SIAM J. Optim.
Vol. 30, No. 4, pp. 3359–3386

SCALABLE ALGORITHMS FOR THE SPARSE RIDGE REGRESSION*

WEIJUN XIE† AND XINWEI DENG‡

Abstract. Sparse regression and variable selection for large-scale data have been rapidly developed in the past decades. This work focuses on sparse ridge regression, which enforces the sparsity by use of the L_0 norm. We first prove that the continuous relaxation of the mixed integer second order conic (MISOC) reformulation using perspective formulation is equivalent to that of the convex integer formulation proposed in recent work. We also show that the convex hull of the constraint system of the MISOC formulation is equal to its continuous relaxation. Based upon these two formulations (i.e., the MISOC formulation and convex integer formulation), we analyze two scalable algorithms, the greedy and randomized algorithms, for sparse ridge regression with desirable theoretical properties. The proposed algorithms are proved to yield near-optimal solutions under mild conditions. We further propose integrating the greedy algorithm with the randomized algorithm, which can greedily search the features from the nonzero subset identified by the continuous relaxation of the MISOC formulation. The merits of the proposed methods are illustrated through numerical examples in comparison with several existing ones.

Key words. approximation algorithm, chance constraint, conic program, mixed integer, ridge regression

AMS subject classifications. 90C11, 90C15, 62J07

DOI. 10.1137/19M1245414

1. Introduction.

This paper considers the following optimization problem:

$$v^* = \min_{\boldsymbol{\beta}} \left\{ \frac{1}{n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 : \|\boldsymbol{\beta}\|_0 \leq k \right\}. \tag{F0}$$

We refer to such an optimization problem as the *sparse ridge regression*, which is also studied by [5, 29, 38, 45]. In (F0), $\boldsymbol{y} \in \mathbb{R}^n$ denotes the response vector, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p] \in \mathbb{R}^{n \times p}$ represents the model matrix, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of regression coefficients (i.e., estimand), and $\lambda > 0$ is a positive tuning parameter for the ridge penalty (i.e., squared L_2 penalty). Additionally, $\|\boldsymbol{\beta}\|_0$ is the L_0 norm, which counts the number of nonzero entries of vector $\boldsymbol{\beta}$. The value of k represents the number of features to be chosen. In (F0), we aim to find the best k-sparse estimator, which minimizes the least squares error with a squared L_2 penalty. Without loss of generality, let us assume that $k \leq \min(n, p)$.

Note that formulation (F0) is quite general and is equivalent to the following convex quadratic program with an L_0 constraint,

$$\min_{\boldsymbol{\beta}} \left\{ \boldsymbol{\beta}^\top \boldsymbol{Q} \boldsymbol{\beta} - 2\boldsymbol{a}^\top \boldsymbol{\beta} + b : \|\boldsymbol{\beta}\|_0 \leq k \right\}, \tag{QP}$$

where $\boldsymbol{Q}$ is a symmetric and positive definite matrix. Formulation (QP) is equivalent to (F0) through the choice of λ as a positive number that is less than the smallest eigenvalue of $\boldsymbol{Q}$ and $\boldsymbol{X} = \sqrt{n}(\boldsymbol{Q} - \lambda \boldsymbol{I})^{1/2}, \boldsymbol{y} = \sqrt{n}(\boldsymbol{Q} - \lambda \boldsymbol{I})^{-1/2}\boldsymbol{a}, b = \boldsymbol{a}^\top (\boldsymbol{Q} - \lambda \boldsymbol{I})^{-1}\boldsymbol{a}$.

*Received by the editors February 19, 2019; accepted for publication (in revised form) September 14, 2020; published electronically December 17, 2020.
https://doi.org/10.1137/19M1245414
†Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA USA 24061 (wxie@vt.edu).
‡Department of Statistics, Virginia Tech, Blacksburg, VA USA 24061 (xdeng@vt.edu).

Sparse ridge regression (F0) can be reformulated as a chance constrained program (CCP) with finite support [1, 34]. That is, we consider p scenarios with equal probability $\frac{1}{p}$, where the ith scenario set is $S^i := \{\boldsymbol{\beta} : \beta_i = 0\}$ for $i \in [p]$. The constraint $\|\boldsymbol{\beta}\|_0 \le k$ means that at most k out of the p scenarios can be violated. Hence, we can reformulate (F0) as a CCP,

$$\text{(F0-CCP)} \qquad v^* = \min_{\boldsymbol{\beta}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 : \frac{1}{p}\sum_{i\in[p]} \mathbb{I}(|\beta_i| \le 0) \ge 1 - \frac{k}{p} \right\},$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. In section 2, we will investigate the extent of the recent progress on CCP (e.g., [1, 34, 43]) which can be used to solve (F0-CCP). It appears that many existing approaches may not work well due to the scalability issue or may result in trivial solutions. In section 4, we conduct and analyze two scalable algorithms as well as an integration of these two algorithms to solve the sparse ridge regression with theoretical guarantees.

Relevant literature. The ridge regression has been extensively studied in statistics [19, 36, 55]. It has been shown in the existing literature [19, 36, 55] that the additional ridge penalty $\lambda\|\boldsymbol{\beta}\|_2^2$ in (F0) has several desirable advantages, including stable solution, estimator variance reduction, and efficient computation. Recent progress in [7, 24] shows that under a certain distributional ambiguity set, the optimal regression coefficients found in (F0) are more robust than those from the conventional sparse regression model if the data $(\boldsymbol{X}, \boldsymbol{y})$ are insufficient or are subject to noise. However, although it has many desirable properties, the ridge estimator is often not sparse. Enabling sparsity in regression has been the focus of a significant amount of work, including the L_1 penalty [52], the Bridge estimator using the L_q $(q > 0)$ penalty [30], the nonconvex smoothly clipped absolute deviations (SCAD) penalty [58], and the minimax concave penalty [59], among many others. Several excellent and comprehensive reviews of sparse regression can be found in [8, 20, 27]. In particular, it is worth mentioning that in [61], the authors proposed a well-known "elastic net" approach, which integrates the ridge penalty (i.e., the squared L_2 penalty) and the L_1 penalty into the ordinary least squares objective to obtain a sparse estimator. However, similar to the L_1 penalty method, the elastic net might not consistently find an exact k-sparse estimator. In contrast, we introduce a constraint $\|\boldsymbol{\beta}\|_0 \le k$ in (F0), which strictly enforces the sparsity on $\boldsymbol{\beta}$ and therefore can obtain the best k-sparse estimator.

It has been proven that exact sparse linear regression (F0) with $\lambda = 0$ is NP-hard (cf. [42]), as is the sparse ridge regression (F0). Various effective approximation algorithms or heuristics have been introduced for solving sparse regression [17, 18, 23, 32, 33, 39]. For example, in [14] the authors studied the greedy approach (or forward stepwise selection method) and proved its approximation guarantee when the covariance matrix is nearly identity and has a constant bandwidth. In [15], the authors relaxed this assumption and showed that to maximize the R^2 statistic for linear regression, the greedy approach yields a constant approximation ratio under appropriate conditions. However, the greedy approach has been found to be prohibitively expensive when the number of features (i.e., p) becomes large [48]. Recently, [29] integrated coordinate descent with local combinatorial search and reported that the proposed method can numerically outperform existing ones. However, this method does not provide any provable guarantee on the global optimality. Many researchers have also attempted to solve sparse regression by developing exact algorithms (e.g., branch and cut) or using mixed integer program (MIP) solvers. It has been shown that for certain large-scale instances with large signal-to-noise ratios, MIP approaches with warm start (a good initial solution) work quite well and can yield very high quality solutions [2, 4, 5, 37, 38, 40, 41]. In particular, in [5] the authors also studied sparse ridge regression and developed a branch and cut algorithm. However, through our numerical study, these exact approaches can only solve medium-sized instances to near-optimality, and their performances highly rely on the speed of commercial solvers and can vary significantly from one dataset to another. In this work, our emphasis is on developing fast approximation algorithms with attractive scalability and theoretical performance guarantees.

Our approaches and contributions. In this work, we will focus on studying sparse ridge regression (F0) and deriving scalable algorithms. We will first investigate various existing approaches of CCP to solve (F0-CCP). One particular approach, which has been used to solve sparse regressions [4], is to introduce one binary variable for each indicator function in (F0-CCP) and linearize it with the big-M coefficient. However, such a method can be very slow in computation, particularly for large-scale datasets. To overcome this challenge, we develop a *big-M free* mixed integer second order conic (MISOC) reformulation for (F0-CCP). We further show that its continuous relaxation is equivalent to that of the mixed integer convex (MIC) formulation in [5, 17]. Moreover, these two formulations motivate us to construct a greedy approach (i.e., forward selection) in a much more efficient way than previously proposed in the literature. The performance guarantee of our greedy approach is also established. A randomized algorithm is studied by investigating the continuous relaxations of the proposed MISOC formulation. The numerical study shows that the proposed methods work quite well. In particular, the greedy approach outperforms the other methods in both running time and accuracy of variable selection. Our contributions are summarized as follows:

(i) We investigate theoretical properties of three existing approaches of CCP to solve (F0-CCP), i.e., the big-M method, the conditional-value-at-risk (i.e., **CVaR**) approach [43], and the heuristic algorithm from [1], and we shed some light on why those methods may not be amenable to solving the sparse ridge regression (F0).

(ii) We establish a mixed integer second order conic (MISOC) reformulation for (F0-CCP) from the perspective formulation of [26] and prove that our formulation's continuous relaxation is equivalent to that of the mixed integer convex (MIC) formulation in [5, 17]. We prove that the convex hull of the MISOC formulation is equivalent to its continuous relaxation. We also show that the proposed MISOC formulation can be stronger than the naive big-M formulation.

(iii) Based on the reformulations, we develop an efficient greedy approach for solving (F0-CCP) and prove its performance guarantee under a mild condition. The proposed greedy approach is theoretically sound and computationally efficient.

(iv) By establishing a relationship between the continuous relaxation value of the MISOC formulation and the optimal value of (F0-CCP) (i.e., v^*), we analyze a randomized algorithm based on the optimal continuous relaxation solution of the MISOC formulation and derive its theoretical properties. Such a continuous relaxation solution can help reduce the number of potential features and thus can be integrated with the greedy approach.

(v) Our numerical study shows that the proposed methods work quite well, particularly for large-scale instances, and that the proposed greedy approach can outperform other methods in both running time and accuracy.

The remainder of the paper is organized as follows. Section 2 investigates the applicability of several existing approaches of CCP for solving the sparse ridge regression (F0). Section 3 develops two big-M free MIC program formulations and proves their equivalence. Section 4 proposes and analyzes two scalable algorithms and proves their performance guarantees. Section 5 introduces the generalized cross validation for selecting a proper tuning parameter and presents a generalization of the proposed formulations to the sparse matrix estimation. The numerical experiments of the proposed scalable algorithms are presented in section 6. We conclude this work with some discussion in section 7.

The following notation is used throughout the paper. We use boldface letters (e.g., $\boldsymbol{x}, \boldsymbol{A}$) to denote vectors or matrices, and we use corresponding nonbold letters to denote their components. Given a positive integer number t, we let $[t]=\{1,\ldots,t\}$, and we let $\boldsymbol{I}_t$ denote the $t\times t$ identity matrix. Given a subset $S\subseteq[p]$, we let $\boldsymbol{\beta}_S$ denote the subvector of $\boldsymbol{\beta}$ with entries from a subset S, and $\boldsymbol{X}_S$ is a submatrix of $\boldsymbol{X}$ with columns from a subset S. For a matrix $\boldsymbol{Y}$, we let $\sigma_{\min}(\boldsymbol{Y})$ and $\sigma_{\max}(\boldsymbol{Y})$ denote its smallest and largest singular values, respectively. Given a vector $\boldsymbol{x}$, we let $\operatorname{diag}(\boldsymbol{x})$ be a diagonal matrix with diagonal entries from $\boldsymbol{x}$. For a matrix $\boldsymbol{W}$, we let $\boldsymbol{W}_{\bullet i}$ denotes its ith column. Given a set T, we let $\operatorname{conv}(T)$ denote its convex hull. Given a finite set S, we let $|S|$ denote its cardinality. Given two sets S,T, we let $S\setminus T$ denote the set of elements in S but not in T, let $S\cup T$ denote the union of S and T, and let $S\Delta T$ be their symmetric difference, i.e., $S\Delta T=(S\setminus T)\cup(T\setminus S)$.

2. Investigating existing solution approaches for solving CCP.

In this section, we investigate three commonly used approaches for solving (F0-CCP).

2.1. Big-M method.

One typical method for a CCP is formulating it as a MIP by introducing a binary variable z_i for each scenario $i\in[p]$, i.e., $\mathbb{I}(\beta_i\neq 0)\le z_i$, and then using a big-M method to linearize it; i.e., if $|\beta_i|\le M_i$ with a large positive number M_i, then $z_i\ge\mathbb{I}(\beta_i\neq 0)$ is equivalent to $|\beta_i|\le M_i z_i$. Therefore, (F0-CCP) can be reformulated as the following MIP:

(F0-big-M)

$$v^*=\min_{\boldsymbol{\beta},\boldsymbol{z}}\left\{\frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_2^2:\sum_{i\in[p]}z_i\le k,|\beta_i|\le M_i z_i,\boldsymbol{z}\in\{0,1\}^n\right\}.$$

Formulation (F0-big-M) has been widely used in recent works on sparse regression (see, e.g., [4, 5, 37, 38, 40, 41]). The advantage of (F0-big-M) is that it can be directly solved by off-the-shelf solvers (e.g., CPLEX, Gurobi). However, one has to choose the vector $\boldsymbol{M}=(M_1,\ldots,M_p)^\top$ properly.

