Consistent Second-Order Conic Integer Programming for Learning Bayesian Networks

Simge Küçükyavuz[*†] Ali Shojaie[† ‡] Hasan Manzour[§] Linchuan Wei[¶]
Hao-Hsiang Wu [||]

May 6, 2022

Abstract

Bayesian Networks (BNs) represent conditional probability relations among a set of random variables (nodes) in the form of a directed acyclic graph (DAG), and have found diverse applications in knowledge discovery. We study the problem of learning the sparse DAG structure of a BN from continuous observational data. The central problem can be modeled as a mixed-integer program with an objective function composed of a convex quadratic loss function and a regularization penalty subject to linear constraints. The optimal solution to this mathematical program is known to have desirable statistical properties under certain conditions. However, the state-of-the-art optimization solvers are not able to obtain provably optimal solutions to the existing mathematical formulations for medium-size problems within reasonable computational times. To address this difficulty, we tackle the problem from both computational and statistical perspectives. On the one hand, we propose a concrete early stopping criterion to terminate the branch-and-bound process in order to obtain a near-optimal solution to the mixed-integer program, and establish the consistency of this approximate solution. On the other hand, we improve the existing formulations by replacing the linear "big-M" constraints that represent the relationship between the continuous and binary indicator variables with second-order conic constraints. Our numerical results demonstrate the effectiveness of the proposed approaches.

Keywords: Mixed-integer conic programming, Bayesian networks, directed acyclic graphs, early stopping criterion, consistency.

[*] Department of Industrial Engineering and Management Sciences, Northwestern University (simge@northwestern.edu).

[†] These authors contributed equally to this work.

[‡] Department of Biostatistics, University of Washington (ashojaie@uw.edu).

[§] Department of Industrial and Systems Engineering, University of Washington (hmanzour@uw.edu).

[¶] Department of Industrial Engineering and Management Sciences, Northwestern University (Linchuan-Wei2022@u.northwestern.edu).

[||] Department of Management Science, National Yang Ming Chiao Tung University (hhwu2@nycu.edu.tw).

1 Introduction

A Bayesian network (BN) is a probabilistic graphical model consisting of a labeled directed acyclic graph (DAG) $\mathcal{G} = (V, E)$, in which the vertex set $V = \{V_1, \ldots, V_m\}$ corresponds to m random variables, and the edge set E prescribes a decomposition of the joint probability distribution of the random variables based on their parents in $\mathcal{G}$. The edge set E encodes Markov relations on the nodes in the sense that each node is conditionally independent of its non-descendents given its parents. BNs have been used in knowledge discovery [49, 9], classification [1], feature selection [22], latent variable discovery [29] and genetics [37]. They also play a vital part in causal inference [40].

In this paper, we propose reformulations of the mixed-integer quadratic programs (MIQP) for learning the optimal DAG structure of BNs given n continuous observations from a system of linear structural equation models (SEMs). While there exist exact integer-programming (IP) formulations for learning DAG structure with *discrete* data [11, 12, 27, 50, 4, 34, 35, 5, 13, 14], the development of tailored computational tools for learning the optimal DAG structure from *continuous* data has received less attention. In principle, exact methods developed for discrete data can be applied to continuous data. However, such methods result in exponentially sized formulations in terms of the number of binary variables. A common practice to circumvent the exponential number of binary variables is to limit the in-degree of each node [12, 14, 5]. But, this may result in sub-optimal solutions. On the contrary, MIQP formulations for learning DAGs corresponding to linear SEMs require a *polynomial* number of binary variables. This is because for BNs with linear SEMs, the score function — i.e., the penalized negative log-likelihood (PNL) — can be explicitly written as a function of the coefficients of linear SEMs [45, 52, 38, 32]. In contrast to the existing MIQPs [38, 32], our reformulations exploit the convex quadratic objective and the relationship between the continuous and binary variables to improve the strength of the continuous relaxations.

Continuous BNs with linear SEMs have witnessed a growing interest in the statistics and computer science communities [52, 43, 31, 23, 47]. In particular, it has been shown that the solution obtained from solving the PNL augmented by ℓ_0 regularization, which introduces a penalty on the number of non-zero arc weights in the estimated DAG, achieves desirable statistical properties [41, 52, 31]. Moreover, if the model is *identifiable* [41, 31], that is when the true causal graph can be identified from the joint distribution, then such a solution is guaranteed to uncover the true causal DAG when the sample size n is large enough. However, given the difficulty of obtaining exact solutions, existing approaches for learning DAGs from linear SEMs have primarily relied on *heuristics*, using techniques such as coordinate descent [21, 2, 26] and non-convex continuous optimization [61]. Unfortunately, these heuristics are not guaranteed to achieve the desirable properties of the global optimal solution. Moreover, it is difficult to evaluate the statistical properties of a sub-optimal solution with no optimality guarantees [28]. To bridge this gap, in this paper we develop mathematical formulations for learning optimal BNs from linear SEMs using a PNL objective with ℓ_0 regularization. By connecting the optimality gap of the mixed-integer program to the statistical properties of the solution, we also establish an *early stopping criterion* under which we can terminate the branch-and-bound procedure and attain a solution which asymptotically recovers the true parameters with high probability.

Our work is related to recent efforts to develop exact tailored methods for DAG learning from continuous data. [57] show that A^*-lasso algorithm tailored for DAG structure learning from continuous data with ℓ_1-regularization, which introduces a penalty on the sum of absolute values of the arc weights, is more effective than the previous approaches based on dynamic programming [e.g., 46] that are suitable for both discrete and continuous data. [38] develop a mathematical program for DAG structure learning with ℓ_1 regularization. [32] improve and extend the formulation by [38] for DAG learning from continuous data with both ℓ_0 and ℓ_1 regularizations. The numerical experiments by [32] demonstrate that as the number of nodes grows, their MIQP formulation outperforms A^*-lasso and the existing IP methods; this improvement is both in terms of reducing the IP optimality gap, when the algorithm is stopped due to a time limit, and in terms of computational time, when the instances can be solved to optimality. In light of these recent efforts, the current paper makes important contributions to this problem at the intersection of statistics and optimization.

- The statistical properties of *optimal* PNL with ℓ_0 regularization have been studied extensively [31, 52]. However, it is often difficult to obtain an optimal solution and no results have been established on the statistical properties of approximate solutions. In this paper, we give an early stopping criterion for the branch-and-bound process under which the approximate solution gives consistent estimates of the true coefficients of the linear SEM. Our result leverages the statistical consistency of the PNL estimate with ℓ_0 regularization [52, 41] along with the properties of the branch-and-bound method wherein both lower and upper bound values on the objective function are available at each iteration. By connecting these two properties, we obtain a concrete early stopping criterion, as well as a proof of consistency of the approximate solution. To the best of our knowledge, this result is the first of its kind for DAG learning.

