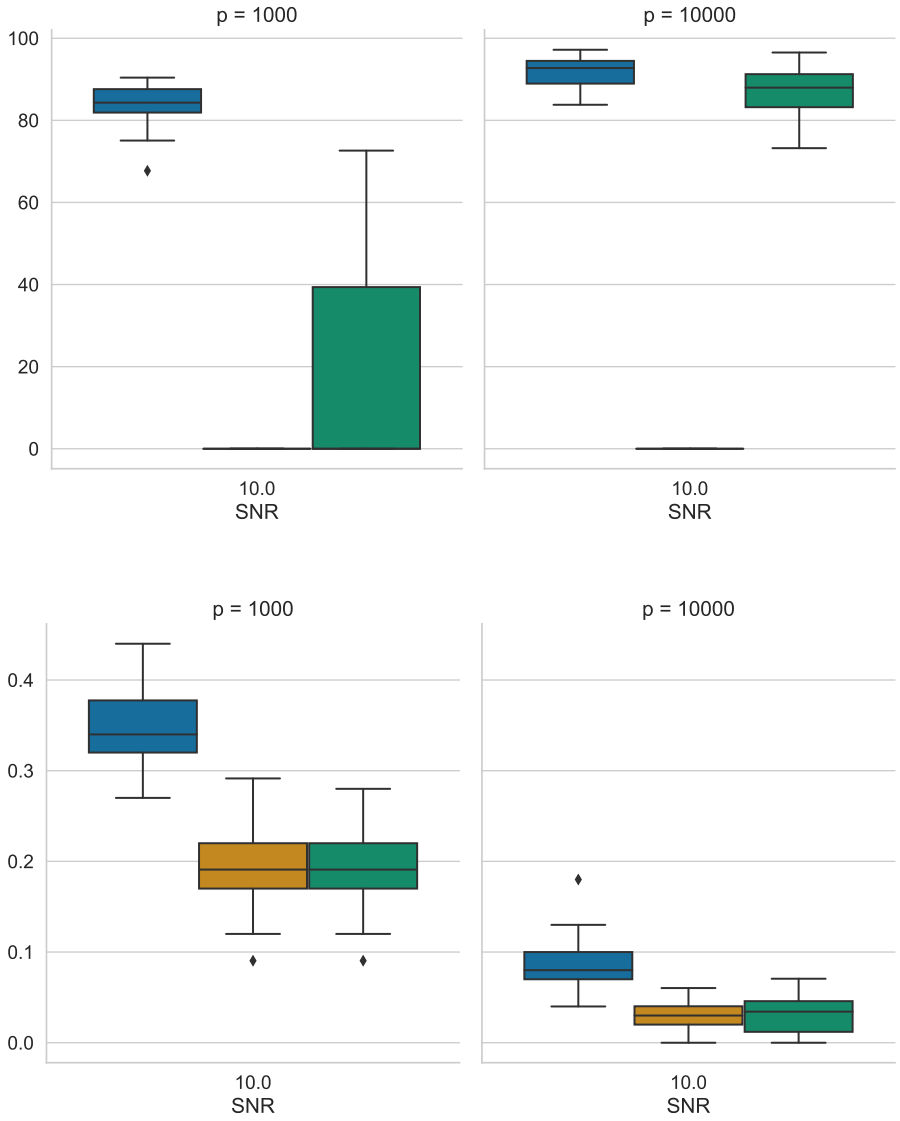

FIGURE 7. Gap (top) and selection accuracy (bottom) between PADM (blue), L0Learn (yellow), and finetuned L0Learn (green) for dim-deg-1 and dim-deg-2.

another variant of our method that uses a perspective reformulation of the portfolio optimization problems only improved the obtained objective function values while it significantly slows down the method. For the best subset selection problem, we compared our method with an exact approach and two other heuristic approaches that are state-of-the-art in the current literature. Our numerical study shows that we are at least competitive to these methods. In particular, the statistical quality of our solutions are superior to those of the other methods.

Let us finally mention some open topics for future research. First, we "only" considered two exemplary instantiations of cardinality-constrained models and further examples can be studied as well. Moreover, one could also embed the

proposed method as a primal heuristic in a branch-and-bound algorithm to further enhance the solution process for obtaining globally optimal solutions.