It is known that (F0-big-M) with big-M coefficients typically has a very weak continuous relaxation value. Consequently, there has been significant research on improving the big-M coefficients of (F0-big-M); see, for example, [1, 4, 44, 46, 51]. However, the tightening procedures tend to be time-consuming, particularly for large-scale datasets. In section 3, we introduce two big-M free MIP formulations whose continuous relaxation is proved to be stronger than that of (F0-big-M).

2.2. CVaR approximation.

Another well-known approximation of CCP is the so-called conditional-value-at-risk (**CVaR**) approximation (see [43] for details), which is to replace the nonconvex probabilistic constraint by a convex **CVaR** constraint.

For the sparse ridge regression in (F0-CCP), the resulting formulation is

$$v^{\mathbf{CVaR}} = \min_{\boldsymbol{\beta}} \left\{ \frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 : \inf_t \left[-\frac{k}{p}t + \frac{1}{p}\sum_{i\in[p]} (|\beta_i|+t)_+ \right] \le 0 \right\}, \tag{1}$$

where $(w)_+ = \max(w,0)$. It is seen that (1) is a convex optimization problem and provides a feasible solution to (F0-CCP). Thus $v^{\mathbf{CVaR}} \ge v^*$. However, we observe that the only feasible solution to (1) is $\boldsymbol{\beta}=0$.

PROPOSITION 1. *The only feasible solution to* (1) *is* $\boldsymbol{\beta}=0$, *i.e.,* $v^{\mathbf{CVaR}} = \frac{1}{n}\|\boldsymbol{y}\|_2^2$.

Proof. We first observe that the infimum in (1) must be achievable. Indeed, $h(t) := -\frac{k}{p}t + \frac{1}{p}\sum_{i\in[p]}(|\beta_i|+t)_+$ is continuous and convex in t, and $\lim_{t\to\infty} h(t)=\infty$ and $\lim_{t\to-\infty} h(t)=\infty$. Therefore, the infimum in (1) must exist. Hence, in (1), we can replace the infimum by the following existence operator:

$$v^{\mathbf{CVaR}} = \min_{\boldsymbol{\beta}} \left\{ \frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 : \exists t, -\frac{k}{p}t + \frac{1}{p}\sum_{i\in[p]} (|\beta_i|+t)_+ \le 0 \right\}.$$

Since $\frac{1}{p}\sum_{i\in[p]}(|\beta_i|+t)_+ \ge 0$ and $\frac{k}{p}>0$, therefore $t\ge 0$, i.e.,

$$v^{\mathbf{CVaR}} = \min_{\boldsymbol{\beta}} \left\{ \frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 : \exists t\ge 0, \frac{p-k}{p}t + \frac{1}{p}\sum_{i\in[p]} |\beta_i| \le 0 \right\},$$

which implies that $t=0$ and $\beta_i=0$ for each $i\in[p]$. ∎

Therefore, the **CVaR** approach yields a trivial solution for (F0-CCP). Hence, it is not a desirable approach, and other alternatives are preferable.

2.3. Heuristic algorithm in [1]. In the recent work of [1], the authors proposed a heuristic algorithm for a CCP with a discrete distribution. It was reported that such a method could solve most of their numerical instances to near-optimality (i.e., within a 4% optimality gap). The key idea of the heuristic algorithm in [1] is to minimize the sum of infeasibilities for all scenarios when the objective value is upper bounded by v^U. Specifically, they considered the following optimization problem:

$$\min_{\boldsymbol{\beta}} \left\{ \sum_{i\in[p]} |\beta_i| : \frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \le v^U \right\}. \tag{2}$$

Let $\boldsymbol{\beta}_U^*$ be an optimal solution to (2) given an upper bound v^U of v^*. The heuristic algorithm is to decrease the value of v^U if $\|\boldsymbol{\beta}_U^*\|_0 \le k$ and increase it otherwise. This bisection procedure will terminate after a finite number of iterations. The detailed procedure is described in Algorithm 1. Let v^{heur} denote the output solution from Algorithm 1. Then clearly, we have the following.

PROPOSITION 2. *For Algorithm* 1*, the following two properties hold:*

(i) *It terminates with at most* $\lfloor \log_2(\frac{\|\boldsymbol{y}\|_2^2}{n\widehat{\delta}}) \rfloor + 1$ *iterations; and*

(ii) *it generates a feasible solution to* (F0-CCP)*, i.e.,* $v^* \le v^{\text{heur}}$.

Proof.

(i) To prove the first part, Algorithm 1 will terminate if and only if $U - L \leq \widehat{\delta}$. After one iteration, the difference between U and L is halved. Suppose Algorithm 1 will terminate within at most T steps. Then we must have

$$\frac{\|\boldsymbol{y}\|_2^2}{n2^{T-1}} > \widehat{\delta},$$

i.e., $T < 1 + \log_2\left(\frac{\|\boldsymbol{y}\|_2^2}{n\widehat{\delta}}\right)$.

(ii) We start with a feasible solution $\boldsymbol{\beta} = 0$ to (F0-CCP). In Algorithm 1, we keep track of the feasible solutions from iteration to iteration. Thus, the output of Algorithm 1 is a feasible solution to (F0-CCP), i.e., $v^* \leq v^{\text{heur}}$. □

Algorithm 1. Heuristic algorithm in [1].

Let $L = 0$ and $U = \frac{\|\boldsymbol{y}\|_2^2}{n}$ be known lower and upper bounds for (F0-CCP), let $\widehat{\delta} > 0$ be the stopping tolerance parameter.
while $U - L > \widehat{\delta}$ **do**
    $q \leftarrow (L + U)/2$.
    Let $\widehat{\boldsymbol{\beta}}$ be an optimal solution of (2) and set $\widehat{z}_i = \mathbb{I}(\widehat{\beta}_i = 0)$ for all $i \in [p]$.
    if $\sum_{i\in[p]} \widehat{z}_i \geq p - k$ **then**
        $U \leftarrow q$.
    else
        $L \leftarrow q$.
    end if
end while
Output $v^{\text{heur}} \leftarrow U$.

It is worth mentioning that for any given upper bound v^U, formulation (2) is similar to the elastic net proposed by [61], which can be interpreted as a Lagrangian relaxation of (2). The difference between Algorithm 1 and elastic net is that this iterative procedure simultaneously guarantees the sparsity and reduces the regression error, while elastic net seeks a trade-off among the regression error, squared L_2 penalty, and L_1 penalty of $\boldsymbol{\beta}$. We also note that Algorithm 1 might not be computationally efficient since it requires solving (2) multiple times, but a warm start from the solution of the previous iteration might help speed up the algorithm. Although there has been much development of the statistical properties of the elastic net method [16, 61], to the best of our knowledge, there is no known performance guarantee (i.e., approximation ratio) for Algorithm 1.

3. Investigating two big-M free reformulations and their formulation comparison. Note that the Big-M formulation in (F0-big-M) is quite compact since it only involves $2p$ variables (i.e., $\boldsymbol{\beta}, \boldsymbol{z}$). However, it is usually a weak formulation in the sense that the continuous relaxation value of (F0-big-M) can be quite far from the optimal value v^*. In this section, we propose two big-M free reformulations of (F0-CCP) that arise from two distinct perspectives and prove their equivalence.

3.1. Mixed integer second order conic formulation. In this subsection, we will present a MISOC formulation and its analytical properties. To begin, we first make an observation from the perspective formulation in [12, 18, 21, 26]; in [18], the authors introduced perspective relaxation for sparse regression with the L_0 penalty term, where they convexified a quadratic term using perspective formulation. Let us consider a nonconvex set

$$W_i := \left\{(\beta_i, \mu_i, z_i) : \beta_i^2 \le \mu_i, z_i \ge \mathbb{I}(\beta_i \ne 0), z_i \in \{0,1\}\right\} \tag{3}$$

for each $i \in [p]$. The result in [26] shows that the convex hull of W_i, denoted as $\operatorname{conv}(W_i)$, can be characterized as follows.

Lemma 1 (Lemma 3.1 in [26]). *For each $i \in [p]$, the convex hull of the set W_i is*

$$\operatorname{conv}(W_i) = \left\{(\beta_i, \mu_i, z_i) : \beta_i^2 \le \mu_i z_i, z_i \in [0,1]\right\}. \tag{4}$$

Lemma 1 suggests an extended formulation for (F0-CCP) without big-M coefficients. To achieve this goal, we first introduce a variable μ_i as the upper bound of β_i^2 for each $i \in [p]$, and then introduce a binary variable $z_i \ge \mathbb{I}(\beta_i \ne 0)$. Thus, (F0-CCP) is equal to

$$v^* = \min_{\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \lambda\|\boldsymbol{\mu}\|_1 : \sum_{i\in[p]} z_i \le k, (\beta_i, \mu_i, z_i) \in W_i \ \forall i \in [p] \right\},$$

which can be equivalently reformulated as

$$v^* = \min_{\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \lambda\|\boldsymbol{\mu}\|_1 : (\beta_i, \mu_i, z_i) \in \operatorname{conv}(W_i), z_i \in \{0,1\}, \forall i \in [p], \right.$$
$$\left. \sum_{i\in[p]} z_i \le k \right\}. \tag{5}$$

Note that (i) in (5), we replace W_i by $\operatorname{conv}(W_i)$ and enforce z_i to be binary for each $i \in [p]$; and (ii) from Lemma 1, $\operatorname{conv}(W_i)$ can be described by (4).

The above result is summarized in the following theorem.

Theorem 1. *The formulation* (F0-CCP) *is equivalent to*

(F0-MISOC)
$$v^* = \min_{\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \lambda\|\boldsymbol{\mu}\|_1 : \sum_{i\in[p]} z_i \le k, \beta_i^2 \le \mu_i z_i, z_i \in \{0,1\} \ \forall i \in [p] \right\}.$$

Formulation (F0-MISOC) introduces p more variables $\{\mu_i\}_{i\in[p]}$ than (F0-big-M), but it does not require any big-M coefficients.

Next, we show that the convex hull of the feasible region of (F0-MISOC) is equal to that of its continuous relaxation. Therefore, it suggests that we might not be able to improve the formulation by simply exploring the constraint system of (F0-MISOC). For notational convenience, let T denote the feasible region of (F0-MISOC), i.e.,

$$T = \left\{ (\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{z}) : \sum_{i\in[p]} z_i \le k, \beta_i^2 \le \mu_i z_i, z_i \in \{0,1\} \ \forall i \in [p] \right\}. \tag{6}$$

The following result indicates that the continuous relaxation of the set T is equivalent to $\operatorname{conv}(T)$.

Proposition 3. *Let T denote the feasible region of* (F0-MISOC). *Then*

$$\operatorname{conv}(T)=\left\{(\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}):\sum_{i\in[p]} z_i\leq k,\beta_i^2\leq \mu_i z_i, z_i\in[0,1]\ \forall i\in[p]\right\}.$$

Proof. Let $\widehat{T}$ be the continuous relaxation set of T, i.e.,

$$\widehat{T}=\left\{(\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}):\sum_{i\in[p]} z_i\leq k,\beta_i^2\leq \mu_i z_i, z_i\in[0,1]\ \forall i\in[p]\right\}.$$

We would like to show that $\operatorname{conv}(T)=\widehat{T}$. We separate the proof into two steps, i.e., prove $\operatorname{conv}(T)\subseteq\widehat{T}$ and $\widehat{T}\subseteq\operatorname{conv}(T)$.

(i) It is clear that $\operatorname{conv}(T)\subseteq\widehat{T}$.