- In spite of recent progress, a key challenge in learning DAGs from linear SEMs is enforcing bounds on arc weights. This is commonly modeled using the standard "big-M constraint" approach [38, 32]. As shown by [32], this strategy leads to poor continuous relaxations for the problem, which in turn results in slow lower bound improvement in the branch-and-bound tree. In particular, [32] establish that all existing big-M formulations achieve the same continuous relaxation objective function under a mild condition (see Proposition 4). To circumvent this issue, we present a mixed-integer second-order cone program (MISOCP), which gives a tighter continuous relaxation than existing big-M formulations under certain conditions discussed in detail in Section 5.3. This formulation can be solved by powerful state-of-the-art optimization packages. Our numerical results show the superior performance of

MISOCP compared to the existing big-M formulations in terms of improving the lower bound and reducing the optimality gap. We also compare our method against the state-of-the-art benchmarks [9, 24] both for identifiable and non-identifiable instances, and show that our method provides the best estimation among all methods in most of the networks, especially for the non-identifiable cases.

The rest of the paper is organized as follows. In Section 2, we define the DAG structure learning problem corresponding to linear SEMs, and give a general framework for the problem. In Section 3, we present our early stopping criterion and establish the asymptotic properties of the solution obtained under this stopping rule. We review existing mathematical formulations in Section 4, and present our proposed mathematical formulations in Section 5. Results of comprehensive numerical studies are presented in Section 6. We end the paper with a summary in Section 7.

2 Problem setup: Penalized DAG estimation with linear SEMs

Let $\mathcal{M} = (V, E)$ be an undirected and possibly cyclic super-structure graph with node set $V = \{1, 2, \ldots, m\}$ and edge set $E \subseteq V \times V$; let $\overrightarrow{\mathcal{M}} = (V, \overrightarrow{E})$ be the corresponding bi-directional graph with $\overrightarrow{E} = \{(j,k),(k,j)|(j,k) \in E\}$. We refer to undirected edges as *edges* and directed edges as *arcs*.

We assume that causal effects of continuous random variables in a DAG $\mathcal{G}_0$ are represented by m linear regressions of the form

$$X_k = \sum_{j \in pa_k^{\mathcal{G}_0}} \beta_{jk} X_j + \epsilon_k, \quad k = 1, \ldots, m, \tag{1}$$

where X_k is the random variable associated with node k, $pa_k^{\mathcal{G}_0}$ represents the parents of node k in $\mathcal{G}_0$, i.e., the set of nodes with arcs pointing to k; the latent random variable ϵ_k denotes the unexplained variation in node k; and BN parameter β_{jk} specifies the effect of node j on k for $j \in pa_k^{\mathcal{G}_0}$. The above model is known as a linear SEM [40].

Let $\mathcal{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_m)$ be the $n \times m$ data matrix with n rows representing i.i.d. samples from each random variable, and m columns representing random variables $X_1, \ldots, X_m$. The linear SEM (1) can be compactly written in matrix form as $\mathcal{X} = \mathcal{X}B + \mathcal{E}$, where $B = [\beta] \in \mathbb{R}^{m \times m}$ is a matrix with $\beta_{kk} = 0$ for $k = 1, \ldots, m$, $\beta_{jk} = 0$ for all $(j,k) \notin E$, and $\mathcal{E}$ is the $n \times m$ 'noise' matrix. Then, $\mathcal{G}(B)$ denotes the directed graph on m nodes such that arc (j,k) appears in $\mathcal{G}(B)$ if and only if $\beta_{jk} \neq 0$. Throughout the paper, we will use B and β to denote the matrix of coefficients and its vectorized version.

A key challenge when estimating DAGs by minimizing the loss function (2) is that the true DAG is generally not identifiable from observational data. However, for certain

SEM distributions, the true DAG is *identifiable* from observational data; that is when the true causal graph can be identified from the joint distribution. Two important examples are linear SEMs with possibly non-Gaussian homoscedastic noise variables [41], as well as linear SEMs with unequal noise variances that are known up to a constant [31]. In these special cases, the true DAG can be identified from observational data, without requiring the (strong) 'faithfulness' assumption, which is known to be restrictive in high dimensions [51, 48]. Given these important implications, in this paper we focus on learning Bayesian networks corresponding to the above *identifiable* linear SEMs, i.e., settings where the error variances are either equal, or known up to a constant.

The negative log likelihood for an identifiable linear SEM (1) with equal noise variances is proportional to

$$l(\beta; \mathcal{X}) = n \operatorname{tr}\left\{(I-B)(I-B)^\top \widehat{\Sigma}\right\}; \tag{2}$$

here $\widehat{\Sigma} = n^{-1}\mathcal{X}^\top\mathcal{X}$ is the empirical covariance matrix, and I is the identity matrix [45, 52].

To learn *sparse* DAGs, van de Geer and Bühlmann [52] propose to augment the negative log likelihood with an ℓ_0 regularization term. Given a super-structure $\mathcal{M}$, the optimization problem corresponding to this penalized negative log-likelihood (PNL$\mathcal{M}$) is given by

$$\textbf{PNL}\mathcal{M} \quad \min_{B \in \mathbb{R}^{m\times m}} \quad \mathcal{L}(\beta) := l(\beta; \mathcal{X}) + \lambda_n \|\beta\|_0 \tag{3a}$$

$$\text{s.t. } \mathcal{G}(B) \text{ induces a DAG from } \overrightarrow{\mathcal{M}}, \tag{3b}$$

where the tuning parameter λ_n controls the degree of the ℓ_0 regularization

$$\|\beta\|_0 := \sum_{(j,k)\in\overrightarrow{E}} \mathbb{1}(\beta_{jk}),$$

where $\mathbb{1}(\beta_{jk})$ is an indicator function with value one if $\beta_{jk} \neq 0$, and 0 otherwise. The constraint (3b) stipulates that the resulting directed subgraph is a DAG induced from $\overrightarrow{\mathcal{M}}$. When $\mathcal{M}$ corresponds to a complete graph, PNL$\mathcal{M}$ reduces to the original PNL of van de Geer and Bühlmann [52].

The choice of ℓ_0 regularization in (3) is deliberate. Although ℓ_1 regularization has attractive computational and statistical properties in high-dimensional regression [8], many of these advantages disappear in the context of DAG structure learning [21, 2]. By considering ℓ_0 regularization, [52] establish the consistency of PNL under appropriate assumptions. More specifically, for a Gaussian SEM, they show that the estimated DAG has (asymptotically) the same number of edges as the DAG with minimal number of edges (minimal-edge I-MAP), and establish the consistency of PNL for learning sparse DAGs. These results are formally stated in Proposition 1 in the next section.

Remark 1. A Tikhonov (ℓ_2) regularization term, $\mu\|\beta\|_2^2$, for a given $\mu > 0$, can also be added to the objective (3a) to obtain more stable solutions [7].