Acknowledgments

The first author thanks the DFG for their support within RTG 2126 "Algorithmic Optimization". The third author thanks the DFG for their support within project A05 and B08 in the "Sonderforschungsbereich/Transregio 154 Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks". Finally, we are very grateful to two anonymous reviewers whose comments contributed significantly to improving the quality of this paper.

Appendix A. Omitted Proofs and Remarks

A.1. **Proof of Proposition 5.**

Proof. If $e^\top \bar{x}_S = 1$ we have proven the result because $\tilde{w}_S = \bar{x}_S$ must hold for $\tilde{w}$ to be an optimal solution. Hence, we only have to consider the case $e^\top \bar{x}_S < 1$ because $e^\top \bar{x}_S > 1$ is infeasible. First, we show that $\tilde{w}_i \geq \bar{x}_i$ holds for all $i \in S$.

Note that if $e^\top \tilde{w} = e^\top \tilde{w}_S = 1$ and $e^\top \bar{x}_S < 1$ holds, then there must be at least one index $i \in S$ with $\tilde{w}_i > \bar{x}_i$. By contradiction, assume that there also exists an index $j \in S$ with $\tilde{w}_j < \bar{x}_j$. Next, we define $\gamma := \min\{\tilde{w}_i - \bar{x}_i, \bar{x}_j - \tilde{w}_j\} > 0$ and consider $\hat{w} \in \mathbb{R}^n$ such that

$$\hat{w}_l := \begin{cases} \tilde{w}_i - \gamma, & \text{if } l = i, \\ \tilde{w}_j + \gamma, & \text{if } l = j, \\ \tilde{w}_l, & \text{otherwise,} \end{cases} \tag{15}$$

holds. Clearly, $e^\top \hat{w} = 1$ and $\operatorname{supp}(\hat{w}) = S$ hold. Thus, $\hat{w}$ is feasible for (7). Furthermore, it holds that $\hat{w}_i \geq \bar{x}_i$ and $\hat{w}_j \leq \bar{x}_j$. From this and (15), it follows

$$\begin{aligned} \|\hat{w} - \bar{x}\|_1 &= \sum_{l=1}^{n} |\hat{w}_l - \bar{x}_l| = \sum_{\substack{l=1 \\ l \notin \{i,j\}}}^{n} |\hat{w}_l - \bar{x}_l| + |\hat{w}_i - \bar{x}_i| + |\hat{w}_j - \bar{x}_j| \\ &= \sum_{\substack{l=1 \\ l \notin \{i,j\}}}^{n} |\hat{w}_l - \bar{x}_l| + \hat{w}_i - \bar{x}_i + \bar{x}_j - \hat{w}_j \\ &= \sum_{\substack{l=1 \\ l \notin \{i,j\}}}^{n} |\tilde{w}_l - \bar{x}_l| + \tilde{w}_i - \gamma - \bar{x}_i + \bar{x}_j - \tilde{w}_j - \gamma \\ &= \|\tilde{w} - \bar{x}\|_1 - 2\gamma < \|\tilde{w} - \bar{x}\|_1, \end{aligned}$$

which is a contradiction to $\tilde{w}$ being an optimal solution of (7). Thus, $\tilde{w}_i \geq \bar{x}_i$ holds for all $i \in S$.

Consider now $w^* \in \mathbb{R}^n$ with

$$w^*_l := \begin{cases} \bar{x}_l / (e^\top \bar{x}_S), & \text{if } l \in S, \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, w^* is feasible for (7) and $w^*_i \geq \bar{x}_i$ holds for all $i \in S$. We are now ready to show that w^* is also an optimal solution of (7):

$$\begin{aligned} \|\tilde{w} - \bar{x}\|_1 &= \sum_{l \notin S} |\bar{x}_l| + \sum_{l \in S} \tilde{w}_l - \bar{x}_l = \sum_{l \notin S} |\bar{x}_l| + 1 - e^\top \bar{x}_s \\ &= \sum_{l \notin S} |\bar{x}_l| + \sum_{l \in S} w^*_l - \bar{x}_l = \|w^* - \bar{x}\|_1. \end{aligned}$$

