The Annals of Statistics
2016, Vol. 44, No. 2, 813–852
DOI: 10.1214/15-AOS1388

BEST SUBSET SELECTION VIA A MODERN OPTIMIZATION LENS

BY DIMITRIS BERTSIMAS, ANGELA KING AND RAHUL MAZUMDER[1]

Massachusetts Institute of Technology

In the period 1991–2015, algorithmic advances in Mixed Integer Optimization (MIO) coupled with hardware improvements have resulted in an astonishing 450 billion factor speedup in solving MIO problems. We present a MIO approach for solving the classical best subset selection problem of choosing k out of p features in linear regression given n observations. We develop a discrete extension of modern first-order continuous optimization methods to find high quality feasible solutions that we use as warm starts to a MIO solver that finds provably optimal solutions. The resulting algorithm (a) provides a solution with a guarantee on its suboptimality even if we terminate the algorithm early, (b) can accommodate side constraints on the coefficients of the linear regression and (c) extends to finding best subset solutions for the least absolute deviation loss function. Using a wide variety of synthetic and real datasets, we demonstrate that our approach solves problems with n in the 1000s and p in the 100s in minutes to provable optimality, and finds *near* optimal solutions for n in the 100s and p in the 1000s in minutes. We also establish via numerical experiments that the MIO approach performs better than `Lasso` and other popularly used sparse learning procedures, in terms of achieving sparse solutions with good predictive power.

1. Introduction. We consider the linear regression model with response vector $\mathbf{y}_{n\times 1}$, model matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$, regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^{p\times 1}$ and errors $\boldsymbol{\varepsilon} \in \mathbb{R}^{n\times 1}$: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. We will assume that the columns of $\mathbf{X}$ have been standardized to have zero means and unit ℓ_2-norm. In many important classical and modern statistical applications, it is desirable to obtain a parsimonious fit to the data by finding the best k-feature fit to the response $\mathbf{y}$. Especially in the high-dimensional regime with $p \gg n$, in order to conduct statistically meaningful inference, it is desirable to assume that the true regression coefficient $\boldsymbol{\beta}$ is sparse. Quite naturally, the last few decades have seen a flurry of activity in estimating

Received June 2014; revised August 2015.

[1]Supported by ONR-N00014-15-1-2342 and an Interface grant from the Moore Sloan Foundation.

MSC2010 subject classifications. Primary 62J05, 62J07, 62G35; secondary 90C11, 90C26, 90C27.

Key words and phrases. Sparse linear regression, best subset selection, ℓ_0-constrained minimization, lasso, least absolute deviation, algorithms, mixed integer programming, global optimization, discrete optimization.

sparse linear models with good explanatory power. Central to this statistical task lies the best subset problem [Miller (2002)] with subset size k, which is given by the following optimization problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \le k, \tag{1.1}$$

where the ℓ_0 (pseudo)norm of a vector $\boldsymbol{\beta}$ counts the number of nonzeros in $\boldsymbol{\beta}$ and is given by $\|\boldsymbol{\beta}\|_0 = \sum_{i=1}^{p} 1(\beta_i \neq 0)$, where $1(\cdot)$ denotes the indicator function. The cardinality constraint makes problem (1.1) NP-hard [Natarajan (1995)]. Indeed, state-of-the-art algorithms to solve problem (1.1), as implemented in popular statistical packages, like `leaps` in `R`, do not scale to problem sizes larger than $p = 30$. Due to this reason, it is not surprising that the best subset problem has been widely dismissed as being *intractable* by the greater statistical community.

In this paper, we address problem (1.1) using modern optimization methods, specifically mixed integer optimization (MIO) and a discrete extension of first-order continuous optimization methods. Using a wide variety of synthetic and real datasets, we demonstrate that our approach solves problems with n in the 1000s and p in the 100s in minutes to provable optimality, and finds near optimal solutions for n in the 100s and p in the 1000s in minutes. To the best of our knowledge, this is the first time that MIO has been demonstrated to be a tractable solution method for problem (1.1). We note that we use the term tractability not to mean the usual polynomial solvability for problems, but rather the ability to solve problems of realistic size with associated certificates of optimality, in times that are appropriate for the applications we consider.

Brief context and background. As there is a vast literature on the best subset problem, we present a brief and selective overview of related approaches. To overcome the computational difficulties of the best subset problem, computationally friendlier convex optimization based methods like `Lasso` [Chen, Donoho and Saunders (1998), Tibshirani (1996)] have been proposed as a surrogate for problem (1.1). For the linear regression problem, the Lagrangian form of `Lasso` solves

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{1.2}$$

where the ℓ_1 penalty on $\boldsymbol{\beta}$, that is, $\|\boldsymbol{\beta}\|_1 = \sum_i |\beta_i|$ shrinks the coefficients toward zero and produces a sparse solution by setting many coefficients to be exactly zero. There has been a substantial amount of impressive work on `Lasso` [Bickel, Ritov and Tsybakov (2009), Candès and Plan (2009), Donoho (2006), Efron et al. (2004), Greenshtein and Ritov (2004), Knight and Fu (2000), Meinshausen and Bühlmann (2006), Wainwright (2009), Zhang and Huang (2008), Zhao and Yu (2006)] in terms of algorithms and understanding of its theoretical properties; see also the excellent books or surveys [Bühlmann and van de Geer (2011), Hastie,

Tibshirani and Friedman (2009), Tibshirani (2011)]. Indeed, Lasso enjoys several attractive statistical properties. Under various conditions on the model matrix $\mathbf{X}$ and $n, p, \boldsymbol{\beta}$, it can be shown that Lasso delivers a sparse model with good predictive performance [Bühlmann and van de Geer (2011), Hastie, Tibshirani and Friedman (2009)]. In order to perform exact variable selection, much stronger assumptions are required [Bühlmann and van de Geer (2011)]. Sufficient conditions under which Lasso gives a sparse model with good predictive performance are the restricted eigenvalue conditions and compatibility conditions [Bühlmann and van de Geer (2011)]. These involve statements about the spectrum of submatrices of $\mathbf{X}$ and are difficult to verify [Bandeira et al. (2013)] for a given data-matrix $\mathbf{X}$. An important reason behind the popularity of Lasso is perhaps its computational efficiency and scalability to practical sized problems. Problem (1.2) is a convex quadratic optimization problem and there are several efficient solvers for it; see, for example, Efron et al. (2004), Friedman et al. (2007), Nesterov (2013).

In spite of its favorable statistical properties, Lasso has several shortcomings. In the presence of noise and correlated variables, in order to deliver a model with good predictive accuracy, Lasso brings in a large number of nonzero coefficients (all of which are shrunk toward zero) including noise variables. Lasso leads to biased regression coefficient estimates, since the ℓ_1-norm penalizes the large coefficients more severely than the smaller coefficients. In contrast, if the best subset selection procedure decides to include a variable in the model, it brings it in without any shrinkage thereby draining the effect of its correlated surrogates. Upon increasing the degree of regularization, Lasso sets more coefficients to zero, but in the process ends up leaving out true predictors from the active set. Thus, as soon as certain sufficient regularity conditions on the data are violated, Lasso becomes suboptimal as (a) a variable selector and (b) in terms of delivering a model with good predictive performance. The shortcomings of Lasso are also known in the statistics literature. In fact, there is a significant gap between what can be achieved via best subset selection and Lasso: this is supported by empirical (for small problem sizes, i.e., $p \leq 30$) and theoretical evidence; see, for example, Greenshtein (2006), Mazumder, Friedman and Hastie (2011), Raskutti, Wainwright and Yu (2011), Shen et al. (2013), Zhang, Wainwright and Jordan (2014), Zhang and Zhang (2012) and the references therein. Some discussion is also presented in Section 4.

To address the shortcomings, nonconvex penalized regression is often used to "bridge" the gap between the convex ℓ_1 penalty and the combinatorial ℓ_0 penalty [Candès, Wakin and Boyd (2008), Fan and Li (2001), Frank and Friedman (1993), Friedman (2008), Mazumder, Friedman and Hastie (2011), Zhang (2010a, 2010b), Zhang and Huang (2008), Zou (2006), Zou and Li (2008)]. Written in Lagrangian form, this gives rise to continuous nonconvex optimization problems of the form

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_i p\big(|\beta_i|; \gamma; \lambda\big), \tag{1.3}$$

where $p(|\beta|;\gamma;\lambda)$ is a nonconvex function in β with λ and γ denoting the degree of regularization and nonconvexity, respectively. Typical examples of nonconvex penalties include the minimax concave penalty (MCP), the smoothly clipped absolute deviation (SCAD) and ℓ_γ penalties [see, e.g., Fan and Li (2001), Frank and Friedman (1993), Mazumder, Friedman and Hastie (2011), Zou and Li (2008)]. There is strong statistical evidence indicating the usefulness of estimators obtained as minimizers of nonconvex penalized problems (1.3) over `Lasso`; see, for example, Fan and Lv (2011, 2013), Loh and Wainwright (2013), Lv and Fan (2009), van de Geer, Bühlmann and Zhou (2011), Zhang (2010a), Zhang and Zhang (2012), Zheng, Fan and Lv (2014). In a recent paper, Zheng, Fan and Lv (2014) discuss the usefulness of nonconvex penalties over convex penalties (like `Lasso`) in identifying important covariates, leading to efficient estimation strategies in high dimensions. They describe interesting connections between ℓ_0 regularized least squares and least squares with the hard thresholding penalty; and in the process develop comprehensive global properties of hard thresholding regularization in terms of various metrics. Fan and Lv (2013) establish asymptotic equivalence of a wide class of regularization methods in high dimensions with comprehensive sampling properties on both global and computable solutions. Problem (1.3) mainly leads to a family of continuous and nonconvex optimization problems. Various effective nonlinear optimization based methods [see, e.g., Candès, Wakin and Boyd (2008), Fan and Li (2001), Loh and Wainwright (2013), Mazumder, Friedman and Hastie (2011), Zhang (2010a), Zou and Li (2008) and the references therein] have been proposed in the literature to obtain good local minimizers to problem (1.3). In particular, Mazumder, Friedman and Hastie (2011) propose `Sparsenet`, a coordinate-descent procedure to trace out a surface of local minimizers for problem (1.3) for the MCP penalty using effective warm start procedures.

The Lagrangian version of (1.1) given by

$$\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\sum_{i=1}^{p}1(\beta_i\neq 0), \tag{1.4}$$

may be seen as a special case of (1.3). Note that, due to nonconvexity, problems (1.4) and (1.1) are *not* equivalent. Problem (1.1) allows one to control the exact level of sparsity via the choice of k, unlike (1.4) where there is no clear correspondence between λ and k. Problem (1.4) is a discrete optimization problem unlike continuous optimization problems (1.3) arising from continuous nonconvex penalties. Insightful statistical properties of problem (1.4) have been explored from a theoretical viewpoint in Greenshtein (2006), Greenshtein and Ritov (2004), Shen et al. (2013), Zhang and Zhang (2012), for example. Shen et al. (2013) points out that (1.1) is preferable over (1.4) in terms of superior statistical properties of the resulting estimator. The aforementioned papers, however, do not discuss methods to obtain provably optimal solutions to problems (1.4) or (1.1), and to the best of our knowledge, computing optimal solutions to problems (1.4) and (1.1) is deemed as intractable.

Our approach. In this paper, we propose a novel framework via which the best subset selection problem can be solved to optimality or near optimality in problems of practical interest within a reasonable time frame. At the core of our proposal is a computationally tractable framework that brings to bear the power of modern discrete optimization methods: discrete first-order methods motivated by first-order methods in convex optimization [Nesterov (2004)] and mixed integer optimization (MIO); see Bertsimas and Weismantel (2005). We do not guarantee polynomial time solution times as these do not exist for the best subset problem unless P = NP. Rather, our view of computational tractability is the ability of a method to solve problems of practical interest in times that are appropriate for the application addressed. An advantage of our approach is that it adapts to variants of the best subset regression problem of the form

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_q^q \quad \text{s.t.} \quad \|\boldsymbol{\beta}\|_0 \leq k, \mathbf{A}\boldsymbol{\beta} \leq \mathbf{b},$$

where $\mathbf{A}\boldsymbol{\beta} \leq \mathbf{b}$ represents polyhedral constraints and $q \in \{1, 2\}$ refers to a least absolute deviation or the least squares loss function on the residuals $\mathbf{r} := \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$.

Approaches in the mathematical optimization literature. In a seminal paper Furnival and Wilson (1974), the authors describe a leaps and bounds procedure for computing global solutions to problem (1.1) (for the classical $n > p$ case) which can be achieved with computational effort significantly less than complete enumeration. `leaps`, a state-of-the-art `R` package uses this principle to perform best subset selection for problems with $n > p$ and $p \leq 30$. Bertsimas and Shioda (2009) proposed a tailored branch-and-bound scheme that can be applied to problem (1.1) using ideas from Furnival and Wilson (1974) and techniques in quadratic optimization, extending and enhancing the proposal of Bienstock (1996). The proposal of Bertsimas and Shioda (2009) concentrates on obtaining high quality upper bounds for problem (1.1) and is less scalable than the methods presented in this paper.

Contributions. We summarize our contributions in this paper below:

1. We use MIO to find a provably optimal solution for the best subset problem. Our approach has the appealing characteristic that if we terminate the algorithm early, we obtain a solution with a guarantee on its suboptimality. Furthermore, our framework can accommodate side constraints on $\boldsymbol{\beta}$ and also extends to finding best subset solutions for the least absolute deviation loss function.
2. We introduce a general algorithmic framework based on a discrete extension of modern first-order continuous optimization methods that provide near-optimal solutions for the best subset problem. The MIO algorithm significantly benefits from solutions obtained by the first-order methods and problem specific information that can be computed in a data-driven fashion.

3. We report computational results with both synthetic and real-world datasets that show that our proposed framework can deliver provably optimal solutions for problems of size n in the 1000s and p in the 100s in minutes. For high-dimensional problems with $n \in \{50, 100\}$ and $p \in \{1000, 2000\}$, with the aid of warm starts and further problem-specific information, our approach finds nearly optimal solutions in minutes but takes hours to provide certificates on the quality of the solution.

4. We investigate the statistical properties of best subset selection procedures for practical problem sizes, which to the best of our knowledge, have remained largely unexplored to date. We demonstrate the favorable predictive performance and sparsity-inducing properties of the best subset selection procedure over its competitors in a wide variety of real and synthetic examples for the least squares and absolute deviation loss functions.

The structure of the paper is as follows. In Section 2, we present a brief overview of MIO, including a summary of the computational advances it has enjoyed in the last twenty-five years. We present the proposed MIO formulations for the best subset problem as well as some connections with the compressed sensing literature for estimating parameters and providing lower bounds for the MIO formulations that improve their computational performance. In Section 3, we develop a discrete extension of first-order methods in convex optimization to obtain near optimal solutions for the best subset problem and establish its convergence properties—the proposed algorithm and its properties may be of independent interest. Section 4 briefly reviews some of the statistical properties of the best-subset solution, highlighting the performance gaps in prediction error, over regular `Lasso`-type estimators. In Section 5, we perform a variety of computational tests on synthetic and real datasets to assess the algorithmic and statistical performances of our approach for the least squares loss function for both the classical overdetermined case $n > p$, and the high-dimensional case $p \gg n$. In Section 6, we report computational results for the least absolute deviation loss function. In Section 7, we include our concluding remarks. Due to space limitations, some of the material has been relegated to the supplemental article [Bertsimas, King and Mazumder (2015)].

2. Mixed integer optimization formulations. We present a brief overview of MIO, including the simply astonishing advances it has enjoyed in the last twenty-five years. We then present the proposed MIO formulations for the best subset problem as well as some connections with the compressed sensing literature for estimating parameters. We also present completely data driven methods to estimate parameters in the MIO formulations that improve their computational performance.