In our earlier work [32], we observe that existing mathematical formulations are slow to converge to a provably optimal solution, $\beta^\star$, of (3) using the state-of-the-art optimization solvers. Therefore, the solution process needs to be terminated early to yield a feasible solution, $\hat{\beta}$ with a positive optimality gap, i.e., a positive difference between the upper bound on $\mathcal{L}(\beta^\star)$ provided by $\mathcal{L}(\hat{\beta})$ and a lower bound on $\mathcal{L}(\beta^\star)$ provided by the best continuous relaxation obtained by the branch-and-bound algorithm upon termination. However, statistical properties of such a sub-optimal solution are not well-understood. Therefore, there exists a gap between theory and computation: while the optimal solution has nice statistical properties, the properties of the solutions obtained from approximate computational algorithms are not known. Moreover, due to the non-convex and complex nature of the problem, characterizing the properties of the solutions provided by heuristics is especially challenging. In the next section, we bridge this gap by developing a concrete early stopping criterion and establishing the consistency of the solution obtained using this criterion.

3 Early stopping criterion for DAG learning

In this section, we establish a sufficient condition for the approximate solution of PNL$\mathcal{M}$, $\hat{\beta}$ to be consistent for the true coefficients, β^0; that is $\|\beta^0 - \hat{\beta}\|_2^2 = \mathcal{O}\left(s^0 \log(m)/n\right)$, where s^0 is the number of arcs in the true DAG, and $x \asymp y$ means that x converges to y asymptotically. This result is obtained by leveraging an important property of the branch-and-bound process for integer programming that provides both lower and upper bounds on the objective function $\mathcal{L}(\beta^\star)$ upon early stopping, as well as the consistency results of the PNL estimate with ℓ_0 regularization. Using the insight from this new result, we then propose a concrete stopping criterion for terminating the branch-and-bound process that results in consistent parameter estimates.

Let LB and UB, respectively, denote the lower and upper bounds on the optimal objective function value (3a) obtained from solving (3) under an early stopping criterion (i.e., when the obtained solution is not necessarily optimal). We define the difference between the upper and lower bounds as the *absolute* optimality gap: GAP = UB − LB. Let $\hat{\mathcal{G}}$ and $\hat{\beta}$ denote the structure of the DAG and coefficients of the arcs from optimization model (3) under the early stopping condition with sample size n and regularization parameter λ_n. Let $\mathcal{G}^\star$ and $\beta^\star$ denote the DAG structure and coefficients of arcs obtained from the optimal solution of (3), and $\mathcal{G}^0$ and β^0 denote the true DAG structure and the coefficient of arcs, respectively. We denote the number of arcs in $\hat{\mathcal{G}}$, $\mathcal{G}^0$, and $\mathcal{G}^\star$ by $\hat{s}$, s^0, and $s^\star$, respectively. The score value in (3a) of each solution is denoted by $\mathcal{L}(\phi)$ where $\phi \in \{\beta^\star, \hat{\beta}, \beta^0\}$.

Next, we present our main result. Our result extends van de Geer and Bühlmann's result on consistency of PNL$\mathcal{M}$ for the optimal, but computationally unattainable, estimator, $\beta^\star$ to an approximate estimator, $\hat{\beta}$, obtained from early stopping. In the following (including the statement of our main result, Proposition 2), we assume that the super-structure $\mathcal{M}$ is known *a priori*. The setting where $\mathcal{M}$ is estimated from data is discussed at the end of the section. We begin by stating the key result from [52] and the required assumptions. Throughout, we consider a Gaussian linear SEM of the form (1). We denote the variance of error terms, ϵ_j, by σ_{jj}^2 and the true covariance matrix of the set of random variables, $(X_1, \ldots, X_m)$ by the $m \times m$ matrix Σ.

Assumption 1. Suppose $m < c_0 n/\log(n)$ for some constant $c_0 > 0$ and for some constant σ_0^2, it holds that $\max_{j=1,\ldots,m} \sigma_{jj}^2 \leq \sigma_0^2$. Moreover, the smallest and largest eigenvalues of Σ, $\kappa_{\min}(\Sigma)$ and $\kappa_{\max}(\Sigma)$, satisfy

$$\left(\frac{c_0}{\log(n)}\right)^{1/2} < \underline{\kappa} \leq \kappa_{\min}(\Sigma) < \kappa_{\max}(\Sigma) \leq \overline{\kappa} < \infty$$

for constants $\underline{\kappa}$ and $\overline{\kappa}$.

Assumption 2. Let, as in [52], $\widetilde{\Omega}(\pi)$ be the precision matrix of the vector of noise variables for an SEM given permutation π of nodes. Denoting the diagonal entries of this matrix by $\tilde{\omega}_{jj}$, there exists a constant $\omega_0 > 0$ such that if $\widetilde{\Omega}(\pi)$ is not a multiple of the identity matrix, then

$$m^{-1} \sum_{j=1}^{m} \left((\tilde{\omega}_{jj})^2 - 1\right)^2 > 1/\omega_0.$$

Proposition 1. (Theorem 5.1 in [52]) Suppose Assumptions 1 and 2 hold and let $\alpha_0 := \min\{\frac{4}{m}, 0.05\}$. Then for an ℓ_0 regularization parameter $\lambda \asymp \log(m)/n$, it holds with probability at least $1 - \alpha_0$ that

$$\|\beta^\star - \beta^0\|_2^2 + \lambda s^\star = \mathcal{O}\left(\lambda s^0\right).$$

Here, $\lambda = \lambda_n/n$, because the loss function (2) is that of [52] scaled by the sample size n. The next result establishes the consistency of the approximate estimator, $\hat{\beta}$, obtained using our proposed early stopping strategy.

Proposition 2. Suppose Assumptions 1 and 2 hold and let $\alpha_0 = \min\{\frac{4}{m}, 0.05\}$ and $\lambda \asymp \log(m)/n$. Then, the estimator $\hat{\beta}$ obtained from early stopping of the branch-and-bound process such that GAP $\asymp \mathcal{O}\left(n\lambda s^0\right) = \mathcal{O}\left(\log(m)s^0\right)$ satisfies

$$\left\|\hat{\beta} - \beta^0\right\|_2^2 \asymp \mathcal{O}\left(\frac{\log(m)}{n} s^0\right)$$

with probability $(1 - \alpha_0)$.

Proof. First, by the triangle inequality and the fact that $2ab \leq a^2 + b^2, \forall a, b \in \mathbb{R}$,

$$\begin{aligned} \|\hat{\beta} - \beta^0\|_2^2 &\leq \left(\|\hat{\beta} - \beta^\star\|_2 + \|\beta^\star - \beta^0\|_2\right)^2 \\ &= \|\hat{\beta} - \beta^\star\|_2^2 + \|\beta^\star - \beta^0\|_2^2 + 2\|\hat{\beta} - \beta^\star\|_2 \|\beta^\star - \beta^0\|_2 \\ &\leq 2\|\hat{\beta} - \beta^\star\|_2^2 + 2\|\beta^\star - \beta^0\|_2^2. \end{aligned} \tag{4}$$

Recall that β denotes the vectorized coefficient matrix B. Then, in a slight abuse of notation, we denote by $\mathcal{X}$ both the vectorized and a block diagonal version of $\mathcal{X}$ and by $\mathcal{E}$ a vectorized version of error $\mathcal{E}$. Then $\ell(\beta; \mathcal{X})$ can be written as $\ell(\beta; \mathcal{X}) = \|\mathcal{X} - \mathcal{X}\beta\|_2^2$ (see Eq. 1). Then, we can write a Taylor series expansion of $\ell\left(\hat{\beta}; \mathcal{X}\right)$ around $\ell\left(\beta^\star; \mathcal{X}\right)$ to get

$$\|\mathcal{X}(\hat{\beta} - \beta^\star)\|_2^2 = \ell(\hat{\beta}; \mathcal{X}) - \ell(\beta^\star; \mathcal{X}) - 2(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{X}(\beta^\star - \beta^0) + 2(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{E}. \tag{5}$$

But, Proposition 2.1 in [53] states that for every $0 < \xi < 1$,

$$\left\|\widehat{\Sigma} - \Sigma\right\|_2 \leq \left(\frac{m}{n}\right)^{1/2}, \tag{6}$$

with probability $1 - \xi$. Thus, letting $\xi = \alpha_0$, (6) holds in our setting with probability $1 - \alpha_0$.