□

A.2. **Proof of Proposition 6.**

Proof. Let $S^* = \{i_1, \ldots, i_k\}$ and $w^* \in \mathbb{R}^n$ with

$$w_l^* := \begin{cases} \bar{x}_l/(e^\top \bar{x}_{S^*}), & \text{if } l \in S^*, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, assume that there is an optimal solution $\tilde{w}$ of (7) with $\tilde{S} = \operatorname{supp}(\tilde{w})$. Due to Proposition 5, we can assume without loss of generality that $\tilde{w}$ has the form

$$\tilde{w}_l := \begin{cases} \bar{x}_l/(e^\top \bar{x}_{\tilde{S}}), & \text{if } l \in \tilde{S}, \\ 0, & \text{otherwise,} \end{cases}$$

and, hence, that $\tilde{w} \geq \bar{x}$ holds. Then, it follows that

$$\begin{aligned} \|\tilde{w} - \bar{x}\|_1 &= \sum_{l \notin \tilde{S}} \bar{x}_l + \sum_{l \in \tilde{S}} \tilde{w}_l - \bar{x}_l = \sum_{l \notin \tilde{S}} \bar{x}_l + 1 - \sum_{l \in \tilde{S}} \bar{x}_l \geq \sum_{l \notin S^*} \bar{x}_l + 1 - \sum_{l \in S^*} \bar{x}_l \\ &= \sum_{l \notin S^*} \bar{x}_l + \sum_{l \in S^*} w_l^* - \bar{x}_l = \|w^* - \bar{x}\|_1. \end{aligned}$$

Thus, w^* is an optimal solution of (7). □

A.3. **Proof of Proposition 7.**

Proof. We need to prove two properties: feasibility and optimality of $\delta^{t,l+1}$. To prove feasibility, we need to check whether $\delta^{t,l+1} \in \bar{V}$. This is obvious since there are at most k non-zero entries in $\delta^{t,l+1}$ and all the others $p-k$ entries are zero. Now, we prove that $\delta^{t,l+1}$ is optimal. Suppose that δ in Problem (11) is not restricted to the set $\bar{V}$, i.e., the problem is an unconstrained problem. Then, the minimum objective value that one can obtain is zero and this is reached by setting $\delta^{t,l+1} = \gamma^{t,l+1}$. However, if δ is restricted to the set $\bar{V}$ and $\gamma^{t,l+1}$ is not the zero vector, at most k entries of δ are allowed to be non-zero and then one needs to choose the k out of p that minimize the sum of the absolute values of the 1 norm. Clearly, the best choice are the k largest entries of $\gamma^{t,l+1}$, because then the k corresponding terms in the sum are zero and the other $p-k$ are the smallest entries of $\gamma^{t,l+1}$. This concludes the proof. □

A.4. **Remark on The Robustness of Our Heuristic.** The PADM can be understood as cutting off the smallest coefficients in each iteration to construct a sparse solution. This idea is similar to the trimmed Lasso [7]. Trimmed Lasso is a generalization of Lasso. For the latter, the coefficients are penalized by an ℓ_1 norm term, while for the trimmed Lasso, the regularization is given by

$$T_k(\beta) = \min_{\delta:\, |\operatorname{supp}(\delta)| \leq k} \|\beta - \delta\|_1,$$

i.e., the trimmed Lasso is the optimization problem

$$\min_{\beta \in \mathbb{R}^p} \|X\beta - y\|_2^2 + \mu T_k(\beta). \tag{16}$$

Interestingly, the two components of the trimmed Lasso, i.e., the least-squares part and the regularization, are the two subproblems that we identified to be the directions for the PADM. In other words, the result of an inner loop of the PADM can be considered a heuristic solution to the trimmed Lasso. The connection of the PADM theory and the theory behind the trimmed Lasso reveals an interesting relation. In [7] it is shown that for some sufficiently large μ, Problem (16) is equivalent to the best subset selection problem. This insight can as well be derived

from Theorem 4. Moreover, the optimization problem (16) can be reformulated as a robust optimization problem. The authors in [7] prove that problem

$$\min_{\beta\in\mathbb{R}^p} \min_{\substack{I\subseteq[p]\\|I|=p-k}} \max_{\Delta\in\mathcal{L}_I^\mu} \|(X+\Delta)\beta - y\|_2^2 \tag{17}$$

with

$$\mathcal{L}_I^\mu := \{\Delta \in \mathbb{R}^{n\times p} : \|\Delta_i\|_2 \leq \mu \text{ for all } i \in [p] \text{ for } \Delta_i = 0 \text{ for all } i \notin I\},$$

is equivalent to (16). Hence, the inner loop solution of the PADM can be considered a solution to (17)—i.e., a robustification of the usual least-squares problem. Therefore, each outer iteration of the PADM increases the robustification severity. Thus, even though we are only guaranteed to obtain a partial minimum, the effects of the underlying robustification can lead to results which are still very good from a statistical point of view.

