Scalable Algorithms for the Sparse Ridge Regression

Weijun Xie*1 and Xinwei Deng†2

1Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061
2Department of Statistics, Virginia Tech, Blacksburg, VA 24061

January 18, 2020

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 pave out a theoretical foundation to understand why existing approaches may not work well for this problem, in particular on large-scale datasets. By reformulating the problem as a chance-constrained program, we derive a novel mixed integer second order conic (MISOC) reformulation using perspective formulation and prove that its continuous relaxation is equivalent to that of the convex integer formulation proposed in recent work. Based upon these two formulations (i.e., the proposed MISOC formulation and an existing convex integer formulation), we develop two new 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 to integrate 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.

Approximation Algorithm, Chance Constraint, Conic Program, Mixed Integer, Ridge Regression

1 Introduction

Modern statistical analysis often encounters regressions with a large number of explanatory variables (also known as features). Hence, sparse regression and variable selection have been studied intensively in the past decades. Ridge regression, first proposed by [28], has several desirable properties including stability of the solution, reduced variance, and efficient computation. However, its solution is usually not sparse [24].

To overcome these issues, we consider the sparse ridge regression problem:

\[
\nu^* = \min_{\beta} \left\{ \frac{1}{n} \| y - X\beta \|_2^2 + \lambda \|\beta\|_2^2 : \|\beta\|_0 \leq k \right\},
\]

(F0)

*Email: wxie@vt.edu.
†Email: xdeng@vt.edu.
where $\mathbf{y} \in \mathbb{R}^n$ denotes the response vector, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$ represents the model matrix, $\mathbf{\beta} \in \mathbb{R}^p$ 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). Here, $||\mathbf{\beta}||_0$ is the $L_0$ norm, which counts the number of nonzero entries of vector $\mathbf{\beta}$. 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 $k \leq \min(n, p)$.

Sparse ridge regression (F0) can be reformulated as a chance constrained program (CCP) with finite support [1, 33]. That is, we consider $p$ scenarios with equal probability $\frac{1}{p}$, where the $i$th scenario set is $S_i := \{\mathbf{\beta} : \beta_i = 0\}$ for $i \in [p]$. The constraint $||\mathbf{\beta}||_0 \leq k$ means that at most $k$ out of the $p$ scenarios can be violated. Hence, we can reformulate (F0) as a CCP below

$$
\nu^* = \min_{\mathbf{\beta}} \left\{ \frac{1}{n} ||\mathbf{y} - \mathbf{X}\mathbf{\beta}||_2^2 + \lambda||\mathbf{\beta}||_2^2 : \frac{1}{p} \sum_{i \in [p]} \mathbb{I}(||\beta_i|| \leq 0) \geq 1 - \frac{k}{p} \right\},
$$

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, 33, 42]) 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 propose two novel 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 [16, 35, 52]. It has been shown from existing literature [16, 35, 52] that the additional ridge penalty $\lambda||\mathbf{\beta}||_2^2$ in (F0) has several desirable advantages including stable solution, estimator variance reduction, and efficient computation. Some recent progress in [6, 21] 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 $(\mathbf{X}, \mathbf{y})$ are insufficient or are subject to some noises. 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 [49], the Bridge estimator using the $L_2$ penalty [1], the minimax concave penalty [56], among many others. Several excellent and comprehensive reviews of sparse regression can be found in [7], [25], and [17]. In particular, it is worth mentioning that in [58], the authors proposed a well-known “elastic net” approach, which integrates the ridge penalty (i.e., squared $L_2$ penalty) and $L_1$ 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 exactly $k$-sparse estimator. On the contrary, instead, we introduce a constraint $||\mathbf{\beta}||_0 \leq k$ in (F0), which strictly enforces the sparsity on $\mathbf{\beta}$, 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., [41]), so is the sparse ridge regression (F0). Various effective approximation algorithms or heuristics have been introduced to solve sparse regression [14, 15, 20, 31, 32, 38]. For example, in [11], 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 [12], 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 prohibitively expensive when the number of features (i.e., $p$) becomes large [26]. Recently, [27] 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 solution quality. 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-sized 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, 36, 37, 39, 40]. 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 to develop 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, in particular for large-scale datasets. To overcome the aforementioned 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 a mixed integer convex (MIC) formulation in [5, 14]. 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 both in running time and accuracy of variable selection. The contributions are summarized below:

(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 [42], and the heuristic algorithm from [1], and shed some lights on why those methods may not be amenable to solve the sparse ridge regression (F0).

(ii) We establish a mixed integer second order conic (MISOC) formulation for (F0-CCP) from perspective formulation [23] and prove its continuous relaxation is equivalent to that of a mixed integer convex formulation in the work of [5, 14]. 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 to solve (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 develop 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.
The remainder of the paper is organized as follows. Section 2 investigates the applicability of several existing approaches of CCP to solve the sparse ridge regression (F0). Section 3 develops two big-M free mixed integer convex 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 to select a proper tuning parameter and 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 bold-letters (e.g., \( \mathbf{x}, \mathbf{A} \)) to denote vectors or matrices, and use corresponding non-bold letters to denote their components. Given a positive integer number \( t \), we let \( \{1, \ldots, t\} = \{t\} \) and let \( \mathbf{I}_t \) denote the \( t \times t \) identity matrix. Given a subset \( S \subseteq [p] \), we let \( \mathbf{X}_S \) be a submatrix of \( \mathbf{X} \) with columns from a subset \( S \). For a matrix \( \mathbf{Y} \), we let \( \sigma_{\min}(\mathbf{Y}) \) and \( \sigma_{\max}(\mathbf{Y}) \) denote its smallest and largest singular values, respectively. Given a vector \( \mathbf{x} \), we let \( \text{diag}(\mathbf{x}) \) be a diagonal matrix with diagonal entries from \( \mathbf{x} \). For a matrix \( \mathbf{W} \), we let \( \mathbf{W}_{\cdot i} \) denote its \( i \)th column. Given a set \( T \), we let \( \text{conv}(T) \) denote its convex hull. Given a finite set \( S \), we let \( |S| \) denote its cardinality.

\[2 \text{ Existing Solution Approaches}\]

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

\[2.1 \text{ Big-M Method}\]

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

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

(F0-big-M)

The above formulation (F0-big-M) has been used widely in recent works on sparse regression (see, e.g., [4, 5, 36, 37, 39, 40]). The advantage of (F0-big-M) is that it can be directly solved by the off-the-shelf solvers (e.g., CPLEX, Gurobi). However, one has to choose the vector \( \mathbf{M} = (M_1, \ldots, M_p)^\top \) properly.

There are many ways to choose the big-M coefficients (i.e., \( \{M_i\}_{i \in [p]} \)). One typical way [4] is that for each \( i \in [p] \), one can let \( M_i \) be equal to the largest value of \(|\beta_i| \) given that the optimal value \( v^* \) of (F0-CCP) is bounded by \( v^U \), i.e., let \( M_i \) be equal to the largest optimal value of the following two convex quadratic programs

\[
M_i = \max_{\beta} \left\{ \beta_i : \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_2^2 \leq v^U \right\} ,
\]
\[
\max_{\beta} \left\{ -\beta_i : \frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \leq v^U \right\}.
\]

(1)

To solve (1), it needs to compute an upper bound \(v^U\) of \(v^*\). Note that the objective value of any feasible solution to (F0-CCP) suffices. A naive upper bound is \(v^U = \|y\|_2^2\) since \(\beta = 0\) is feasible to (F0-CCP). On the other hand, to obtain vector \(M\), one has to solve two convex quadratic programs in (1) for each \(i \in [p]\), which can be very time-consuming, in particular when \(p\) is large.