(ii) To prove $\widehat{T}\subseteq\operatorname{conv}(T)$, we only need to show that for any given point $(\widehat{\boldsymbol{\beta}},\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{z}})\in\widehat{T}$, we have $(\widehat{\boldsymbol{\beta}},\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{z}})\in\operatorname{conv}(T)$. Since $\widehat{\boldsymbol{z}}\in\{\boldsymbol{z}:\sum_{i\in[p]}z_i\leq k,\boldsymbol{z}\in[0,1]^p\}$, which is an integral polytope, there exists K integral extreme points $\{\bar{\boldsymbol{z}}^t\}_{t\in[K]}\subseteq\mathbb{Z}_+^p$ such that $\widehat{\boldsymbol{z}}=\sum_{t\in[K]}\lambda_t\bar{\boldsymbol{z}}^t$ with $\lambda_t\in(0,1)$ for all t and $\sum_{t\in[K]}\lambda_t=1$. Now we construct $(\bar{\boldsymbol{\beta}}^t,\bar{\boldsymbol{\mu}}^t)$ for each $t\in[K]$ as follows:

$$\bar{\mu}_i^t=\begin{cases}\frac{\widehat{\mu}_i}{\widehat{z}_i} & \text{if } \ \bar{z}_i^t=1,\\ 0 & \text{otherwise},\end{cases}\qquad \bar{\beta}_i^t=\begin{cases}\frac{\widehat{\beta}_i}{\widehat{z}_i} & \text{if } \ \bar{z}_i^t=1\\ 0 & \text{otherwise}\end{cases}\ \forall i\in[p].$$

First, we claim that $(\bar{\boldsymbol{\beta}}^t,\bar{\boldsymbol{\mu}}^t,\bar{\boldsymbol{z}}^t)\in T$ for all $t\in[K]$. Indeed, for any $t\in[K]$,

$$(\bar{\beta}_i^t)^2=\begin{cases}\frac{(\widehat{\beta}_i)^2}{\widehat{z}_i^2} & \text{if } \ \bar{z}_i^t=1\\ 0 & \text{otherwise}\end{cases}\leq\bar{\mu}_i^t\bar{z}_i^t=\begin{cases}\frac{\widehat{\mu}_i}{\widehat{z}_i} & \text{if } \ \bar{z}_i^t=1\\ 0 & \text{otherwise}\end{cases}\ \forall i\in[p],$$
$$\sum_{i\in[p]}\bar{z}_i^t\leq k,$$
$$\bar{\boldsymbol{z}}^t\in\{0,1\}^p.$$

As $\widehat{\boldsymbol{z}}=\sum_{t\in[K]}\lambda_t\bar{\boldsymbol{z}}^t$, thus for each $i\in[p]$, we have

$$\sum_{t\in[K]}\lambda_t\bar{\mu}_i^t=\sum_{t\in[K]}\lambda_t\frac{\widehat{\mu}_i}{\widehat{z}_i}\bar{z}_i^t=\widehat{\mu}_i,$$
$$\sum_{t\in[K]}\lambda_t\bar{\beta}_i^k=\sum_{t\in[K]}\lambda_t\frac{\widehat{\beta}_i}{\widehat{z}_i}\bar{z}_i^t=\widehat{\beta}_i.$$

Thus, $(\widehat{\boldsymbol{\beta}},\widehat{\boldsymbol{\mu}},\widehat{\boldsymbol{z}})\in\operatorname{conv}(T)$. ∎

Finally, we remark that if an upper bound $\boldsymbol{M}$ of $\boldsymbol{\beta}$ is known, then (F0-MISOC) can be further strengthened by adding the constraints $|\beta_i|\leq M_iz_i$ for each $i\in[p]$. This result is summarized in the following proposition.

Proposition 4. *The formulation* (F0-CCP) *is equivalent to*

$$\text{(F0-MISOC-M)}\qquad v^*=\min_{(\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z})\in T}\left\{\frac{1}{n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\mu}\|_1:|\beta_i|\leq M_iz_i\ \forall i\in[p]\right\}$$

where $\boldsymbol{M}=(M_1,\dots,M_p)^\top$ denotes the vector of big-M coefficients, and the set T is defined as in (6).

Please note that the results in Proposition 3 and Proposition 4 can be generalized to a convex quadratic program with side constraints and an L_0 constraint [6] as in the portfolio optimization problem.

3.2. Mixed integer convex formulation. In this subsection, we will introduce an equivalent MIC formulation to (F0-CCP). The main idea is to separate the optimization in (F0-CCP) into two steps: (i) we optimize over $\boldsymbol{\beta}$ by fixing its nonzero entries with at most k, and (ii) we select the best subset of nonzero entries with size at most k. After the first step, it turns out that we can arrive at a convex integer program, which is big-M free. This result has been observed in the recent work of [5, 17].

Proposition 5 (see [5, 17]). *The formulation* (F0-CCP) *is equivalent to*

(F0-MIC)
$$v^* = \min_{\boldsymbol{z}} \left\{ f(\boldsymbol{z}) := \lambda \boldsymbol{y}^\top \left[n\lambda \boldsymbol{I}_n + \sum_{i\in[p]} z_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \right]^{-1} \boldsymbol{y} : \sum_{i\in[p]} z_i \leq k, \boldsymbol{z} \in \{0,1\}^p \right\}.$$

Note that in [5], the authors proposed a branch and cut algorithm for solving (F0-MIC), which was shown to be effective in solving some large-scale instances. In the next subsection, we will show that the continuous relaxation of (F0-MIC) is equivalent to that of (F0-MISOC). Therefore, it can be more appealing to solve (F0-MISOC) directly by MISOC solvers (e.g., CPLEX, Gurobi). Indeed, we numerically compare the branch and cut algorithm with directly solving (F0-MISOC) in section 6.

Finally, we remark that given the set of selected features $S \subseteq [p]$, its corresponding estimator $\widehat{\boldsymbol{\beta}}$ can be computed by the following formula:

$$\begin{cases} \widehat{\boldsymbol{\beta}}_S = \left(\boldsymbol{X}_S^\top \boldsymbol{X}_S + n\lambda \boldsymbol{I}_{|S|}\right)^{-1} \boldsymbol{X}_S^\top \boldsymbol{y} \\ \widehat{\beta}_i = 0 \quad \text{if } i \in [p] \setminus S \end{cases}, \tag{7}$$

where $\widehat{\boldsymbol{\beta}}_S$ denotes a subvector of $\widehat{\boldsymbol{\beta}}$ with entries from subset S.

3.3. Formulation comparisons. In this subsection, we will focus on comparing (F0-big-M), (F0-MISOC), (F0-MISOC-M), and (F0-MIC) according to their continuous relaxation bounds. First, let v_1, v_2, v_3, v_4 denote the continuous relaxation of (F0-big-M), (F0-MISOC), (F0-MISOC-M), and (F0-MIC), respectively, i.e.,

(8a)
$$v_1 = \min_{\boldsymbol{\beta},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 : \sum_{i\in[p]} z_i \leq k, |\beta_i| \leq M_i z_i, \boldsymbol{z} \in [0,1]^p \right\},$$

(8b)
$$v_2 = \min_{\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\mu}\|_1 : \beta_i^2 \leq \mu_i z_i \ \forall i \in [p], \sum_{i\in[p]} z_i \leq k, \boldsymbol{z} \in [0,1]^p \right\},$$

(8c)
$$v_3 = \min_{\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\mu}\|_1 : \beta_i^2 \leq \mu_i z_i, |\beta_i| \leq M_i z_i \ \forall i \in [p], \right.$$
$$\left. \sum_{i\in[p]} z_i \leq k, \boldsymbol{z} \in [0,1]^p \right\},$$

(8d)
$$v_4 = \min_{\boldsymbol{z}} \left\{ f(\boldsymbol{z}) = \lambda \boldsymbol{y}^\top \left[n\lambda \boldsymbol{I}_n + \sum_{i\in[p]} z_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \right]^{-1} \boldsymbol{y} : \sum_{i\in[p]} z_i \leq k, \boldsymbol{z} \in [0,1]^p \right\}.$$

Next, in the following theorem we will show a comparison of proposed formulations, i.e., (F0-big-M), (F0-MISOC), (F0-MISOC-M), and (F0-MIC). In particular, we prove that $v_2 = v_4$, i.e., the continuous relaxation bounds of (F0-MISOC) and (F0-MIC) coincide. In addition, we show that by adding big-M constraints $|\beta_i| \leq M_i z_i$ for each $i \in [p]$ into (F0-MISOC), we arrive at a tighter relaxation bound than that of (F0-big-M), i.e., $v_3 \geq v_1$.

THEOREM 2. *Let* v_1, v_2, v_3, v_4 *denote optimal values of* (8a), (8b), (8c), *and* (8d), *respectively. Then*
(i) $v_2 = v_4 \leq v_3$, *and*
(i) $v_1 \leq v_3$.

Proof. We separate the proof into three steps.

(1) We will prove $v_2 = v_4$ first. By Lemma A.1 of [47], we note that (8d) is equivalent to

$$\begin{aligned} v_4 = \min_{\boldsymbol{\gamma}_0, \boldsymbol{\gamma}, \boldsymbol{z}} \quad & \lambda \left(\|\boldsymbol{\gamma}_0\|_2^2 + \sum_{i\in[p]} \frac{\gamma_i^2}{z_i} \right) \\ \text{s.t.} \quad & \sqrt{\lambda n}\boldsymbol{\gamma}_0 + \sum_{i\in[p]} \boldsymbol{x}_i \gamma_i = \boldsymbol{y}, \\ & \sum_{i\in[p]} z_i \leq k, \\ & \boldsymbol{z} \in [0,1]^p, \boldsymbol{\gamma}_0 \in \mathbb{R}^n, \gamma_i \in \mathbb{R} \quad \forall i \in [p], \end{aligned}$$

where, by default, we let $\frac{0}{0} = 0$. Now let $\beta_i = \gamma_i$, and introduce a new variable μ_i to denote $\mu_i \geq \frac{\beta_i^2}{z_i}$ for each $i \in [p]$. Then the above formulation is equivalent to

$$\begin{aligned} v_4 = \min_{\boldsymbol{\gamma}_0, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{z}} \quad & \lambda \left(\|\boldsymbol{\gamma}_0\|_2^2 + \|\boldsymbol{\mu}\|_1 \right) \\ \text{s.t.} \quad & \sqrt{\lambda n}\boldsymbol{\gamma}_0 + \sum_{i\in[p]} \boldsymbol{x}_i \beta_i = y, \\ & \beta_i^2 \leq \mu_i z_i \quad \forall i \in [p], \\ & \sum_{i\in[p]} z_i \leq k, \\ & \boldsymbol{z} \in [0,1]^p, \boldsymbol{\gamma}_0 \in \mathbb{R}^n, \mu_i \in \mathbb{R}_+ \quad \forall i \in [p]. \end{aligned}$$

Finally, in the above formulation, substitute

$$\boldsymbol{\gamma}_0 = \frac{1}{\sqrt{\lambda n}} \left(\boldsymbol{y} - \sum_{i\in[p]} \boldsymbol{x}_i \beta_i \right) = \frac{1}{\sqrt{\lambda n}} \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \right)$$

into the objective function. Then we arrive at (8b).

(2) Next, we will prove $v_2 \leq v_3$. Note that the set of the constraints in (8b) is a subset of those in (8c). Thus, $v_2 \leq v_3$.
(3) Third, we will prove $v_1 \leq v_3$. We first note that v_1 is equivalent to

$$v_1 = \min_{\boldsymbol{\beta},\boldsymbol{\mu},\boldsymbol{z}} \left\{ \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\mu}\|_1 : \beta_i^2 \leq \mu_i, |\beta_i| \leq M_i z_i \ \forall i \in [p], \sum_{i\in[p]} z_i \leq k, \boldsymbol{z} \in [0,1]^p \right\}.$$

The result $v_1 \geq v_3$ follows directly by observing that the constraints $\beta_i^2 \leq \mu_i z_i$ for each $i \in [p]$ imply that $\beta_i^2 \leq \mu_i$ for each $i \in [p]$. ◻

Based on the results established in Theorem 2, we could directly solve the second order conic program (8b) to obtain the continuous relaxation of MIC (F0-MIC), which can be solved quite efficiently by existing solvers (e.g., CPLEX, Gurobi). In addition, adding big-M constraints $|\beta_i| \leq M_i z_i$ for each $i \in [p]$ into (8b), we see that the relaxation bound can be further improved.

Finally, we would like to elaborate that by choosing the vector $\boldsymbol{M}$ differently, the continuous relaxation bound v_2 of (F0-MISOC) can dominate v_1, the continuous relaxation bound of (F0-big-M), and vice versa.

Example 1. Consider the following instance of (F0-CCP) with $n = 2, p = 2, k = 1$ and $\boldsymbol{y} = (1,1)^\top, \boldsymbol{X} = \boldsymbol{I}_2$. Thus, in this case, we have $v^* = \frac{\lambda}{1+2\lambda} + \frac{1}{2}, v_2 = \frac{4\lambda}{1+4\lambda}$. There are two different choices available for $\boldsymbol{M} = (M_1, M_2)^\top$:

(i) If we choose $\boldsymbol{M}$ loosely, i.e., $M_1 = M_2 = \sqrt{\frac{\|\boldsymbol{y}\|_2^2}{n\lambda}} = \sqrt{\frac{1}{\lambda}}$, then

$$v_1 = \frac{2\lambda}{1+2\lambda} < v_2 < v^*$$

given that $\lambda > 0$.

(ii) If we choose $\boldsymbol{M}$ to be the tightest bound of the optimal solutions of (F0-CCP), i.e., $M_1 = M_2 = \frac{1}{1+2\lambda}$, then

$$v_2 < v_1 = \frac{8\lambda + 1}{8\lambda + 4} < v^*$$

given that $\lambda \in (0, 1/4)$.

4. Two scalable algorithms and their performance guarantees. In this section, we will study two scalable algorithms based upon two equivalent formulations (F0-MISOC) and (F0-MIC), i.e., the greedy approach based on (F0-MIC), and the randomized algorithm based on (F0-MISOC).

4.1. The greedy approach based on mixed integer convex formulation. The greedy approach (i.e., forward selection) has been commonly used as a heuristic to conduct the best subset selection [15, 50, 60]. The idea of the greedy approach is to select the feature that minimizes the marginal decrement of the objective value in (F0-MIC) at each iteration until the number of selected features reaches k. Note that when we are given a selected subset $S \subseteq [p]$ and an index $j \notin S$, the marginal objective value difference by adding j to S can be computed explicitly via the

Sherman–Morrison formula [49] as

$$\lambda \boldsymbol{y}^\top \left[\boldsymbol{A}_S + \boldsymbol{x}_j \boldsymbol{x}_j^\top\right]^{-1} \boldsymbol{y} - \lambda \boldsymbol{y}^\top A_S^{-1} \boldsymbol{y} = -\frac{\lambda \left(\boldsymbol{y}^\top \boldsymbol{A}_S^{-1} \boldsymbol{x}_j\right)^2}{1 + \boldsymbol{x}_j^\top \boldsymbol{A}_S^{-1} \boldsymbol{x}_j},$$

$$\left[\boldsymbol{A}_S + \boldsymbol{x}_j \boldsymbol{x}_j^\top\right]^{-1} = \boldsymbol{A}_S^{-1} - \frac{\boldsymbol{A}_S^{-1} \boldsymbol{x}_j \boldsymbol{x}_j^\top \boldsymbol{A}_S^{-1}}{1 + \boldsymbol{x}_j^\top \boldsymbol{A}_S^{-1} \boldsymbol{x}_j},$$

where $\boldsymbol{A}_S = n\lambda \boldsymbol{I}_n + \sum_{i\in S} \boldsymbol{x}_i \boldsymbol{x}_i^\top$.