References

[1] M. S. Aktürk, A. Atamtürk, and S. Gürel. "A strong conic quadratic reformulation for machine-job assignment with controllable processing times." In: *Operations Research Letters* 37.3 (2009), pp. 187–191. DOI: 10.1016/j.orl.2008.12.009.

[2] A. Atamturk and A. Gomez. "Safe screening rules for L0-regression from Perspective Relaxations." In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 421–430.

[3] A. Atamtürk and A. Gómez. "Strong formulations for quadratic optimization with M-matrices and indicator variables." In: *Mathematical Programming* 170.1 (2018), pp. 141–176. DOI: 10.1007/s10107-018-1301-5.

[4] J. E. Beasley. "OR-Library: Distributing Test Problems by Electronic Mail." In: *The Journal of the Operational Research Society* 41.11 (1990), pp. 1069–1072. DOI: 10.2307/2582903.

[5] P. Belotti, P. Bonami, M. Fischetti, A. Lodi, M. Monaci, A. Nogales-Gómez, and D. Salvagnin. "On handling indicator constraints in mixed integer programming." In: *Computational Optimization and Applications* 65.3 (2016), pp. 545–566. DOI: 10.1007/s10589-016-9847-8.

[6] D. Bertsimas and M. S. Copenhaver. "Characterization of the equivalence of robustification and regularization in linear and matrix regression." In: *European Journal of Operational Research* 270.3 (2018), pp. 931–942. DOI: 10.1016/J.EJOR.2017.03.051.

[7] D. Bertsimas, M. S. Copenhaver, and R. Mazumder. *The Trimmed Lasso: Sparsity and Robustness*. 2017. arXiv: 1708.04527.

[8] D. Bertsimas and R. Cory-Wright. *A Scalable Algorithm For Sparse Portfolio Selection*. 2020. arXiv: 1811.00138.

[9] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. *A unified approach to mixed-integer optimization problems with logical constraints*. 2019. arXiv: 1907.02109.

[10] D. Bertsimas, R. Cory-Wright, and J. Pauphilet. "Solving large-scale sparse PCA to certifiable (near) optimality." In: (2020). arXiv: 2005.05195.

[11] D. Bertsimas, C. Darnell, and R. Soucy. "Portfolio Construction Through Mixed-Integer Programming at Grantham, Mayo, Van Otterloo and Company." In: *INFORMS Journal on Applied Analytics* 29.1 (1999), pp. 49–66. DOI: 10.1287/inte.29.1.49.

[12] D. Bertsimas and V. Digalakis Jr. "The Backbone Method for Ultra-High Dimensional Sparse Machine Learning." In: (2020). arXiv: 2006.06592.

[13] D. Bertsimas, A. King, and R. Mazumder. "Best subset selection via a modern optimization lens." In: *The Annals of Statistics* 44.2 (2016), pp. 813–852. DOI: 10.1214/15-AOS1388.

[14] D. Bertsimas and B. van Parys. "Sparse high-dimensional regression: Exact scalable algorithms and phase transitions." In: *Annals of Statistics* 48.1 (2020), pp. 300–323. DOI: 10.1214/18-AOS1804.

[15] D. Bertsimas and R. Shioda. "Algorithm for cardinality-constrained quadratic optimization." In: *Computational Optimization & Applications* 43.1 (2009), pp. 1–22. DOI: 10.1007/s10589-007-9126-9.

[16] D. Bienstock. "Computational study of a family of mixed-integer quadratic programming problems." In: *Mathematical Programming* 74 (1996), pp. 121–140. DOI: 10.1007/BF02592208.