Here we derive a slightly weaker but a closed-form vector \(M\). Note that all the convex programs in (1) share the same constraint \(\frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \leq v^U\). Thus, the key proof idea is to relax the constraint in (1) into a weaker one, which is more amenable for a closed-form upper bound of vector \(|\beta|\). This result is summarized below.

**Proposition 1** Suppose that \(v^U\) is an upper bound to \(v^*\). Then the vector \(M = (M_1, \ldots, M_p)^\top\) can be chosen as

\[
M_i = \max \left\{ \frac{1}{n\rho} x_i^\top y + \sqrt{\frac{1}{n^2\rho^2}\|X^\top y\|_2^2 + \frac{v^U}{\rho} - \frac{1}{n\rho}\|y\|_2^2}, \right. \\
\left. \frac{1}{n\rho} x_i^\top y - \sqrt{\frac{1}{n^2\rho^2}\|X^\top y\|_2^2 + \frac{v^U}{\rho} - \frac{1}{n\rho}\|y\|_2^2} \right\}
\]

for all \(i \in [p]\), where \(\rho = \sigma_{\min}(X^\top X) + \lambda\).

**Proof.** In (1), it is sufficient to find a component-wise upper bound of vector \(|\beta|\) of any feasible \(\beta\) satisfies

\[
\frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \leq v^U.
\]

First, the above constraint implies that

\[
\frac{1}{n}\|y\|_2^2 - \frac{2}{n} y^\top X\beta + \left(\frac{1}{n} \sigma_{\min}(X^\top X) + \lambda\right) \|\beta\|_2^2 \leq v^U,
\]

where the first inequality is due to the inequality \(X^\top X \succeq \sigma_{\min}(X^\top X)I_p\). Let \(\rho = \sigma_{\min}(X^\top X) + \lambda\). Therefore, (3) further implies

\[
\rho \sum_{i \in [p]} \left( \beta_i - \frac{1}{n\rho} x_i^\top y \right)^2 \leq \frac{1}{n^2\rho^2}\|X^\top y\|_2^2 + v^U - \frac{1}{n\rho}\|y\|_2^2.
\]

Thus, we have

\[
|\beta_i| \leq \max \left\{ \frac{1}{n\rho} x_i^\top y + \sqrt{\frac{1}{n^2\rho^2}\|X^\top y\|_2^2 + \frac{v^U}{\rho} - \frac{1}{n\rho}\|y\|_2^2}, \right. \\
\left. \frac{1}{n\rho} x_i^\top y - \sqrt{\frac{1}{n^2\rho^2}\|X^\top y\|_2^2 + \frac{v^U}{\rho} - \frac{1}{n\rho}\|y\|_2^2} \right\}
\]

for each \(i \in [p]\). \qed
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), for example, [1, 4, 43, 44, 48]. However, the tightening procedures tend to be time-consuming in particular for large-scale datasets. In Section 3, we will introduce two big-M free MIP formulations, whose continuous relaxation can be proven 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 [42] 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

\[ v_{\text{CVaR}} = \min_{\beta} \left\{ \frac{1}{n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 : \inf t \left[ -\frac{k}{p} t + \frac{1}{p} \sum_{i \in [p]} (|\beta_i| + t)_+ \right] \leq 0 \right\} , \]  

where \((w)_+ = \max(w, 0)\). It is seen that (4) is a convex optimization problem and provides a feasible solution to (F0-CCP). Thus \(v_{\text{CVaR}} \geq v^*\). However, we observe that the only feasible solution to (4) is \(\beta = 0\).

**Proposition 2** The only feasible solution to (4) is \(\beta = 0\), i.e., \(v_{\text{CVaR}} = \frac{1}{n} \|y\|_2^2\).

**Proof.** We first observe that the infimum in (4) 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 (4) must exist. Hence, in (4), we can replace the infimum by the existence operator:

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

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

\[ v_{\text{CVaR}} = \min_{\beta} \left\{ \frac{1}{n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 : \exists t \geq 0, -\frac{k}{p} t + \frac{1}{p} \sum_{i \in [p]} |\beta_i| \leq 0 \right\} , \]

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 more preferred.

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 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

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

(5)

Let $\beta^*_U$ be an optimal solution to (5) given an upper bound $v^U$ of $v^*$. The heuristic algorithm is to decrease the value of $v^U$ if $\|\beta^*_U\|_0 \leq k$, and increase it, otherwise. This bisection procedure will terminate after a finite number of iterations. The detailed procedure is described in Algorithm 1. Let $v^{\text{heur}}$ denote the output solution from Algorithm 1. Then clearly,

**Proposition 3** For Algorithm 1, the following two properties hold:

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

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

Proof.

(i) To prove the first part, Algorithm 1 will terminate if and only if $U - L \leq \hat{\delta}$. And 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

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

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

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

It is worth mentioning that for any given upper bound $v^U$, the formulation (5) is similar to the elastic net proposed by [58], which can be interpreted as a Lagrangian relaxation of (5). 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 $\beta$. We also note that Algorithm 1 might not be computationally efficient since it requires solving (5) multiple times but a warm start from the solution of the previous iteration might help speed up the algorithm. Although there have been much development of statistical properties of the elastic net method [13, 58], to the best of our knowledge, there is not a known performance guarantee (i.e., approximation ratio) for Algorithm 1.

## 3 Two Big-M Free Reformulations of Sparse Ridge Regression

Note that the Big-M formulation in (F0-big-M) is quite compact since it only involves $2p$ variables (i.e., $\beta, z$). 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.
Algorithm 1 Heuristic Algorithm in [1]

1: Let $L = 0$ and $U = \frac{\|y\|_2^2}{n}$ be known lower and upper bounds for (F0-CCP), let $\delta > 0$ be the stopping tolerance parameter.
2: while $U - L > \delta$ do
3: $q \leftarrow (L + U)/2$.
4: Let $\hat{\beta}$ be an optimal solution of (5) and set $\hat{z}_i = \mathbb{I}(\hat{\beta}_i = 0)$ for all $i \in [p]$.
5: if $\sum_{i \in [p]} \hat{z}_i \geq p - k$ then
6: $U \leftarrow q$.
7: else
8: $L \leftarrow q$.
9: end if
10: end while
11: Output $v_{\text{heur}} \leftarrow U$.

3.1 Mixed Integer Second Order Conic (MISOC) Formulation

In this subsection, we will derive a MISOC formulation and its analytical properties. To begin with, we first make an observation from the perspective formulation in [9, 15, 18, 23], where in [15], the authors introduced perspective relaxation for sparse regression with $L_0$ penalty term, where they convexified a quadratic term using perspective formulation. Let us consider a nonconvex set

$$W_i := \{(\beta_i, \mu_i, z_i) : \beta_i^2 \leq \mu_i, z_i \geq \mathbb{I}(\beta_i \neq 0), z_i \in \{0, 1\}\}, \quad (6)$$

for each $i \in [p]$. The results in [23] shows that the convex hull of $W_i$, denoted as $\text{conv}(W_i)$, can be characterized as below.

**Lemma 1** (Lemma 3.1. in [23]) For each $i \in [p]$, the convex hull of the set $W_i$ is

$$\text{conv}(W_i) = \{(\beta_i, \mu_i, z_i) : \beta_i^2 \leq \mu_i z_i, z_i \in [0, 1]\}. \quad (7)$$

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

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

which can be equivalently reformulated as

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

Note that (i) in (8), we replace $W_i$ by $\text{conv}(W_i)$ and enforce $z_i$ to be binary for each $i \in [p]$; and (ii) from Lemma 1, $\text{conv}(W_i)$ can be described by (7).