Since, by Assumption 1, $\kappa_{\min}(\Sigma) \geq \underline{\kappa} > \frac{c_0}{\log(n)}$, by Weyl's theorem we have

$$\kappa_{\min}(\mathcal{X}^\top \mathcal{X}) = n\kappa_{\min}\left(\widehat{\Sigma}\right) > n\underline{\kappa} - (mn)^{1/2} > n\underline{\kappa} - n\left(\frac{c_0}{\log(n)}\right)^{1/2} = n\left(\underline{\kappa} - \left(\frac{c_0}{\log(n)}\right)^{1/2}\right),$$

which means that $\kappa_{\min}(\mathcal{X}^\top \mathcal{X}) > 0$ with probability $1 - \alpha_0$.

Denoting $c_n' \equiv \left(\underline{\kappa} - \left(\frac{c_0}{\log(n)}\right)^{1/2}\right)^{-1}$, for large enough n, we have that, with probability $1 - \alpha_0$,

$$\|\hat{\beta} - \beta^\star\|_2^2 \leq n^{-1}c_n'\|\mathcal{X}(\hat{\beta} - \beta^\star)\|_2^2. \tag{7}$$

Combining (5) and (7), and using triangle inequality again, we get

$$\begin{aligned}
\|\hat{\beta} - \beta^\star\|_2^2 &\leq n^{-1}c_n'\|\mathcal{X}(\hat{\beta} - \beta^\star)\|_2^2 \\
&= n^{-1}c_n'\left(\ell(\hat{\beta}; \mathcal{X}) - \ell(\beta^\star; \mathcal{X}) - 2(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{X}(\beta^\star - \beta^0) + 2(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{E}\right) \\
&\leq n^{-1}c_n'\left|\ell(\hat{\beta}; \mathcal{X}) - \ell(\beta^\star; \mathcal{X}) - 2(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{X}(\beta^\star - \beta^0) + 2(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{E}\right| \\
&\leq n^{-1}c_n'\left|\ell(\hat{\beta}; \mathcal{X}) - \ell(\beta^\star; \mathcal{X})\right| + 2n^{-1}c_n'(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{X}(\beta^\star - \beta^0) + 2n^{-1}c_n'\left|(\hat{\beta} - \beta^\star)^\top \mathcal{X}^\top \mathcal{E}\right| \\
&\leq n^{-1}c_n'\left|\ell(\hat{\beta}; \mathcal{X}) - \ell(\beta^\star; \mathcal{X})\right| + 2n^{-1}c_n'\kappa_{\max}(\mathcal{X}^\top \mathcal{X})\|\hat{\beta} - \beta^\star\|_2\|\beta^\star - \beta^0\|_2 + 2n^{-1}c_n'\|\hat{\beta} - \beta^\star\|_2\|\mathcal{X}^\top \mathcal{E}\|_2,
\end{aligned} \tag{8}$$

where, as before, $\kappa_{\max}$ denotes the maximum eigenvalue of the matrix.

Using a similar argument as the one used above for the minimum eigenvalue of $\mathcal{X}^\top \mathcal{X}$, by (6) we have that, with probability $1 - \alpha_0$,

$$\kappa_{\max}(\mathcal{X}^\top \mathcal{X}) = n\kappa_{\max}\left(\widehat{\Sigma}\right) \leq n\kappa_{\max}(\Sigma) + n\left(\frac{c_0}{\log(n)}\right)^{1/2} \leq n\left(\overline{\kappa} + \left(\frac{c_0}{\log(n)}\right)^{1/2}\right).$$

Plugging the above bound into (8) we get

$$\begin{aligned}\|\hat{\beta}-\beta^{\star}\|_2^2 \leq{}& n^{-1}c'_n\left|\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})\right| \\ &+2c'_n\left(\overline{\kappa}+\left(\frac{c_0}{\log(n)}\right)^{1/2}\right)\|\hat{\beta}-\beta^{\star}\|_2\|\beta^{\star}-\beta^0\|_2+2n^{-1}c'_n\|\hat{\beta}-\beta^{\star}\|_2\|\mathcal{X}^{\top}\mathcal{E}\|_2.\end{aligned}\tag{9}$$

Now, let $Z=\left\|\hat{\beta}-\beta^{\star}\right\|_2$, $\Pi=2c'_n\left[\left(\overline{\kappa}+\left(\frac{c_0}{\log(n)}\right)^{1/2}\right)\|\beta^{\star}-\beta^0\|_2+n^{-1}\|\mathcal{X}^{\top}\mathcal{E}\|_2\right]$, and $\Gamma=n^{-1}c'_n\left|\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})\right|$. Then, the inequality in (9) can be written as $Z^2\leq\Pi Z+\Gamma$. Solving for Z and noting that Z, Γ and Π are non-negative, in order to have $Z^2\leq\Pi Z+\Gamma$, we must have $Z\leq\left(\Pi+\sqrt{\Pi^2+4\Gamma}\right)/2$.

Next, let $\mathcal{T}$ be the event under which $\Pi=o(1)$. Then, on this set, we have $Z\leq\left(o(1)+\sqrt{o(1)+4\Gamma}\right)/2$, or, $Z^2\leq\Gamma+o(1)$; that is

$$\left\|\hat{\beta}-\beta^{\star}\right\|_2^2\leq n^{-1}c'_n\left|\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})\right|+o(1).\tag{10}$$

Plugging (10) into (4), on the set $\mathcal{T}$ we have

$$\begin{aligned}\|\hat{\beta}-\beta^0\|_2^2&\leq 2n^{-1}c'_n\left|\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})\right|+2\|\beta^{\star}-\beta^0\|_2^2+o(1)\\ &=2n^{-1}c'_n\left|\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})+(\lambda\hat{s}-\lambda s^{\star})-(\lambda\hat{s}-\lambda s^{\star})\right|+2\|\beta^{\star}-\beta^0\|_2^2+o(1)\\ &\leq 2n^{-1}c'_n\left|\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})+\lambda\hat{s}-\lambda s^{\star}\right|+2n^{-1}c'_n\left|\lambda\hat{s}-\lambda s^{\star}\right|+2\|\beta^{\star}-\beta^0\|_2^2+o(1).\end{aligned}\tag{11}$$