[17] D. Bienstock. "Eigenvalue Techniques for Convex Objective, Nonconvex Optimization Problems." In: *International Conference on Integer Programming and Combinatorial Optimization*. Ed. by F. Eisenbrand and F. B. Shepherd. Springer Berlin Heidelberg, 2010, pp. 29–42.

[18] P. Bonami and M. A. Lejeune. "An Exact Solution Approach for Portfolio Optimization Problems Under Stochastic and Integer Constraints." In: *Operations Research* 57.3 (2009), pp. 650–670. DOI: 10.1287/opre.1080.0599.

[19] P. Bonami, A. Lodi, A. Tramontani, and S. Wiese. "On mathematical programming with indicator constraints." In: *Mathematical Programming* 151.1 (2015), pp. 191–223. DOI: 10.1007/s10107-015-0891-4.

[20] P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011. DOI: 10.1080/02664763.2012.694258.

[21] O. P. Burdakov, C. Kanzow, and A. Schwartz. "Mathematical Programs with Cardinality Constraints: Reformulation by Complementarity-Type Conditions and a Regularization Method." In: *SIAM Journal on Optimization* 26.1 (2016), pp. 397–425. DOI: 10.1137/140978077.

[22] S. Burer and R. D. C. Monteiro. "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization." In: *Mathematical Programming* Ser. B.95 (2003), pp. 329–357. DOI: 10.1007/s10107-002-0352-8.

[23] J. P. Burgard, C. M. Costa, and M. Schmidt. *Decomposition Methods for Robustified k-Means Clustering Problems: If Less Conservative Does Not Mean Less Bad*. Tech. rep. 2020. URL: http://www.optimization-online.org/DB_HTML/2020/05/7799.html.

[24] E. Candes and T. Tao. "Decoding by linear programming." In: *IEEE Transactions on Information Theory* 51.12 (2005), pp. 4203–4215. DOI: 10.1109/TIT.2005.858979.

[25] S. B. Çay. *Random Portfolio Dataset Generator*. DOI: 10.5281/zenodo.53204.

[26] T. Chang, N. Meade, J. Beasley, and Y. Sharaiha. "Heuristics for cardinality constrained portfolio optimisation." In: *Computers & Operations Research* 27.13 (2000), pp. 1271–1302. DOI: 10.1016/S0305-0548(99)00074-X.

[27] X. Cui, X. Zheng, S. Zhu, and X. Sun. "Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems." In: *Journal of Global Optimization* 56.4 (2013), pp. 1409–1423. DOI: 10.1007/s10898-012-9842-2.

[28] A. Dedieu, H. Hazimeh, and R. Mazumder. "Learning sparse classifiers: Continuous and mixed integer optimization perspectives." In: *arXiv preprint arXiv:2001.06471* (2020).

[29] S. S. Dey, R. Mazumder, M. Molinaro, and G. Wang. *Sparse principal component analysis and its l_1-relaxation*. 2017. arXiv: 1712.00800.

[30] E. Dobriban and J. Fan. "Regularity properties for sparse regression." In: *Communications in Mathematics and Statistics* 4.1 (2016), pp. 1–19. DOI: 10.1007/s40304-015-0078-6.

[31] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles." In: *Mathematical programming* 91.2 (2002), pp. 201–213. DOI: 10.1007/s101070100263.

[32] H. Dong, K. Chen, and J. Linderoth. *Regularization vs. Relaxation: A conic optimization perspective of statistical variable selection*. 2015. arXiv: 1510.06083.

[33] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. "Least angle regression." In: *Annals of Statistics* 32.2 (2004), pp. 407–499. DOI: 10.1214/009053604000000067.

[34] A. Frangioni and C. Gentile. "A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes." In: *Operations Research Letters* 37.3 (2009), pp. 206–210. DOI: 10.1016/j.orl.2009.02.003.

[35] A. Frangioni and C. Gentile. "Perspective cuts for a class of convex 0-1 mixed integer programs." In: *Mathematical Programming* 106.2 (2006), pp. 225–236. DOI: 10.1007/s10107-005-0594-3.

[36] A. Frangioni, F. Furini, and C. Gentile. "Approximated perspective relaxations: a project and lift approach." In: *Computational Optimization and Applications* 63.3 (2016), pp. 705–735. DOI: 10.1007/s10589-015-9787-8.