This motivates an efficient implementation of the greedy approach, which is described in Algorithm 2. Note that in Algorithm 2, at each iteration, we only need to keep track of $\{\boldsymbol{A}_S^{-1}\boldsymbol{x}_j\}_{j\in[p]}$, $\{\boldsymbol{x}_j\boldsymbol{A}_S^{-1}\boldsymbol{x}_j\}_{j\in[p]}$, and $\{\boldsymbol{y}\boldsymbol{A}_S^{-1}\boldsymbol{x}_j\}_{j\in[p]}$, which have space complexity $O(np)$, and update them from one iteration to another, which costs $O(np)$ operations per iteration. Therefore, the space and time complexities of Algorithm 2 are $O(np)$ and $O(npk)$, respectively.

Algorithm 2. Proposed greedy approach for solving (F0-MIC).

1: Initialize $S = \emptyset$ and $\boldsymbol{A}_S = n\lambda \boldsymbol{I}_n$
2: **for** $i = 1, \ldots, k$ **do**
3: Let $j^* \in \arg\min_{j\in[p]\setminus S} \left\{ -\frac{\lambda(\boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{x}_j)^2}{1+\boldsymbol{x}_j^\top \boldsymbol{A}_S^{-1}\boldsymbol{x}_j} \right\}$
4: Let $S = S \cup \{j^*\}$ and $\boldsymbol{A}_S = \boldsymbol{A}_S + \boldsymbol{x}_{j^*}\boldsymbol{x}_{j^*}^\top$, $\boldsymbol{A}_S^{-1} = \boldsymbol{A}_S^{-1} - \frac{\boldsymbol{A}_S^{-1}\boldsymbol{x}_{j^*}\boldsymbol{x}_{j^*}^\top \boldsymbol{A}_S^{-1}}{1+\boldsymbol{x}_{j^*}^\top \boldsymbol{A}_S^{-1}\boldsymbol{x}_{j^*}}$
5: **end for**
6: Output $v^G \leftarrow \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1} \boldsymbol{y}$.

From our empirical study, we see that the greedy approach work quite well. Indeed, we will investigate the greedy solution and prove that it can be very close to the true optimal, particularly when λ is not too small. To begin, let us define θ_s to be the largest singular value of all the matrices $\boldsymbol{X}_S\boldsymbol{X}_S^\top$ with $|S| = s$, i.e.,

$$\theta_s := \max_{|S|=s} \sigma_{\max}^2(\boldsymbol{X}_S) = \max_{|S|=s} \sigma_{\max}(\boldsymbol{X}_S\boldsymbol{X}_S^\top), \tag{9}$$

for each $s \in [p]$. By definition (9), we have $\theta_1 \le \theta_2 \le \cdots \le \theta_p$, and by default, we let $\theta_0 = 0$.

Our main results of near-optimality of the greedy approach are stated as below. That is, if $p \ge k$, then the solution of the greedy approach will be quite close to any optimal estimator from (F0-CCP) as λ grows.

Theorem 3. *Suppose $p \ge k$. Then the output (i.e., v^G) of the greedy approach (i.e., Algorithm 2) is bounded by*

$$v^* \le v^G \le \frac{n\lambda + \theta_k}{n\lambda} \left(1 - \frac{n^2\lambda^2 \underline{\theta}}{(n\lambda+\theta_1)(n\lambda+\theta_k)^2} \log\left(\frac{p+1}{p+1-k}\right)\right) v^*, \tag{10}$$

where θ defined as in (9) *and*

$$\underline{\theta} = \min_{T\subseteq[p],|T|\ge p-k+1} \sigma_{\min}(\boldsymbol{X}_T\boldsymbol{X}_T^\top).$$

Proof. First, suppose that $\boldsymbol{z}^*$ is an optimal solution to (F0-MIC). According to the definition of θ_k, we have $n\lambda \boldsymbol{I}_n + \sum_{i\in[p]} z_i^* \boldsymbol{x}_i\boldsymbol{x}_i^\top \preceq (n\lambda+\theta_k)\boldsymbol{I}_n$. Thus,

$$v^* = \lambda \boldsymbol{y}^\top \left(n\lambda \boldsymbol{I}_n + \sum_{i\in[p]} z_i^* \boldsymbol{x}_i\boldsymbol{x}_i^\top\right) \boldsymbol{y} \geq \frac{\lambda}{n\lambda+\theta_k}\|\boldsymbol{y}\|_2^2. \tag{11}$$

On the other hand, according to step 3 of Algorithm 2, for any given S such that $|S| = s < k$, and $\boldsymbol{A}_S = n\lambda \boldsymbol{I}_n + \sum_{i\in S} \boldsymbol{x}_i\boldsymbol{x}_i^\top$ and $j \in [p]\setminus S$, we observe that

$$\lambda \boldsymbol{y}^\top \left[\boldsymbol{A}_S + \boldsymbol{x}_j\boldsymbol{x}_j^\top\right]^{-1} \boldsymbol{y} - \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{y} = -\frac{\lambda\left(\boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{x}_j\right)^2}{1 + x_j^\top \boldsymbol{A}_S^{-1} x_j}. \tag{12}$$

Thus, using the identity (12), we can prove by induction that the greedy value is upper bounded by

$$v^G \leq \left(1 - \frac{n^2\lambda^2\underline{\theta}}{(n\lambda+\theta_1)(n\lambda+\theta_k)^2}\sum_{i\in[k]}\frac{1}{p+1-i}\right)\frac{1}{n}\|\boldsymbol{y}\|_2^2. \tag{13}$$

Indeed, if $k = 0$, then (13) holds. Suppose that if $k = t \geq 0$, (13) holds. Now let $k = t+1$, and let S be the selected subset at iteration t. By induction, we have

$$\lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{y} \leq \left(1 - \frac{n^2\lambda^2\underline{\theta}}{(n\lambda+\theta_1)(n\lambda+\theta_k)^2}\sum_{i\in[t]}\frac{1}{p+1-i}\right)\frac{1}{n}\|\boldsymbol{y}\|_2^2.$$

By the greedy selection procedure, we further have

$$\begin{aligned}
v^G &= \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{y} + \min_{j\in[p]\setminus S} \lambda \boldsymbol{y}^\top \left[\boldsymbol{A}_S + \boldsymbol{x}_j\boldsymbol{x}_j^\top\right]^{-1} y - \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{y}\\
&\leq \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{y} + \frac{1}{p-t}\sum_{j\in[p]\setminus S}\left[-\frac{\lambda\left(\boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{x}_j\right)^2}{1 + x_j^\top \boldsymbol{A}_S^{-1} x_j}\right]\\
&\leq \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1}\boldsymbol{y} - \frac{n\lambda^2}{(p-t)(n\lambda+\theta_1)}\boldsymbol{y}^\top \boldsymbol{A}_S^{-1}(\boldsymbol{X}_{[p]\setminus S}\boldsymbol{X}_{[p]\setminus S}^\top)\boldsymbol{A}_S^{-1}\boldsymbol{y}\\
&\leq \left(1 - \frac{n^2\lambda^2\underline{\theta}}{(n\lambda+\theta_1)(n\lambda+\theta_k)^2}\sum_{i\in[t+1]}\frac{1}{p+1-i}\right)\frac{1}{n}\|\boldsymbol{y}\|_2^2,
\end{aligned}$$

where the first equality is due to (12), the first inequality is due to the minimum being no larger than the average, the second inequality is due to $\boldsymbol{A}_S \succeq n\lambda \boldsymbol{I}_n$ and $\|\boldsymbol{x}_j\|_2^2 \leq \theta_1$, and the third inequality is due to the induction and the facts that $p \geq k$, $A_S \preceq (n\lambda+\theta_k)\boldsymbol{I}_n$, and $\underline{\theta} \leq \sigma_{\min}(\boldsymbol{X}_{[p]\setminus S}\boldsymbol{X}_{[p]\setminus S}^\top)$.

Combining (11) and (12) and using the fact that $\sum_{i\in[k]}\frac{1}{p+1-i} \geq \int_0^k \frac{1}{p+1-t}dt = \log(\frac{p+1}{p+1-k})$, we see that the conclusion follows. ∎

We make the following remarks about Theorem 3.

(i) If $p < n+k$, then according to the definition, $\underline{\theta} = 0$.

(ii) If we normalize $\|\boldsymbol{x}_i\|_2^2 = n$ for each $i \in [p]$, we must have $\theta_k \leq kn$ and thus $\frac{n\lambda}{n\lambda+\theta_k} \leq \frac{\lambda}{\lambda+k}$. Therefore, we can see that the objective value of the greedy approach is closer to the true optimal value if the tuning parameter becomes larger.

(iii) Additionally, our analysis and asymptotic optimality of the greedy approach are new, without any assumption on the data, and thus are quite different from the existing results for sparse regression [10, 11, 14, 15, 31, 60]. For example, the results in [10, 11] require the well-known restricted isometry property (RIP), stated as

$$(1-\delta_s)\|\boldsymbol{\beta}\|_2^2 \leq \|\boldsymbol{X}\boldsymbol{\beta}\|_2^2 \leq (1+\delta_s)\|\boldsymbol{\beta}\|_2^2 \quad \forall s \in [p], \boldsymbol{\beta}: \|\boldsymbol{\beta}\|_0 = s,$$

where $\boldsymbol{\delta} \in (0,1)^p$ is a constant. This is quite a strong assumption, but our Theorem 3 does not require such an assumption. On the other hand, if the tuning parameter $\lambda \to 0_+$, then our performance guarantee can be arbitrarily bad. Therefore, our analysis cannot trivially extend to sparse regression.

In the next subsection, we will investigate a randomized algorithm and prove its approximation guarantee under a weaker condition of λ.

In addition, we remark that the estimator $\boldsymbol{\beta}^G$ of the greedy approach can be computed by (7), where S denotes the set of features selected by the greedy approach. In the next theorem, we will show that the derived estimator from the greedy approach (i.e., $\boldsymbol{\beta}^G$) can be also quite close to an optimal solution $\boldsymbol{\beta}^*$ of (F0-CCP).

THEOREM 4. *Let $\boldsymbol{\beta}^*$ be an optimal solution to* (F0-CCP) *with a set of selected features S^*, and let $\boldsymbol{\beta}^G$ be the estimator from the greedy approach with a set of selected features S^G. Suppose that $p \geq k$; then we have*

$$\|\boldsymbol{\beta}^G - \boldsymbol{\beta}^*\|_2 \leq \frac{\sqrt{4n\theta_{|S^G\setminus S^*|}v^*}}{n\lambda + \sigma_{\min}(\boldsymbol{X}_{S^U}^\top \boldsymbol{X}_{S^U})} + \sqrt{\frac{n\nu v^*}{n\lambda + \sigma_{\min}(\boldsymbol{X}_{S^U}^\top \boldsymbol{X}_{S^U})}},$$

where $S^U = S^G \cup S^*$, i.e., the union of set S^G and set S^*, and

$$\nu = \frac{n\lambda + \theta_k}{n\lambda}\left(1 - \frac{n^2\lambda^2\underline{\theta}}{(n\lambda+\theta_1)(n\lambda+\theta_k)^2}\log\left(\frac{p+1}{p+1-k}\right)\right) - 1.$$

Proof. Note that the greedy estimator $\boldsymbol{\beta}^G$ can be computed through (7) by setting S as S^G, the set of selected features by the greedy approach. Moreover, we define $\tilde{\boldsymbol{X}}$ as follows:

$$\begin{cases} \tilde{\boldsymbol{X}}_{S^G\setminus S^*} = \boldsymbol{X}_{S^G\setminus S^*}, \\ \tilde{\boldsymbol{X}}_{\bullet i} = 0 \quad \text{if } i \in [p] \setminus (S^G \setminus S^*). \end{cases}$$