2.1. *Brief background on MIO*. The general form of a Mixed Integer Quadratic Optimization (MIO) problem is as follows:

$$\begin{aligned} &\min \ \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{a} \\ &\text{s.t.} \quad \mathbf{A}\boldsymbol{\alpha} \leq \mathbf{b}, \end{aligned}$$

$$\alpha_i \in \{0, 1\}, \qquad i \in \mathcal{I},$$
$$\alpha_j \geq 0, \qquad j \notin \mathcal{I},$$

where $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{k\times m}$, $\mathbf{b} \in \mathbb{R}^k$ and $\mathbf{Q} \in \mathbb{R}^{m\times m}$ (positive semidefinite) are the given parameters of the problem; the symbol "$\leq$" denotes element-wise inequalities and we optimize over $\boldsymbol{\alpha} \in \mathbb{R}^m$ containing both discrete ($\alpha_i, i \in \mathcal{I}$) and continuous ($\alpha_i, i \notin \mathcal{I}$) variables, with $\mathcal{I} \subset \{1, \ldots, m\}$. For background on MIO, see Bertsimas and Weismantel (2005). Subclasses of MIO problems include convex quadratic optimization problems ($\mathcal{I} = \varnothing$), mixed integer ($\mathbf{Q} = \mathbf{0}_{m\times m}$) and linear optimization problems ($\mathcal{I} = \varnothing, \mathbf{Q} = \mathbf{0}_{m\times m}$). Some examples of modern integer optimization solvers include CPLEX, GLPK, MOSEK and GUROBI.

In the period 1991–2015, the computational power of MIO solvers has increased at an astonishing rate. In Bixby (2012), to measure the speedup of MIO solvers, the same set of MIO problems were tested on the same computers using twelve consecutive versions of CPLEX and version-on-version speedups were reported. The versions tested ranged from CPLEX 1.2, released in 1991 to CPLEX 11, released in 2007. Each version released in these years produced a speed improvement on the previous version, leading to a total speedup factor of more than 29,000 between the first and last version tested [see Bixby (2012), Nemhauser (2013) for details]. GUROBI 1.0, an MIO solver which was first released in 2009, was measured to have similar performance to CPLEX 11. Version-on-version speed comparisons of successive GUROBI releases have shown a speedup factor of nearly 27 between GUROBI 6.0, released in 2015, and GUROBI 1.0 [Bixby (2012), Nemhauser (2013), Optimization Inc. (2015)]. The combined machine-independent speedup factor in MIO solvers between 1991 and 2015 is 780,000. This impressive speedup factor is due to incorporating both theoretical and practical advances into MIO solvers. Cutting plane theory, disjunctive programming for branching rules, improved heuristic methods, techniques for preprocessing MIOs, using linear optimization as a black box to be called by MIO solvers, and improved linear optimization methods have all contributed greatly to the speed improvements in MIO solvers [Bixby (2012)]. In addition, the past twenty years have also brought dramatic improvements in hardware. Figure 1 shows the exponentially increasing speed of supercomputers over the past twenty years, measured in billion floating point operations per second [Top500 Supercomputer Sites (2015)]. The hardware speedup from 1994 to 2015 is approximately $10^{5.75} \sim 570{,}000$. When both hardware and software improvements are considered, the overall speedup is approximately 450 billion. Note that the speedup factors cited here refer to mixed integer linear optimization problems. The speedup factors for mixed integer quadratic problems are similar. MIO solvers provide both feasible solutions as well as lower bounds to the optimal value. As the MIO solver progresses toward the optimal solution, the lower bounds improve and provide an increasingly better guarantee of suboptimality, which is especially useful if the MIO solver is stopped before reaching the global optimum. In contrast, heuristic methods do not provide such a certificate of suboptimality.

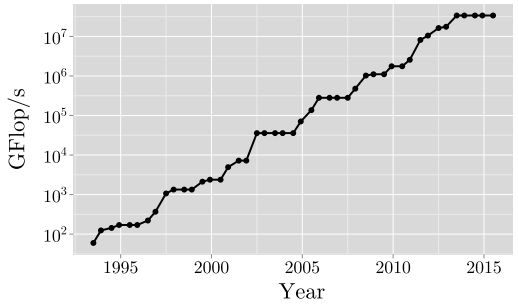

FIG. 1. *Peak supercomputer speed in GFlop/s (log scale) from* 1994–2015.

The belief that MIO approaches to problems in statistics are not practically relevant was formed in the 1970s and 1980s and it was at the time justified. Given the astonishing speedup of MIO solvers and computer hardware in the last twenty-five years, the mindset of MIO as theoretically elegant but practically irrelevant perhaps needs to be revisited. In this paper, we provide empirical evidence of this fact in the context of the best subset selection problem.

2.2. *MIO formulations for the best subset selection problem.* We first present a simple reformulation to problem (1.1) as a MIO (in fact, a mixed integer quadratic optimization) problem:

$$
\begin{aligned}
Z_1 = \min_{\boldsymbol{\beta},\mathbf{z}} \ & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \\
\text{s.t.} \quad & -\mathcal{M}_U z_i \le \beta_i \le \mathcal{M}_U z_i, \qquad i = 1, \ldots, p, \\
& z_i \in \{0, 1\}, \qquad i = 1, \ldots, p, \\
& \sum_{i=1}^{p} z_i \le k,
\end{aligned}
\tag{2.1}
$$

where $\mathbf{z} \in \{0, 1\}^p$ is a binary variable and $\mathcal{M}_U$ is a constant such that if $\widehat{\boldsymbol{\beta}}$ is a minimizer of problem (2.1), then $\mathcal{M}_U \ge \|\widehat{\boldsymbol{\beta}}\|_\infty$. If $z_i = 1$, then $|\beta_i| \le \mathcal{M}_U$ and if $z_i = 0$, then $\beta_i = 0$. Thus, $\sum_{i=1}^{p} z_i$ is an indicator of the number of nonzeros in $\boldsymbol{\beta}$.

Provided that $\mathcal{M}_U$ is chosen to be sufficiently large with $\mathcal{M}_U \ge \|\widehat{\boldsymbol{\beta}}\|_\infty$, a solution to problem (2.1) will be a solution to problem (1.1). Of course, $\mathcal{M}_U$ is not known a priori, and a small value of $\mathcal{M}_U$ may lead to a solution different from (1.1). The choice of $\mathcal{M}_U$ affects the strength of the formulation and is critical for obtaining good lower bounds in practice. In Section 2.3, we describe how to find appropriate values for $\mathcal{M}_U$. Note that there are other MIO formulations, presented herein [see problem (2.4)] that do not rely on a-priori specifications of $\mathcal{M}_U$. However, we will stick to formulation (2.1) for the time being, since it provides

some interesting connections to the Lasso. Formulation (2.1) leads to interesting insights, especially via the structure of the convex hull of its constraints:

$$\begin{aligned}&\operatorname{Conv}\left(\left\{\boldsymbol{\beta} : |\beta_i| \le \mathcal{M}_U z_i, z_i \in \{0,1\}, i = 1,\dots,p, \sum_{i=1}^{p} z_i \le k\right\}\right)\\ &\quad = \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_\infty \le \mathcal{M}_U, \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_U k\} \subseteq \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_U k\}.\end{aligned}$$

Thus, the minimum of problem (2.1) is lower-bounded by the optimum objective value of both the following convex optimization problems:

$$Z_2 := \min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_\infty \le \mathcal{M}_U, \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_U k, \tag{2.2}$$

$$Z_3 := \min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_U k, \tag{2.3}$$

where (2.3) is the familiar Lasso in constrained form. This is a weaker relaxation than formulation (2.2), which in addition to the ℓ_1 constraint on $\boldsymbol{\beta}$, has box-constraints controlling the values of the β_i's. It is easy to see that the following ordering exists: $Z_3 \le Z_2 \le Z_1$, with the inequalities being strict in most instances. In terms of approximating the optimal solution to problem (2.1), the MIO solver begins by first solving a continuous relaxation of problem (2.1). The Lasso formulation (2.3) is weaker than this root node relaxation. Additionally, MIO is typically able to significantly improve the quality of the root node solution as the MIO solver progresses toward the optimal solution. To motivate the reader, we provide an example of the evolution (see Figure 2) of the MIO formulation (2.4) for the Diabetes dataset [Efron et al. (2004)], with $n = 350$, $p = 64$ (for further details on the dataset see Section 5).

Since formulation (2.1) is sensitive to the choice of $\mathcal{M}_U$, we consider an alternative MIO formulation based on Specially Ordered Sets [Bertsimas and Weismantel (2005)] as described next.

Formulations via specially ordered sets. Any feasible solution to formulation (2.1) will have $(1 - z_i)\beta_i = 0$ for every $i \in \{1,\dots,p\}$. This constraint can be modeled via integer optimization using Specially Ordered Sets of Type 1 [Bertsimas and Weismantel (2005)] (SOS-1), as follows:

$$(1 - z_i)\beta_i = 0 \quad \Longleftrightarrow \quad (\beta_i, 1 - z_i) : \text{SOS-1},$$

for every $i = 1,\dots,p$. This leads to the following formulation of (1.1):

$$\begin{aligned}\min_{\boldsymbol{\beta},\mathbf{z}} \ & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2\\ \text{s.t.} \ & (\beta_i, 1 - z_i) : \text{SOS-1}, \qquad i = 1,\dots,p,\\ & z_i \in \{0,1\}, \qquad i = 1,\dots,p,\\ & \sum_{i=1}^{p} z_i \le k.\end{aligned} \tag{2.4}$$

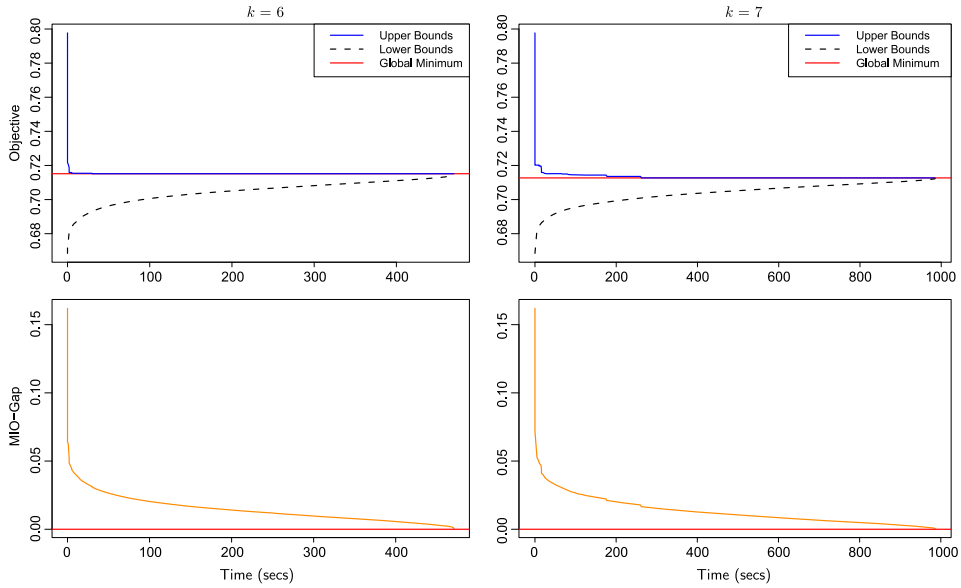

FIG. 2. *The typical evolution of the MIO formulation* (2.4) *for the diabetes dataset with* $n = 350$, $p = 64$ *with* $k = 6$ (*left panel*) *and* $k = 7$ (*right panel*). *The top panel shows the evolution of upper and lower bounds with time. The lower panel shows the evolution of the corresponding MIO gap with time. Optimal solutions for both the problems are found in a few seconds in both examples, but it takes* 10–20 *minutes to certify optimality via the lower bounds. Note that the time taken for the MIO to certify convergence to the global optimum increases with increasing* k.

Problem (2.4) can also be used obtain global solutions to problem (1.1)—problem (2.4) unlike problem (2.1) does not require any specification of the parameter $\mathcal{M}_U$.

We now present a more structured representation of problem (2.4):

$$
\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{z}} \ & \frac{1}{2}\boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X})\boldsymbol{\beta} - \langle \mathbf{X}'\mathbf{y}, \boldsymbol{\beta}\rangle + \frac{1}{2}\|\mathbf{y}\|_2^2 \\
\text{s.t.} \quad & (\beta_i, 1 - z_i) : \text{SOS-1}, \qquad i = 1, \ldots, p, \\
& z_i \in \{0, 1\}, \qquad i = 1, \ldots, p, \\
& \sum_{i=1}^{p} z_i \leq k, \\
& -\mathcal{M}_U \leq \beta_i \leq \mathcal{M}_U, \qquad i = 1, \ldots, p, \\
& \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_\ell.
\end{aligned}
\tag{2.5}
$$

We also provide problem-dependent constants $\mathcal{M}_U$ and $\mathcal{M}_\ell \in [0, \infty]$. $\mathcal{M}_U$ provides an upper bound on the absolute value of the regression coefficients and $\mathcal{M}_\ell$ provides an upper bound on the ℓ_1-norm of $\boldsymbol{\beta}$. Adding these bounds typically leads

to improved performance of the MIO, especially in delivering lower bound certificates. In Section 2.3, we describe several approaches to compute these parameters from the data.

We also consider another formulation for (2.5):

$$
\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{z},\boldsymbol{\zeta}} \ & \frac{1}{2}\boldsymbol{\zeta}^T\boldsymbol{\zeta} - \langle \mathbf{X}'\mathbf{y}, \boldsymbol{\beta}\rangle + \frac{1}{2}\|\mathbf{y}\|_2^2 \\
\text{s.t.} \quad & \boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta}, \\
& (\beta_i, 1 - z_i) : \text{SOS-1}, \qquad i = 1, \ldots, p, \\
& z_i \in \{0, 1\}, \qquad i = 1, \ldots, p, \\
& \sum_{i=1}^{p} z_i \le k, \\
& -\mathcal{M}_U \le \beta_i \le \mathcal{M}_U, \qquad i = 1, \ldots, p, \\
& \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_\ell, \\
& -\mathcal{M}_U^\zeta \le \zeta_i \le \mathcal{M}_U^\zeta, \qquad i = 1, \ldots, n, \\
& \|\boldsymbol{\zeta}\|_1 \le \mathcal{M}_\ell^\zeta,
\end{aligned}
\tag{2.6}
$$

where the optimization variables are $\boldsymbol{\beta} \in \mathbb{R}^p$, $\boldsymbol{\zeta} \in \mathbb{R}^n$, $\mathbf{z} \in \{0, 1\}^p$ and $\mathcal{M}_U, \mathcal{M}_\ell$, $\mathcal{M}_U^\zeta, \mathcal{M}_\ell^\zeta \in [0, \infty]$ are problem specific parameters. Problem (2.6) is equivalent to the following variant of the best subset problem:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}} \ & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \\
\text{s.t.} \quad & \|\boldsymbol{\beta}\|_0 \le k, \\
& \|\boldsymbol{\beta}\|_\infty \le \mathcal{M}_U, \qquad \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_\ell, \\
& \|\mathbf{X}\boldsymbol{\beta}\|_\infty \le \mathcal{M}_U^\zeta, \qquad \|\mathbf{X}\boldsymbol{\beta}\|_1 \le \mathcal{M}_\ell^\zeta.
\end{aligned}
\tag{2.7}
$$

Formulations (2.5) and (2.6) differ in the size of the quadratic forms that are involved. The current state-of-the-art MIO solvers are better equipped to handle mixed integer linear over quadratic optimization problems. Formulation (2.5) has fewer variables but has a quadratic form in p variables—we find this formulation more useful in the $n > p$ regime, with p in the 100s. Formulation (2.6) on the other hand, has more variables, but involves a quadratic form in n variables—making it more useful for high-dimensional problems $p \gg n$, with n in the 100s and p in the 1000s.