The above result is summarized in the following theorem.
Theorem 1 The formulation (F0-CCP) is equivalent to

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

(F0-MISC)

This formulation (F0-MISC) 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-MISC) 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-MISC). For notational convenience, let \( T \) denote the feasible region of (F0-MISC), i.e.,

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

(9)

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

Proposition 4 Let \( T \) denote the feasible region of (F0-MISC). Then

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

Proof. Let \( \hat{T} \) be the continuous relaxation set of \( T \), i.e.,

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

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

(i) It is clear that \( \text{conv}(T) \subseteq \hat{T} \).

(ii) To prove \( \hat{T} \subseteq \text{conv}(T) \), we only need to show that for any given point \( (\hat{\beta}, \hat{\mu}, \hat{z}) \in \hat{T} \), we have \( (\beta, \mu, z) \in \text{conv}(T) \). Since \( \hat{z} \in \{z : \sum_{i \in [p]} z_i \leq k, z \in [0, 1]^p\} \), which is an integral polytope, there exists \( K \) integral extreme points \( \{\hat{z}_t\}_{t \in [K]} \subseteq \mathbb{Z}_+^p \) such that \( \hat{z} = \sum_{t \in [K]} \lambda_t \hat{z}_t \) with \( \lambda_t \in (0, 1) \) for all \( t \) and \( \sum_{t \in [K]} \lambda_t = 1 \). Now we construct \( (\beta_t', \hat{\mu}_t') \) for each \( t \in [K] \) as follows:

\[ \hat{\mu}_t' = \begin{cases} \frac{\hat{\mu}_t}{\hat{z}_t} & \text{if } \hat{z}_t = 1 \\ 0 & \text{otherwise} \end{cases}, \quad \hat{\beta}_t' = \begin{cases} \frac{\hat{\beta}_t}{\hat{z}_t} & \text{if } \hat{z}_t = 1 \\ 0 & \text{otherwise} \end{cases}, \forall i \in [p]. \]

First of all, we claim that \( (\beta_t', \hat{\mu}_t', \hat{z}_t') \in T \) for all \( t \in [K] \). Indeed, for any \( t \in [K] \),

\[ (\beta_t')^2 = \begin{cases} \frac{(\hat{\beta}_t)^2}{\hat{z}_t} & \text{if } \hat{z}_t = 1 \\ 0 & \text{otherwise} \end{cases} \leq \hat{\mu}_t' \hat{z}_t' = \begin{cases} \frac{\hat{\mu}_t}{\hat{z}_t} & \text{if } \hat{z}_t = 1 \\ 0 & \text{otherwise} \end{cases}, \forall i \in [p] \]
\[
\sum_{i \in [p]} \bar{z}_i^t \leq k \\
\bar{z}_i^t \in \{0, 1\}^p.
\]

As \(\bar{z} = \sum_{t \in [K]} \lambda_t \bar{z}_i^t\), thus, for each \(i \in [p]\), we have

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

Thus, \((\hat{\beta}, \hat{\mu}, \bar{z}) \in \text{conv}(T)\). \(\square\)

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

**Proposition 5** The formulation (F0-CCP) is equivalent to

\[
v^* = \min_{(\beta, \mu, z) \in T} \left\{ \frac{1}{n} \|y - X \beta\|_2^2 + \lambda \|\mu\|_1 : \sum_{i \in [p]} z_i \leq k, z \in \{0, 1\}^p \right\}
\]

(F0-MISOC-M)

where the vector \(M = (M_1, \ldots, M_p)^\top\) can be chosen according to Proposition 1 and the set \(T\) is defined in (9).

### 3.2 Mixed Integer Convex (MIC) 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 \(\beta\) 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 recent work of [5] and [14].

**Proposition 6** ([5] and [14]) The formulation (F0-CCP) is equivalent to

\[
v^* = \min_z \left\{ f(z) := \lambda y^\top \left[ n \lambda I_n + \sum_{i \in [p]} z_i x_i x_i^\top \right]^{-1} y : \sum_{i \in [p]} z_i \leq k, z \in \{0, 1\}^p \right\}.
\]

(F0-MIC)

Note that in [5], the authors proposed a branch and cut algorithm to solve (F0-MIC), which was shown to be effective in solving some large-sized 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 $\hat{\beta}$ can be computed by the following formula:

$$\begin{cases}
\hat{\beta}_S = (X_S^\top X_S + n\lambda I_{|S|})^{-1} X_S^\top y, \\
\hat{\beta}_i = 0 \quad \text{if } i \in [p] \setminus S,
\end{cases}$$

(10)

where $\hat{\beta}_S$ denotes a sub-vector of $\hat{\beta}$ 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.,

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

(11a)

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

(11b)

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

(11c)

$$v_4 = \min_z \left\{ f(z) = \lambda y^\top \left[ n\lambda I + \sum_{i \in [p]} z_i x_i x_i^\top \right]^{-1} y : \sum_{i \in [p]} z_i \leq k, z \in [0,1]^p \right\}.$$

(11d)

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 (11a), (11b), (11c) and (11d), respectively. Then

(i) $v_2 = v_4 \leq v_3$; and

(ii) $v_1 \leq v_3$.

**Proof.** We separate the proof into three steps.

1. We will prove $v_2 = v_4$ first. By Lemma A.1. [45], we note that (11d) is equivalent to

$$v_4 = \min_{\gamma_0, \gamma, z} \lambda \left( \| \gamma_0 \|_2^2 + \sum_{i \in [p]} \frac{\gamma_i^2}{z_i} \right),$$

11
for each $i$.

Vice versa, 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 (11b), the relaxation bound can be further improved.

Finally, we would like to elaborate that by choosing the vector $M$ 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.

\[
\begin{align*}
\text{s.t. } & \sqrt{\lambda_n \gamma_0} + \sum_{i \in [p]} x_i \gamma_i = y, \\
& \sum_{i \in [p]} z_i \leq k, \\
& z \in [0,1]^p, \gamma_0 \in \mathbb{R}^n, \gamma_i \in \mathbb{R}, \forall i \in [p],
\end{align*}
\]

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

\[
\begin{align*}
v_4 &= \min_{\gamma_0, \beta, \mu, z} \lambda \left( \|\gamma_0\|_2^2 + \|\mu\|_1 \right), \\
\text{s.t. } & \sqrt{\lambda_n \gamma_0} + \sum_{i \in [p]} x_i \beta_i = y, \\
& \beta_i^2 \leq \mu_i z_i, \forall i \in [p], \\
& \sum_{i \in [p]} z_i \leq k, \\
& z \in [0,1]^p, \gamma_0 \in \mathbb{R}^n, \mu_i \in \mathbb{R}_+, \forall i \in [p].
\end{align*}
\]

Finally, in the above formulation, replace $\gamma_0 = \frac{1}{\sqrt{\lambda_n}} \left( y - \sum_{i \in [p]} x_i \beta_i \right) = \frac{1}{\sqrt{\lambda_n}} (y - X\beta)$. Then we arrive at (11b).

2. Next, we will prove $v_2 \leq v_3$. Note that the set of the constraints in (11b) is a subset of those in (11c). Thus, $v_2 \leq v_3$.

3. Third, we will prove $v_1 \leq v_3$. Let $(\beta^*, \mu^*, z^*)$ be optimal solution of (11c). Clearly, we must have $\mu^*_i = \frac{(\beta^*_i)^2}{z^*_i}$ for each $i \in [p]$, otherwise, suppose that there exists $i_0 \in [p]$ such that $\mu^*_{i_0} > \frac{(\beta^*_i)^2}{z^*_i}$, then the objective value of (11c) can be strictly less than $v_2$ by letting $\beta = \beta^*$ and $\mu = \mu^*$ for each $i \neq i_0$ and $\mu_{i_0} = \frac{(\beta^*_i)^2}{z^*_i}$, a contradiction of the optimality of $(\beta^*, \mu^*, z^*)$.