Then we have

$$\begin{aligned}
&\frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^G\|_2^2 + \lambda\|\boldsymbol{\beta}^G\|_2^2 - \left[\frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_2^2\right] \leq \nu v^* \\
&(\Leftrightarrow) - 2\left(\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\right)^\top \left[-\frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*) + \lambda\boldsymbol{\beta}^*\right] \\
&\quad + \left(\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\right)^\top \left[\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I}_p\right]\left(\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\right) \leq \nu v^* \\
&(\Leftrightarrow) - 2\left(\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\right)^\top \left[-\frac{1}{n}\tilde{\boldsymbol{X}}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*)\right] \\
&\quad + \left(\boldsymbol{\beta}^*_{S^U} - \boldsymbol{\beta}^G_{S^U}\right)^\top \left[\frac{1}{n}\boldsymbol{X}_{S^U}^\top\boldsymbol{X}_{S^U} + \lambda\boldsymbol{I}_{|S^U|}\right]\left(\boldsymbol{\beta}^*_{S^U} - \boldsymbol{\beta}^G_{S^U}\right) \leq \nu v^* \\
&(\Rightarrow) - \frac{2}{n}\|\tilde{\boldsymbol{X}}\|_2\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2\|\boldsymbol{\beta}^*_{S^G\setminus S^*} - \boldsymbol{\beta}^G_{S^G\setminus S^*}\|_2
\end{aligned}$$

$$
\begin{aligned}
&\quad + \left(\lambda + \frac{\sigma_{\min}(\boldsymbol{X}_{S^U}^\top \boldsymbol{X}_{S^U})}{n}\right) \left\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\right\|_2^2 \le \nu v^* \\
&(\Rightarrow) - \sqrt{\frac{4\theta_{|S^G \setminus S^*|} v^*}{n}} \|\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\|_2 + \left(\lambda + \frac{\sigma_{\min}(\boldsymbol{X}_{S^U}^\top \boldsymbol{X}_{S^U})}{n}\right) \left\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^G\right\|_2^2 \le \nu v^* \\
&(\Rightarrow) \left\|\boldsymbol{\beta}^G - \boldsymbol{\beta}^*\right\|_2 \\
&\quad \le \frac{\sqrt{4n\theta_{|S^G \setminus S^*|} v^*}}{n\lambda + \sigma_{\min}(\boldsymbol{X}_{S^U}^\top \boldsymbol{X}_{S^U})} + \sqrt{\frac{n\nu v^*}{n\lambda + \sigma_{\min}(\boldsymbol{X}_{S^U}^\top \boldsymbol{X}_{S^U})}},
\end{aligned}
$$

where the second equivalence is due to the optimality condition of β^*, i.e., $-\frac{1}{n}\boldsymbol{X}_{S^*}^\top(\boldsymbol{y} - \boldsymbol{X}_{S^*}\boldsymbol{\beta}_{S^*}^*) + \lambda\boldsymbol{\beta}_{S^*}^* = 0$, and the nonzero entries of $\boldsymbol{\beta}^* - \boldsymbol{\beta}^G$ are only from subset $S^U := S^G \cup S^*$. The first implication is due to submultiplicativity of matrix norm and $\|\boldsymbol{A}\|_2 \ge \sigma_{\min}(\boldsymbol{A})$, the second implication is due to $\|\tilde{\boldsymbol{X}}\|_2 \le \sqrt{\theta_k}$, $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2 \le \sqrt{nv^*}$, and the last implication is due to any solution of the quadratic inequality $at^2 - bt - c \le 0$ with $a, b, c > 0$ being upper bounded by $\frac{b}{a} + \sqrt{\frac{c}{a}}$. ∎

Note that in Theorem 4, the first term of the error bound vanishes when $S^G = S^*$, i.e., when the greedy approach can exactly identify all the features.

4.2. The randomized algorithm based on mixed integer second order conic formulation. In this subsection, we investigate a randomized algorithm based on the continuous relaxation solution of (F0-MISOC), i.e., the optimal solution to (8b), which can be efficiently solved via the interior point method or other convex optimization approaches [3].

Suppose that $\widehat{\boldsymbol{z}}$ is the optimal solution of the continuous relaxation model (8b). For each $i \in [p]$, the column $\boldsymbol{x}_i$ will be picked by probability $\widehat{z}_i$. The detailed implementation is illustrated in Algorithm 3.

Algorithm 3. Proposed randomized algorithm.

1: Let $\widehat{\boldsymbol{z}}$ be the optimal solution to (8b)
2: Initialize set $S = \emptyset$ and vector $\tilde{\boldsymbol{z}} = 0 \in \mathbb{R}^p$
3: **for** $i = 1, \ldots, p$ **do**
4: Sample a standard uniform random variable U
5: **if** $U \le \widehat{z}_i$ **then**
6: Let $S = S \cup \{i\}$ and $\tilde{z}_i = 1$
7: **end if**
8: **end for**
9: Output $S, \tilde{\boldsymbol{z}}$

Next, we will show that if λ is not too small, then with high probability, the output S of Algorithm 3 yields its corresponding objective value close to the optimal value v^*. To begin, we present the following matrix concentration bound.

LEMMA 2 (Theorem 1.4 of [53]). *Consider a finite sequence $\{\boldsymbol{Y}_k\}$ of independent, random, symmetric matrices with dimension d. Assume that each random matrix satisfies $E[\boldsymbol{Y}_k] = 0$ and $\|\boldsymbol{Y}_k\|_2^2 \le R^2$ almost surely. Then, for all $t \ge 0$, we have*

$$
\mathbb{P}\left\{\left\|\sum_k \boldsymbol{Y}_k\right\|_2 \ge t\right\} \le d \exp\left(-\frac{t^2}{2\nu^2 + 2/3Rt}\right), \tag{14}
$$

where $\nu^2 := \|\sum_k \mathbb{E}[\boldsymbol{Y}_k^2]\|_2$.

Lemma 2 implies that if λ is not too small, then with high probability, $\lambda n\boldsymbol{I}_n + \sum_{i\in S}\boldsymbol{x}_i\boldsymbol{x}_i^\top$ has eigenvalues similar to those of $\lambda n\boldsymbol{I}_n + \sum_{i\in[p]}\widehat{z}_i\boldsymbol{x}_i\boldsymbol{x}_i^\top$, where $\widehat{\boldsymbol{z}}$ is the optimal solution to (8b) and S is the output of Algorithm 3.

LEMMA 3. *Let $\widehat{\boldsymbol{z}}$ be the optimal solution to* (8b)*, and let S be the output of Algorithm* 3*. Given that $\alpha\in(0,1)$ and*

$$\lambda \geq \frac{\log(2n/\alpha)\sqrt{\theta_1}}{3n\epsilon} + \frac{\sqrt{2\theta_k\log(2n/\alpha)}}{2n\epsilon},$$

then with probability at least $1-\frac{\alpha}{2}$, we have

$$(1-\epsilon)\boldsymbol{u}^\top\boldsymbol{\Sigma}_*\boldsymbol{u} \leq \boldsymbol{u}^\top\widehat{\boldsymbol{\Sigma}}\boldsymbol{u} \leq (1+\epsilon)\boldsymbol{u}^\top\boldsymbol{\Sigma}_*\boldsymbol{u} \quad \forall \boldsymbol{u}\in\mathbb{R}^n,$$

where $\boldsymbol{\Sigma}_* = \lambda n\boldsymbol{I}_n + \sum_{i\in[p]}\widehat{z}_i\boldsymbol{x}_i\boldsymbol{x}_i^\top$ and $\widehat{\boldsymbol{\Sigma}} = \lambda n\boldsymbol{I}_n + \sum_{i\in S}\boldsymbol{x}_i\boldsymbol{x}_i^\top$.

Proof. Let $\widehat{\boldsymbol{z}}$ be the optimal solution to (8b), and let $\{r_i\}_{i\in[p]}$ be independent Bernoulli random variables with $\mathbb{P}\{r_i=1\}=\widehat{z}_i$ for each $i\in[p]$. Consider the random matrix defined for each $i\in[p]$ as

$$\boldsymbol{A}_i = (r_i - \widehat{z}_i)\boldsymbol{x}_i\boldsymbol{x}_i^\top.$$

Clearly, we have $\mathbb{E}[\boldsymbol{A}_i]=0$. On the other hand, by definition we have $\|\boldsymbol{x}_i\|_2^2\leq\theta_1$ for each $i\in[p]$, and thus

$$\|\boldsymbol{A}_i\|_2 = |r_i-\widehat{z}_i|\|\boldsymbol{x}_i\|_2^2 \leq \theta_1 := R^2.$$

Also,

$$\begin{aligned}\left\|\sum_{i\in[p]}\mathbb{E}[\boldsymbol{A}_i^2]\right\|_2 &= \left\|\sum_{i\in[p]}\widehat{z}_i\left(1-\widehat{z}_i\right)\|\boldsymbol{x}_i\|_2^2\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|_2 = \left\|\sum_{i\in[p]}\widehat{z}_i\left(1-\widehat{z}_i\right)\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|_2\\ &\leq \left\|\sum_{i\in[p]}\widehat{z}_i\boldsymbol{x}_i\boldsymbol{x}_i^\top\right\|_2 \leq \theta_k,\end{aligned}$$

where the first inequality is due to the triangle inequality and $\|\boldsymbol{x}_i\|_2^2=1$ for each $i\in[p]$, the second inequality is due to $1-\widehat{z}_i\in[0,1]$ for all $i\in[p]$, and the last one is due to

$$\begin{aligned}&\max_{\boldsymbol{z}\in[0,1]^p}\left\{\sigma_{\max}\left(z_i\boldsymbol{x}_i\boldsymbol{x}_i^\top\right) : \sum_{i\in[p]}z_i = k\right\}\\ &= \max_{\boldsymbol{z}\in\{0,1\}^p}\left\{\sigma_{\max}\left(z_i\boldsymbol{x}_i\boldsymbol{x}_i^\top\right) : \sum_{i\in[p]}z_i = k\right\} := \theta_k.\end{aligned}$$

Now by Lemma 2 with $\sigma_{\min}(\boldsymbol{\Sigma}_*)$ denoting the smallest eigenvalue of $\boldsymbol{\Sigma}_*$ and $t=\epsilon\sigma_{\min}(\boldsymbol{\Sigma}_*)$, we have

$$\mathbb{P}\left\{\left\|\sum_{i\in[p]}\left(\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}_*\right)\right\|_2 \geq \epsilon\sigma_{\min}(\boldsymbol{\Sigma}_*)\right\} \leq n\exp\left(-\frac{\epsilon^2\sigma_{\min}^2(\boldsymbol{\Sigma}_*)}{2\theta_k + 2/3\epsilon\sqrt{\theta_1}\sigma_{\min}(\boldsymbol{\Sigma}_*)}\right).$$

We would like to ensure that the right-hand side of the above inequality is at most $\frac{\alpha}{2}$.

Thus,

$$\begin{aligned} &\mathbb{P}\left\{\left\|\sum_{i\in[p]}\left(\widehat{\mathbf{\Sigma}}-\mathbf{\Sigma}_*\right)\right\|_2 \geq \epsilon\sigma_{\min}(\mathbf{\Sigma}_*)\right\} \leq \frac{\alpha}{2} \\ (\Leftarrow)\quad & n\exp\left(-\frac{\epsilon^2\sigma_{\min}^2(\mathbf{\Sigma}_*)}{2\theta_k+2/3\epsilon\sigma_{\min}(\mathbf{\Sigma}_*)}\right)\leq\frac{\alpha}{2} \\ (\Leftarrow)\quad & \sigma_{\min}(\mathbf{\Sigma}_*)\geq\frac{\log(2n/\alpha)\sqrt{\theta_1}}{3\epsilon}+\frac{\sqrt{2\theta_k\log(2n/\alpha)}}{2\epsilon} \\ (\Leftarrow)\quad & \lambda\geq\frac{\log(2n/\alpha)\sqrt{\theta_1}}{3n\epsilon}+\frac{\sqrt{2\theta_k\log(2n/\alpha)}}{2n\epsilon} \end{aligned}$$

where the second implication is due to the quadratic inequality $at^2 - bt - c \geq 0$ with $a, b, c > 0$ being satisfied if $t \geq \frac{b}{a} + \sqrt{\frac{c}{a}}$, and the third implication is due to $\lambda n \leq \sigma_{\min}(\mathbf{\Sigma}_*)$.

Then the conclusion follows directly by Weyl's theorem [22, 57]. ∎

Based on Lemma 3, we can imply the following bicriteria approximation of (F0).

THEOREM 5. *Let $(S, \tilde{\boldsymbol{z}})$ be the output of Algorithm 3. Given that $\alpha \in (0,1)$ and*

$$\lambda\geq\frac{\log(2n/\alpha)\sqrt{\theta_1}}{3n\epsilon}+\frac{\sqrt{2\theta_k\log(2n/\alpha)}}{2n\epsilon},$$

with probability at least $1-\alpha$, we have

$$v^R := \lambda\boldsymbol{y}^\top\left[\lambda n\boldsymbol{I}_n+\sum_{i\in[p]}\tilde{z}_i\boldsymbol{x}_i\boldsymbol{x}_i^\top\right]^{-1}\boldsymbol{y}\leq(1+\epsilon)v^* \tag{15}$$

and

$$\sum_{i\in[p]}\tilde{z}_i\leq\left(1+\sqrt{\frac{3\log(2/\alpha)}{k}}\right)k. \tag{16}$$

Proof. Note that (15) follows from Lemma 3. The result in (16) holds due to the Chernoff bound [13], i.e.,

$$\mathbb{P}\left\{\sum_{i\in[p]}\tilde{z}_i\leq\left(1+\sqrt{\frac{3\log(2/\alpha)}{k}}\right)k\right\}\geq 1-e^{-\frac{\left(\sqrt{\frac{3\log(2/\alpha)}{k}}\right)^2 k}{3}}\geq 1-\frac{\alpha}{2}.$$

Therefore, by Boole's inequality, we arrive at the conclusion. ∎

When revising this paper, we discovered the very interesting paper [45], which also studied the same randomized rounding algorithms. Our results are distinguished from the work in [45] through two aspects: (i) We propose a second order conic program for obtaining the continuous relaxation solution, while [45] proposed a gradient decent method to solve it; and (ii) our approximation ratio is multiplicative and does not depend on p, while Theorem 3 in [45] derived an additive approximation bound, which is proportional to the square root of support of the continuous relaxation solution and thus can be $O(\sqrt{p})$. That is, using our notation, our approximation ratio is

$$v^R \leq \left(1 + \frac{\log(2n/\alpha)\sqrt{\theta_1}}{3n\lambda} + \frac{\sqrt{2\theta_k \log(2n/\alpha)}}{2n\lambda}\right) v^*,$$

and the approximation bound v^p in [45] is

$$v^p - v^* \leq c_4 \frac{\sqrt{r \log(\min\{r, n\})}}{n\lambda},$$

where $r = \|\widehat{\boldsymbol{z}}\|_0$ with $\widehat{\boldsymbol{z}}$ denoting the continuous relaxation solution, and c_4 is a "sufficient large constant." Clearly, if c_4 is very large or $\|\widehat{\boldsymbol{z}}\|_0$ is close to p, then our bound is much tighter than [45].