[37] A. Frangioni, F. Furini, and C. Gentile. "Improving the Approximated Projected Perspective Reformulation by dual information." In: *Operations Research Letters* 45.5 (2017), pp. 519–524. DOI: 10.1016/j.orl.2017.08.001.

[38] A. Frangioni and C. Gentile. "SDP diagonalizations and perspective cuts for a class of nonseparable MIQP." In: *Operations Research Letters* 35.2 (2007), pp. 181–185. DOI: 10.1016/j.orl.2006.03.008.

[39] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. "Pathwise coordinate optimization." In: *The Annals of Applied Statistics* 1.2 (2007), pp. 302–332. DOI: 10.1214/07-aoas131.

[40] J. Gao and D. Li. "Optimal Cardinality Constrained Portfolio Selection." In: *Operations Research* 61.3 (2013), pp. 745–761. DOI: 10.1287/opre.2013.1170.

[41] J. Ge, X. Li, H. Jiang, H. Liu, T. Zhang, M. Wang, and T. Zhao. "Picasso: A Sparse Learning Library for High Dimensional Data Analysis in R and Python." In: *Journal of Machine Learning Research* 20.44 (2019), pp. 1–5. URL: http://jmlr.org/papers/v20/17-722.html.

[42] S. van de Geer. "The Deterministic Lasso." In: *Seminar für Statistik, Eidgenössische Technische Hochschule (ETH) Zürich* 140 (2007).

[43] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Penalty Alternating Direction Methods for Mixed-Integer Optimization: A New View on Feasibility Pumps." In: *SIAM Journal on Optimization* 27 (2017). DOI: 10.1137/16M1069687.

[44] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Solving Highly Detailed Gas Transport MINLPs: Block Separability and Penalty Alternating Direction Methods." In: *INFORMS Journal on Computing* 30.2 (2018), pp. 309–323. DOI: 10.1287/ijoc.2017.0780.

[45] B. Geißler, A. Morsi, L. Schewe, and M. Schmidt. "Solving power-constrained gas transportation problems using an MIP-based alternating direction method." In: *Computers & Chemical Engineering* 82 (2015), pp. 303–317. DOI: 10.1016/j.compchemeng.2015.07.005.

[46] J. Gorski, F. Pfeuffer, and K. Klamroth. "Biconvex sets and optimization with biconvex functions: a survey and extensions." In: *Mathematical Methods of Operations Research* 66.3 (2007), pp. 373–407. DOI: 10.1007/s00186-007-0161-1.

[47] O. Günlük and J. Linderoth. "Perspective reformulations of mixed integer nonlinear programs with indicator variables." In: *Mathematical Programming* 124.1-2 (2010), pp. 183–205. DOI: 10.1007/s10107-010-0360-z.

[48] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh. "Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares." In: *Journal of Machine Learning Research* 16.104 (2015), pp. 3367–3402. URL: http://jmlr.org/papers/v16/hastie15a.html.

[49] T. Hastie, R. Tibshirani, and R. Tibshirani. "Best Subset, Forward Stepwise or Lasso? Analysis and Recommendations Based on Extensive Comparisons." In: *Statistical Science* 35.4 (2020), pp. 579–592. DOI: 10.1214/19-STS733.

[50] H. Hazimeh and R. Mazumder. "Fast Best Subset Selection: Coordinate Descent and Local Combinatorial Optimization Algorithms." In: *Operations Research* 68.5 (2020), pp. 1517–1537. DOI: 10.1287/opre.2019.1919.

[51] M. Hirschberger, Y. Qi, and R. E. Steuer. "Randomly generating portfolio-selection covariance matrices with specified distributional characteristics." In: *European Journal of Operational Research* 177.3 (2007), pp. 1610–1625. DOI: 10.1016/j.ejor.2005.10.014.

[52] Y. Jin, R. Qu, and J. Atkin. "Constrained portfolio optimisation: the state-of-the-art Markowitz models." In: *Proceedings of 5th the International Conference on Operations Research and Enterprise Systems*. 2016, pp. 388–395. DOI: 10.5220/0005758303880395.

[53] T. Kleinert and M. Schmidt. "Computing Feasible Points of Bilevel Problems with a Penalty Alternating Direction Method." In: *INFORMS Journal on Computing* (2019). DOI: 10.1287/ijoc.2019.0945. Forthcoming.