As we said earlier, the bounds on $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ are not required, but if these constraints are provided, they improve the strength of the MIO formulation. In other words, formulations with tightly specified bounds provide better lower bounds to

the global optimization problem in a specified amount of time, when compared to a MIO formulation with loose bound specifications. We next show how these bounds can be computed from given data.

2.3. *Specification of parameters*. We present herein methods to estimate the quantities $\mathcal{M}_U, \mathcal{M}_\ell, \mathcal{M}_U^\zeta, \mathcal{M}_\ell^\zeta$ such that an optimal solution to problem (2.7) is also an optimal solution to problem (1.1).

2.3.1. *Specification of parameters via coherence and restricted eigenvalues*. Herein, we describe methods relating the parameters to the notions of coherence and restricted strong convexity [Candes and Tao (2006), Donoho and Huo (2001)].

Coherence and restricted eigenvalues of a model matrix. Given a model matrix $\mathbf{X}$, Tropp (2006) introduced the cumulative coherence function

$$\mu[k] := \max_{|I|=k} \max_{j \notin I} \sum_{i \in I} \big|\langle \mathbf{X}_j, \mathbf{X}_i \rangle\big|,$$

where, $\mathbf{X}_j$, $j = 1, \ldots, p$ represent the columns of $\mathbf{X}$, that is, features. For $k = 1$, we obtain the notion of coherence introduced in Donoho and Elad (2003), Donoho and Huo (2001) as a measure of the maximal pairwise correlation in absolute value of the columns of $\mathbf{X}$: $\mu := \mu[1] = \max_{i \neq j} |\langle \mathbf{X}_i, \mathbf{X}_j \rangle|$. Candès (2008), Candes and Tao (2006) [see also Bühlmann and van de Geer (2011) and references therein] introduced the notion that a matrix $\mathbf{X}$ satisfies a restricted eigenvalue condition if

$$\lambda_{\min}(\mathbf{X}_I' \mathbf{X}_I) \geq \gamma_k \qquad \text{for every } I \subset \{1, \ldots, p\} \text{ such that } |I| \leq k, \tag{2.8}$$

where $\lambda_{\min}(\mathbf{X}_I' \mathbf{X}_I)$ denotes the smallest eigenvalue of the matrix $\mathbf{X}_I' \mathbf{X}_I$. An inequality linking $\mu[k]$ and γ_k is as follows.

PROPOSITION 1. *The following bounds hold*:

(a) [*Tropp* (2006)]: $\mu[k] \leq \mu \cdot k$.
(b) [*Donoho and Elad* (2003)]: $\gamma_k \geq 1 - \mu[k-1] \geq 1 - \mu \cdot (k-1)$.

The computations of $\mu[k]$ and γ_k for general k are difficult, while μ is simple to compute. Proposition 1 provides bounds for $\mu[k]$ and γ_k in terms of the coherence μ.

Operator norms of submatrices. The (p, q) operator norm of matrix $\mathbf{A}$ is given by $\|\mathbf{A}\|_{p,q} := \max_{\|\mathbf{u}\|_q = 1} \|\mathbf{A}\mathbf{u}\|_p$. We will use extensively here the $(1, 1)$ operator norm. We assume that each column vector of $\mathbf{X}$ has unit ℓ_2-norm. The results derived in the next proposition (for a proof see Section 8.3 in the supplementary material [Bertsimas, King and Mazumder (2015)]) borrow and enhance techniques developed by Tropp (2006) in the context of analyzing the ℓ_1-ℓ_0 equivalence in compressed sensing.

PROPOSITION 2. *For any* $I \subset \{1, \ldots, p\}$ *with* $|I| = k$, *we have*:

(a) $\|\mathbf{X}_I'\mathbf{X}_I - \mathbf{I}\|_{1,1} \leq \mu[k-1]$.

(b) *If* $\mathbf{X}_I'\mathbf{X}_I$ *is invertible and* $\|\mathbf{X}_I'\mathbf{X}_I - \mathbf{I}\|_{1,1} < 1$, *then* $\|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\|_{1,1} \leq \frac{1}{1-\mu[k-1]}$.

We note that part (b) also appears in Tropp (2006) for the operator norm $\|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\|_{\infty,\infty}$.

Given a set $I \subset \{1, \ldots, p\}$ with $|I| = k$, we let $\widehat{\boldsymbol{\beta}}_I$ denote the least squares regression coefficients obtained by regressing $\mathbf{y}$ on $\mathbf{X}_I$, that is, $\widehat{\boldsymbol{\beta}}_I = (\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'\mathbf{y}$. If we append $\widehat{\boldsymbol{\beta}}_I$ with zeros in the remaining coordinates we obtain $\widehat{\boldsymbol{\beta}}$, where, we suppress the dependence on I for notational convenience.

Recall that $\mathbf{X}_j$, $j = 1, \ldots, p$ represent the columns of $\mathbf{X}$; and we will use $\mathbf{x}_i$, $i = 1, \ldots, n$ to denote the rows of $\mathbf{X}$. As discussed above $\|\mathbf{X}_j\| = 1$. We order the correlations $|\langle \mathbf{X}_j, \mathbf{y}\rangle|$:

$$|\langle \mathbf{X}_{(1)}, \mathbf{y}\rangle| \geq |\langle \mathbf{X}_{(2)}, \mathbf{y}\rangle| \geq \cdots \geq |\langle \mathbf{X}_{(p)}, \mathbf{y}\rangle|. \tag{2.9}$$

We finally denote by $\|\mathbf{x}_i\|_{1:k}$ the sum of the top k absolute values of the entries of x_{ij}, $j \in \{1, 2, \ldots, p\}$. The following theorem (for a proof, see Section 8.4 [Bertsimas, King and Mazumder (2015)]).

THEOREM 2.1. *For any* $k \geq 1$ *such that* $\mu[k-1] < 1$ *any optimal solution* $\widehat{\boldsymbol{\beta}}$ *to* (1.1) *satisfies*:

$$\text{(a)} \quad \|\widehat{\boldsymbol{\beta}}\|_1 \leq \frac{1}{1-\mu[k-1]} \sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|, \tag{2.10}$$

$$\text{(b)} \quad \|\widehat{\boldsymbol{\beta}}\|_\infty \leq \min\left\{ \frac{1}{\gamma_k} \sqrt{\sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|^2}, \frac{1}{\sqrt{\gamma_k}} \|\mathbf{y}\|_2 \right\}, \tag{2.11}$$

$$\text{(c)} \quad \|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1 \leq \min\left\{ \sum_{i=1}^{n} \|\mathbf{x}_i\|_\infty \|\widehat{\boldsymbol{\beta}}\|_1, \sqrt{k}\|\mathbf{y}\|_2 \right\}, \tag{2.12}$$

$$\text{(d)} \quad \|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty \leq \Big(\max_{i=1,\ldots,n} \|\mathbf{x}_i\|_{1:k} \Big) \|\widehat{\boldsymbol{\beta}}\|_\infty. \tag{2.13}$$

Note that, above, the upper bound in part (a) becomes infinite as soon as $\mu[k-1] \geq 1$. In such a case, we can obtain bounds by using techniques described in Section 2.3.2. The interesting message conveyed by Theorem 2.1 is that the upper bounds on $\|\widehat{\boldsymbol{\beta}}\|_1$, $\|\widehat{\boldsymbol{\beta}}\|_\infty$, $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1$ and $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty$, corresponding to the problem (2.7) can all be obtained in terms of γ_k and $\mu[k-1]$, quantities of fundamental interest appearing in the analysis of ℓ_1 regularization methods and understanding how close they are to ℓ_0 solutions [Candès (2008), Candes and Tao (2006), Donoho

and Elad (2003), Donoho and Huo (2001), Tropp (2006)]. On a different note, Theorem 2.1 arises from a purely computational motivation and quite curiously, involves the same quantities: cumulative coherence and restricted eigenvalues. While quantities $\mu[k-1]$, γ_k are difficult to compute exactly, they can be approximated by Proposition 1 which provides bounds commonly used in the compressed sensing literature.

2.3.2. *Specification of parameters via convex quadratic optimization.* We present alternative purely data-driven way to compute the upper bounds to the parameters by solving several simple convex quadratic optimization problems.

Bounds on $\widehat{\beta}_i$'s. For the case $n > p$, upper and lower bounds on $\widehat{\beta}_i$ can be obtained by solving the following pair of convex optimization problems:

$$\begin{aligned} u_i^+ := \max_{\boldsymbol{\beta}} \beta_i \qquad & \qquad u_i^- := \min_{\boldsymbol{\beta}} \beta_i \\ \text{s.t.} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \le \text{UB}, \qquad & \qquad \text{s.t.} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \le \text{UB}, \end{aligned} \tag{2.14}$$

for $i = 1, \ldots, p$. Above, UB is an upper bound to the minimum of problem (1.1). u_i^+ is an upper bound to $\widehat{\beta}_i$, since the cardinality constraint $\|\boldsymbol{\beta}\|_0 \le k$ does not appear in the optimization problem. Similarly, u_i^- is a lower bound to $\widehat{\beta}_i$. The quantity $\mathcal{M}_U^i = \max\{|u_i^+|, |u_i^-|\}$ serves as an upper bound to $|\widehat{\beta}_i|$. A reasonable choice for UB is obtained by using the discrete first-order methods (as described in Section 3) in combination with the MIO formulation (2.4) (for a predefined amount of time). Having obtained $\mathcal{M}_U^i$ as described above, we can obtain an upper bound to $\|\widehat{\boldsymbol{\beta}}\|_\infty$ and $\|\widehat{\boldsymbol{\beta}}\|_1$ as follows: $\mathcal{M}_U = \max_i \mathcal{M}_U^i$ and $\|\widehat{\boldsymbol{\beta}}\|_1 \le \sum_{i=1}^k \mathcal{M}_U^{(i)}$ where, $\mathcal{M}_U^{(1)} \ge \mathcal{M}_U^{(2)} \ge \cdots \ge \mathcal{M}_U^{(p)}$. Similarly, bounds corresponding to parts (c) and (d) in Theorem 2.1 can be obtained by using the upper bounds on $\|\widehat{\boldsymbol{\beta}}\|_\infty$, $\|\widehat{\boldsymbol{\beta}}\|_1$ as described above.

Note that the quantities u_i^+ and u_i^- are finite when the level sets of the least squares loss function are bounded—the bounds are loose when $p > n$. In the following, we describe methods to obtain nontrivial bounds on $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle$, for $i = 1, \ldots, n$ that apply for arbitrary n, p.

Bounds on $\langle \mathbf{x}_i, \widehat{\boldsymbol{\beta}} \rangle$'s. Consider the following convex quadratic optimization problems:

$$\begin{aligned} v_i^+ := \max_{\boldsymbol{\beta}} \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle \qquad & \qquad v_i^- := \min_{\boldsymbol{\beta}} \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle \\ \text{s.t.} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \le \text{UB}, \qquad & \qquad \text{s.t.} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \le \text{UB}, \end{aligned} \tag{2.15}$$

for $i = 1, \ldots, n$. Note that the bounds obtained from (2.15) are nontrivial for both the under-determined and overdetermined cases. These bounds are upper

and lower bounds since we drop the cardinality constraint on $\boldsymbol{\beta}$. The quantity $v_i = \max\{|v_i^+|, |v_i^-|\}$ serves as an upper bound to $|\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle|$. In particular, this leads to simple upper bounds on $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty \le \max_i v_i$ and $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1 \le \sum_i v_i$ and can be thought of completely data-driven methods to estimate bounds appearing in (2.12) and (2.13). Problems (2.14) and (2.15) can be computed efficiently, as we discuss in Section 8.1 in the supplementary material [Bertsimas, King and Mazumder (2015)].

2.3.3. *Parameter specifications from advanced warm-starts*. The methods in Sections 2.3.1 and 2.3.2 lead to *provable* bounds on the parameters: with these bounds problem (2.7) provides an optimal solution to problem (1.1). We now describe some alternatives that lead to excellent parameter specifications in practice.

The discrete first-order methods described in Section 3 provide good upper bounds to problem (1.1). These solutions when supplied as a warm-start to the MIO formulation (2.4) are often improved by MIO, thereby leading to high quality solutions to problem (1.1) within several minutes. If $\widehat{\boldsymbol{\beta}}_{\text{hyb}}$ denotes an estimate obtained from this hybrid approach, then $\mathcal{M}_U := \tau\|\widehat{\boldsymbol{\beta}}_{\text{hyb}}\|_\infty$ with τ a multiplier greater than one (e.g., $\tau \in \{1.5, 2, 5\}$) provides a good estimate for the parameter $\mathcal{M}_U$. A reasonable upper bound to $\|\widehat{\boldsymbol{\beta}}\|_1$ is $k\mathcal{M}_U$. Bounds on the other quantities: $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1$, $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty$ can be derived by using expressions appearing in Theorem 2.1, with aforementioned bounds on $\|\widehat{\boldsymbol{\beta}}\|_1$ and $\|\widehat{\boldsymbol{\beta}}\|_\infty$.

2.3.4. *Some generalizations and variants*. Some variations and improvements of the procedures described above are presented in Section 8.2 in the supplementary material [Bertsimas, King and Mazumder (2015)].

Recommendations. We summarize our observations about the parameter choices based on some numerical experiments. For $n > p$ examples, when $\mathbf{X}$ is full rank, methods in Sections 2.3.1, 2.3.2 are often quite similar—we thus recommend computing bounds via both these methods and taking the tighter of the two. For $n < p$ examples, when k is small, Section 2.3.1 provides useful bounds on $\boldsymbol{\beta}$, which are not available via Section 2.3.2. We recommend computing the implied bounds on all parameters appearing in parts (a)–(b) (Theorem 2.1) and taking the tightest bound. Bounds obtained via Section 2.3.3 are generally always tighter (provided τ is small) and are readily available as a by-product of our algorithmic framework—we recommend these bounds in practice, unless provably optimal bounds are of the essence. We remind the reader that these bounds are particularly useful while proving optimality of the solutions obtained via MIO. They are not as critical in obtaining good upper bounds to problem (1.1).

3. Discrete first-order algorithms. We develop a discrete extension of first-order methods in convex optimization [Nesterov (2004, 2013)] to obtain near opti-

mal solutions for problem (1.1) and the least absolute deviation (LAD) loss function. Our approach applies to the problem of minimizing any smooth convex function subject to cardinality constraints. In Section 5, we demonstrate how these methods enhance the performance of MIO.

Our framework borrows ideas from projected gradient descent methods in first-order convex optimization problems [Nesterov (2004)] and generalizes them to the discrete optimization problem (3.1). We also derive new global convergence results for our proposed algorithms as presented in Theorem 3.1. In the signal processing literature [Blumensath and Davies (2008, 2009)] proposed iterative hard-thresholding algorithms for problem (1.4). The authors establish convergence properties of the algorithm when $\mathbf{X}$ satisfies a coherence [Blumensath and Davies (2008)] or Restricted Isometry Property [Blumensath and Davies (2009)]. In the context of problem (1.1), our algorithm and its convergence analysis do not require any such condition on $\mathbf{X}$. Our framework, with some novel modifications also applies to the nonsmooth least absolute deviation loss with cardinality constraints as discussed in Section 3.2.

Consider the following optimization problem:

$$\min_{\boldsymbol{\beta}} g(\boldsymbol{\beta}) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \le k, \tag{3.1}$$

where, $g(\boldsymbol{\beta}) \ge 0$ is[2] convex and has Lipschitz continuous gradient:

$$\|\nabla g(\boldsymbol{\beta}) - \nabla g(\widetilde{\boldsymbol{\beta}})\| \le \ell \|\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}}\|. \tag{3.2}$$

The first ingredient of our approach is the observation that when $g(\boldsymbol{\beta}) = \|\boldsymbol{\beta} - \mathbf{c}\|_2^2$ for a given $\mathbf{c}$, problem (3.1) admits a closed form solution (for completeness we present a proof in Section 9.2 of the supplementary material [Bertsimas, King and Mazumder (2015)]).