Next, let $\boldsymbol{\beta}^R$ be the estimator from Algorithm 3, which can be computed according to (7) by letting S be the output from Algorithm 3. Then we can show that the distance between $\boldsymbol{\beta}^R$ and $\boldsymbol{\beta}^*$ (i.e., $\|\boldsymbol{\beta}^R - \boldsymbol{\beta}^*\|_2$) can be also quite small, where $\boldsymbol{\beta}^*$ is an optimal solution to (F0).

THEOREM 6. *Let $\boldsymbol{\beta}^*$ be an optimal solution to* (F0) *with a set of selected features S^*, and let $\boldsymbol{\beta}^R$ be the estimator from Algorithm* 3 *with a set of selected features S^R. Given $\alpha \in (0,1)$, if $\lambda \geq \frac{\log(2n/\alpha)\sqrt{\theta_1}}{3n\epsilon} + \frac{\sqrt{2\theta_k \log(2n/\alpha)}}{2n\epsilon}$, then with probability at least $1-\alpha$, we have*

$$\|\boldsymbol{\beta}^R - \boldsymbol{\beta}^*\|_2 \leq \frac{\sqrt{4n\theta_{|S^R \setminus S^*|} v^*}}{n\lambda + \sigma_{\min}(\boldsymbol{X}_{S^R \cup S^*}^\top \boldsymbol{X}_{S^R \cup S^*})} + \sqrt{\frac{n\epsilon v^*}{n\lambda + \sigma_{\min}(\boldsymbol{X}_{S^R \cup S^*}^\top \boldsymbol{X}_{S^R \cup S^*})}}.$$

Proof. The proof is almost identical to that of Theorem 4 and thus is omitted here. ∎

Finally, we remark that we can integrate the greedy approach with the randomized algorithm by applying the greedy approach based on the support of the continuous relaxation solution of (F0-MISOC). That is, given that $\widehat{\boldsymbol{z}}$ is the optimal solution to (8b) and $\delta > 0$ is a positive constant, we first let a set $\mathcal{C} := \{i \in [p] : \widehat{z}_i \geq \delta\}$ and apply the greedy approach (Algorithm 2) to set $\mathcal{C}$ rather than $[p]$, which could save a significant amount of computational time, particularly when the continuous relaxation solution $\widehat{\boldsymbol{z}}$ is very sparse. The detailed description can be found in Algorithm 4.

Algorithm 4. Proposed restricted greedy approach.

1: Let $\widehat{\boldsymbol{z}}$ be the optimal solution to (8b)
2: Initialize $\delta > 0$ (e.g., $\delta = 0.01$), $\mathcal{C} := \{i \in [p] : \widehat{z}_i \geq \delta\}$
3: Let $S = \emptyset$ and $\boldsymbol{A}_S = n\lambda \boldsymbol{I}_n$
4: **for** $i = 1, \ldots, k$ **do**
5: Let $j^* \in \arg\min_{j \in \mathcal{C} \setminus S} \left\{ -\frac{\lambda(\boldsymbol{y}^\top \boldsymbol{A}_S^{-1} \boldsymbol{x}_j)^2}{1 + \boldsymbol{x}_j^\top \boldsymbol{A}_S^{-1} \boldsymbol{x}_j} \right\}$
6: Let $S = S \cup \{j^*\}$ and $\boldsymbol{A}_S = \boldsymbol{A}_S + \boldsymbol{x}_{j^*}\boldsymbol{x}_{j^*}^\top$, $\boldsymbol{A}_S^{-1} = \boldsymbol{A}_S^{-1} - \frac{\boldsymbol{A}_S^{-1}\boldsymbol{x}_{j^*}\boldsymbol{x}_{j^*}^\top \boldsymbol{A}_S^{-1}}{1 + \boldsymbol{x}_{j^*}^\top \boldsymbol{A}_S^{-1} \boldsymbol{x}_{j^*}}$
7: **end for**
8: Output $v^{RG} \leftarrow \lambda \boldsymbol{y}^\top \boldsymbol{A}_S^{-1} \boldsymbol{y}$.

5. Selection of tuning parameter and generalization to sparse matrix estimation. In this section, we will discuss how to select the tuning parameter λ using generalized cross validation and show that our proposed approaches can be extended to sparse matrix estimation.

5.1. Selection of tuning parameter by generalized cross validation. For a given k, we can adopt the commonly used GCV [25, 54] to choose the best λ in the ridge regression. Specifically, the GCV can be defined as

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \widehat{y}_i}{1-(\boldsymbol{H}_S)_{ii}}\right)^2, \tag{17}$$

where $\boldsymbol{H}_S = \boldsymbol{X}_S(\boldsymbol{X}_S^\top \boldsymbol{X}_S + n\lambda I)^{-1}\boldsymbol{X}_S^\top$ denotes the hat matrix of the ridge regression, and $\widehat{\boldsymbol{y}} = \boldsymbol{H}_S\boldsymbol{y}$ is the vector of the fitted responses. With a sequence of λ values in $\{\lambda_1, \ldots, \lambda_m\}$, we can choose the one having the smallest $GCV(\lambda)$ value. It is worth mentioning that the original GCV [25, 54] was proposed for the ridge regression without a sparsity requirement, and thus GCV used in this paper is a heuristic procedure for the sparse ridge regression problem.

5.2. Generalization to sparse matrix estimation. In this subsection, we consider a sparse matrix estimation proposed by [9]. In that problem, the authors were trying to estimate the inverse of covariance matrix $\widehat{\boldsymbol{\Sigma}} \in \mathbb{R}^{t\times t}$ and choose the sparest estimator. In their model, they optimized the L_1 norm of the estimator given that the estimation error is within a constant. Similar to (F0), we can instead directly optimize the estimation error given that only k sparse elements can be chosen, which can be formulated as

$$v^* = \min_{\boldsymbol{\Omega}}\left\{\|\boldsymbol{I}_t - \widehat{\boldsymbol{\Sigma}}\Omega\|_F^2 + \lambda\|\boldsymbol{\Omega}\|_F^2 : \|\boldsymbol{\Omega}\|_0 \le k\right\}. \tag{18}$$

To view this model as a special case of (F0), we rewrite matrix Ω as a vector $\boldsymbol{\beta} \in \mathbb{R}^{t^2\times 1}$ and $\widehat{\boldsymbol{\Sigma}}$ as $\boldsymbol{X} \in \mathbb{R}^{t\times t^2}$, where

$$\begin{aligned}
&\beta(j + t(i-1)) = \Omega(i,j) \quad \forall i, j \in [t],\\
&X(s,r) = \begin{cases} \Sigma(s, r - t(s-1)) & \text{if } 1 \le r - t(s-1) \le t \\ 0 & \text{otherwise} \end{cases} \qquad \forall s \in [t], r \in [t^2],\\
&y_{j+t(i-1)} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \forall i, j \in [t].
\end{aligned}$$

Thus, (18) reduces to (F0). Then the results for sparse ridge regression in the previous sections hold for (18).

6. Experimental verification. In this section, we illustrate the different algorithms proposed in this paper and how to choose the tuning parameters. Particularly, subsection 6.1 focuses on a comparison of the branch and cut algorithm in [5], MISOC formulation (F0-MISOC), heuristic Algorithm 1 in [1], greedy Algorithm 2, randomized Algorithm 3, and restricted greedy Algorithm 4. Subsection 6.2 focuses on MISOC formulation (F0-MISOC) and greedy Algorithm 2 to illustrate that although greedy Algorithm 2 is fast and close to optimality, it might be able to provide near-optimal solutions. Subsection 6.3 demonstrates how to choose the tuning parameter λ using GCV via a real-world application. The code for the greedy algorithm can be found at https://github.com/xwj06/Sparse_Ridge_Regression.git.

6.1. Comparison of the branch and cut algorithm in [5], mixed integer second order conic formulation (F0-MISOC), heuristic Algorithm 1 in [1], greedy Algorithm 2, randomized Algorithm 3, and restricted greedy Algorithm 4 via large-scale synthetic datasets. In this subsection, we conduct experimental studies to evaluate the performance of the proposed methods in comparison with several existing ones for solving sparse ridge regression problems. The data are generated from the linear model

$$y = \boldsymbol{x}^\top \boldsymbol{\beta}^0 + \tilde{\epsilon},$$

where $\tilde{\epsilon} \sim N(0, \sigma^2)$. The independent and identically distributed (i.i.d.) samples of $\boldsymbol{x}$ are generated from a multivariate normal distribution with

$$\boldsymbol{x}_i \sim N(0, \boldsymbol{\Sigma}),\ i = 1, \ldots, n,$$

where $\boldsymbol{\Sigma}$ is the covariance matrix with $\sigma_{ij} = \rho^{|i-j|}$ for each $i, j \in [p]$, and $\rho = 0.5$. The first k entries of $\boldsymbol{\beta}^0 = (\beta_1^0, \ldots, \beta_p^0)^\top$ are nonzero, and their values are drawn randomly from the uniform distribution Unif$(-3, 3)$. To control the signal-to-noise ratio (SNR), we choose the value of σ^2 such that SNR $= \text{var}(\boldsymbol{x}^\top \boldsymbol{\beta}^0)/\text{var}(\tilde{\epsilon}) = 9$. By generating an i.i.d. sample of noise $\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_n$ with $\tilde{\epsilon}_i \sim N(0, \sigma^2)$ for each $i \in [n]$, we simulate the response values, i.e., $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}^0 + \tilde{\epsilon}_i$ for each $i \in [n]$.

Recall that the goal is to find a best k-sparse estimator for a given k. The compared performances of the methods are evaluated by the selection accuracy and computational time. Here we consider different combinations of k, n, p for generating the simulation data, where $p \in \{1000, 5000\}$, $n \in \{500, 1000, 5000\}$, and $k \in \{10, 20, 30\}$. Each simulation setting is repeated 10 times, i.e., for each tuple (k, n, p), we generate 10 repetitions.[1] For simplicity, for all the testing instances, we set the tuning parameter to $\lambda = 0.08$.

The compared methods include the branch and cut algorithm proposed by [5] based on (F0-MIC),[2] directly solving (F0-MISOC), the heuristic Algorithm 1 in [1], the proposed greedy Algorithm 2, the proposed randomized Algorithm 3, and the proposed restricted greedy Algorithm 4. Note that the heuristic Algorithm 1 in [1] is similar to the least absolute shrinkage and selection (LASSO) method in the use of the L_1 norm to achieve sparsity. The commercial solver Gurobi 7.5 with its default setting is used to solve (F0-MISOC) and its continuous relaxation. We set the time limit to be one hour (3600 seconds). Due to out-of-memory and out-of-time-limit issues, in the case of $p = 5000$, we only compute two of the most effective algorithms: the proposed greedy Algorithm 2 and the proposed restricted greedy Algorithm 4. The comparison results are listed in Tables 1–3, where the *Avg. obj. value*, *Avg. gap*, *Avg. comp. time*, and *Avg. false alarm rate* denote the average objective function value, average optimality gap (of exact methods) from Guorbi, average computational time (in seconds), and average percentage of falsely detected features, respectively. For most of the test instances, the optimal value v^* can be very difficult to obtain. Therefore, we only compare the objective function values of different algorithms, where the smaller objective function value implies that the output of the algorithm is more accurate. All computations were executed on a MacBook Pro with a 2.80 GHz processor and 16GB RAM.

[1]We restrict the simulation to 10 replications because certain existing methods are very slow in computation.

[2]Please note that [5] proposed a sophisticated warm-start procedure. However, for the sake of fair comparison, we directly implemented the branch and cut algorithm without any warm-start procedure.

TABLE 1
Comparison of the branch and cut algorithm in [5] *and directly solving* (F0-MISOC) *with* $p = 1000$.

p	k	n	Branch and cut algorithm				Solving (F0-MISOC)			
			Avg. obj. value	Avg. comp. time(s)	Avg. gap	Avg. false alarm rate	Avg. obj. value	Avg. comp. time(s)	Avg. gap	Avg. false alarm rate
1000	10	500	9.71	3438.51	47.2%	26.0%	6.83	3505.82	7.1%	5.0%
		1000	7.11	2451.47	10.4%	5.0%	7.27	3562.61	9.7%	7.0%
		5000	NA*	NA	NA	NA	6.67	387.44	0.0%	0.0%
	20	500	23.02	3600.00	141.5%	45.0%	11.98	3600.00	21.4%	20.0%
		1000	31.52	3600.00	131.2%	50.5%	11.55	3600.00	11.7%	18.0%
		5000	NA	NA	NA	NA	11.30	2434.64	0.3%	0.5%
	30	500	39.62	3600.00	189.3%	51.3%	20.42	3600.00	31.4%	27.0%
		1000	50.63	3600.00	175.9%	55.0%	19.16	3600.00	18.1%	22.3%
		5000	NA	NA	NA	NA	17.79	3600.00	1.3%	5.0%

*NA represents out-of-memory instances.