Then, using the fact that $\ell(\hat{\beta};\mathcal{X})-\ell(\beta^{\star};\mathcal{X})+\lambda\hat{s}-\lambda s^{\star}=\mathcal{L}(\hat{\beta})-\mathcal{L}(\beta^{\star})\leq\mathrm{GAP}$ we can write (11) as

$$\begin{aligned}\|\hat{\beta}-\beta^0\|_2^2&\leq 2n^{-1}c'_n\left|\mathcal{L}(\hat{\beta};\mathcal{X})-\mathcal{L}(\beta^{\star};\mathcal{X})\right|+2n^{-1}c'_n\lambda s^{\star}+2\|\beta^{\star}-\beta^0\|_2^2+o(1),\\ &\leq 2n^{-1}c'_n\mathrm{GAP}+2\|\beta^{\star}-\beta^0\|_2^2+2n^{-1}c'_n\lambda s^{\star}+o(1),\end{aligned}\tag{12}$$

where, in the first inequality, we also use the fact that the penalized estimation procedure chooses at most n nonzero entries. Hence, since $m<n$ by Assumption 1,

$$2n^{-1}c'_n\left|\lambda\hat{s}-\lambda s^{\star}\right|\leq 2n^{-1}c'_n\lambda s^{\star}+2c'_n\lambda=2n^{-1}c'_n\lambda s^{\star}+\mathcal{O}\left(\frac{\log(m)}{n}\right)=2n^{-1}c'_n\lambda s^{\star}+o(1).$$

Now, by Proposition 1, we know that with probability at least $1-\alpha_0$, $\|\beta^{\star}-\beta^0\|_2^2=\mathcal{O}\left(s^0\log(m)/n\right)$, and $\lambda s^{\star}=\mathcal{O}\left(s^0\log(m)/n\right)$. Moreover, using Gaussian concentration inequalities for $n^{-1}\|\mathcal{X}^{\top}\mathcal{E}\|_2$ [e.g., Lemma 6.2 in 8], the probability of the set $\mathcal{T}$ is lower bounded by the probability that $\|\beta^\star - \beta^0\|_2^2 = \mathcal{O}\left(s^0 \log(m)/n\right)$, which is $1-\alpha_0$. Thus, if we stop the branch-and-bound algorithm when

$$\text{GAP} = \mathcal{O}\left(n\lambda s^0\right) = \mathcal{O}\left(\log(m) s^0\right),$$

then the first two terms in (12) would both be of order $\mathcal{O}\left(s^0 \log(m)/n\right)$, while the third term, $2n^{-1}c_n'\lambda s^\star$ would be of a smaller order (by an n^{-1} factor). This guarantees that, with probability at least $(1-\alpha_0)$, $\|\hat{\beta} - \beta^0\|_2^2 = \mathcal{O}\left(s^0 \log(m)/n\right)$, as desired. □

Proposition 2 suggests that the branch-and-bound algorithm can be stopped by setting a threshold $c^\star n\lambda s^0$ on the value of GAP $= |\text{UB} - \text{LB}|$ for a constant $c^\star > 0$, say $c^\star = 1$. Such a solution will then achieve the same desirable statistical properties (in terms of parameter consistency) as the optimal solution $\beta^\star$. While λ can be chosen data-adaptively (as discussed in Section 6), both of these choices depend on the value of s^0, which is not known in practice. However, one can find an upper bound for s^0 based on the number of edges in the super-structure $\mathcal{M}$. In particular, if $\mathcal{M}$ is the moral graph [40] with s_m edges, then $s^0 \leq s_m$. However, while in many applications $s_m \asymp s^0$, this is not always guaranteed. Thus, to ensure consistent estimation when replacing s^0 with s_m and setting $c^\star = 1$ in practice, we use the more conservative threshold of $\lambda s^0 \asymp s^0 \log(m)/n$. With this choice the first and third terms in (12) would be of the same (vanishing) order, and the consistency rate would be driven by the convergence rate of $\|\beta^\star - \beta^0\|_2^2$. We investigate the performance of this choice in Section 6.4.

The above results, including the specific choice of early stopping criterion, are also valid if the super-structure $\mathcal{M}$ corresponding to the moral graph is not known *a priori*. That is because the moral graph can be consistently estimated from data using recent developments in graphical modeling; see Drton and Maathuis [16] for a review of the literature. While some of the existing algorithms based on ℓ_1-penalty require an additional *irrepresentability* condition [33, 44], this assumption can be relaxed by using instead an adaptive lasso penalty or by thresholding the initial lasso estimates [8].

In light of Proposition 2, it is of great interest to develop algorithms that converge to a solution with a small optimality gap expeditiously. To achieve this, one approach is to obtain better lower bounds using the branch-and-bound process from strong mathematical formulations for (3). To this end, we next review existing formulations of (3).

4 Existing Formulations of DAG Learning with Linear SEMs

In this section, we review known mathematical formulations for DAG learning with linear SEMs. We first outline the necessary notation below.

Index Sets
$V = \{1, 2, \dots, m\}$: index set of random variables;

$\mathcal{D} = \{1, 2, \ldots, n\}$: index set of samples.

Input
$\mathcal{M} = (V, E)$: an undirected super-structure graph (e.g., the moral graph);
$\overrightarrow{\mathcal{M}} = (V, \overrightarrow{E})$: the bi-directional graph corresponding to the undirected graph $\mathcal{M}$;
$\mathcal{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_m)$, where $\mathcal{X}_v = (x_{1v}, x_{2v}, \ldots, x_{nv})^\top$ and x_{dv} denotes dth sample ($d \in \mathcal{D}$) of random variable X_v; note $\mathcal{X} \in \mathbb{R}^{n \times m}$;
λ_n : tuning parameter (penalty coefficient for ℓ_0 regularization).

Continuous optimization variables
β_{jk}: weight of arc (j, k) representing the regression coefficients $\forall (j, k) \in \overrightarrow{E}$.

Binary optimization variables
$z_{jk} = 1$ if arc (j, k) exists in a DAG; otherwise 0, $\forall (j, k) \in \overrightarrow{E}$,
$g_{jk} = 1$ if $\beta_{jk} \neq 0$; otherwise 0, $\forall (j, k) \in \overrightarrow{E}$.

Let $F(\beta, g) = \sum_{k \in V} \sum_{d \in \mathcal{D}} \left(x_{dk} - \sum_{(j,k) \in \overrightarrow{E}} \beta_{jk} x_{dj} \right)^2 + \lambda_n \sum_{(j,k) \in \overrightarrow{E}} g_{jk}$. The PNL$\mathcal{M}$ problem can be cast as the following optimization model:

$$\min_{B \in \mathbb{R}^{m \times m}, g \in \{0,1\}^{|\overrightarrow{E}|}} \quad F(\beta, g), \tag{13a}$$

$$\mathcal{G}(B) \text{ induces a DAG from } \overrightarrow{\mathcal{M}}, \tag{13b}$$

$$\beta_{jk}(1 - g_{jk}) = 0, \qquad \forall (j, k) \in \overrightarrow{E}, \tag{13c}$$

$$g_{jk} \in \{0, 1\}, \qquad \forall (j, k) \in \overrightarrow{E}. \tag{13d}$$

The objective function (13a) is an expanded version of $\mathcal{L}(\beta)$ in PNL$\mathcal{M}$, where we use the indicator variable g_{jk} to encode the ℓ_0 regularization. The constraints in (13b) rule out cycles. The constraints in (13c) are non-linear and stipulate that $\beta_{jk} \neq 0$ only if $g_{jk} = 1$.