Therefore, we have

\[
\frac{1}{n} \|y - X\beta^*\|_2^2 + \lambda \|\beta^*\|_2^2 \leq \frac{1}{n} \|y - X\beta^*\|_2^2 + \lambda \sum_{i \in [p]} \frac{(\beta^*_i)^2}{z^*_i} = \frac{1}{n} \|y - X\beta^*\|_2^2 + \lambda \|\mu^*\|_1 = v_3,
\]

where the inequality is because $z^*_i \in [0,1]$ for each $i \in [p]$.

Hence, $(\beta^*, z^*)$ is feasible to (11a) with a smaller objective value. Thus, $v_1 \leq v_3$. \qed

Based on the results established in Theorem 2, we could directly solve the second order conic program (11b) 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 (11b), the relaxation bound can be further improved.

Finally, we would like to elaborate that by choosing the vector $M$ 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 $y = (1, 1)^T, X = I_2$. 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 about $M = (M_1, M_2)^T$:

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

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

given that $\lambda > 0$.

(ii) If we choose $M$ 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 MIC Formulation

The greedy approach (i.e., forward selection) has been commonly used as a heuristic to conduct the best subset selection [12, 47, 57]. 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 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 [46] as below:

$$\lambda y^T \left[ A_S + x_j x_j^T \right]^{-1} y - \lambda y^T A_S^{-1} y = -\frac{\lambda (y^T A_S^{-1} x_j)^2}{1 + x_j^T A_S^{-1} x_j},$$

$$\left[ A_S + x_j x_j^T \right]^{-1} = A_S^{-1} - \frac{A_S^{-1} x_j x_j^T A_S^{-1}}{1 + x_j^T A_S^{-1} x_j},$$

where $A_S = n\lambda I_n + \sum_{i \in S} x_i x_i^T$.

This motivates us 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 $\{A_S^{-1} x_j\}_{j \in [p]}$, $\{x_j A_S^{-1} x_j\}_{j \in [p]}$ and $\{y A_S^{-1} x_j\}_{j \in [p]}$, which has space complexity $O(np)$ and update them from one iteration to another iteration, which costs $O(np)$ operations per iteration. Therefore, the space and time complexity of Algorithm 2 are $O(np)$ and $O(npk)$, respectively.

From our empirical study, the greedy approach work quite appealing. Indeed, we will investigate the greedy solution and prove that it can be very close to the true optimal, in particular when $\lambda$
is not too small. To begin with, let us define $\theta_s$ to be the largest singular value of all the matrices $X_S X_S^T$ with $|S| = s$, i.e.,

$$\theta_s := \max_{|S| = s} \sigma_{\text{max}}^2(X_S) = \max_{|S| = s} \sigma_{\text{max}}(X_S X_S^T), \quad (12)$$

for each $s \in [p]$. By definition (12), we have $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_p$, 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 $p \geq k$, then the solution of greedy approach will be quite close to any optimal estimator from (F0-CCP) as $\lambda$ grows.

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

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

where $\theta$ defined in (12) and

$$\theta = \min_{T \subseteq [p], |T| \geq p - k + 1} \sigma_{\text{min}}(X_T X_T^T).$$

**Proof.**

First of all, suppose that $z^*$ is an optimal solution to (F0-MIC). According to the definition of $\theta_k$, we have $n\lambda I_n + \sum_{i \in [p]} z_i^* x_i x_i^T \leq (n\lambda + \theta_k) I_n$. Thus,

$$v^* = \lambda y^T \left(n\lambda I_n + \sum_{i \in [p]} z_i^* x_i x_i^T \right) y \geq \frac{\lambda}{n\lambda + \theta_k} \|y\|^2. \quad (14)$$

On the other hand, according to Step 3 of Algorithm 2, for any given $S$ such that $|S| = s < k$, and $A_S = n\lambda I_n + \sum_{i \in S} x_i x_i^T$ and $j \in [p] \setminus S$, we observe that

$$\lambda y^T \left[A_S + x_j x_j^T\right]^{-1} y - \lambda y^T A_S^{-1} y = -\frac{\lambda (y^T A_S^{-1} x_j)^2}{1 + x_j^T A_S^{-1} x_j}.$$ \quad (15)

Thus, using the identity (15), we can prove by induction that the greedy value is upper bounded by

$$v^G \leq \left(1 - \frac{n^2 \lambda^2 \theta}{(n\lambda + \theta_1)(n\lambda + \theta_k)^2} \sum_{i \in [k]} \frac{1}{p + 1 - i}\right) \frac{1}{n} \|y\|^2.$$ \quad (16)
Indeed, if \( k = 0 \), then (16) holds. Suppose that \( k = t \geq 0 \), (16) holds. Now let \( k = t + 1 \) and let \( S \) be the selected subset at iteration \( t \). By induction, we have

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

And by the greedy selection procedure, we further have

\[
v^G = \lambda y^\top A_S^{-1} y + \min_{j \in [p] \setminus S} \lambda y^\top \left[ A_S + x_j x_j^\top \right]^{-1} y - \lambda y^\top A_S^{-1} y
\]

\[
\leq \lambda y^\top A_S^{-1} y + \frac{1}{p - t} \sum_{j \in [p] \setminus S} \left[ -\frac{\lambda (y^\top A_S^{-1} x_j)^2}{1 + x_j^\top A_S^{-1} x_j} \right]
\]

\[
\leq \lambda y^\top A_S^{-1} y - \frac{n\lambda^2}{(p - t)(n\lambda + \theta_1)} y^\top A_S^{-1} (X_{[p] \setminus S} X_{[p] \setminus S}^\top) A_S^{-1} y
\]

\[
\leq \left( 1 - \frac{n^2 \lambda^2 \theta}{(n\lambda + \theta_1)(n\lambda + \theta_k)^2} \sum_{i \in [t+1]} \frac{1}{p + 1 - i} \right) \frac{1}{n} \|y\|^2,
\]

where the first equality is due to (15), the first inequality is because the minimum is no larger than the average, the second inequality is because \( A_S \succeq n\lambda I_n \) and \( \|x_j\|^2 \leq \theta_1 \), and the third inequality is due to the induction and the facts that \( p \geq k \), \( A_S \succeq (n\lambda + \theta_k) I_n \), \( \theta \leq \sigma_{\min}(X_{[p] \setminus S} X_{[p] \setminus S}^\top) \).

Combining (14) and (15) and using the fact that \( \sum_{i \in [k]} \frac{1}{p + 1 - i} \geq \int_0^k \frac{1}{p + 1 - t} dt = \log \left( \frac{p + 1}{p + 1 - k} \right) \), the conclusion follows.

We make the following remarks about Theorem 3.

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

(ii) If we normalize \( \|x_i\|^2 = n \) for each \( i \in [p] \), we must have \( \theta_k \leq kn \), thus \( \frac{n\lambda}{n\lambda + \theta_k} \leq \frac{\lambda}{\lambda + k} \).

Therefore, we can see that the objective value of greedy approach is closer to the true optimal value if the tuning parameter becomes larger.

(iii) Besides, our analysis and asymptotic optimality of the greedy approach is new without any assumption on the data and thus is quite different from the existing ones for sparse regression \([11, 12, 30, 57]\). 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 derive a randomized algorithm and prove its approximation guarantee under a weaker condition of \( \lambda \).