Table 1 reports the comparison results between directly solving (F0-MISOC) and the branch and cut algorithm based on (F0-MIC). It is seen that directly solving (F0-MISOC) outperforms the branch and cut algorithm for most of the instances, particularly when k becomes large. This is because (i) we proved in Theorem 2 that continuous relaxations of (F0-MIC) and (F0-MISOC) are equivalent, and thus directly solving (F0-MISOC) should perform at least as well as the branch and cut algorithm; and (ii) the branch and cut algorithm needs to compute the gradient of the objective function in (F0-MIC), which involves a very time-consuming $n \times n$ matrix inversion. However, both approaches reach the time limit for most of the cases, and the average false alarm rates are higher than those for the approximation algorithms in Table 2. Therefore, for large-scale instances, these approaches might not be very desirable.

TABLE 2
Comparison of heuristic Algorithm 1 *in* [1], *greedy Algorithm* 2, *randomized Algorithm* 3, *and restricted greedy Algorithm* 4 *with* $p = 1000$.

p	k	n	Heuristic Algorithm 1 in [1]			Proposed greedy Algorithm 2		
			Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate	Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate
1000	10	500	9.59	579.36	3.0%	6.60	0.47	0.0%
		1000	7.88	45.78	0.0%	6.54	0.59	0.0%
		5000	7.24	737.06	0.0%	6.67	1.41	0.0%
	20	500	15.87	589.66	14.5%	10.86	0.79	9.0%
		1000	13.42	47.92	11.5%	10.91	2.02	4.0%
		5000	12.66	738.55	4.5%	11.30	2.37	0.0%
	30	500	28.87	583.98	17.0%	16.88	1.13	10.7%
		1000	23.53	43.92	12.7%	17.19	1.43	6.7%
		5000	19.74	678.10	6.0%	17.74	3.28	2.0%

p	k	n	Proposed randomized Algorithm 3			Proposed restricted greedy Algorithm 4		
			Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate	Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate
1000	10	500	7.79	4.06	14.0%	6.60	3.84	0.0%
		1000	6.86	11.21	6.0%	6.54	10.58	0.0%
		5000	6.67	181.77	0.0%	6.67	186.81	0.0%
	20	500	12.88	4.01	23.5%	10.86	3.80	9.0%
		1000	11.68	10.84	18.0%	10.91	13.81	4.0%
		5000	11.40	199.31	6.5%	11.30	202.66	0.0%
	30	500	20.89	4.21	26.3%	16.89	4.06	11.0%
		1000	19.89	10.58	24.0%	17.19	11.94	6.7%
		5000	18.11	167.95	10.0%	17.74	170.14	2.0%

TABLE 3
Comparison of greedy Algorithm 2 and restricted greedy Algorithm 4 with $p = 5000$.

p	k	n	Proposed greedy Algorithm 2			Proposed restricted greedy Algorithm 4		
			Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate	Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate
5000	10	500	4.57	2.31	0.0%	4.57	15.81	0.0%
		1000	4.59	3.13	0.0%	4.59	39.06	0.0%
		5000	4.68	9.04	0.0%	4.68	1451.78	0.0%
	20	500	12.86	4.31	8.0%	12.86	15.69	8.0%
		1000	13.35	5.41	2.5%	13.35	38.14	2.5%
		5000	13.27	14.58	0.0%	13.27	1426.93	0.0%
	30	500	14.02	5.98	20.7%	14.02	16.24	20.7%
		1000	14.97	8.21	12.7%	14.97	39.41	12.7%
		5000	15.60	20.52	3.3%	15.60	1503.48	3.3%

From Tables 1 and 2, we see that the proposed greedy Algorithm 2 and restricted greedy Algorithm 4 apparently perform best among all compared methods. We see that for the instances with $k = 10$, the heuristic Algorithm 1, greedy Algorithm 2, and restricted greedy Algorithm 4 find almost all the features, while the randomized Algorithm 3 performs slightly worse. When the number of active features k grows, all the compared methods have relatively larger false alarm rates. Their ability to identify the right features improves as the sample size n increases, i.e., they provide more information. The heuristic Algorithm 1 in [1] is less accurate and takes a much longer. Thus, it might not be a good option for large-scale instances either. In contrast, we note that the greedy Algorithm 2 is much more accurate. It runs very fast with the computation time, which is proportional to n, p, k. But the randomized Algorithm 3, which depends on the solution time of solving the continuous relaxation of (F0-MISOC), is quite insensitive to k in terms of computation time. Therefore, by integrating these two algorithms, we see that the restricted greedy Algorithm 4 can be advantageous for large k, providing accurate estimation with fast computation. For the numerical study with $p = 5000$ below, we choose these two algorithms for comparison because they are the most efficient.

In Table 3, we observe that the greedy Algorithm 2 and the restricted greedy Algorithm 4 have exactly the same false alarm rates. But the greedy Algorithm 2 is much faster than the restricted greedy Algorithm 4. This is mainly because it takes much longer to solve the continuous relaxation to the optimality, and for these instances, k is relatively small. In particular, for large-scale datasets (e.g., $n = p = 5000$), the computation time of the restricted greedy Algorithm 4 is much longer than in the case with $p = 1000$. But the greedy Algorithm 2 can still find very high quality solutions within 30 seconds of computation time. On the other hand, we note that the accuracy of both approaches grows when the sample size increases. Thus, we recommend finding a reasonable sample size so that the greedy methods can work efficiently and identify the features accurately.

We have numerically compared our implementation with the state-of-art R package posted by [28]. Table 4 summarize the comparison in terms of computational time. It is seen that our implementation can outperform the one in [28]. The advantage appears to be more striking as n becomes larger. Thus, we envision that our implementation for the greedy approach (or forward selection) is efficient and can be interesting to the reader.

TABLE 4
A comparison with the forward selection algorithm proposed in [28]. *Note that the solver in* [28] *only works for sparse regression. Thus, we only compare the computational times.*

p	k	n	Greedy Algorithm 2 Time (s)	Forward selection in [28] Time (s)
1000	10	500	0.47	0.79
		1000	0.59	1.10
		5000	1.41	11.41
	20	500	0.79	1.33
		1000	1.01	2.86
		5000	2.37	26.49
	30	500	1.13	2.17
		1000	1.43	4.01
		5000	3.28	38.98

6.2. Further investigation of mixed integer second order conic formulation (F0-MISOC) and greedy Algorithm 2 with varying signal-to-noise ratio and tuning parameter $\boldsymbol{\lambda}$ via medium-size synthetic datasets. Following the same data-generating procedure in the previous subsection, we conduct a thorough comparison of MISOC formulation (F0-MISOC) and greedy Algorithm 2. In particular, we generate 16 instances with $n = 100, p = 40, k \in \{5, 10, 15, 20\}, \text{SNR} \in \{0.5, 1, 2, 4\}$, and to illustrate the effects of tuning parameter λ, we let it vary from the range $\{0.01, 0.1, 1, 10\}$. Similarly, each simulation setting is repeated 10 times, and the average results are reported in Tables 5 and 6.

In Tables 5 and 6, it is seen that for these instances, formulation (F0-MISOC) can be solved to optimality within 2 minutes, while greedy Algorithm 2 can find the very near-optimal solutions within 0.1 second. We also see that in terms of average objective value and average false alarm rate, greedy Algorithm 2 in this case is slightly worse than formulation (F0-MISOC), since the latter is able to provide exact solutions. Thus, if the instances are not large, we suggest solving exact formulation (F0-MISOC), which indeed provides the best performance. As for the SNR, we see that the false alarm rates of both approaches decrease as SNR increases, which is consistent with the intuition since higher SNR implies stronger signal and thus a more accurate prediction. In terms of tuning parameter, we see that the computational time of formulation (F0-MISOC) changes significantly as λ increases. On the other hand, if the tuning parameter λ is too big, the false alarm rate will increase significantly. Thus, a proper choice of the tuning parameter λ will be critical for formulation (F0-MISOC). In the next subsection, we will use GCV to choose a proper tuning parameter λ for a real-world case.

6.3. A real-world case study using the dataset in [56]. In this subsection, we conduct a case study using the dataset in [56], which attempted to map the loci on the third chromosome of Drosophila melanogaster that will influence an index of wing shape. The dataset has $n = 701$ recombinant inbred lines (i.e., observations) and genotypes of 48 markers, where 11 markers are highly correlated with others and are thus removed. The selected 37 markers and their corresponding indices can be found at https://www4.stat.ncsu.edu/~boos/var.select/wing.shape.html. Similar to [56], we also consider the interactions of the remaining 37 markers, and thus there are $p = 37 + \binom{37}{2} = 703$ features in total. We use GCV to choose a proper tuning parameter λ from the list $\{10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1, 0.2, 0.5, 1\}$ for each $k \in \{10, 20, 40\}$. We use greedy Algorithm 2 to solve all the instances, and the total running time is within 1 minute. Table 7 shows the feature selection results.

Table 5

Comparison of MISOC formulation (F0-MISOC) *and greedy Algorithm* 2 *with* $n = 100, p = 40, k \in \{5, 10\}$.

k	SNR	λ	MISOC formulation (F0-MISOC) Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate	Proposed greedy Algorithm 2 Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate
5	0.5	0.01	29.75	81.72	44.0%	29.88	0.013	46.0%
		0.1	31.65	4.44	38.0%	31.72	0.014	38.0%
		1	39.98	0.28	36.0%	39.99	0.013	36.0%
		10	49.13	0.29	44.0%	49.13	0.010	44.0%
	1	0.01	15.07	54.24	38.0%	15.10	0.011	38.0%
		0.1	16.56	2.11	38.0%	16.57	0.012	38.0%
		1	23.90	0.29	44.0%	23.90	0.014	44.0%
		10	32.73	0.28	52.0%	32.73	0.014	52.0%
	2	0.01	7.11	64.68	26.0%	7.11	0.014	26.0%
		0.1	8.40	2.80	26.0%	8.40	0.014	26.0%
		1	14.71	0.29	32.0%	14.71	0.011	32.0%
		10	21.90	0.28	44.0%	21.90	0.010	44.0%
	4	0.01	3.86	20.76	20.0%	3.86	0.010	20.0%
		0.1	5.13	1.17	18.0%	5.14	0.013	20.0%
		1	11.17	0.28	34.0%	11.17	0.014	34.0%
		10	18.15	0.28	40.0%	18.15	0.014	40.0%
10	0.5	0.01	49.00	43.43	49.0%	49.58	0.025	47.0%
		0.1	53.08	3.00	48.0%	53.30	0.021	46.0%
		1	68.60	0.25	50.0%	68.61	0.019	51.0%
		10	85.37	0.24	55.0%	85.37	0.019	55.0%
	1	0.01	24.62	12.44	46.0%	24.75	0.023	40.0%
		0.1	27.81	0.68	40.0%	27.89	0.024	39.0%
		1	41.93	0.23	44.0%	41.94	0.025	44.0%
		10	58.06	0.27	50.0%	58.06	0.022	51.0%
	2	0.01	13.02	7.70	29.0%	13.08	0.022	29.0%
		0.1	15.84	0.51	28.0%	15.91	0.020	30.0%
		1	28.53	0.21	34.0%	28.53	0.025	34.0%
		10	42.95	0.25	42.0%	42.95	0.024	42.0%
	4	0.01	6.70	2.51	27.0%	6.75	0.024	30.0%
		0.1	9.22	0.40	30.0%	9.23	0.020	30.0%
		1	20.67	0.21	34.0%	20.67	0.020	34.0%
		10	34.09	0.25	49.0%	34.09	0.021	49.0%

In Table 7, we see that using GCV from section 5.1, the best tuning parameter λ tends to be small, particularly when k increases. In general, a proper k can be determined by biologists or engineers, and as long as k is not very large, we are able to deliver near-optimal feature selections efficiently. In fact, we see that if $k = 40$, then we can identify all the necessary markers listed in the Table 3 of [35] except x23. This demonstrates that our proposed method is indeed effective for feature selection problems.