There are two sources of difficulty in solving (13a)-(13d): (i) the acyclic nature of DAG imposed by the combinatorial constraints in (13b); (ii) the set of *nonlinear* constraints in (13c), which stipulates that $\beta_{jk} \neq 0$ only if there exists an arc (j, k) in $\mathcal{G}(B)$. In Section 4.1, we discuss related studies to address the former, whereas in Section 4.2 we present relevant literature for the latter.

4.1 Linear encodings of the acyclicity constraints (13b)

There are several ways to ensure that the estimated graph does not contain any cycles. The first approach is to add a constraint for each cycle in the graph, so that at least one arc in this cycle must not exist in $\mathcal{G}(B)$. A *cutting plane* (CP) method is used to solve such a formulation which may require generating an exponential number of constraints.

Another way to rule out cycles is by imposing constraints such that the nodes follow a topological order [38]. A topological ordering is a unique ordering of the nodes of a graph from 1 to m such that the graph contains an arc (j,k) if node j appears before node k in the order. We refer to this formulation as *topological ordering* (TO). The TO formulation has $\mathcal{O}(m^2)$ variables and $\mathcal{O}(|\overrightarrow{E}|)$ constraints. We give these formulations in the Appendix, for completeness.

The *layered network* (LN) formulation for learning from continuous data proposed by [32] is shown to perform better, empirically, than the TO formulation because it reduces the number of binary variables and is proven to obtain the same continuous relaxation bounds. Therefore, smaller quadratic programs are solved that provide the same relaxation bounds as larger quadratic programs. This formulation is closely related to the generation number approach proposed in [11] for discrete data. In this paper, we focus on the LN formulation and refer the reader to the Appendix and [32] for comparisons of these formulations and their sizes in detail. Next, we give the LN encoding of the acyclicity constraints. Define decision variables $z_{jk} \in \{0,1\}$ for all $(j,k) \in \overrightarrow{E}$, where the variable z_{jk} takes value 1 if there is an arc (j,k) in the network

$$\textbf{LN} \quad g_{jk} \leq z_{jk} \qquad \forall (j,k) \in \overrightarrow{E}, \tag{14a}$$

$$z_{jk} - (m-1) z_{kj} \leq \psi_k - \psi_j \quad \forall (j,k) \in \overrightarrow{E}. \tag{14b}$$

Let ψ_k be the *layer value* for node k. The set of constraints in (14b) ensures that if the layer of node j appears before that of node k (i.e., there is a direct path from node j to node k), then $\psi_k \geq \psi_j + 1$. This rules out any cycles.

Constraint (14b) written for $(k,i) \in \overrightarrow{E}$ imposes that if $z_{ij} = 1$ and $z_{jk} = 1$, i.e., if $\psi_k > \psi_i$, then $z_{ik} = 1$, even if $\beta_{ik} = g_{ik} = 0$ in an optimal solution. Thus, additional binary vector z along with the set of constraints in (14a) is needed to correctly encode the ℓ_0 regularization. The LN formulation has $\mathcal{O}(|\overrightarrow{E}|)$ variables and constraints. Note that $|\overrightarrow{E}|$ is much smaller than m^2 for sparse skeleton/moral graphs.

4.2 Linear encodings of the non-convex constraints (13c)

The nonconvexity of the set of constraints in (13c) causes challenges in obtaining provably optimal solutions with existing optimization software. Therefore, we consider convex representations of this set of constraints. First, we present a linear encoding of the constraints in (13c). Although the existing compact (i.e., polynomial sized) TO and LN formulations discussed in Section 4.1 differ in their approach to ruling out cycles, one commonality among them is that they replace the non-linear constraint (13c) by so called *big-M constraints* given by

$$-M g_{jk} \leq \beta_{jk} \leq M g_{jk}, \forall (j,k) \in \overrightarrow{E}, \tag{15}$$

for a large enough M. Unfortunately, these big-M constraints (15) are poor approximations of (13c), especially in this problem, because no natural and tight value for M exists.

Although a few techniques have been proposed for obtaining the big-M parameter for sparse regression problem [e.g., 7, 6, 25, 39], the resulting parameters are often too large in practice. Further, finding a tight big-M parameter itself is a difficult problem to solve for DAG structure learning.

Consider (13a)-(13d) by replacing (13c) with the linear big-M constraints (15) and writing the objective function in a matrix form. We denote the resulting formulation, which has a convex quadratic objective and linear constraints, by the following MIQP.

$$\textbf{MIQP} \quad \min_{B\in\mathbb{R}^{m\times m}, g\in\{0,1\}^{|\overrightarrow{E}|}} \quad \mathrm{tr}\left[(I-B)(I-B)^\top \mathcal{X}^\top \mathcal{X}\right] + \lambda_n \sum_{(j,k)\in\overrightarrow{E}} g_{jk} \tag{16a}$$

$$(13\text{b}), (15) \tag{16b}$$

$$g_{jk} \in \{0,1\} \quad \forall (j,k) \in \overrightarrow{E}. \tag{16c}$$

Depending on which types of constraints are used in lieu of (13b), as explained in Section 4.1, MIQP (16) results in three different formulations: MIQP+CP, which uses (23), MIQP+TO, which uses (24), and MIQP+LN, which uses (14), respectively.

To discuss the challenges of the big-M approach, we give a definition followed by two propositions.

Definition 1. A formulation A is said to be *stronger* than formulation B if $\mathcal{R}(A) \subset \mathcal{R}(B)$ where $\mathcal{R}(A)$ and $\mathcal{R}(B)$ correspond to the feasible regions of continuous relaxations of A and B, respectively.

Proposition 3. (Proposition 3 in [32]) *The MIQP+TO and MIQP+CP formulations are stronger than the MIQP+LN formulation.*

As a consequence of Definition 1, the optimal objective function value of the continuous relaxation of the stronger formulation provides a lower bound on the true optimal objective function of the MIQP that is greater than or equal to the optimal objective function value of the continuous relaxation of the weaker formulation due to the smaller set of feasible solutions. However, next proposition shows that, perhaps surprisingly, the continuous relaxations of MIQP+TO and MIQP+CP formulations, while stronger according to Definition 1, give the same optimal objective function value (and the same lower bound on the true optimal objective).