PROPOSITION 3. *If $\widehat{\boldsymbol{\beta}}$ is an optimal solution to the following problem*:

$$\widehat{\boldsymbol{\beta}} \in \operatorname*{arg\,min}_{\|\boldsymbol{\beta}\|_0 \le k} \|\boldsymbol{\beta} - \mathbf{c}\|_2^2, \tag{3.3}$$

then it can be computed as follows: $\widehat{\boldsymbol{\beta}}$ *retains the* k *largest* (*in absolute value*) *elements of* $\mathbf{c} \in \mathbb{R}^p$ *and sets the rest to zero*, *i.e.*, *if* $|c_{(1)}| \ge |c_{(2)}| \ge \cdots \ge |c_{(p)}|$, *denote the ordered values of the absolute values of the vector* $\mathbf{c}$, *then*

$$\widehat{\beta}_i = \begin{cases} c_i, & \text{if } i \in \{(1), \ldots, (k)\}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.4}$$

where, $\widehat{\beta}_i$ *is the* i*th coordinate of* $\widehat{\boldsymbol{\beta}}$. *We will denote the set of solutions to problem* (3.3) *by the notation* $\mathbf{H}_k(\mathbf{c})$.

[2]The lower bound of zero in $g(\boldsymbol{\beta}) \ge 0$, can be relaxed to be any finite number.

The notation "argmin" [appearing in problem (3.3) and other places that follow] denotes the set of minimizers. Operator (3.4) is also known as the hard-thresholding operator [Donoho and Johnstone (1994)]—a notion that arises in the context of the following related optimization problem:

$$\widehat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta} - \mathbf{c}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0, \tag{3.5}$$

where, $\widehat{\boldsymbol{\beta}}$ admits a simple closed form expression given by $\widehat{\beta}_i = c_i$ if $|c_i| > \sqrt{\lambda}$ and $\widehat{\beta}_i = 0$ otherwise, for $i = 1, \ldots, p$.

REMARK 1. There is a subtle difference between the minimizers of problems (3.3) and (3.5). For problem (3.5), the smallest (in absolute value) nonzero element in $\widehat{\boldsymbol{\beta}}$ is greater than $\sqrt{\lambda}$ in absolute value. On the other hand, in problem (3.3) there is no lower bound to the minimum (in absolute value) nonzero element of a minimizer. This issue arises while analyzing the convergence properties of Algorithm 1 (see Section 3.1).

Given a current solution $\boldsymbol{\beta}$, the second ingredient of our approach is to upper bound the function $g(\boldsymbol{\eta})$ around $g(\boldsymbol{\beta})$. To do so, we use ideas from projected gradient descent methods in first-order convex optimization problems [Nesterov (2004, 2013)].

PROPOSITION 4 [Nesterov (2013, 2004)]. *For a convex function $g(\boldsymbol{\beta})$ satisfying condition (3.2) and for any $L \geq \ell$, we have*

$$g(\boldsymbol{\eta}) \leq Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) := g(\boldsymbol{\beta}) + \frac{L}{2}\|\boldsymbol{\eta} - \boldsymbol{\beta}\|_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta} - \boldsymbol{\beta}\rangle \tag{3.6}$$

for all $\boldsymbol{\beta}, \boldsymbol{\eta}$ with equality holding at $\boldsymbol{\beta} = \boldsymbol{\eta}$.

Applying Proposition 3 to the upper bound $Q_L(\boldsymbol{\eta}, \boldsymbol{\beta})$ in Proposition 4, we obtain

$$\begin{aligned}
&\arg\min_{\|\boldsymbol{\eta}\|_0 \leq k} Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) \\
&\quad = \arg\min_{\|\boldsymbol{\eta}\|_0 \leq k} \left(\frac{L}{2} \left\| \boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}) \right) \right\|_2^2 - \frac{1}{2L}\|\nabla g(\boldsymbol{\beta})\|_2^2 + g(\boldsymbol{\beta}) \right) \\
&\quad = \arg\min_{\|\boldsymbol{\eta}\|_0 \leq k} \left\| \boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}) \right) \right\|_2^2 \\
&\quad = \mathbf{H}_k\left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta}) \right),
\end{aligned} \tag{3.7}$$

where $\mathbf{H}_k(\cdot)$ is defined in (3.4). In light of (3.7), we are now ready to present Algorithm 1 to find a stationary point (see Definition 1) of problem (3.1).

ALGORITHM 1. *Input*: $g(\boldsymbol{\beta})$, parameter: L and convergence tolerance: ε.
Output: A first-order stationary solution $\boldsymbol{\beta}^*$.

1. Initialize with $\boldsymbol{\beta}_1 \in \mathbb{R}^p$ such that $\|\boldsymbol{\beta}_1\|_0 \leq k$.
2. For $m \geq 1$, apply (3.7) with $\boldsymbol{\beta} = \boldsymbol{\beta}_m$ to obtain $\boldsymbol{\beta}_{m+1}$ as:

$$\boldsymbol{\beta}_{m+1} \in \mathbf{H}_k\Big(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\Big). \tag{3.8}$$

3. Repeat step 2, until $g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1}) \leq \varepsilon$.

3.1. *Convergence analysis of Algorithm 1.* We first define the notion of first-order optimality for problem (3.1).

DEFINITION 1. Given an $L \geq \ell$, the vector $\boldsymbol{\eta} \in \mathbb{R}^p$ is said to be a first-order stationary point of problem (3.1) if $\|\boldsymbol{\eta}\|_0 \leq k$ and it satisfies the following fixed-point equation:

$$\boldsymbol{\eta} \in \mathbf{H}_k\Big(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta})\Big). \tag{3.9}$$

We provide some intuition associated with the above definition. Consider $\boldsymbol{\eta}$ as in Definition 1. Since $\|\boldsymbol{\eta}\|_0 \leq k$, there is a set $I \subset \{1, \ldots, p\}$ such that $\eta_i = 0, i \in I$ and the size of I^c (complement of I) is k. Since $\boldsymbol{\eta} \in \mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta}))$, for $i \notin I$ we have: $\eta_i = \eta_i - \frac{1}{L}\nabla_i g(\boldsymbol{\eta})$, where, $\nabla_i g(\boldsymbol{\eta})$ is the ith coordinate of $\nabla g(\boldsymbol{\eta})$. It thus follows: $\nabla_i g(\boldsymbol{\eta}) = 0, i \notin I$. Since $g(\boldsymbol{\eta})$ is convex in $\boldsymbol{\eta}$, this means that $\boldsymbol{\eta}$ solves the following convex optimization problem:

$$\min_{\boldsymbol{\eta}} g(\boldsymbol{\eta}) \quad \text{s.t.} \quad \eta_i = 0, i \in I. \tag{3.10}$$

Note, however, that the converse of the above statement is not true. That is, if $\tilde{I} \subset \{1, \ldots, p\}$ is an arbitrary subset with $|\tilde{I}^c| = k$ then a solution $\widehat{\boldsymbol{\eta}}_{\tilde{I}}$ to the restricted convex problem (3.10) with $I = \tilde{I}$ need *not* correspond to a first-order stationary point. Any global minimizer to problem (3.1) is also a first-order stationary point (see Proposition 7). The following proposition (for its proof see Section 9.3 in the supplementary material [Bertsimas, King and Mazumder (2015)]) sheds light on a first-order stationary point $\boldsymbol{\eta}$ for which $\|\boldsymbol{\eta}\|_0 < k$ (i.e., the inequality is strict).

PROPOSITION 5. *If $\boldsymbol{\eta}$ satisfies the first-order stationary condition* (3.9) *and* $\|\boldsymbol{\eta}\|_0 < k$, *then* $\boldsymbol{\eta} \in \arg\min_{\boldsymbol{\beta}} g(\boldsymbol{\beta})$.

We define the notion of an ε-approximate first-order stationary point of problem (3.1).

DEFINITION 2. Given an $\varepsilon > 0$ and $L \geq \ell$ we say that $\boldsymbol{\eta}$ satisfies an ε-approximate first-order optimality condition of problem (3.1) if $\|\boldsymbol{\eta}\|_0 \leq k$ and for some $\widehat{\boldsymbol{\eta}} \in \mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta}))$, we have $\|\boldsymbol{\eta} - \widehat{\boldsymbol{\eta}}\|_2 \leq \varepsilon$.

We now introduce some notation. Let $\boldsymbol{\beta}_m = (\beta_{m1}, \ldots, \beta_{mp})$ and $\mathbf{1}_m = (e_1, \ldots, e_p)$ with $e_j = 1$, if $\beta_{mj} \neq 0$, and $e_j = 0$, if $\beta_{mj} = 0$, $j = 1, \ldots, p$, that is, $\mathbf{1}_m$ represents the sparsity pattern of the support of $\boldsymbol{\beta}_m$. Suppose, we order the coordinates of $\boldsymbol{\beta}_m$ by their absolute values: $|\beta_{(1),m}| \geq |\beta_{(2),m}| \geq \cdots \geq |\beta_{(p),m}|$. Note that by definition (3.8), $\beta_{(i),m} = 0$ for all $i > k$ and $m \geq 2$. We denote $\alpha_{k,m} = |\beta_{(k),m}|$ to be the kth largest (in absolute value) entry in $\boldsymbol{\beta}_m$ for all $m \geq 2$. Clearly, if $\alpha_{k,m} > 0$ then $\|\boldsymbol{\beta}_m\|_0 = k$ and if $\alpha_{k,m} = 0$ then $\|\boldsymbol{\beta}_m\|_0 < k$. Let $\overline{\alpha}_k := \limsup_{m\to\infty} \alpha_{k,m}$ and $\underline{\alpha}_k := \liminf_{m\to\infty} \alpha_{k,m}$.

The following proposition, the proof of which can be found in Section 9.1 [Bertsimas, King and Mazumder (2015)], describes the asymptotic convergence properties of Algorithm 1.

PROPOSITION 6. *Consider $g(\boldsymbol{\beta})$ and ℓ as defined in (3.2). Let $\boldsymbol{\beta}_m, m \geq 1$ be the sequence generated by Algorithm 1. Then we have*:

(a) *For any $L \geq \ell$, the sequence $g(\boldsymbol{\beta}_m)$ is decreasing, converges and satisfies*

$$g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1}) \geq \frac{L-\ell}{2}\|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2^2. \tag{3.11}$$

(b) *If $L > \ell$, then $\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m \to \mathbf{0}$ as $m \to \infty$.*

(c) *If $L > \ell$ and $\underline{\alpha}_k > 0$, then the sequence $\mathbf{1}_m$ converges after finitely many iterations, that is, there exists an iteration index M^* such that $\mathbf{1}_m = \mathbf{1}_{m+1}$ for all $m \geq M^*$. Furthermore, the sequence $\boldsymbol{\beta}_m$ is bounded and converges to a first-order stationary point.*

(d) *If $L > \ell$ and $\underline{\alpha}_k = 0$ then $\liminf_{m\to\infty} \|\nabla g(\boldsymbol{\beta}_m)\|_\infty = 0$.*

(e) *Let $L > \ell$, $\overline{\alpha}_k = 0$ and suppose that the sequence $\boldsymbol{\beta}_m$ has a limit point. Then $g(\boldsymbol{\beta}_m) \to \min_{\boldsymbol{\beta}} g(\boldsymbol{\beta})$.*

REMARK 2. Note that the existence of a limit point in Proposition 6, part (e) is guaranteed under fairly weak conditions. One such condition is that $\sup\{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_0 \leq k, f(\boldsymbol{\beta}) \leq f_0\} < \infty$, for any finite value f_0. In words, this means that the k-sparse level sets of the function $g(\boldsymbol{\beta})$ is bounded. In the special case where $g(\boldsymbol{\beta})$ is the least squares loss function, the above condition is equivalent to every k-submatrix $(\mathbf{X}_J)$ of $\mathbf{X}$ comprising of k columns being full rank. In particular, this holds with probability one when the entries of $\mathbf{X}$ are drawn from a continuous distribution and $k < n$.

REMARK 3. Parts (d) and (e) of Proposition 6 correspond to unregularized solutions of the problem $\min g(\boldsymbol{\beta})$. The conditions assumed in part (c) imply that

the support of $\boldsymbol{\beta}_m$ stabilizes and Algorithm 1 behaves like vanilla gradient descent thereafter. The support of $\boldsymbol{\beta}_m$ need not stabilize for parts (d), (e) and thus Algorithm 1 may not behave like vanilla gradient descent after finitely many iterations. However, the objective values (under minor regularity assumptions) converge to $\min g(\boldsymbol{\beta})$.

The following proposition, the proof of which can be found in Section 9.4, establishes some additional properties of the fixed-point equation (3.9).

PROPOSITION 7. *Suppose $L > \ell$. We have the following*:

(a) *If $\boldsymbol{\eta}$ satisfies a first-order stationary point as in Definition 1, then the set $\mathbf{H}_k(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta}))$ has exactly one element: $\boldsymbol{\eta}$.*

(b) *If $\widehat{\boldsymbol{\beta}}$ is a global minimizer of problem (3.1), then it is a first-order stationary point.*

While Proposition 6 establishes the asymptotic convergence properties of Algorithm 1, the following theorem, the proof of which can be found in Section 9.5 [Bertsimas, King and Mazumder (2015)], characterizes the rate of convergence of the algorithm to a first-order stationary point.

THEOREM 3.1. *Let $L > \ell$ and $\boldsymbol{\beta}^*$ denote a first-order stationary point of Algorithm 1. After M iterations, Algorithm 1 satisfies*

$$\min_{m=1,\ldots,M} \|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2^2 \leq \frac{2(g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}^*))}{M(L-\ell)}, \tag{3.12}$$

where $g(\boldsymbol{\beta}_m) \downarrow g(\boldsymbol{\beta}^*)$ as $m \to \infty$.

Theorem 3.1 implies that for any $\varepsilon > 0$ there exists $M = O(\frac{1}{\varepsilon})$ such that for some $1 \leq m^* \leq M$, we have $\|\boldsymbol{\beta}_{m^*+1} - \boldsymbol{\beta}_{m^*}\|_2^2 \leq \varepsilon$. Note that the convergence rates derived above apply for a large class of problems (3.1), where, the function $g(\boldsymbol{\beta}) \geq 0$ is convex with Lipschitz continuous gradient (3.2). Tighter rates may be obtained under additional structural assumptions on $g(\cdot)$. For example, the adaptation of Algorithm 1 for problem (1.4) was analyzed in Blumensath and Davies (2008, 2009) with $\mathbf{X}$ satisfying coherence [Blumensath and Davies (2008)] or Restricted Isometry Property (RIP) [Blumensath and Davies (2009)]. In these cases, the algorithm can be shown to have a linear convergence rate [Blumensath and Davies (2008, 2009)], where the rate depends upon the RIP constants.

Note that by Proposition 6 the support of $\boldsymbol{\beta}_m$ stabilizes after finitely many iterations, after which Algorithm 1 behaves like gradient descent on the stabilized support. If $g(\boldsymbol{\beta})$ restricted to this support is strongly convex, then Algorithm 1 will enjoy a linear rate of convergence [Nesterov (2004)], as soon as the support

stabilizes. This behavior is adaptive, that is, Algorithm 1 does not need to be modified after the support stabilizes. We next describe practical schemes via which first-order stationary points of Algorithm 1 can be obtained by solving a low dimensional convex optimization problem, as soon as the support stabilizes. In our experiments, this algorithm (with multiple starting points) took at most 1–2 minutes for $p = 2000$ and a few seconds for smaller values of p.

Polishing coefficients on the active set. Algorithm 1 *detects* the active set after a few iterations. Once the active set stabilizes, the algorithm may take a number of iterations to estimate the values of the regression coefficients on the active set to a high accuracy level. In this context, we found the following simple method to be quite useful. When the algorithm has converged to a tolerance of ε ($\approx 10^{-4}$), we fix the current active set, $\mathcal{I}$, and solve a lower-dimensional convex optimization problem (3.10) with $I = \mathcal{I}^c$—for the least squares and least absolute deviation problems, this can be solved very efficiently.