7. Conclusion. This paper studies sparse ridge regression with the use of an exact L_0 norm for the sparsity. It is known that imposing the L_0 norm for the sparsity in regression can often become an NP-hard problem in variable selection and estimation. We present a mixed integer second order conic (MISOC) formulation, which is big-M free and is based on perspective formulation. We prove that the continuous relaxation of this MISOC reformulation is equivalent to the convex integer program (CIP) formulation studied in the literature, and it can be stronger than the straightforward big-M formulation. Based on these two formulations, we propose two scalable algorithms, the greedy and randomized algorithms, for solving sparse ridge regression. Under mild conditions, both algorithms find near-optimal solutions

Table 6
Comparison of MISOC formulation (F0-MISOC) *and greedy Algorithm* 2 *with* $n = 100, p = 40, k \in \{15, 20\}$.

k	SNR	λ	MISOC formulation (F0-MISOC)			Proposed greedy Algorithm 2		
			Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate	Avg. obj. value	Avg. comp. time(s)	Avg. false alarm rate
15	0.5	0.01	66.81	110.75	46.7%	68.61	0.035	48.7%
		0.1	74.92	4.54	46.7%	75.61	0.035	46.0%
		1	101.07	0.27	42.0%	101.09	0.029	42.0%
		10	126.31	0.27	46.7%	126.31	0.028	46.7%
	1	0.01	34.42	162.55	37.3%	35.15	0.033	38.0%
		0.1	39.85	5.54	36.7%	40.13	0.035	36.7%
		1	60.56	0.33	41.3%	60.64	0.035	42.0%
		10	83.68	0.29	49.3%	83.68	0.032	49.3%
	2	0.01	18.85	34.03	25.3%	19.09	0.025	27.3%
		0.1	23.72	1.66	25.3%	23.88	0.028	26.0%
		1	42.64	0.27	32.7%	42.69	0.036	34.0%
		10	61.73	0.25	41.3%	61.73	0.040	40.7%
	4	0.01	9.50	11.84	22.7%	9.61	0.026	23.3%
		0.1	14.01	0.53	24.0%	14.10	0.028	22.7%
		1	32.67	0.26	30.0%	32.67	0.033	30.7%
		10	52.66	0.28	39.3%	52.66	0.037	39.3%
20	0.5	0.01	66.04	81.72	43.0%	66.60	0.040	43.5%
		0.1	74.61	4.44	41.5%	74.72	0.037	41.0%
		1	103.88	0.28	38.0%	103.90	0.048	38.5%
		10	136.87	0.29	41.5%	136.87	0.044	41.5%
	1	0.01	28.59	54.24	34.5%	28.88	0.035	34.5%
		0.1	33.63	2.11	33.5%	33.80	0.037	32.5%
		1	52.82	0.29	34.0%	52.84	0.045	35.5%
		10	76.04	0.28	38.5%	76.04	0.046	38.5%
	2	0.01	15.95	64.68	29.0%	16.22	0.034	27.5%
		0.1	20.16	2.80	27.5%	20.27	0.039	28.0%
		1	36.68	0.29	27.0%	36.68	0.051	27.5%
		10	58.11	0.28	35.0%	58.11	0.052	35.5%
	4	0.01	8.24	20.76	27.0%	8.32	0.058	24.5%
		0.1	11.78	1.17	24.0%	11.79	0.052	23.5%
		1	27.25	0.28	32.0%	27.25	0.055	32.0%
		10	48.07	0.28	38.0%	48.07	0.061	38.0%

Table 7
Feature selection results using the dataset in [56] *and GCV in subsection* 5.1. *Here,* x*i denotes ith marker and* x*i*.x*j represents the interaction of markers i and j.*

λ	k	Selected features
10^{-4}	10	x1, x18, x48, x1.x18, x1.x48, x5.x15, x11.x42, x16.x33, x17.x48, x42.x45
10^{-5}	20	x1, x18, x37, x48, x1.x4, x1.x18, x1.x48, x5.x15, x11.x42, x14.x37, x16.x33, x16.x45, x17.x27, x17.x48, x34.x40, x34.x48, x36.x40, x36.x48, x40.x45, x42.x45
10^{-5}	40	x1, x10, x18, x37, x40, x48, x1.x4, x1.x10, x1.x18, x1.x48, x3.x44, x5.x15, x5.x48, x7.x10, x9.x10, x9.x13, x9.x18, x10.x13, x10.x18, x10.x30, x11.x40, x11.x42, x12.x36, x13.x33, x14.x37, x16.x33, x16.x45, x17.x27, x17.x48, x18.x36, x34.x40, x34.x45, x34.x48, x35.x45, x35.x48, x36.x40, x36.x48, x40.x45, x42.x45, x46.x48

with performance guarantees. Our numerical study demonstrates that the proposed algorithms can indeed solve large-scale instances efficiently. In general, we recommend solving the MISOC formulation first, which might be efficient; otherwise, we suggest using the scalable algorithms studied in this paper.

Acknowledgments. We thank the two anonymous referees and the associate editor for their valuable comments which improved this paper.

REFERENCES

[1] S. Ahmed, J. Luedtke, Y. Song, and W. Xie, *Nonanticipative duality, relaxations, and formulations for chance-constrained stochastic programs*, Math. Program., 162 (2017), pp. 51–81.

[2] A. Atamturk and A. Gomez, *Rank-One Convexification for Sparse Regression*, preprint, https://arxiv.org/abs/1901.10334, 2019.

[3] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MOS-SIAM Ser. Optim. 2, SIAM, 2001, https://doi.org/10.1137/1.9780898718829.

[4] D. Bertsimas, A. King, and R. Mazumder, *Best subset selection via a modern optimization lens*, Ann. Statist., 44 (2016), pp. 813–852.

[5] D. Bertsimas and B. Van Parys, *Sparse High-Dimensional Regression: Exact Scalable Algorithms and Phase Transitions*, preprint, https://arxiv.org/abs/1709.10029, 2017.

[6] D. Bienstock, *Computational study of a family of mixed-integer quadratic programming problems*, Math. Program., 74 (1996), pp. 121–140.

[7] J. Blanchet, Y. Kang, and K. Murthy, *Robust Wasserstein profile inference and applications to machine learning*, J. Appl. Probab., 56 (2019), pp. 830–857.

[8] P. Bühlmann and S. Van De Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*, Springer Ser. Statist., Springer-Verlag, 2011.

[9] T. Cai, W. Liu, and X. Luo, *A constrained l_1 minimization approach to sparse precision matrix estimation*, J. Amer. Statist. Assoc., 106 (2011), pp. 594–607.

[10] E. Candès and T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n*, Ann. Statist., 35 (2007), pp. 2313–2351.

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

[12] S. Ceria and J. Soares, *Convex programming for disjunctive convex optimization*, Math. Program., 86 (1999), pp. 595–614.

[13] H. Chernoff, *A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations*, Ann. Math. Statist., 23 (1952), pp. 493–507.

[14] A. Das and D. Kempe, *Algorithms for subset selection in linear regression*, in Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, ACM, 2008, pp. 45–54.

[15] A. Das and D. Kempe, *Submodular Meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection*, preprint, https://arxiv.org/abs/1102.3975, 2011.

[16] C. De Mol, E. De Vito, and L. Rosasco, *Elastic-net regularization in learning theory*, J. Complex., 25 (2009), pp. 201–230.

[17] H. Dong, *On the Exact Recovery of Sparse Signals via Conic Relaxations*, preprint, https://arxiv.org/abs/1603.04572, 2016.

[18] H. Dong, K. Chen, and J. Linderoth, *Regularization vs. Relaxation: A Conic Optimization Perspective of Statistical Variable Selection*, preprint, https://arxiv.org/abs/1510.06083, 2015.

[19] N. R. Draper and R. C. Van Nostrand, *Ridge regression and James-Stein estimation: Review and comments*, Technometrics, 21 (1979), pp. 451–466.

[20] J. Fan and J. Lv, *Nonconcave penalized likelihood with np-dimensionality*, IEEE Trans. Inform. Theory, 57 (2011), pp. 5467–5484.

[21] A. Frangioni and C. Gentile, *Perspective cuts for a class of convex 0–1 mixed integer programs*, Math. Program., 106 (2006), pp. 225–236.

[22] J. N. Franklin, *Matrix Theory*, Courier Corporation, 1968.

[23] J. H. Friedman, *Fast sparse regression and classification*, Internat. J. Forecasting, 28 (2012), pp. 722–738.

[24] R. Gao, X. Chen, and A. J. Kleywegt, *Wasserstein Distributional Robustness and Regularization in Statistical Learning*, preprint, https://arxiv.org/abs/1712.06050v1, 2017.

[25] D. A. Girard, *Asymptotic optimality of the fast randomized versions of GVC and C_L in ridge regression and regularization*, Ann. Statist., 19 (1991), pp. 1950–1963.

[26] O. Günlük and J. Linderoth, *Perspective reformulation and applications*, in Mixed Integer Nonlinear Programming, IMA Vol. Math. Appl. 154, Springer, 2012, pp. 61–89.

[27] T. Hastie, R. Tibshirani, and J. Friedman, *Unsupervised learning*, in The Elements of

Statistical Learning, Springer, 2009, pp. 485–585.
[28] T. Hastie, R. Tibshirani, and R. J. Tibshirani, *Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso*, preprint, https://arxiv.org/abs/1707.08692, 2017.
[29] H. Hazimeh and R. Mazumder, *Fast Best Subset Selection: Coordinate Descent and Local Combinatorial Optimization Algorithms*, preprint, https://arxiv.org/abs/1803.01454, 2018.
[30] J. Huang, J. L. Horowitz, and S. Ma, *Asymptotic properties of bridge estimators in sparse high-dimensional regression models*, Ann. Statist., 36 (2008), pp. 587–613.
[31] R. Khanna, E. Elenberg, A. G. Dimakis, S. Negahban, and J. Ghosh, *Scalable Greedy Feature Selection via Weak Submodularity*, preprint, https://arxiv.org/abs/1703.02723, 2017.
[32] M. Kowalski, *Sparse regression using mixed norms*, Appl. Comput. Harmon. Anal., 27 (2009), pp. 303–324.
[33] L. Kuo and B. Mallick, *Variable selection for regression models*, Sankhyā Ser. B, 60 (1998), pp. 65–81.
[34] J. Luedtke, *A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support*, Math. Program., 146 (2014), pp. 219–244.
[35] S. Mak and C. F. J. Wu, *cmenet: A new method for bi-level variable selection of conditional main effects*, J. Amer. Statist. Assoc., 114 (2019), pp. 844–856.
[36] D. W. Marquardt and R. D. Snee, *Ridge regression in practice*, Amer. Statist., 29 (1975), pp. 3–20.
[37] R. Mazumder and P. Radchenko, *The discrete Dantzig selector: Estimating sparse linear models via mixed integer linear optimization*, IEEE Trans. Inform. Theory, 63 (2017), pp. 3053–3075.
[38] R. Mazumder, P. Radchenko, and A. Dedieu, *Subset Selection with Shrinkage: Sparse Linear Modeling When the SNR Is Low*, preprint, https://arxiv.org/abs/1708.03288, 2017.
[39] A. Miller, *Subset Selection in Regression*, CRC Press, 2002.
[40] R. Miyashiro and Y. Takano, *Mixed integer second-order cone programming formulations for variable selection in linear regression*, Eur. J. Oper. Res., 247 (2015), pp. 721–731.
[41] R. Miyashiro and Y. Takano, *Subset selection by Mallows' C_p: A mixed integer programming approach*, Expert Syst. Appl., 42 (2015), pp. 325–331.
[42] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM J. Comput., 24 (1995), pp. 227–234, https://doi.org/10.1137/S0097539792240406.
[43] A. Nemirovski and A. Shapiro, *Convex approximations of chance constrained programs*, SIAM J. Optim., 17 (2006), pp. 969–996, https://doi.org/10.1137/050622328.
[44] K. Pavlikov, A. Veremyev, and E. L. Pasiliao, *Optimization of value-at-risk: Computational aspects of MIP formulations*, J. Oper. Res. Soc., 69 (2018), pp. 676-690.
[45] M. Pilanci, M. J. Wainwright, and L. El Ghaoui, *Sparse learning via Boolean relaxations*, Math. Program., 151 (2015), pp. 63–87.
[46] F. Qiu, S. Ahmed, S. S. Dey, and L. A. Wolsey, *Covering linear programming with violations*, INFORMS J. Comput., 26 (2014), pp. 531–546.
[47] G. Sagnol and R. Harman, *Computing exact D-optimal designs by mixed integer second-order cone programming*, Ann. Statist., 43 (2015), pp. 2198–2224.
[48] M. Seeger, C. Williams, and N. Lawrence, *Fast Forward Selection to Speed Up Sparse Gaussian Process Regression*, Technical report, University of Edinburgh, 2003.
[49] J. Sherman and W. J. Morrison, *Adjustment of an inverse matrix corresponding to a change in one element of a given matrix*, Ann. Math. Statist., 21 (1950), pp. 124–127.
[50] A. J. Smola and P. L. Bartlett, *Sparse greedy Gaussian process regression*, in Advances in Neural Information Processing Systems 13, MIT Press, 2001, pp. 619–625.
[51] Y. Song, J. R. Luedtke, and S. Küçükyavuz, *Chance-constrained binary packing problems*, INFORMS J. Comput., 26 (2014), pp. 735–747.
[52] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[53] J. A. Tropp, *User-friendly tail bounds for sums of random matrices*, Found. Comput. Math., 12 (2012), pp. 389–434.
[54] W. N. van Wieringen, *Lecture Notes on Ridge Regression*, preprint, https://arxiv.org/abs/1509.09169, 2015.
[55] H. D. Vinod, *A survey of ridge regression and related techniques for improvements over ordinary least squares*, Rev. Econom. Statist., 60 (1978), pp. 121–131.
[56] K. Weber, R. Eisman, L. Morey, A. Patty, J. Sparks, M. Tausek, and Z.-B. Zeng, *An analysis of polygenes affecting wing shape on chromosome 3 in Drosophila melanogaster*, Genetics, 153 (1999), pp. 773–786, https://www.genetics.org/content/153/2/773.

[57] H. Weyl, *The asymptotic distribution law of the eigenvalues of linear partial differential equations (with an application to the theory of cavity radiation)*, Math. Ann., 71 (1912), pp. 441–479.
[58] H. Xie and J. Huang, *SCAD-penalized regression in high-dimensional partially linear models*, Ann. Statist., 37 (2009), pp. 673–696.
[59] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Statist., 38 (2010), pp. 894–942.
[60] T. Zhang, *Adaptive forward-backward greedy algorithm for learning sparse representations*, IEEE Trans. Inform. Theory, 57 (2011), pp. 4689–4708.
[61] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, J. R. Stat. Soc. Ser. B Stat. Methodol., 67 (2005), pp. 301–320.