In addition, we remark that the estimator \( \beta^G \) of the greedy approach can be computed by (10), 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., \( \beta^G \)) can be also quite close to an optimal solution \( \beta^* \) of (F0-CCP).
Theorem 4 Let $\beta^*$ be an optimal solution to (F0-CCP) with set of selected features $S^*$ and $\beta^G$ be the estimator from the greedy approach with set of selected features $S^G$. Suppose that $p \geq k$, then we have

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

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

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

Proof. Note that the greedy estimator $\beta^G$ can be computed through (10) by setting $S$ to be $S^G$, the set of selected features by greedy approach. Moreover, we define $\tilde{X}$ as follows:

$$\begin{align*}
\{ \tilde{X}_{S^G\setminus S^*} &= X_{S^G\setminus S^*}, \\
\tilde{X}_i &= 0 \quad \text{if } i \in [p] \setminus (S^G \setminus S^*)
\end{align*}$$

Then we have,

$$\frac{1}{n}\|y - X\beta^G\|_2^2 + \lambda\|\beta^G\|_2^2 - \left[\frac{1}{n}\|y - X\beta^*\|_2^2 + \lambda\|\beta^*\|_2^2\right] \leq n\nu v^* \tag{\text{X}}$$

$$\Leftrightarrow -2(\beta^* - \beta^G)^\top \left[-\frac{1}{n}X^\top (y - X\beta^*) + \lambda\beta^*\right]$$

$$+ (\beta^* - \beta^G)^\top \left[\frac{1}{n}X^\top X + \lambda I_p\right] (\beta^* - \beta^G) \leq n\nu v^*$$

$$\Leftrightarrow -2(\beta^* - \beta^G)^\top \left[-\frac{1}{n}\tilde{X}^\top (y - X\beta^*)\right]$$

$$+ (\beta^*_S - \beta^G_S)^\top \left[\frac{1}{n}\tilde{X}_{SU}^\top \tilde{X}_{SU} + \lambda I_{SU}\right] (\beta^*_S - \beta^G_S) \leq n\nu v^* \tag{\text{Y}}$$

$$\Rightarrow -\frac{2}{n}\|\tilde{X}\|_2\|y - X\beta^*\|_2\|\beta^*_S - \beta^G_S\|_2$$

$$+ \left(\lambda + \sigma_{\min}(X_{S^G\setminus S^*}) \right)\|\beta^* - \beta^G\|_2 \leq n\nu v^*$$

$$\Rightarrow -\sqrt{\frac{4\theta_{S^G\setminus S^*}}{n}}\|\beta^* - \beta^G\|_2 + \left(\lambda + \sigma_{\min}(X_{S^G\setminus S^*}) \right)\|\beta^* - \beta^G\|_2 \leq n\nu v^*$$

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

where the second equivalence is due to the optimality condition of $\beta^*$, i.e.,

$$-\frac{1}{n}X_{S^*}^\top (y - X_{S^*}\beta^*_S) + \lambda\beta^*_S = 0$$

and the nonzero entries of $\beta^* - \beta^G$ are only from subset $S^U := S^G \cup S^*$. The first implication is due to sub-multiplicativity of matrix norm and $\|A\|_2 \geq \sigma_{\min}(A)$,
the second implication is because of \( \|\tilde{X}\|_2 \leq \sqrt{k}, \|y - X\beta^*\|_2 \leq \sqrt{nv^*} \), and the last implication is because any solution of the following quadratic inequality \( at^2 - bt - c \leq 0 \) with \( a, b, c > 0 \) is 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 MISOC Formulation

In this subsection, we develop a randomized algorithm based on the continuous relaxation solution of (F0-MISOC), i.e., the optimal solution to (11b), which can be efficiently solved via the interior point method or other convex optimization approaches [3].

Suppose that \( \hat{z} \) is the optimal solution of the continuous relaxation model (11b). For each \( i \in [p] \), the column \( x_i \) will be picked by probability \( \hat{z}_i \). The detailed implementation is illustrated in Algorithm 3.

Algorithm 3 Proposed Randomized Algorithm

1: Let \( \hat{z} \) be the optimal solution to (11b)
2: Initialize set \( S = \emptyset \) and vector \( \tilde{z} = 0 \in \mathbb{R}^p \)
3: for \( i = 1, \ldots, p \) do
4: Sample a standard uniform random variable \( U \)
5: if \( U \leq \hat{z}_i \) then
6: Let \( S = S \cup \{i\} \) and \( \tilde{z}_i = 1 \)
7: end if
8: end for
9: Output \( S, \tilde{z} \)

Next, we will show that if \( \lambda \) 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 with, we present the following matrix concentration bound.

Lemma 2 (Theorem 1.4., [50]) Consider a finite sequence \( \{Y_k\} \) of independent, random, symmetric matrices with dimension \( d \). Assume that each random matrix satisfies \( E[Y_k] = 0 \) and \( \|Y_k\|_2^2 \leq R^2 \) almost surely. Then, for all \( t \geq 0 \), we have

\[
\mathbb{P}\left\{ \left\| \sum_k Y_k \right\|_2 \geq t \right\} \leq d \exp \left( -\frac{t^2}{2\nu^2 + 2/3Rt} \right), \tag{17}
\]

where \( \nu^2 := \| \sum_k E[Y_k^2] \|_2 \).

Lemma 2 implies that if \( \lambda \) is not too small, then with high probability, \( \lambda nI_n + \sum_{i \in S} \hat{z}_i x_i x_i^\top \) has the similar eigenvalues as \( \lambda nI_n + \sum_{i \in [p]} \hat{z}_i x_i x_i^\top \), where \( \hat{z} \) is the optimal solution to (11b) and \( S \) is the output of Algorithm 3.

Lemma 3 Let \( \hat{z} \) be the optimal solution to (11b) and \( S \) be the output of Algorithm 3. Given that \( \alpha \in (0, 1) \) and

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

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

$$(1 - \epsilon)u^\top \Sigma_* u \leq u^\top \hat{\Sigma} u \leq (1 + \epsilon)u^\top \Sigma_* u, \forall u \in \mathbb{R}^n,$$

where $\Sigma_* = \lambda n I_n + \sum_{i \in [p]} \hat{z}_i x_i x_i^\top$ and $\hat{\Sigma} = \lambda n I_n + \sum_{i \in S} x_i x_i^\top$.

**Proof.** Let $\hat{z}$ be the optimal solution to $(11b)$ and let $\{r_i\}_{i \in [p]}$ be independent Bernoulli random variables with $\mathbb{P}\{r_i = 1\} = \hat{z}_i$ for each $i \in [p]$. Consider the random matrix defined as for each $i \in [p]$, $A_i = (r_i - \hat{z}_i) x_i x_i^\top$ and $\mathbb{E}[A_i] = 0$. On the other hand, by definition, we have $\|x_i\|^2 \leq \theta_1$ for each $i \in [p]$, thus

$$\|A_i\|_2 = |r_i - \hat{z}_i| \|x_i\|^2 \leq \theta_1 := R^2.$$

Also,

$$\left\| \sum_{i \in [p]} \mathbb{E}[A_i^2] \right\|_2 \leq \left\| \sum_{i \in [p]} \hat{z}_i (1 - \hat{z}_i) \|x_i\|^2 x_i x_i^\top \right\|_2 = \left\| \sum_{i \in [p]} \hat{z}_i (1 - \hat{z}_i) x_i x_i^\top \right\|_2 \leq \theta_k,$$

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

$$\max_{\mathbf{z} \in [0, 1]^p} \left\{ \sigma_{\max} \left( \sum_{i \in [p]} \hat{z}_i x_i x_i^\top \right) : \sum_{i \in [p]} \hat{z}_i = k \right\} = \max_{\mathbf{z} \in [0, 1]^p} \left\{ \sigma_{\max} \left( \hat{z}_i x_i x_i^\top : \sum_{i \in [p]} \hat{z}_i = k \right) : \hat{z}_i \right\} = \theta_k.$$

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

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

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

Thus,

$$\mathbb{P} \left\{ \left\| \sum_{i \in [p]} \left( \hat{\Sigma} - \Sigma_* \right) \right\|_2 \geq \epsilon \sigma_{\min}(\Sigma_*) \right\} \leq \frac{\alpha}{2},$$

$$\Leftrightarrow n \exp \left( -\frac{\epsilon^2 \sigma_{\min}(\Sigma_*)^2}{2 \theta_k + 2/3 \epsilon \sigma_{\min}(\Sigma_*)} \right) \leq \frac{\alpha}{2},$$

18
\[ \sigma_{\min}(\Sigma_*) \geq \frac{\log(2n/\alpha)\sqrt{\theta_1}}{3\epsilon} + \frac{\sqrt{2}\theta_k \log(2n/\alpha)}{2\epsilon}, \]

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

where the second implication is because the following quadratic inequality \( at^2 - bt - c \geq 0 \) with \( a, b, c > 0 \) is satisfied if \( t \geq \frac{b}{a} + \sqrt{\frac{b^2 - 4ac}{a}} \), and the third implication is due to \( \lambda n \leq \sigma_{\min}(\Sigma_*) \).