We observed in our experiments that Algorithm 2, a minor variant of Algorithm 1 had better empirical performance. Algorithm 2 modifies step 2 of Algorithm 1 by using a simple line search, as described below:

ALGORITHM 2. Replace step 2 in Algorithm 1 by:

2. $\boldsymbol{\beta}_{m+1} = \lambda_m \boldsymbol{\eta}_m + (1 - \lambda_m)\boldsymbol{\beta}_m$, where, $\boldsymbol{\eta}_m \in \mathbf{H}_k(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m))$, with $\lambda_m \in \arg\min_\lambda g(\lambda \boldsymbol{\eta}_m + (1 - \lambda)\boldsymbol{\beta}_m)$.

$\boldsymbol{\eta}_m$ produced by Algorithm 2 is k-sparse and the updates satisfy: $g(\boldsymbol{\beta}_{m+1}) \leq g(\boldsymbol{\eta}_m)$. Algorithm 2 may be perceived as one that *restarts* Algorithm 1 with multiple starting points: $\boldsymbol{\beta}_m$. We observed empirically that $\lambda_m \approx 0$ after a few iterations in which case, $\boldsymbol{\eta}_m, \boldsymbol{\beta}_{m+1}$ have the same support, and thus Algorithm 2 *behaves* like Algorithm 1. For Algorithm 2, we use the following convergence criterion: we exit if $|g(\boldsymbol{\eta}_{m+1}) - g(\boldsymbol{\eta}_m)| \leq \varepsilon$ or run the algorithm for a maximum of N iterations and exit with $\boldsymbol{\eta}_{m^*}$, where, $m^* \in \arg\min_{1 \leq m \leq N} g(\boldsymbol{\eta}_m)$. Algorithm 1 is then run with the resultant estimate—this usually takes at most a few additional iterations to converge.[3]

3.2. *Application to least squares subset selection.* For problem (1.1), we have $g(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$, $\nabla g(\boldsymbol{\beta}) = -\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ and $\ell = \lambda_{\max}(\mathbf{X}'\mathbf{X})$—and the framework described above, applies readily. The polishing of coefficients in the active set can be performed via a least squares problem on $\mathbf{y}, \mathbf{X}_J$, where J denotes the support of the k-sparse regression coefficient.

[3]We note that this step is hardly necessary in practice, but might be used to ensure a convergent algorithm.

3.3. *Application to least absolute deviation subset selection.* We consider the LAD problem with support constraints in $\boldsymbol{\beta}$:

$$\min_{\boldsymbol{\beta}} g_1(\boldsymbol{\beta}) := \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{\beta}\|_0 \le k. \tag{3.13}$$

Since $g_1(\boldsymbol{\beta})$ is nonsmooth, our framework does not apply directly. We smooth the non-differentiable $g_1(\boldsymbol{\beta})$ so that we can apply Algorithms 1 and 2. Observing that $g_1(\boldsymbol{\beta}) = \sup_{\|\mathbf{w}\|_\infty \le 1} \langle \mathbf{y} - \mathbf{X}\boldsymbol{\beta}, \mathbf{w}\rangle$ we make use of the smoothing technique of Nesterov (2005) to obtain $g_1(\boldsymbol{\beta}; \tau) = \sup_{\|\mathbf{w}\|_\infty \le 1} (\langle \mathbf{y} - \mathbf{X}\boldsymbol{\beta}, \mathbf{w}\rangle - \frac{\tau}{2}\|\mathbf{w}\|_2^2)$. $g_1(\boldsymbol{\beta}; \tau)$ is a smooth approximation of $g_1(\beta)$, with $\ell = \lambda_{\max}(\mathbf{X}'\mathbf{X})/\tau$ and our algorithmic framework applies.

In order to obtain a good approximation to problem (3.13), we found the following strategy to be useful in practice:

1. Fix $\tau > 0$, initialize with $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ and repeat the following steps 2–3 till convergence:
2. Apply Algorithm 1 (or Algorithm 2) to the smooth function $g_1(\boldsymbol{\beta}; \tau)$. Let $\boldsymbol{\beta}_\tau^*$ be the limiting solution.
3. Decrease $\tau \leftarrow \tau\gamma$ for some predefined constant $\gamma = 0.8$ (say), and go back to step 1 with $\boldsymbol{\beta}_0 = \boldsymbol{\beta}_\tau^*$. Exit if $\tau <$ TOL, for some predefined tolerance.

4. A brief tour of statistical properties of problem (1.1). For the sake of completeness, we briefly review some statistical properties of problem (1.1) and contrast it with `Lasso` based solutions. Suppose, data is generated via a linear model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0 + \boldsymbol{\varepsilon}$, with $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, \sigma^2)$ and let $\widehat{\boldsymbol{\beta}}$ be a solution to (1.1). In terms of the expected (worst case) predictive risk, it is well known [Bunea, Tsybakov and Wegkamp (2007), Raskutti, Wainwright and Yu (2011), Zhang, Wainwright and Jordan (2014)] that the following upper bound holds:

$$\max_{\boldsymbol{\beta}^0 : \|\boldsymbol{\beta}^0\|_0 \le k} \frac{1}{n}\mathbb{E}\big(\|\mathbf{X}\boldsymbol{\beta}^0 - \mathbf{X}\widehat{\boldsymbol{\beta}}\|_2^2\big) \lesssim \sigma^2 \frac{k\log(p)}{n}, \tag{4.1}$$

where, "$\lesssim$" stands for "$\le$" up to universal constants. A natural question is: how do the bounds for `Lasso`-based solutions compare with (4.1)? Following Zhang, Wainwright and Jordan (2014), we define, for any subset $S \in \{1, 2, \ldots, p\}$, with size $|S| = k$; $C(S) := \{\boldsymbol{\beta} : \sum_{j \notin S} |\beta_j| \le 3\sum_{j \in S} |\beta_j|\}$. $\mathbf{X}$ is said to satisfy a restricted eigenvalue type condition with parameter $\gamma(X)$ if it satisfies:[4] $\frac{1}{n}\|\mathbf{X}\boldsymbol{\beta}\|_2^2 \ge \gamma(\mathbf{X})\|\boldsymbol{\beta}\|_2^2$ for $\boldsymbol{\beta} \in \bigcup_S C(S)$. Suppose $\widehat{\boldsymbol{\beta}}_{\ell_1}$ solves (1.2) with $\lambda = 4n\sigma\sqrt{\frac{\log p}{n}}$ and let $\widehat{\boldsymbol{\beta}}_{\mathrm{TL}}$ denote its thresholded version, which retains the top k entries of $\widehat{\boldsymbol{\beta}}_{\ell_1}$ in absolute value and sets the remaining to zero. Zhang, Wainwright and Jordan (2014)

[4]Note that $\gamma(\mathbf{X}) \le 1$ and $\gamma(\mathbf{X})$ is related to the so called compatibility condition [Bühlmann and van de Geer (2011)].

show that under such restricted eigenvalue type conditions the following holds:

$$\begin{aligned}\frac{\sigma^2}{\gamma(\mathbf{X}_{\text{bad}})^2}\frac{k^{1-\delta}\log(p)}{n} &\lesssim \max_{\boldsymbol{\beta}^0:\|\boldsymbol{\beta}^0\|_0\le k}\frac{1}{n}\mathbb{E}(\|\mathbf{X}\boldsymbol{\beta}^0-\mathbf{X}\widehat{\boldsymbol{\beta}}_{\text{TL}}\|_2^2)\\ &\lesssim \frac{\sigma^2}{\gamma(\mathbf{X})^2}\frac{k\log(p)}{n}.\end{aligned} \tag{4.2}$$

In particular, the lower bounds apply to *bad* design matrices $\mathbf{X}_{\text{bad}}$ for some arbitrarily small scalar $\delta > 0$. In light of (4.1) and (4.2), there is a significant gap between the predictive performances of subset selection procedures and `Lasso` based k-sparse solutions.[5] If $\gamma(\mathbf{X})$ is small (occurring, e.g., if the pairwise correlations between the features is quite high) this gap can be quite large.

Zhang and Zhang (2012) study statistical properties of solutions to problem (1.4). Raskutti, Wainwright and Yu (2011), Zhang and Zhang (2012) study estimation errors in regression coefficients, under additional minor assumptions on $\mathbf{X}$. Shen et al. (2013), Zhang and Zhang (2012) provide interesting theoretical analysis of the variable selection properties of (1.1) and (1.4), demonstrating their superior variable selection properties over `Lasso` based methods. In passing, we remark that Zhang and Zhang (2012) develop statistical properties of *inexact* solutions to problem (1.4). This may serve as theoretical support for *near global* solutions to problem (1.1), where the certificates of suboptimality are delivered by our MIO framework in terms of global lower bounds.

5. Computational experiments for subset selection with least squares loss. We present a variety of computational experiments to assess the algorithmic and statistical performances of our approach. We consider both the classical overdetermined case with $n > p$ (Section 5.2) and the high-dimensional $p \gg n$ case (Section 5.3) for the least squares loss function with support constraints.

5.1. *Description of experimental data.* We perform a series of experiments on both synthetic and real data.

Synthetic datasets. We consider a collection of problems where $\mathbf{x}_i \sim \mathrm{N}(\mathbf{0}, \Sigma)$, $i = 1, \ldots, n$ are independent realizations from a p-dimensional multivariate normal distribution with mean zero and covariance matrix $\Sigma := (\sigma_{ij})$. The columns of the $\mathbf{X}$ matrix were subsequently standardized to have unit ℓ_2 norm. For a fixed $\mathbf{X}$, we generated $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0 + \boldsymbol{\varepsilon}$, with $\varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. We denote the number of nonzeros in $\boldsymbol{\beta}^0$ by k_0. We define the Signal-to-Noise Ratio (SNR) of the problem as: $\text{SNR} = \frac{\text{var}(\mathbf{x}'\boldsymbol{\beta}^0)}{\sigma^2}$. We consider the following examples.

[5] In fact Zhang, Wainwright and Jordan (2014) establish a result stronger than (4.2), where $\widehat{\boldsymbol{\beta}}_{\text{TL}}$ can be replaced by a k-sparse estimate delivered by a polynomial time method.

EXAMPLE 1. We took $\sigma_{ij} = \rho^{|i-j|}$ for $i, j \in \{1, \ldots, p\} \times \{1, \ldots, p\}$. We consider different values of $k_0 \in \{5, 10\}$ and $\beta_i^0 = 1$ for k_0 equispaced values; rounding the indices to the nearest large integer value of $i \in \{1, 2, \ldots, p\}$ when required.

EXAMPLE 2. We took $\mathbf{\Sigma} = \mathbf{I}_{p\times p}$, $k_0 = 5$ and $\boldsymbol{\beta}^0 = (\mathbf{1}'_{5\times 1}, \mathbf{0}'_{p-5\times 1})' \in \mathbb{R}^p$.

EXAMPLE 3. We took $\mathbf{\Sigma} = \mathbf{I}_{p\times p}$, $k_0 = 10$ and $\beta_i^0 = \frac{1}{2} + (10 - \frac{1}{2})\frac{(i-1)}{k_0}$, $i = 1, \ldots, 10$ and $\beta_i^0 = 0, \forall i > 10$—i.e., a vector with ten nonzero entries, with the nonzero values being equally spaced in the interval $[\frac{1}{2}, 10]$.

EXAMPLE 4. We took $\mathbf{\Sigma} = \mathbf{I}_{p\times p}$, $k_0 = 6$ and $\boldsymbol{\beta}^0 = (-10, -6, -2, 2, 6, 10, \mathbf{0}_{p-6})$, that is, a vector with six nonzero entries, equally spaced in the interval $[-10, 10]$.

Real datasets. We considered the Diabetes dataset [Efron et al. (2004)] with all the second order interactions included in the model, which resulted in 64 predictors. We reduced the sample size to $n = 350$ by taking a random sample and standardized the response and the columns of the model matrix to have zero means and unit ℓ_2-norm.

In addition to the above, we also considered a real microarray dataset: the Leukemia data [Dettling (2004)] downloaded from http://stat.ethz.ch/~dettling/bagboost.html, with $n = 72$ binary responses and more than 3000 predictors. We standardized the response and features to have zero means and unit ℓ_2-norm. We reduced the set of features to 1000 by retaining the features maximally correlated (in absolute value) to the response. From the resulting matrix $\mathbf{X}$ with $n = 72$, $p = 1000$, we generated a semisynthetic dataset with continuous response as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0 + \varepsilon$, where the first five coefficients of $\boldsymbol{\beta}^0$ were taken as one and the rest as zero. The noise was distributed as $\varepsilon_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$, with σ^2 chosen to get a SNR $= 7$.

Computer specifications and software. Computations were carried out in a linux 64 bit server—Intel(R) Xeon(R) eight-core processor @ 1.80 GHz, 16 GB of RAM for the overdetermined $n > p$ case. For the $p > n$ examples, all computations were carried out on Columbia University's high performance computing facility, http://hpc.cc.columbia.edu/, on the `Yeti` cluster computing environment in a Dell Precision T7600 computer with an Intel Xeon E52687 sixteen-core processor @ 3.1 GHz, 128 GB of Ram. The discrete first-order methods were implemented in MATLAB 2012b. We used GUROBI [Gurobi (2013)] version 5.5, for the MIO solvers.

5.2. *The overdetermined regime*: $n > p$. Herein, we study the combined effect of using the discrete first-order methods with the MIO approach using the Diabetes dataset and synthetic datasets. Together, these methods show improvements in obtaining good upper bounds and in closing the MIO gap to certify global optimality. Using synthetic datasets where we know the true linear regression model, we perform side-by-side comparisons of this method with several other state-of-the-art algorithms designed to estimate sparse linear models.

5.2.1. *Obtaining good upper bounds*. We conducted experiments to evaluate the performance of our methods in terms of obtaining high quality solutions for problem (1.1). The following three algorithms were considered:

(a) Algorithm 2 with fifty random initializations.[6] We took the solution corresponding to the best objective value.

(b) MIO with cold start, that is, formulation (2.4) with a time limit of 500 seconds.

(c) MIO with warm start. This was the MIO formulation (2.4) initialized with a solution obtained from (a). The combined run was for a total of 500 seconds.

For the MIO formulation (2.4) above, since $n > p$, we massaged the objective function into the form (2.5), that is, a quadratic problem in p variables.

To compare the different algorithms in terms of the quality of upper bounds, we run for every instance all the algorithms and obtain the best solution among them, say, f_*. If f_{alg} denotes the value of the best subset objective function for method alg $\in \{(a), (b), (c)\}$, we define the relative accuracy of the solution obtained by "alg" as

$$\text{Relative accuracy} = (f_{\text{alg}} - f_*)/f_*. \tag{5.1}$$

Table 1 shows results for the Diabetes dataset for different values of k. For each algorithm, we report the time taken by it to reach the best objective value during the time of 500 seconds. Using the discrete first-order methods in combination with the MIO algorithm resulted in finding the best possible relative accuracy in a matter of a few minutes.

5.2.2. *Improving MIO performance via warm starts*. We performed a series of experiments on the Diabetes dataset to obtain a globally optimal solution to problem (1.1) via our approach and to understand the implications of using advanced warm starts to the MIO formulation in terms of certifying optimality. For each k, we ran Algorithm 2 with fifty random initializations, which took less than a few

[6] we took fifty random starting values around $\mathbf{0}$ of the form $\min(i-1, 1)\varepsilon, i = 1, \ldots, 50$, where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}_{p\times 1}, 4\mathbf{I})$. We chose Algorithm 2 since it provided better upper bounds than Algorithm 1. However, if Algorithm 1 is run with many more initializations, the best solution obtained is similar to Algorithm 2.