Proposition 4. (Proposition 5 in [32]) *Let $\beta^\star_{jk}$ denote the optimal coefficient associated with an arc $(j,k) \in \overrightarrow{E}$ from problem (3). For the same variable branching in the branch-and-bound process, the continuous relaxations of the MIQP+LN formulation for ℓ_0 regularization attain the same optimal objective function value as MIQP+TO and MIQP+CP, if $M \geq 2 \max_{(j,k)\in\overrightarrow{E}} |\beta^\star_{jk}|$.*

Proposition 3 implies that the MIQP+TO and MIQP+CP formulations are stronger than the MIQP+LN formulation. Nonetheless, Proposition 4 establishes that for sufficiently large values of M, stronger formulations for the DAG learning problem attain the same continuous relaxation objective function value as the weaker formulation throughout the branch-and-bound tree. The optimal solution to the continuous relaxation of MIQP formulations of DAG structure learning may not be at an extreme point of the convex hull of feasible points. Hence, stronger formulations do not necessarily ensure better lower bounds for certain formulations of this problem involving the *nonlinear* objective. This is in contrast to a mixed-integer program (MIP) with *linear* objective, whose continuous relaxation is a linear program (LP). In that case, there exists an optimal solution that is an extreme point of the corresponding feasible set. As a result, a better lower bound can be obtained from a stronger formulation that better approximates the convex hull of the set of solutions to a mixed-integer linear program; this generally leads to faster convergence. A prime example is the traveling salesman problem (TSP), for which stronger formulations attain better computational performance [36]. In contrast, the numerical results by [32] empirically show that MIQP+LN has better computational performance because it is a compact formulation with the fewest constraints and the same continuous relaxation bounds.

Our next result, which is adapted from [15] to the DAG structure learning problem, shows that the continuous relaxation of MIQP is equivalent to the optimal solution to the problem where the ℓ_0-regularization term is replaced with an ℓ_1-regularization term (i.e., $\|\beta\|_1 = \sum_{(j,k)\in \overrightarrow{E}} |\beta_{jk}|$) with a particular choice of the ℓ_1 penalty. This motivates us to consider tighter continuous relaxation for MIQP. Let (β^R, g^R) be an optimal solution to the continuous relaxation of MIQP.

Proposition 5. For $M \geq 2 \max\limits_{(j,k)\in \overrightarrow{E}} |\beta^R_{jk}|$, a continuous relaxation of MIQP (16), where the binary variables are relaxed, is equivalent to the problem where the ℓ_0 regularization term is replaced with an ℓ_1-regularization term with penalty parameter $\tilde{\lambda} = \frac{\lambda_n}{M}$.

Proof. For $M \geq 2 \max\limits_{(j,k)\in \overrightarrow{E}} |\beta^R_{jk}|$, the value g^R_{jk} is $\frac{\beta^R_{jk}}{M}$ in an optimal solution to the continuous relaxation of MIQP (16). Otherwise, we can reduce the value of the decision variable g^R without violating any constraints while reducing the objective function. Note that since $M \geq 2 \max\limits_{(j,k)\in \overrightarrow{E}} |\beta^R_{jk}|$, we have $\frac{\beta^R_{jk}}{M} \leq 1$, $\forall (j,k) \in \overrightarrow{E}$. To show that the set of constraints in (13b) is satisfied, we consider the set of CP constraints. In this case, the set of constraints (13b) holds, i.e., $\sum_{(j,k)\in \mathcal{C}_A} \frac{\beta^R_{jk}}{M} \leq |\mathcal{C}_A| - 1, \quad \forall \mathcal{C}_A \in \mathcal{C}$, because $M \geq 2 \max\limits_{(j,k)\in \overrightarrow{E}} |\beta^R_{jk}|$. This implies that $g^R_{jk} = \frac{\beta^R_{jk}}{M}$ is the optimal solution. Thus, the objective function reduces to ℓ_1 regularization with the coefficient $\frac{\lambda_n}{M}$.

Finally, Proposition 4 establishes that for $M \geq 2 \max_{(j,k)\in \overrightarrow{E}} |\beta^{\star}_{jk}|$, the objective function value of the continuous relaxations of MIQP+CP, MIQP+LN and MIQP+TO are equivalent. This implies that the continuous relaxations of all formulations are equivalent, which completes the proof. □

Despite the promising performance of MIQP+LN, its continuous relaxation objective function value provides a weak lower bound due to the big-M constraints. To circumvent this issue, a natural strategy is to improve the big-M value. Nonetheless, existing methods which ensure a valid big-M value or heuristic techniques [38, 25] do not lead to tight big-M values. For instance, the heuristic technique by [38] to obtain big-M values always satisfies the condition in Proposition 5 and exact techniques are expected to produce even larger big-M values. Therefore, we directly develop tighter approximations for (13c) next.

5 New Perspectives for Mathematical Formulations of DAG Learning

In this section, we discuss improved mathematical formulations for learning DAG structure of a BN based on convex (instead of linear) encodings of the constraints in (13c).

Problem (13) is an MIQP with non-convex complementarity constraints (13c), a class of problems which has received a fair amount of attention from the operations research community over the last decade [17, 18, 19, 20, 25, 30, 55, 56]. There has also been recent interest in leveraging these developments to solve sparse regression problems with ℓ_0 regularization [42, 15, 58, 3, 54].

Next, we review applications of MIQPs with complementarity constraints of the form (13c) for solving sparse regression with ℓ_0 regularization. [20] develop a so-called projected perspective relaxation method, to solve the perspective relaxation of mixed-integer nonlinear programming problems with a convex objective function and complementarity constraints. This reformulation requires that the corresponding binary variables are not involved in other constraints. Therefore, it is suitable for ℓ_0 sparse regression, but cannot be applied for DAG structure learning. [42] show how a broad class of ℓ_0-regularized problems, including sparse regression as a special case, can be formulated exactly as optimization problems. The authors use the Tikhonov regularization term $\mu\|\beta\|_2^2$ and convex analysis to construct an improved convex relaxation using the reverse Huber penalty. In a similar vein, [6] exploit the Tikhonov regularization and develop an efficient algorithm by reformulating the sparse regression mathematical formulation as a saddle-point optimization problem with an outer linear integer optimization problem and an inner dual quadratic optimization problem which is capable of solving high-dimensional sparse regressions. [58] apply the perspective formulation of sparse regression optimization problem with both ℓ_0 and the Tikhonov regularizations. The authors establish that the continuous relaxation of the perspective formulation is equivalent to the continuous relaxation of the formulation given by [6]. [15] propose perspective relaxation for ℓ_0 sparse regression optimization formulation and establish that the popular sparsity-inducing concave penalty function known as the minimax concave penalty [59] and the reverse Huber penalty [42] can be obtained as special cases of the perspective relaxation – thus the relaxations of formulations by [59, 42, 6, 58] are equivalent. The authors obtain an optimal perspective relaxation that is no weaker than any perspective relaxation. Among the related approaches, the optimal perspective relaxation by [15] is the only one that does not explicitly require the use of Tikhonov regularization.

The perspective formulation, which in essence is a fractional non-linear program, can be cast either as a mixed-integer second-order cone program (MISOCP) or a semi-infinite mixed-integer linear program (SIMILP). Both formulations can be solved directly by state-of-the-art optimization packages. [15] and [3] solve the continuous relaxations and then use a heuristic approach (e.g., rounding techniques) to obtain a feasible solution (an upper bound). In this paper, we directly solve the MISOCP and SIMILP formulations for learning sparse DAG structures.