Then the conclusion follows directly by Weyl’s theorem \([19, 54]\). \(\square\)

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

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

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

then with probability at least \(1 - \alpha\), we have

\[ \lambda y^\top \left[ \lambda nI + \sum_{i \in [p]} \tilde{z}_i x_i x_i^\top \right]^{-1} y \leq (1 + \epsilon) v^* \]  

and

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

**Proof.** Note that (18) follows from Lemma 3. The result in (19) holds due to the Chernoff bound \([10]\), i.e.,

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

Therefore, by Boole’s inequality, we arrive at the conclusion. \(\square\)

Next, let \(\beta^R\) be the estimator from Algorithm 3, which can be computed according to (10) by letting \(S\) be the output from Algorithm 3. Then we can show that the distance between \(\beta^R\) and \(\beta^*\) (i.e., \(\|\beta^R - \beta^*\|_2\)) can be also quite small, where \(\beta^*\) is an optimal solution to (F0).

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

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

**Proof.** The proof is almost identical to that of Theorem 4, thus is omitted here. \(\square\)

Finally, we remark that we can integrate the greedy approach with the randomized algorithm, which is to apply the greedy approach based upon the support of the continuous relaxation solution

19
of (F0-MISOC). That is, given that \( \widehat{z} \) is the optimal solution to (11b) and \( \delta > 0 \) is a positive constant, then we first let set \( \mathcal{C} := \{ i \in [p] : \widehat{z}_i \geq \delta \} \) and apply greedy approach (Algorithm 2) to set \( \mathcal{C} \) rather than \([p] \), which could save a significant amount of computational time, in particular when continuous relaxation solution \( \widehat{z} \) is very sparse. The detailed description can be found in Algorithm 4.

**Algorithm 4 Proposed Restricted Greedy Approach**

1: Let \( \widehat{z} \) be the optimal solution to (11b)
2: Initialize \( \delta > 0 \) (e.g., \( \delta = 0.01 \)), \( \mathcal{C} := \{ i \in [p] : \widehat{z}_i \geq \delta \} \)
3: Let \( S = \emptyset \) and \( A_S = n\lambda I_n \)
4: for \( i = 1, \ldots, k \) do
5: \hspace{1em} Let \( j^* \in \arg \min_{j \in \mathcal{C} \setminus S} \left\{ -\frac{\lambda (y^\top A_S^{-1} x_j)^2}{1 + x_j^\top A_S^{-1} x_j} \right\} \)
6: \hspace{1em} Let \( S = S \cup \{ j^* \} \) and \( A_S = A_S + x_{j^*} x_{j^*}\top, A_S^{-1} = A_S^{-1} - \frac{A_S^{-1} x_{j^*} x_{j^*}\top A_S^{-1}}{1 + x_{j^*} A_S^{-1} x_{j^*}} \)
7: end for
8: Output \( v_{RG} \leftarrow \lambda y^\top A_S^{-1} y \).

5 Selection of Tuning Parameter and Generalization to Sparse Matrix Estimation

In this section, we will discuss how to select the tuning parameter \( \lambda \) 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 (GCV)

For a given \( k \), we can adopt the commonly-used generalized cross-validation (GCV) [22, 51] to choose the best \( \lambda \) in the ridge regression. Specifically, the GCV can be defined as

\[
GCV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{y_i - \widehat{y}_i}{1 - (H_S)_{ii}} \right)^2,
\]

where \( H_S = X_S (X_S^\top X_S + n\lambda I)^{-1} X_S^\top \) denotes the hat matrix of the ridge regression and \( \widehat{y} = H_S y \) is the vector of the fitted responses. With a sequence of \( \lambda \) values in \( \{ \lambda_1, \ldots, \lambda_m \} \), we can choose the one having the smallest \( GCV(\lambda) \) value.

5.2 Generalization to Sparse Matrix Estimation

In this subsection, we consider a sparse matrix estimation proposed by [8]. In that problem, the authors were trying to estimate the inverse of covariance matrix \( \widehat{\Sigma} \in \mathbb{R}^{t \times t} \) and choose the sparsest estimator. In their model, they optimize the \( L_1 \) norm of the estimator given that the estimation error is within a constant. Similar to (F0), instead we can directly optimize the estimation error given that only \( k \) sparse elements can be chosen, which can be formulated as below

\[
v^* = \min_{\Omega} \left\{ \| I_t - \widehat{\Sigma} \Omega \|_F^2 + \lambda \| \Omega \|_F^2 : \| \Omega \|_0 \leq k \right\},
\]

(21)
To view this model as a special case of (F0), we rewrite matrix $\Omega$ as a vector $\beta \in \mathbb{R}^{t \times t^2}$ and $\hat{\Sigma}$ as $X \in \mathbb{R}^{t \times t^2}$, where

\[
\beta(j + t(i - 1)) = \Omega(i, j), \forall i, j \in [t],
\]

\[
X(s, r) = \begin{cases} 
\Sigma(s, r - t(s - 1)) & \text{if } 1 \leq r - t(s - 1) \leq t, \forall s \in [t], r \in [t^2], \\
0, & \text{otherwise}
\end{cases}
\]

\[
y_{j + t(i - 1)} = \begin{cases} 
1, & \text{if } i = j, \\
0, & \text{otherwise}
\end{cases}, \forall i, j \in [t]
\]

Thus, (21) reduces to (F0). Then the results for sparse ridge regression in the previous sections hold for (21).

6 Experimental Verification

6.1 Comparison of 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 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 on solving sparse ridge regression problems. The data are generated from the linear model

\[ y = x^\top \beta^0 + \tilde{\epsilon}, \]

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

\[ x_i \sim N(0, \Sigma), i = 1, \ldots, n, \]

where $\Sigma$ 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 $\beta^0 = (\beta_1^0, \ldots, \beta_p^0)^\top$ are nonzero, and their values are drawn randomly from the uniform distribution $\text{Unif}(-3, 3)$. To control the signal-to-noise ratio (SNR), we choose the value of $\sigma^2$ such that $\text{SNR} = \text{var}(x^\top \beta^0) / \text{var}(\tilde{\epsilon}) = 9$. 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., $y_i = x_i^\top \beta^0 + \tilde{\epsilon}_i$ for each $i \in [n]$.

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

The methods in comparison 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.

---

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

2Please note that [5] proposed a sophisticated warm-start procedure. However, for the sake of fair comparison, we directly use branch-and-cut algorithm without any warm-start procedure.
Note that the heuristic Algorithm 1 in [1] is similar to the LASSO in the use of $L_1$ norm to achieve the 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 an 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 Table 1 to Table 3, where the Avg. Obj. Value, Avg. Gap, Avg. Comp. Time, and Avg. False Alarm Rate denotes the average objective function value, average optimality gap (of exact methods) from Guorbi, average computational time (in seconds), and average percent 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 the computations were executed on a MacBook Pro with a 2.80 GHz processor and 16GB RAM.