TABLE 1

Quality of upper bounds for problem (1.1) for the Diabetes dataset, for different values of k. We observe that MIO equipped with warm starts deliver the best upper bounds in the shortest overall times. The run time for the MIO with warm start includes the time taken by the discrete first-order method

	Discrete first-order		MIO cold start		MIO warm start	
k	Accuracy	Time	Accuracy	Time	Accuracy	Time
9	0.1306	1	0.0036	500	0	346
20	0.1541	1	0.0042	500	0	77
49	0.1915	1	0.0015	500	0	87
57	0.1933	1	0	500	0	2

seconds to run. We used the best solution as an advanced warm start to the MIO formulation (2.5). The MIO solver was provided with problem-specific bounds obtained via Section 2.3.3 with $\tau = 2$. For each of these examples, we also ran the MIO formulation without any such additional problem-specific information, that is, formulation (2.4)—we refer to this as "Cold Start." Figure 3 presents a representative subset of the results. We also experimented (not reported here, for brevity) with bounds implied by Sections 2.3.1, 2.3.2 and observed that the MIO formulation (2.5) armed with warm-starts and additional bounds closed the optimality gap faster than their "Cold Start" counterpart.

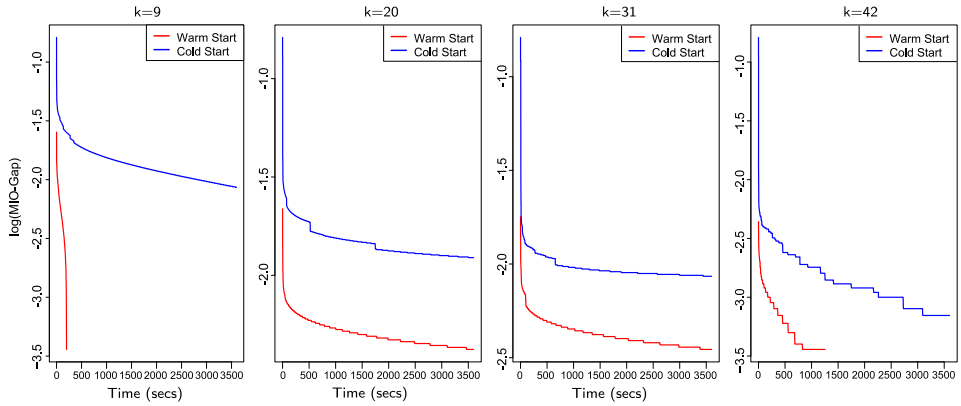

FIG. 3. *The evolution of the MIO optimality gap [in $\log_{10}(\cdot)$ scale] for problem (1.1), for the Diabetes dataset with $n = 350$, $p = 64$, for different values of k. Here, "Warm Start" indicates that the MIO was provided with warm starts and parameter specifications as in Section 2.3; and "Cold Start" indicates that MIO was not provided with any such problem-specific information. MIO ("Warm Start") is found close the optimality gap much faster. In all of these examples, the global optimum was found within a very small fraction of the total time, but the proof of global optimality came later.*

5.2.3. *Statistical performance.* We considered datasets as described in Example 1, Section 5.1—we took different values of n, p with $n > p$, ρ with $k_0 = 10$.

Competing methods and performance measures. For every example, we considered the following learning procedures for comparison purposes: (a) the MIO[7] formulation (2.4) equipped with warm starts from Algorithm 2 (annotated as "MIO" in the figure), (b) the `Lasso`, (c) `Sparsenet` and (d) stepwise regression (annotated as "Step" in the figure). In addition to the above, we have also performed comparisons with an *unshrunk* version of the `Lasso`, that is, performing unrestricted least squares on the `Lasso` support to mitigate the bias imparted by `Lasso` shrinkage.

We used `R` to compute `Lasso`, `Sparsenet` and stepwise regression using the glmnet 1.7.3, `Sparsenet` and Stats 3.0.2 packages, respectively, which were all downloaded from `CRAN` at http://cran.us.r-project.org/.

We note that `Sparsenet` [Mazumder, Friedman and Hastie (2011)] considers a penalized likelihood formulation of the form (1.3), where the penalty is given by the generalized MCP penalty family (indexed by λ, γ) for a family of values of $\gamma \geq 1$ and $\lambda \geq 0$. The family of penalties used by `Sparsenet` is thus given by: $p(t; \gamma; \lambda) = \lambda(|t| - \frac{t^2}{2\lambda\gamma})\mathbf{I}(|t| < \lambda\gamma) + \frac{\lambda^2\gamma}{2}\mathbf{I}(|t| \geq \lambda\gamma)$ for γ, λ described as above. As $\gamma = \infty$ with λ fixed, we get the penalty $p(t; \gamma; \lambda) = \lambda|t|$. This family includes as a special case ($\gamma = 1$), the hard thresholding penalty, recommended in Zheng, Fan and Lv (2014) for its useful statistical properties.

For each procedure, we obtained the "optimal" tuning parameter by selecting the model that achieved the best predictive performance on a held out validation set. Once the model $\widehat{\boldsymbol{\beta}}$ was selected, we obtained the prediction error as

$$\text{Prediction error} = \|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^0\|_2^2 / \|\mathbf{X}\boldsymbol{\beta}^0\|_2^2. \tag{5.2}$$

Note that, if the (sample) features are highly correlated, the selected model, may decide to choose a feature instead of its correlated surrogate—in such cases, variable selection error, measured in terms of Hamming distance with respect to the data-generating model, may be misleading. Size of the optimal model selected serves as a measure of the number of redundant variables selected by the model; and prediction error measures good data-fidelity. Thusly motivated, we report "prediction error" and number of nonzeros in the optimal model in our results. The results were averaged over ten random instances: for every run, the training and validation data had a fixed $\mathbf{X}$ but the noise $\boldsymbol{\varepsilon}$ was random.

Figure 4 presents results for data generated as per Example 1 with $n = 500$ and $p = 100$. We see that the MIO procedure performs very well across all the examples. Among the methods, MIO performs the best, followed by `Sparsenet`,

[7]Note that MIO formulation (2.6) with parameter bounds as in Section 2.3 may also be used here, with similar results.

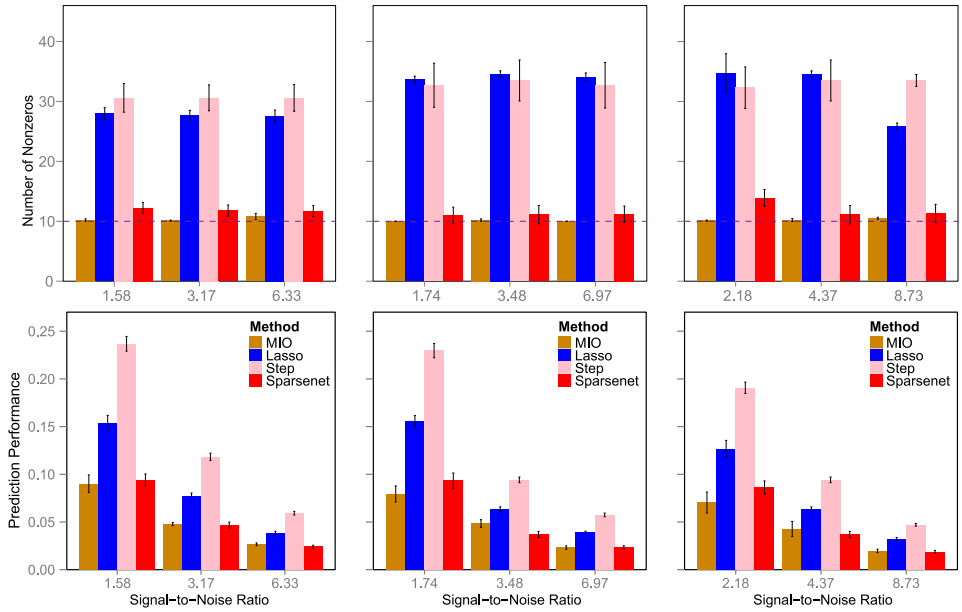

FIG. 4. *Figure showing the sparsity (upper panel) and predictive performances (bottom panel) for different subset selection procedures for the least squares loss. Here, we consider data generated as per Example 1, with* $n = 500$, $p = 100$, $k_0 = 10$, *for three different SNR values with (left panel)* $\rho = 0.5$, *(middle panel)* $\rho = 0.8$, *and (right panel)* $\rho = 0.9$. *The dashed line in the top panel represents the true number of nonzero values. For each of the procedures, the optimal model was selected as the one which produced the best prediction accuracy on a separate validation set, as described in Section 5.2.3.*

`Lasso` with `Step(wise)` exhibiting the worst performance—MIO consistently chose the sparsest model. `Lasso` delivers quite dense models and pays the price in predictive performance too, by selecting wrong variables. As the value of SNR increases, the predictive power of the methods improve, as expected. The differences in predictive errors between the methods diminish with increasing SNR values. With increasing values of ρ (from left panel to right panel in the figure), the number of nonzeros selected by the `Lasso` in the optimal model increases.

We also performed experiments with the *unshrunk* version of the `Lasso`. The unrestricted least squares solution on the optimal model selected by `Lasso` (as shown in Figure 4) had worse predictive performance than the `Lasso`, with the same sparsity pattern. This is probably due to overfitting since the model selected by the `Lasso` is quite dense compared to n, p. We also tried some variants of *unshrunk* `Lasso` which led to models with better performances than the `Lasso` but the results were inferior compared to MIO—we provide a detailed description in Section 10.2 in the supplementary material [Bertsimas, King and Mazumder (2015)].

We also did experiments (not reported here, for brevity) with $n = 1000$, $p = 50$ for Example 1 and found that MIO performed better compared to other competing methods.

5.2.4. *MIO model training*. We trained a sequence of best subset models (indexed by k) by applying the MIO approach with warm starts. Instead of running the MIO solvers from scratch for different values of k, we used the *callback* feature of integer optimization solvers. For each k, the MIO best subset algorithm was terminated the first time either an optimality gap of 1% was reached or a time limit of 15 minutes was reached.[8] Additionally, we considered values of k from 5 through 25.

5.3. *The high-dimensional regime*: $p \gg n$. Herein, we investigate (a) the evolution of upper bounds in the high-dimensional regime (see Section 5.3.1) (b) the effect of a bounding box formulation on closing the optimality gap (see Section 5.3.2) and (c) the statistical performance of the MIO approach in comparison to other sparse learning methods (see Section 5.3.3).

5.3.1. *Obtaining good upper bounds*. We performed experiments similar to those in Section 5.2.1, demonstrating the effectiveness of warm-starting MIO solvers with discrete first-order methods. We considered a synthetic dataset corresponding to Example 2 with $n = 30$, $p = 2000$ for varying SNR values (see Table 2) over a time of 500 s. As before, using the discrete first-order methods in combination with the MIO formulation (2.4) resulted in finding the best possible upper bounds in the shortest possible times.

Figure 5 shows the evolution of the objective value of problem (1.1) for different values of k, for the Leukemia dataset. For each k, we warm-started the MIO solver for formulation (2.4) with the solution obtained by Algorithm 2 and allowed the MIO solver to run for 4000 seconds—the resultant solution is denoted by f_*. We plot Relative Accuracy, that is, $(f_t - f_*)/f_*$, where f_t is the objective value obtained after t seconds. The figure shows that the solution obtained by Algorithm 2 is improved by the MIO on various instances and the time taken to improve the upper bounds depends upon k. In general, for smaller values of k the upper bounds obtained by the MIO algorithm stabilize earlier, that is, MIO finds improved solutions faster than larger values of k.

[8]We observed that it was possible to obtain speedups of a factor of 2–4 by carefully tuning the optimization solver for a particular problem, but chose to maintain generality by solving with default parameters. Thus, we do not report times with the intention of accurately benchmarking the best possible time but rather to show that it is computationally tractable to solve problems to optimality using modern MIO methods.

TABLE 2

The quality of upper bounds for problem (1.1) obtained by Algorithm 2, MIO with cold start and MIO warm-started with Algorithm 2. We consider Example 2 with $n = 30$, $p = 2000$ and different values of SNR. The MIO method, when warm-started with the first-order solution performs the best in terms of getting a good upper bound in the shortest time. Here, "Accuracy" is the same metric as defined in (5.1). The first-order methods work well, but need not lead to best quality solutions on their own. MIO improves the quality of upper bounds delivered by the first-order methods and their combined effect leads to the best performance

		Discrete first-order		**MIO cold start**		**MIO warm start**	
	k	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
SNR = 3	5	0.1647	37.2	1.0510	500	0	72.2
	6	0.6152	41.1	0.2769	500	0	77.1
	7	0.7843	40.7	0.8715	500	0	160.7
	8	0.5515	38.8	2.1797	500	0	295.8
	9	0.7131	45.0	0.4204	500	0	96.0
SNR = 7	5	0.5072	45.6	0.7737	500	0	65.6
	6	1.3221	40.3	0.5121	500	0	82.3
	7	0.9745	40.9	0.7578	500	0	210.9
	8	0.8293	40.5	1.8972	500	0	262.5
	9	1.1879	44.2	0.4515	500	0	254.2

5.3.2. *Bounding box formulation.* With the aid of advanced warm starts as provided by Algorithm 2, the MIO obtains a very high quality solution very quickly—in most of the examples the solution thus obtained turns out to be the

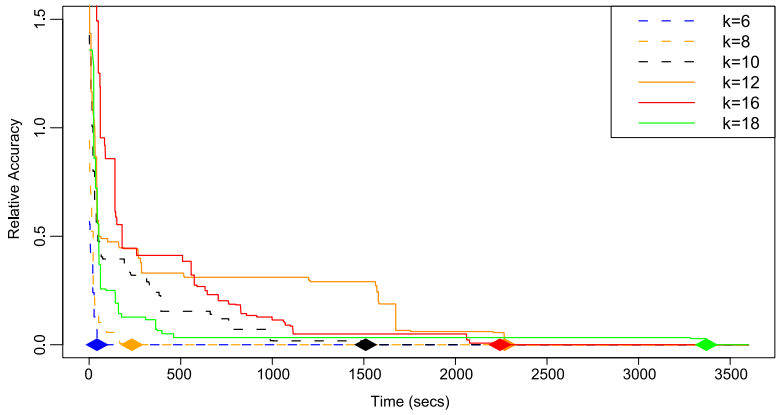

FIG. 5. *Behavior of MIO aided with warm start in obtaining good upper bounds for the Leukemia dataset ($n = 72$, $p = 1000$). The vertical axis shows relative accuracy, that is, $(f_t - f_*)/f_*$, where f_t is the objective value after t seconds and f_* denotes the best objective value obtained by the method after 4000 seconds. The colored diamonds correspond to the locations where the MIO (with warm start) attains the best solution. Note that MIO improves the solution obtained by the first-order method in all the instances.*

global minimum. However, in the typical "high-dimensional" regime, with $p \gg n$, we observe that the certificate of global optimality comes later as the lower bounds of the problem "evolve" slowly. This is observed even in the presence of warm starts and using the implied bounds as developed in Section 2.3 and is aggravated for the cold-started MIO formulation (2.4).