[54] D. Kreber. "Cardinality-Constrained Discrete Optimization for Regression." doctoralthesis. Universität Trier, 2019. DOI: 10.25353/ubtr-xxxx-5723-2109.

[55] Y. Li and W. Xie. "Exact and approximation algorithms for sparse PCA." In: (2020). arXiv: 2008.12438.

[56] H. Markowitz. "Portfolio Selection." In: *Journal of Finance* 7.1 (1952), pp. 77–91. DOI: 10.1111/j.1540-6261.1952.tb01525.x.

[57] R. Mazumder, J. H. Friedman, and T. Hastie. "SparseNet: Coordinate descent with nonconvex penalties." In: *Journal of the American Statistical Association* 106.495 (2011), pp. 1125–1138. DOI: 10.1198/jasa.2011.tm09738.

[58] L. Mencarelli and C. D'Ambrosio. "Complex portfolio selection via convex mixed-integer quadratic programming: a survey." In: *International Transactions in Operational Research* 26.2 (2019), pp. 389–414. DOI: 10.1111/itor.12541.

[59] A. Miller. *Subset Selection in Regression*. Melbourne: Chapman and Hall, 1990.

[60] B. K. Natarajan. "Sparse approximate solutions to linear systems." In: *SIAM Journal on Computing* 24.2 (1995), pp. 227–234. DOI: 10.1137/S0097539792240406.

[61] J. G. Oxley. *Matroid Theory*. Oxford University Press, USA, 2011. DOI: 10.1093/acprof:oso/9780198566946.001.0001.

[62] L. Schewe, M. Schmidt, and D. Weninger. "A decomposition heuristic for mixed-integer supply chain problems." In: *Operations Research Letters* 48.3 (2020), pp. 225–232. DOI: 10.1016/j.orl.2020.02.006.

[63] M. Tawarmalani and N. Sahinidis. "Semidefinite Relaxations of Fractional Programs via Novel Convexification Techniques." In: *Journal of Global Optimization* 20 (2001), pp. 133–154. DOI: 10.1023/A:1011233805045.

[64] Y. Teng, L. Yang, B. Yu, and X. Song. "A penalty PALM method for sparse portfolio selection problems." In: *Optimization Methods and Software* 32.1 (2017), pp. 126–147. DOI: 10.1080/10556788.2016.1204299.

[65] R. Tibshirani. "Regression shrinkage and selection via the Lasso." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[66] R. J. Tibshirani. "The lasso problem and uniqueness." In: *Electronic Journal of Statistics* 7.1 (2013), pp. 1456–1490. DOI: 10.1214/13-EJS815.

[67] A. M. Tillmann and M. E. Pfetsch. "The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing." In: *IEEE Transactions on Information Theory* 60.2 (2014), pp. 1248–1259. DOI: 10.1109/TIT.2013.2290112.

[68] J. P. Vielma, S. Ahmed, and G. L. Nemhauser. "A Lifted Linear Programming Branch-and-Bound Algorithm for Mixed-Integer Conic Quadratic Programs." In: *INFORMS Journal on Computing* 20.3 (2008), pp. 438–450. DOI: 10.1287/ijoc.1070.0256.

[69] R. E. Wendell and A. P. Hurter. "Minimization of a Non-Separable Objective Function Subject to Disjoint Constraints." In: *Operations Research* 24.4 (1976), pp. 643–657. DOI: 10.1287/opre.24.4.643.

[70] C.-H. Zhang. "Nearly unbiased variable selection under minimax concave penalty." In: *The Annals of Statistics* 38.2 (2010), pp. 894–942. DOI: 10.1214/09-AOS729.

[71] X. Zheng, X. Sun, and D. Li. "Improving the Performance of MIQP Solvers for Quadratic Programs with Cardinality and Minimum Threshold Constraints: A Semidefinite Program Approach." In: *INFORMS Journal on Computing* 26.4 (2014), pp. 690–703. DOI: 10.1287/ijoc.2014.0592.

[72] H. Zou and T. Hastie. "Regularization and variable selection via the elastic net." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x.

(C. Moreira Costa, D. Kreber, M. Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany

Email address: costa@uni-trier.de

Email address: kreberd@uni-trier.de

Email address: martin.schmidt@uni-trier.de