Table 1: Comparison of the Branch and Cut algorithm in [5] and directly solving (F0-MISOC) with $p = 1000$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$k$</th>
<th>$n$</th>
<th>Branch and Cut Algorithm</th>
<th>Solving (F0-MISOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>10</td>
<td>1000</td>
<td>9.71</td>
<td>3438.51</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>5000</td>
<td>7.11</td>
<td>2451.47</td>
</tr>
<tr>
<td>1000</td>
<td>20</td>
<td>1000</td>
<td>NA*</td>
<td>NA</td>
</tr>
<tr>
<td>500</td>
<td>20</td>
<td>1000</td>
<td>23.02</td>
<td>3600.00</td>
</tr>
<tr>
<td>500</td>
<td>5000</td>
<td>20</td>
<td>31.52</td>
<td>3600.00</td>
</tr>
<tr>
<td>500</td>
<td>5000</td>
<td>1000</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>500</td>
<td>30</td>
<td>1000</td>
<td>39.62</td>
<td>3600.00</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
<td>30</td>
<td>50.63</td>
<td>3600.00</td>
</tr>
<tr>
<td>500</td>
<td>5000</td>
<td>30</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

The NA* represents for out of memory instances.

Table 1 reports the comparison results between directly solving (F0-MISOC) and the branch-and-cut algorithm based upon (F0-MIC). It is seen that directly solving (F0-MISOC) outperforms the branch-and-cut algorithm for most of the instances, in particular when $k$ becomes large. This is because (i) we proved in Theorem 2 that continuous relaxations of (F0-MIC) and (F0-MISOC) are equivalent, thus directly solving (F0-MISOC) should perform at least as good as 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, for both approaches, they reach the time limit for most of the cases, and the average false alarm rates are higher than the approximation algorithms in Table 2. Therefore, for large-scale instances, these approaches might not be very desirable.

From Table 2 and Table 1, the proposed greedy Algorithm 2 and restricted greedy Algorithm 4 apparently perform best among all comparison 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 methods in comparison have relatively larger false alarm rates. Their performance of identifying right features improves as the sample size $n$ increases, i.e., providing more information. For the heuristic Algorithm 1 in [1], it is less accurate and takes a much longer
Table 2: Comparison of Heuristic Algorithm 1 in [1], Greedy Algorithm 2, Randomized Algorithm 3 and Restricted Greedy Algorithm 4 with \( p = 1000 \)

<table>
<thead>
<tr>
<th>( p )</th>
<th>( k )</th>
<th>( n )</th>
<th>Heuristic Algorithm 1 in [1]</th>
<th>Proposed Greedy Algorithm 2</th>
<th>Proposed Randomized Algorithm 3</th>
<th>Proposed Restricted Greedy Algorithm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>10</td>
<td>500</td>
<td>9.59</td>
<td>579.36</td>
<td>3.0%</td>
<td>6.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>7.88</td>
<td>45.78</td>
<td>0.0%</td>
<td>6.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5000</td>
<td>7.24</td>
<td>737.06</td>
<td>0.0%</td>
<td>6.67</td>
</tr>
<tr>
<td>1000</td>
<td>20</td>
<td>500</td>
<td>15.87</td>
<td>589.66</td>
<td>14.5%</td>
<td>10.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>13.42</td>
<td>47.92</td>
<td>11.5%</td>
<td>10.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5000</td>
<td>12.66</td>
<td>738.55</td>
<td>4.5%</td>
<td>11.30</td>
</tr>
<tr>
<td>1000</td>
<td>30</td>
<td>500</td>
<td>28.87</td>
<td>583.98</td>
<td>17.0%</td>
<td>16.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>23.53</td>
<td>43.92</td>
<td>12.7%</td>
<td>17.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5000</td>
<td>19.74</td>
<td>678.10</td>
<td>6.0%</td>
<td>17.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thus, it might not be a good option for large-scale instances either. In contrast, we note that the greedy Algorithm 2 is much 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-MISO), is quite insensitive to \( k \) in terms of computation time. Therefore, by integrating these two together, 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 most efficient algorithms for comparison.

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 a much longer time to solve the continuous relaxation to the optimality and for these instances, \( k \) is relatively small. In particular, for a large-scale datasets (e.g., \( n = p = 5000 \)), the computation time of the restricted greedy Algorithm 4 is much longer time than those 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 would recommend finding a reasonable sample size that the greedy methods can work efficiently and identify the features accurately.
6.2 Further Investigation of MISOC Formulation (F0-MISOC) and Greedy Algorithm 2 with Varying SNR and Tuning Parameter $\lambda$ 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 $n = 100$, $p = 40$, $k \in \{5, 10, 15, 20\}$, SNR $\in \{0.5, 1, 2, 4\}$ and to illustrate the effects of tuning parameter $\lambda$, 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 Table 4 and Table 5.

In Table 4 and Table 5, 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 more accurate prediction. In terms of tuning parameter, we see that the computational time of formulation (F0-MISOC) changes significantly as $\lambda$ increases. On the other hand, if the tuning parameter $\lambda$ is too big, the false alarm rate will increase significantly. Thus, a proper choice of the tuning parameter $\lambda$ will be critical for formulation (F0-MISOC). In next subsection, we will use the generalized cross validation to choose a proper tuning parameter $\lambda$ for a real-world case.

### Table 3: Comparison of Greedy Algorithm 2 and Restricted Greedy Algorithm 4 with $p = 5000$

<table>
<thead>
<tr>
<th>$p$</th>
<th>$k$</th>
<th>$n$</th>
<th>Proposed Greedy Algorithm 2</th>
<th></th>
<th>Proposed Restricted Greedy Algorithm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5000</td>
<td>4.57</td>
<td>2.31</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1000</td>
<td>4.59</td>
<td>3.13</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>500</td>
<td>4.68</td>
<td>9.04</td>
<td>0.0%</td>
</tr>
<tr>
<td>5000</td>
<td>20</td>
<td>500</td>
<td>12.86</td>
<td>4.31</td>
<td>8.0%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td></td>
<td>13.35</td>
<td>5.41</td>
<td>2.5%</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td></td>
<td>13.27</td>
<td>14.58</td>
<td>0.0%</td>
</tr>
<tr>
<td>30</td>
<td>500</td>
<td></td>
<td>14.02</td>
<td>5.98</td>
<td>20.7%</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td></td>
<td>14.97</td>
<td>8.21</td>
<td>12.7%</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td></td>
<td>15.60</td>
<td>20.52</td>
<td>3.3%</td>
</tr>
</tbody>
</table>

6.3 A Real-world Case Study using the Dataset in [53]

In this subsection, we conduct a case study using the dataset in [53], 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 [53], 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 generalized cross-
Table 4: Comparison of MISOC Formulation (F0-MISOC) and Greedy Algorithm 2 with $n = 100, p = 40, k \in \{5, 10\}$