To address this, we consider a more structured MIO formulation (5.3) (presented below) obtained by adding bounding boxes around a local solution. These restrictions *guide* the MIO in restricting its *search* space and enable the MIO to certify global optimality inside that bounding box. We consider the following additional bounding box constraints to the MIO formulation (2.6):

$$\{\boldsymbol{\beta} : \|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}_0\|_1 \le \mathcal{L}^{\zeta}_{\ell,\mathrm{loc}}\} \cap \{\boldsymbol{\beta} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \le \mathcal{L}^{\beta}_{\ell,\mathrm{loc}}\},$$

where, $\boldsymbol{\beta}_0$ is a candidate sparse solution. The radii of the two ℓ_1-balls above, namely, $\mathcal{L}^{\zeta}_{\ell,\mathrm{loc}}$ and $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}}$ are user-defined parameters and control the size of the feasible set. Using the notation $\boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta}$, we have the following MIO formulation (equipped with the additional bounding boxes):

$$\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{z},\boldsymbol{\zeta}} \quad & \frac{1}{2}\boldsymbol{\zeta}^T\boldsymbol{\zeta} - \langle \mathbf{X}'\mathbf{y}, \boldsymbol{\beta}\rangle + \frac{1}{2}\|\mathbf{y}\|_2^2 \\
\text{s.t.} \quad & \boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta}, \\
& (\beta_i, 1 - z_i) : \text{SOS type-1}, \qquad i = 1, \ldots, p, \\
& z_i \in \{0, 1\}, \qquad i = 1, \ldots, p, \\
& \sum_{i=1}^{p} z_i \le k, \\
& -\mathcal{M}_U \le \beta_i \le \mathcal{M}_U, \qquad i = 1, \ldots, p, \\
& \|\boldsymbol{\beta}\|_1 \le \mathcal{M}_\ell, \\
& -\mathcal{M}^{\zeta}_U \le \zeta_i \le \mathcal{M}^{\zeta}_U, \qquad i = 1, \ldots, n, \\
& \|\boldsymbol{\zeta}\|_1 \le \mathcal{M}^{\zeta}_\ell, \\
& \|\boldsymbol{\zeta} - \boldsymbol{\zeta}_0\|_1 \le \mathcal{L}^{\zeta}_{\ell,\mathrm{loc}}, \\
& \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \le \mathcal{L}^{\beta}_{\ell,\mathrm{loc}}.
\end{aligned} \tag{5.3}$$

For large values of $\mathcal{L}^{\zeta}_{\ell,\mathrm{loc}}$ (resp., $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}}$) the constraints on $\mathbf{X}\boldsymbol{\beta}$ (resp., $\boldsymbol{\beta}$) become ineffective and one gets back formulation (2.6). To see the impact of these additional cutting planes in the MIO formulation, we consider a few examples as shown in Figures 6, 7 and Figure 11 (which can be found in the supplementary material [Bertsimas, King and Mazumder (2015)]).

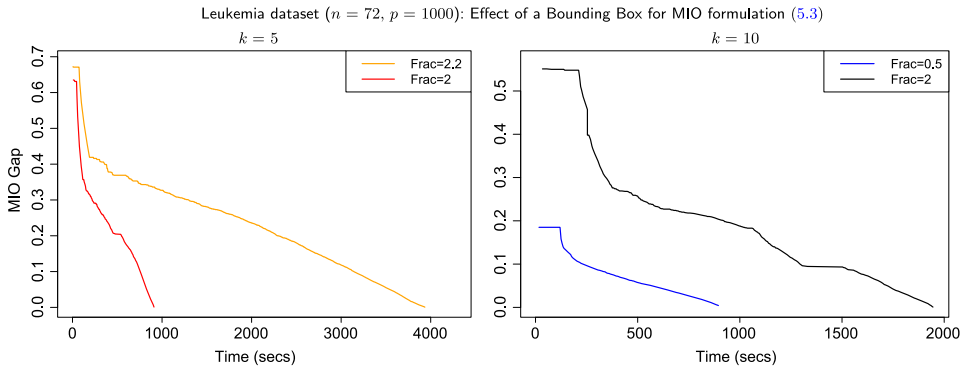

FIG. 6. *The effect of MIO formulation* (5.3) *for the Leukemia dataset. Here,* $\mathcal{L}^{\zeta}_{\ell,\mathrm{loc}} = \infty$ *and* $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}} = \mathrm{Frac}$. *For each* k, *the minimum obtained was the same for the different choices of* $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}}$.

Interpretation of the bounding boxes. A local bounding box in the variable $\boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta}$ directs the MIO solver to seek for candidate solutions that deliver models with predictive accuracy "similar" (controlled by the radius of the ball) to a reference predictive model, given by $\boldsymbol{\zeta}_0$. In our experiments, we typically chose $\boldsymbol{\zeta}_0$ as the solution delivered by running MIO (warm-started with a first-order solution) for a few hundred to a few thousand seconds. More generally, $\boldsymbol{\zeta}_0$ may be selected by any other sparse learning method. In our experiments, we found that the run-time behavior of the MIO depends upon how correlated the columns of $\mathbf{X}$ are—more

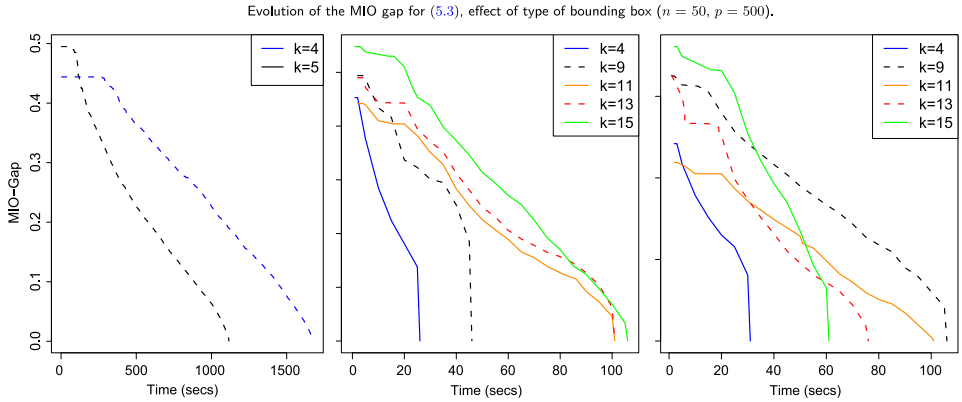

FIG. 7. *The effect of the MIO formulation* (5.3) *for a synthetic dataset as in Example* 1 *with* $\rho = 0.9$, $k_0 = 5$, $n = 50$, $p = 500$, *for different values of* k. (*Left panel*) $\mathcal{L}^{\zeta}_{\ell,\mathrm{loc}} = 0.5\|\mathbf{X}\boldsymbol{\beta}_0\|_1$, $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}} = \infty$ *and* SNR $= 3$. (*Middle panel*) $\mathcal{L}^{\zeta}_{\ell,\mathrm{loc}} = \infty$, $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}} = \|\boldsymbol{\beta}_0\|_1/k$ *and* SNR $= 1$. (*Right panel*) $\mathcal{L}^{\zeta}_{\ell,\mathrm{loc}} = \infty$, $\mathcal{L}^{\beta}_{\ell,\mathrm{loc}} = \|\boldsymbol{\beta}_0\|_1/k$ *and* SNR $= 3$. *The figure shows that the bounding boxes in terms of* $\mathbf{X}\boldsymbol{\beta}$ (*left–panel*) *make the problem harder to solve, when compared to bounding boxes around* $\boldsymbol{\beta}$ (*middle and right panels*). *A possible reason is due to the strong correlations among the columns of* $\mathbf{X}$. *The SNR values do not seem to have a big impact on the run-times of the algorithms* (*middle and right panels*).

correlation leading to longer run-times. Similarly, a bounding box around $\boldsymbol{\beta}$ directs the MIO to look for solutions in the neighborhood of a reference point $\boldsymbol{\beta}_0$. In our experiments, we chose the reference $\boldsymbol{\beta}_0$ as the solution obtained by MIO (warm-started with a first-order solution) and allowing it to run for a few hundred to a few thousand seconds. We observed that the MIO solver in presence of bounding boxes in the $\boldsymbol{\beta}$-space certified optimality and in the process finding better solutions; much faster than the $\boldsymbol{\zeta}$-bounding box method. Note that the $\boldsymbol{\beta}$-bounding box constraint leads to $O(p)$ and the $\boldsymbol{\zeta}$-box leads to $O(n)$ constraints. Thus, when $p \gg n$ the additional $\boldsymbol{\zeta}$ constraints add a fewer number of extra variables when compared to the $\boldsymbol{\beta}$ constraints.

Experiments. In the first set of experiments, we consider the Leukemia dataset with $n = 72$, $p = 1000$. We took two different values of $k \in \{5, 10\}$ and for each case we ran Algorithm 2 with several random restarts. The best solution thus obtained was used to warm start the MIO formulation (2.6), which we ran for an additional 3600 seconds. The solution thus obtained is denoted by $\boldsymbol{\beta}_0$. We then consider formulation (5.3) with $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$ and different values of $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \text{Frac}$ (as annotated in Figure 6)—the results are displayed in Figure 6.

We consider another set of experiments demonstrating the performance of MIO in certifying global optimality for different synthetic datasets with varying n, p, k as well as with different structures on the bounding box. In the first case, we generated data as per Example 1 with $\rho = 0.9$, $k_0 = 5$. We consider the case with $\boldsymbol{\zeta}_0 = \mathbf{X}\boldsymbol{\beta}_0$, $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \infty$ and $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = 0.5\|\mathbf{X}\boldsymbol{\beta}_0\|_1$, where $\boldsymbol{\beta}_0$ is a k-sparse solution obtained from the MIO formulation (2.6) run with a time limit of 1000 seconds, after being warm-started with Algorithm 2. The results are displayed in Figure 7 (left panel). In the second case (with data same as before), we obtained $\boldsymbol{\beta}_0$ in the same fashion as described before—we took a bounding box around $\boldsymbol{\beta}_0$, and left the box constraint around $\mathbf{X}\boldsymbol{\beta}_0$ inactive, that is, we set $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$ and $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \|\boldsymbol{\beta}_0\|_1/k$. We performed two sets of experiments, where the data were generated based on different SNR value—the results are displayed in Figure 7 with SNR $= 1$ (middle panel) and SNR $= 3$ (right panel).

In the same vein, we have Figure 11 [Bertsimas, King and Mazumder (2015)] studying the effect of formulations (5.3) for synthetic datasets generated as per Example 1 with $n = 50$, $p = 1000$, $\rho = 0.9$ and $k_0 = 5$.

5.3.3. *Statistical performance*. To understand the statistical behavior of MIO when compared to other approaches for learning sparse models, we considered synthetic datasets for values of n ranging from 30–50 and values of p ranging from 1000–2000. The following methods were used for comparison purposes (a) Algorithm 2. Here, we used fifty different random initializations around $\mathbf{0}$, of the form $\min(i-1, 1)N(\mathbf{0}_{p\times 1}, 4\mathbf{I})$, $i = 1, \ldots, 50$ and took the solution corresponding to the best objective value; (b) The MIO approach with warm starts from part (a); (c) The `Lasso` solution and (d) The `Sparsenet` solution.

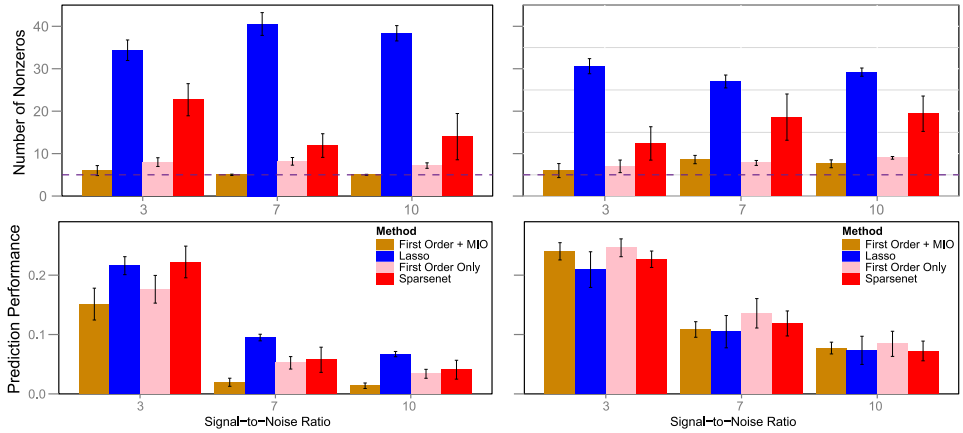

FIG. 8. *The sparsity and predictive performance for different procedures*: (*Left panel*) *shows Example* 1 *with* $n = 50$, $p = 1000$, $\rho = 0.8$, $k_0 = 5$ *and* (*right panel*) *shows Example* 2 *with* $n = 30$, $p = 1000$—*for each instance several SNR values have been shown*.

For methods (a), (b) we considered ten equispaced values of k in the range $[3, 2k_0]$ (including the optimal value of k_0). For each of the methods, the best model was selected in the same fashion as described in Section 5.2.3. For some of the above examples, we also study the performance of the *unshrunk* version of the `Lasso`. In Figures 8 and 9, we present selected representative results from four different examples described in Section 5.1. In Figure 8, the left panel shows the performance of different methods for Example 1 with $n = 50$, $p = 1000$, $\rho = 0.8$, $k_0 = 5$. In this example, there are five nonzero coefficients: the features cor-

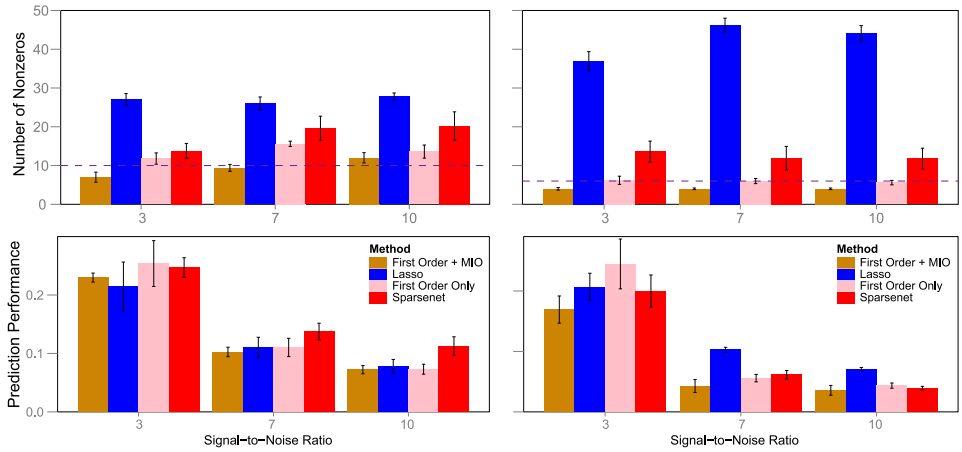

FIG. 9. (*Left panel*) *shows performance for data generated according to Example* 3 *with* $n = 30$, $p = 1000$ *and* (*right panel*) *shows Example* 4 *with* $n = 50$, $p = 2000$.

responding to the nonzero coefficients are weakly correlated and a feature having a nonzero coefficient is highly correlated with a feature having a zero coefficient (in the generating model). In this situation, the Lasso selects a very dense model since it fails to distinguish between a zero and a nonzero coefficient when the variables are correlated—it brings both the coefficients in the model (with shrinkage). MIO (with warm-start) performs the best—both in terms of predictive accuracy and in selecting a sparse set of coefficients. MIO obtains the sparsest model among the four methods and seems to find better solutions in terms of statistical properties than the models obtained by the first-order methods alone. Interestingly, the "optimal model" selected by the first-order methods is more dense than that selected by the MIO. The number of nonzero coefficients selected by MIO remains fairly stable across different SNR values, unlike the other three methods. For this example, we also experimented with the different versions of *unshrunk* Lasso [see results in Bertsimas, King and Mazumder (2015), Section 10.2 for details]. In summary: the best *unshrunk* Lasso models had performance marginally better than Lasso but quite inferior to MIO. Figure 8 (right panel) shows Example 2, with $n = 30$, $p = 1000$, $k_0 = 5$ and all nonzero coefficients equal one. Here, all methods perform similarly in terms of predictive accuracy. In fact, for the smallest value of SNR, the Lasso achieves the best predictive model. In all of the cases, however, the MIO achieves the sparsest model with favorable predictive accuracy. In Figure 9, for both the examples, the model matrix is an i.i.d. Gaussian ensemble. The underlying regression coefficient $\boldsymbol{\beta}^0$ however, is structurally different than Example 2 (as in Figure 8, right-panel). The alternating signs of $\boldsymbol{\beta}^0$ is responsible for different statistical behaviors of the four methods across Figures 8 (right-panel) and Figure 9 (both panels). The MIO (with warm-starts) seems to be the best among all the methods. For Example 3 (Figure 9, left panel), the predictive performances of Lasso and MIO are comparable—the MIO, however, delivers much sparser models than the Lasso.