<table>
<thead>
<tr>
<th>$k$</th>
<th>SNR</th>
<th>$\lambda$</th>
<th>MISOC Formulation (F0-MISOC)</th>
<th>Proposed Greedy Algorithm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.01</td>
<td>29.75</td>
<td>81.72</td>
<td>44.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>31.65</td>
<td>4.44</td>
<td>38.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>39.98</td>
<td>0.28</td>
<td>36.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>49.13</td>
<td>0.29</td>
<td>44.0%</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>15.07</td>
<td>54.24</td>
<td>38.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>16.56</td>
<td>2.11</td>
<td>38.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>23.90</td>
<td>0.29</td>
<td>44.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>32.73</td>
<td>0.28</td>
<td>52.0%</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>7.11</td>
<td>64.68</td>
<td>26.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>8.40</td>
<td>2.80</td>
<td>26.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>14.71</td>
<td>0.29</td>
<td>32.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>21.90</td>
<td>0.28</td>
<td>44.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>3.86</td>
<td>20.76</td>
<td>20.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>5.13</td>
<td>1.17</td>
<td>18.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>11.17</td>
<td>0.28</td>
<td>34.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>18.15</td>
<td>0.28</td>
<td>40.0%</td>
</tr>
<tr>
<td>0.5</td>
<td>0.01</td>
<td>49.00</td>
<td>43.43</td>
<td>49.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>53.08</td>
<td>3.00</td>
<td>48.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>68.60</td>
<td>0.25</td>
<td>50.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>85.37</td>
<td>0.24</td>
<td>55.0%</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>24.62</td>
<td>12.44</td>
<td>46.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>27.81</td>
<td>0.68</td>
<td>40.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>41.93</td>
<td>0.23</td>
<td>44.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>58.06</td>
<td>0.27</td>
<td>50.0%</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>13.02</td>
<td>7.70</td>
<td>29.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>15.84</td>
<td>0.51</td>
<td>28.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>28.53</td>
<td>0.21</td>
<td>34.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>42.95</td>
<td>0.25</td>
<td>42.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>6.70</td>
<td>2.51</td>
<td>27.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>9.22</td>
<td>0.40</td>
<td>30.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>20.67</td>
<td>0.21</td>
<td>34.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>34.09</td>
<td>0.25</td>
<td>49.0%</td>
</tr>
</tbody>
</table>
Table 5: Comparison of MISOC Formulation (F0-MISOC) and Greedy Algorithm 2 with $n = 100, p = 40, k \in \{15, 20\}$

<table>
<thead>
<tr>
<th>$k$</th>
<th>SNR</th>
<th>$\lambda$</th>
<th>MISOC Formulation (F0-MISOC)</th>
<th>Proposed Greedy Algorithm 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Value</td>
<td>Time(s)</td>
</tr>
<tr>
<td>0.5</td>
<td>0.01</td>
<td>66.81</td>
<td>110.75</td>
<td>46.7%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>74.92</td>
<td>4.54</td>
<td>46.7%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>101.07</td>
<td>0.27</td>
<td>42.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>126.31</td>
<td>0.27</td>
<td>46.7%</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>34.42</td>
<td>162.55</td>
<td>37.3%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>39.85</td>
<td>5.54</td>
<td>36.7%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>60.56</td>
<td>0.33</td>
<td>41.3%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>83.68</td>
<td>0.29</td>
<td>49.3%</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>18.85</td>
<td>34.03</td>
<td>25.3%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>23.72</td>
<td>1.66</td>
<td>25.3%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>42.64</td>
<td>0.27</td>
<td>32.7%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>61.73</td>
<td>0.25</td>
<td>41.3%</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>9.50</td>
<td>11.84</td>
<td>22.7%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>14.01</td>
<td>0.53</td>
<td>24.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>32.67</td>
<td>0.26</td>
<td>30.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>52.66</td>
<td>0.28</td>
<td>39.3%</td>
</tr>
<tr>
<td>0.5</td>
<td>0.01</td>
<td>66.04</td>
<td>81.72</td>
<td>43.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>74.61</td>
<td>4.44</td>
<td>41.5%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>103.88</td>
<td>0.28</td>
<td>38.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>136.87</td>
<td>0.29</td>
<td>41.5%</td>
</tr>
<tr>
<td>1</td>
<td>0.01</td>
<td>28.59</td>
<td>54.24</td>
<td>34.5%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>33.63</td>
<td>2.11</td>
<td>33.5%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>52.82</td>
<td>0.29</td>
<td>34.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>76.04</td>
<td>0.28</td>
<td>38.5%</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>15.95</td>
<td>64.68</td>
<td>29.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>20.16</td>
<td>2.80</td>
<td>27.5%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>36.68</td>
<td>0.29</td>
<td>27.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>58.11</td>
<td>0.28</td>
<td>35.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>8.24</td>
<td>20.76</td>
<td>27.0%</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>11.78</td>
<td>1.17</td>
<td>24.0%</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>27.25</td>
<td>0.28</td>
<td>32.0%</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>48.07</td>
<td>0.28</td>
<td>38.0%</td>
</tr>
</tbody>
</table>
Table 6: Feature Selection Results using the Dataset in [53] and GCV in Section 5.1. Here, $x_i$ denotes $i$th marker and $x_i.x_j$ represents the interaction of markers $i$ and $j$.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$k$</th>
<th>Selected Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-4}$</td>
<td>10</td>
<td>$x_1, x_{18}, x_{48}, x_1.x_{48}, x_{5}.x_{15}, x_{11}.x_{33}, x_{17}.x_{48}, x_{42}.x_{45}$</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>$x_1, x_{18}, x_{37}, x_{48}, x_1.x_4, x_1.x_{18}, x_1.x_{48}, x_{5}.x_{15}, x_{11}.x_{42}, x_{14}.x_{37}$, $x_{16}.x_{33}, x_{16}.x_{45}, x_{17}.x_{27}, x_{17}.x_{48}, x_{34}.x_{40}, x_{34}.x_{48}, x_{36}.x_{40}, x_{36}.x_{48}$, $x_{40}.x_{45}, x_{42}.x_{45}$</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>40</td>
<td>$x_1, x_{10}, x_{18}, x_{37}, x_{40}, x_{48}, x_1.x_{10}, x_1.x_{18}, x_1.x_{48}, x_3.x_{44}, x_{5}.x_{15}, x_{5}.x_{48}, x_7.x_{10}, x_{9}.x_{10}, x_{9}.x_{13}, x_9.x_{18}, x_{10}.x_{13}, x_{10}.x_{18}, x_{10}.x_{30}, x_{11}.x_{40}$, $x_{11}.x_{42}, x_{12}.x_{36}, x_{13}.x_{33}, x_{14}.x_{37}, x_{16}.x_{33}, x_{16}.x_{45}, x_{17}.x_{27}, x_{17}.x_{48}$, $x_{18}.x_{36}, x_{34}.x_{40}, x_{34}.x_{45}, x_{34}.x_{48}, x_{35}.x_{45}, x_{35}.x_{48}, x_{36}.x_{40}, x_{36}.x_{48}$, $x_{40}.x_{45}, x_{42}.x_{45}, x_{46}.x_{48}$</td>
</tr>
</tbody>
</table>

In Table 6, we see that using GCV procedure in Section 5.1, the best tuning parameter $\lambda$ tends to be small in particular 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 [34] except $x_{23}$. This demonstrates that our proposed method is indeed effective for feature selection problems.

7 Conclusion

This paper studies the sparse ridge regression with the use of exact $L_0$ norm for the sparsity. It is known that imposing $L_0$ norm for the sparsity in regression can often become an NP-hard problem in variable selection and estimation. We propose a mixed integer second order conic (MISOC) formulation, which is big-M free and is derived based on perspective formulation. We prove that the continuous relaxation of this MISOC reformulation is equivalent to the convex integer program (CIP) formulation studied by literature, and can be stronger than straightforward big-M formulation. Based on these two formulations, we propose two scalable algorithms, the greedy and randomized algorithms, for solving the sparse ridge regression. Under mild conditions, both algorithms can find near-optimal solutions with performance guarantees. Our numerical study demonstrates that the proposed algorithms can indeed solve large-scale instances efficiently. For the small or medium-size instances, we recommend solving MISOC formulation first, which might be efficient as well.

References


30