The key conclusions are as follows:

1. The MIO best subset algorithm has a significant edge in detecting the correct sparsity structure for all examples compared to Lasso, Sparsenet and the stand-alone discrete first-order method.
2. For data generated as per Example 1 with large values of ρ, the MIO best subset algorithm gives better predictive performance compared to its competitors.
3. For data generated as per Examples 2 and 3, MIO delivers similar predictive models like the Lasso, but produces much sparser models. In fact, Lasso seems to perform marginally better than MIO, as a predictive model for small values of SNR.
4. For Example 4, MIO performs the best both in terms of predictive accuracy and delivering sparse models.

6. Computations for subset selection with least absolute deviation loss. Herein, we study the properties of the best subset selection problem with LAD objective (3.13) via a few representative examples. The LAD loss is appropriate when the error follows a heavy-tailed distribution. The datasets used for the experiments parallel those described in Section 5.1, the difference being in the distribution of $\boldsymbol{\varepsilon}$. We took ε_i i.i.d. from a double exponential distribution with variance σ^2. The value of σ^2 was adjusted to get different values of SNR.

Datasets analysed. We consider a set-up similar to Example 1 (Section 5.1) with $k_0 = 5$ and $\rho = 0.9$. Different choices of (n, p) were taken to cover both the overdetermined ($n = 500$, $p = 100$) and high-dimensional cases ($n = 50$, $p = 1000$ and $n = 500$, $p = 1000$).

Other methods used for comparison were (a) discrete first-order method (Section 3.3) (b) MIO warm-started with the first-order solutions and (c) the LAD loss with ℓ_1 regularization:

$$\min \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_1 + \lambda\|\boldsymbol{\beta}\|_1,$$

which we denote by `LAD-Lasso`. The training, validation and testing were done in the same fashion as in the least squares case. For each method, we report the number of nonzeros in the optimal model and associated prediction accuracy (5.2).

Figure 10 (left panel) compares the MIO approach with others for the overdetermined case ($n > p$). Figure 10 (middle and right panels) do the same for the high-dimensional case ($p \gg n$). The conclusions parallel those for the least squares

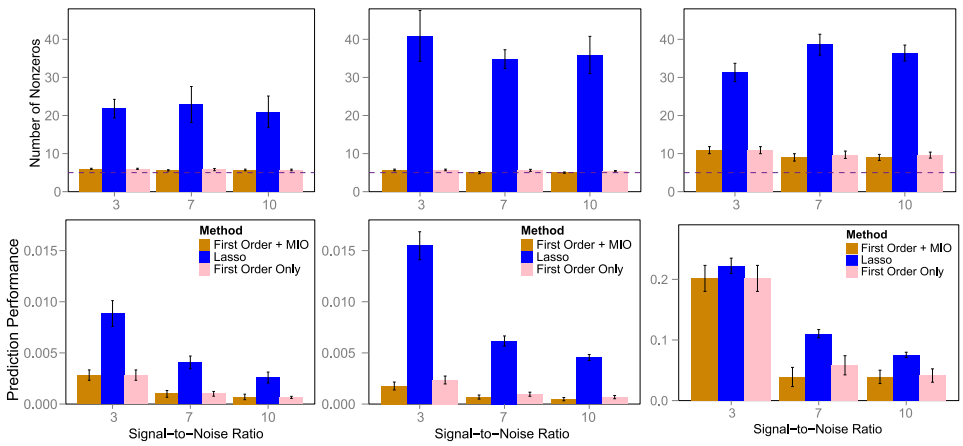

FIG. 10. *Figure showing the number of nonzero values and predictive performance for different values of n and p for problem* (3.13). *The data is generated as per Example* 1 *with* $\rho = 0.9, k_0 = 5$, *for different problem sizes—(left panel)* $n = 500$, $p = 100$; *(middle panel)* $n = 50$, $p = 1000$ *and (right panel)* $n = 500$, $p = 1000$. *The acronym "Lasso" refers to* `LAD-Lasso`. *MIO is seen to deliver sparser models with better predictive accuracy when compared to the* `LAD-Lasso`.

case. Since, in the example considered, the features corresponding to the nonzero coefficients are weakly correlated and a feature having a nonzero coefficient is highly correlated with a feature having a zero coefficient—the `LAD-Lasso` selects an overly dense model and misses out in terms of prediction error. Both the MIO (with warm-starts) and the discrete first-order methods behave similarly—much better than ℓ_1 regularization schemes. As expected, we observed that subset selection with least squares loss leads to inferior models for these examples, due to a heavy-tailed distribution of the errors.

Our findings here are similar to that in the least squares case. The MIO approach provides an edge both in terms of sparsity and predictive accuracy compared to Lasso both for the overdetermined and the high-dimensional case.

7. Conclusions. In this paper, we have revisited the classical best subset selection problem of choosing k out of p features in linear regression given n observations using a modern optimization lens—MIO and a discrete extension of first-order methods from continuous optimization. Exploiting the astonishing progress of MIO solvers in the last twenty-five years, we have shown that this approach solves problems with n in the 1000s and p in the 100s in minutes to provable optimality, and finds near optimal solutions for n in the 100s and p in the 1000s in minutes. Importantly, the solutions provided by the MIO approach often significantly outperform other state of the art methods like `Lasso` in achieving sparse models with good predictive power. Unlike all other methods, the MIO approach always provides a guarantee on its suboptimality even if the algorithm is terminated early. Moreover, it can accommodate side constraints on the coefficients of the linear regression and also extends to finding best subset solutions for the least absolute deviation loss function. While continuous optimization methods have played and continue to play an important role in statistics over the years, discrete optimization methods have not. The evidence in this paper as well as in Bertsimas and Mazumder (2014) suggests that MIO methods are tractable and lead to desirable properties (improved accuracy and sparsity among others) at the expense of higher, but still reasonable, computational times.

Acknowledgments. We would like to thank the Associate Editor and two anonymous reviewers for their comments that helped us improve the paper.

SUPPLEMENTARY MATERIAL

Supplement to "Best subset selection via a modern optimization lens" (DOI: 10.1214/15-AOS1388SUPP; .pdf). Supporting technical material and additional experimental results including some figures and tables are presented in the supplementary material section.

REFERENCES

BANDEIRA, A. S., DOBRIBAN, E., MIXON, D. G. and SAWIN, W. F. (2013). Certifying the restricted isometry property is hard. *IEEE Trans. Inform. Theory* **59** 3448–3450. MR3061257

BERTSIMAS, D., KING, A. and MAZUMDER, R. (2015). Supplement to "Best subset selection via a modern optimization lens." DOI:10.1214/15-AOS1388SUPP.

BERTSIMAS, D. and MAZUMDER, R. (2014). Least quantile regression via modern optimization. *Ann. Statist.* **42** 2494–2525. MR3277669

BERTSIMAS, D. and SHIODA, R. (2009). Algorithm for cardinality-constrained quadratic optimizaiton. *Comput. Optim. Appl.* **43** 1–22. MR2501042

BERTSIMAS, D. and WEISMANTEL, R. (2005). Optimization Over Integers. Dynamic Ideas, Belmont.

BICKEL, P. J., RITOV, Y. and TSYBAKOV, A. B. (2009). Simultaneous analysis of lasso and Dantzig selector. *Ann. Statist.* **37** 1705–1732. MR2533469

BIENSTOCK, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. *Math. Programming* **74** 121–140. MR1406746

BIXBY, R. E. (2012). A brief history of linear and mixed-integer programming computation. *Doc. Math. Extra Volume*: *Optimization Stories* 107–121. MR2991475

BLUMENSATH, T. and DAVIES, M. E. (2008). Iterative thresholding for sparse approximations. *J. Fourier Anal. Appl.* **14** 629–654. MR2461601

BLUMENSATH, T. and DAVIES, M. E. (2009). Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.* **27** 265–274. MR2559726

BÜHLMANN, P. and VAN DE GEER, S. (2011). *Statistics for High-Dimensional Data*: *Methods, Theory and Applications*. *Springer Series in Statistics*. Springer, Heidelberg. MR2807761

BUNEA, F., TSYBAKOV, A. B. and WEGKAMP, M. H. (2007). Aggregation for Gaussian regression. *Ann. Statist.* **35** 1674–1697. MR2351101

CANDÈS, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *C. R. Math. Acad. Sci. Paris* **346** 589–592. MR2412803

CANDÈS, E. J. and PLAN, Y. (2009). Near-ideal model selection by ℓ_1 minimization. *Ann. Statist.* **37** 2145–2177. MR2543688

CANDES, E. J. and TAO, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inform. Theory* **52** 5406–5425. MR2300700

CANDÈS, E. J., WAKIN, M. B. and BOYD, S. P. (2008). Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Appl.* **14** 877–905. MR2461611

CHEN, S. S., DONOHO, D. L. and SAUNDERS, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.* **20** 33–61. MR1639094

DETTLING, M. (2004). Bagboosting for tumor classification with gene expression data. *Bioinformatics* **20** 3583–3593.

DONOHO, D. L. (2006). For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math.* **59** 797–829. MR2217606

DONOHO, D. L. and ELAD, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization. *Proc. Natl. Acad. Sci. USA* **100** 2197–2202 (electronic). MR1963681

DONOHO, D. L. and HUO, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Inform. Theory* **47** 2845–2862. MR1872845

DONOHO, D. L. and JOHNSTONE, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81** 425–455. MR1311089

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* **32** 407–499. MR2060166

FAN, J. and LI, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96** 1348–1360. MR1946581

FAN, J. and LV, J. (2011). Nonconcave penalized likelihood with NP-dimensionality. *IEEE Trans. Inform. Theory* **57** 5467–5484. MR2849368

FAN, Y. and LV, J. (2013). Asymptotic equivalence of regularization methods in thresholded parameter space. *J. Amer. Statist. Assoc.* **108** 1044–1061. MR3174683

FRANK, I. and FRIEDMAN, J. (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics* **35** 109–148.

FRIEDMAN, J. (2008). Fast sparse regression and classification. Technical report, Dept. Statistics, Stanford Univ., Stanford, CA.

FRIEDMAN, J., HASTIE, T., HÖFLING, H. and TIBSHIRANI, R. (2007). Pathwise coordinate optimization. *Ann. Appl. Stat.* **1** 302–332. MR2415737

FURNIVAL, G. and WILSON, R. (1974). Regression by leaps and bounds. *Technometrics* **16** 499–511.

GREENSHTEIN, E. (2006). Best subset selection, persistence in high-dimensional statistical learning and optimization under l_1 constraint. *Ann. Statist.* **34** 2367–2386. MR2291503

GREENSHTEIN, E. and RITOV, Y. (2004). Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli* **10** 971–988. MR2108039

GUROBI, I. (2013). Optimization. Gurobi optimizer reference manual. Available at http://www.gurobi.com.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2009). *The Elements of Statistical Learning*: *Data Mining*, *Inference*, *and Prediction*, 2nd ed. *Springer Series in Statistics*. Springer, New York. MR2722294

KNIGHT, K. and FU, W. (2000). Asymptotics for lasso-type estimators. *Ann. Statist.* **28** 1356–1378. MR1805787

LOH, P.-L. and WAINWRIGHT, M. (2013). Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems* 476–484. Curran Associates, Red Hook, NY.

LV, J. and FAN, Y. (2009). A unified approach to model selection and sparse recovery using regularized least squares. *Ann. Statist.* **37** 3498–3528. MR2549567

MAZUMDER, R., FRIEDMAN, J. H. and HASTIE, T. (2011). *SparseNet*: Coordinate descent with nonconvex penalties. *J. Amer. Statist. Assoc.* **106** 1125–1138. MR2894769

MEINSHAUSEN, N. and BÜHLMANN, P. (2006). High-dimensional graphs and variable selection with the lasso. *Ann. Statist.* **34** 1436–1462. MR2278363

MILLER, A. (2002). *Subset Selection in Regression*, 2nd ed. *Monographs on Statistics and Applied Probability* **95**. Chapman & Hall/CRC, Boca Raton, FL. MR2001193

NATARAJAN, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM J. Comput.* **24** 227–234. MR1320206

NEMHAUSER, G. (2013). Integer programming: The global impact. 2013–12–2013–04, Rome, Italy. Presented at EURO, INFORMS, Accessed. Available at https://smartech.gatech.edu/handle/1853/49829.

NESTEROV, Y. (2004). *Introductory Lectures on Convex Optimization*: *A Basic Course*. *Applied Optimization* **87**. Kluwer Academic, Boston, MA. MR2142598

NESTEROV, YU. (2005). Smooth minimization of non-smooth functions. *Math. Program.* **103** 127–152. MR2166537

NESTEROV, YU. (2013). Gradient methods for minimizing composite functions. *Math. Program.* **140** 125–161. MR3071865

OPTIMIZATION INC. (2015). Gurobi 6.0 performance benchmarks. Available at http://www.gurobi.com/pdfs/benchmarks.pdf. Accessed 5 September 2015.

RASKUTTI, G., WAINWRIGHT, M. J. and YU, B. (2011). Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Trans. Inform. Theory* **57** 6976–6994. MR2882274

SHEN, X., PAN, W., ZHU, Y. and ZHOU, H. (2013). On constrained and regularized high-dimensional regression. *Ann. Inst. Statist. Math.* **65** 807–832. MR3105798

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288. MR1379242

TIBSHIRANI, R. (2011). Regression shrinkage and selection via the lasso: A retrospective. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **73** 273–282. MR2815776

TOP500 SUPERCOMPUTER SITES (2015). Directory page for Top500 lists. In Result for each list since June 1993. Accessed: 09-15-2015. Available at http://www.top500.org/statistics/sublist/.

TROPP, J. A. (2006). Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inform. Theory* **52** 1030–1051. MR2238069

VAN DE GEER, S., BÜHLMANN, P. and ZHOU, S. (2011). The adaptive and the thresholded Lasso for potentially misspecified models (and a lower bound for the Lasso). *Electron. J. Stat.* **5** 688–749. MR2820636

WAINWRIGHT, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Trans. Inform. Theory* **55** 2183–2202. MR2729873

ZHANG, C.-H. (2010a). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.* **38** 894–942. MR2604701

ZHANG, T. (2010b). Analysis of multi-stage convex relaxation for sparse regularization. *J. Mach. Learn. Res.* **11** 1081–1107. MR2629825

ZHANG, C.-H. and HUANG, J. (2008). The sparsity and bias of the LASSO selection in high-dimensional linear regression. *Ann. Statist.* **36** 1567–1594. MR2435448

ZHANG, Y., WAINWRIGHT, M. and JORDAN, M. I. (2014). Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. Preprint. Available at arXiv:1402.1918.

ZHANG, C.-H. and ZHANG, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statist. Sci.* **27** 576–593. MR3025135

ZHAO, P. and YU, B. (2006). On model selection consistency of Lasso. *J. Mach. Learn. Res.* **7** 2541–2563. MR2274449

ZHENG, Z., FAN, Y. and LV, J. (2014). High dimensional thresholded regression and shrinkage effect. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **76** 627–649. MR3210731

ZOU, H. (2006). The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.* **101** 1418–1429. MR2279469

ZOU, H. and LI, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Statist.* **36** 1509–1533. MR2435443

D. BERTSIMAS
R. MAZUMDER
MIT SLOAN SCHOOL OF MANAGEMENT
AND OPERATIONS RESEARCH CENTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS
USA
E-MAIL: dbertsim@mit.edu
rahulmaz@mit.edu

A. KING
OPERATIONS RESEARCH CENTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS
USA
E-MAIL: aking10@mit.edu