Next, we present how perspective formulation can be suitably applied for DAG structure learning with ℓ_0 regularization. We further cast the problem as MISOCP and SIMILP. To this end, we express the objective function (16a) in the following way:

$$\begin{aligned} &\operatorname{tr}[(I-B)(I-B)^\top \mathcal{X}^\top \mathcal{X}] + \lambda_n \sum_{(j,k)\in \overrightarrow{E}} g_{jk} \\ &= \operatorname{tr}[(I-B-B^\top)\mathcal{X}^\top\mathcal{X} + 2BB^\top \mathcal{X}^\top \mathcal{X}] + \lambda_n \sum_{(j,k)\in \overrightarrow{E}} g_{jk}. \end{aligned} \tag{17}$$

Let $\delta \in \mathbb{R}^m_+$ be a vector such that $\mathcal{X}^\top\mathcal{X} - D_\delta \succeq 0$, where $D_\delta = \operatorname{diag}(\delta_1,\ldots,\delta_m)$ and $A \succeq 0$ means that matrix A is positive semi-definite. By splitting the quadratic term $\mathcal{X}^\top\mathcal{X} = (\mathcal{X}^\top\mathcal{X} - D_\delta) + D_\delta$ in (17), the objective function can be expressed as

$$\operatorname{tr}\left[(I-B-B^\top)\mathcal{X}^\top\mathcal{X} + BB^\top(\mathcal{X}^\top\mathcal{X} - D_\delta)\right] + \operatorname{tr}\left(BB^\top D_\delta\right) + \lambda_n \sum_{(j,k)\in \overrightarrow{E}} g_{jk}. \tag{18}$$

Let $Q = \mathcal{X}^\top\mathcal{X} - D_\delta$. (In the presence of Tikhonov regularization with tuning parameter $\mu > 0$, we let $Q = \mathcal{X}^\top\mathcal{X} + \mu I - D_\delta$ as described in Remark 1.) Then, Cholesky decomposition can be applied to decompose Q as $q^\top q$ (note $Q \succeq 0$). As a result, $\operatorname{tr}\left(BB^\top Q\right) = \operatorname{tr}\left(BB^\top q^\top q\right) = \sum_{i=1}^m \sum_{j=1}^m \left(\sum_{(\ell,j)\in\overrightarrow{E}} \beta_{\ell j} q_{i\ell}\right)^2$. The separable component can also be expressed as $\operatorname{tr}\left(BB^\top D_\delta\right) = \sum_{j=1}^m \sum_{(j,k)\in\overrightarrow{E}} \delta_j \beta_{jk}^2$. Using this notation, the objective (18) can be written as

$$\operatorname{tr}\left[(I-B-B^\top)\mathcal{X}^\top\mathcal{X} + BB^\top Q\right] + \sum_{j=1}^m \sum_{(j,k)\in\overrightarrow{E}} \delta_j\beta_{jk}^2 + \lambda_n \sum_{(j,k)\in\overrightarrow{E}} g_{jk}.$$

The Perspective Reformulation (PRef) of MIQP is then given by

$$\textbf{PRef} \quad \min_{B\in\mathbb{R}^{m\times m}, g\in\{0,1\}^{|\overrightarrow{E}|}} \quad \operatorname{tr}\big[(I-B-B^\top)\mathcal{X}^\top\mathcal{X}+BB^\top Q\big]+ \tag{19a}$$

$$\sum_{j=1}^{m}\sum_{(j,k)\in\overrightarrow{E}}\delta_j\frac{\beta_{jk}^2}{g_{jk}}+\lambda_n\sum_{(j,k)\in\overrightarrow{E}}g_{jk},$$

$$(16b)-(16c). \tag{19b}$$

The objective function (19a) is formally undefined when some $g_{jk}=0$. More precisely, we use the convention that $\frac{\beta_{jk}^2}{g_{jk}}=0$ when $\beta_{jk}=g_{jk}=0$ and $\frac{\beta_{jk}^2}{g_{jk}}=+\infty$ when $\beta_{jk}\neq 0$ and $g_{jk}=0$ [19]. The continuous relaxation of PRef, referred to as the perspective relaxation, is much stronger than the continuous relaxation of MIQP under certain conditions discussed in detail in Section 5.3 [42]. However, an issue with PRef is that the objective function is nonlinear due to the fractional term. There are two ways to reformulate PRef. One as a mixed-integer second-order conic program (MISOCP) (see, Section 5.1) and the other as a semi-infinite mixed-integer linear program (SIMILP) (see, Section 5.2).

5.1 Mixed-integer second-order conic program

Let s_{jk} be additional variables representing β_{jk}^2. Then, the MISOCP formulation is given by

$$\textbf{MISOCP} \quad \min_{B\in\mathbb{R}^{m\times m}, s\in\mathbb{R}^{|\overrightarrow{E}|}, g\in\{0,1\}^{|\overrightarrow{E}|}} \quad \operatorname{tr}\left[(I-B-B^\top)\mathcal{X}^\top\mathcal{X}+BB^\top Q\right]+ \tag{20a}$$

$$\sum_{j=1}^{m}\sum_{(j,k)\in\overrightarrow{E}}\delta_j s_{jk}+\lambda_n\sum_{(j,k)\in\overrightarrow{E}}g_{jk},$$

$$s_{jk}g_{jk}\geq\beta_{jk}^2 \quad (j,k)\in\overrightarrow{E}, \tag{20b}$$

$$0\leq s_{jk}\leq M^2 g_{jk} \quad (j,k)\in\overrightarrow{E}, \tag{20c}$$

$$(16b)-(16c). \tag{20d}$$

Here, the constraints in (20b) imply that $\beta_{jk}\neq 0$ only when $g_{jk}=1$. The constraints in (20b) are second-order conic representable because they can be written in the form of $\sqrt{4\beta_{jk}^2+(s_{jk}-g_{jk})^2}\leq s_{jk}+g_{jk}$. The set of constraints in (20c) is valid since $\beta_{jk}\leq Mg_{jk}$ implies $\beta_{jk}^2\leq M^2g_{jk}^2=M^2g_{jk}^2$ and $g_{jk}^2=g_{jk}$ for $g_{jk}\in\{0,1\}$. The set of constraints in (20c) is not required, yet they improve the computational efficiency especially when we restrict the big-M value. [58] report similar behavior for sparse regression. When we relax $g_{jk}\in\{0,1\}$ and let $g_{jk}\in[0,1]$, we obtain the continuous relaxation of MISOCP (20). Let us denote the feasible region of continuous relaxation of MISOCP (20) and MIQP (16) by $\mathcal{R}$MISOCP and $\mathcal{R}$MIQP, and the objective function values by OFV($\mathcal{R}$MISOCP) and OFV($\mathcal{R}$MIQP), respectively. For a more general problem than ours, [10] give a detailed proof establishing that the feasible region of the former is contained in the feasible region of latter i.e., $\mathcal{R}\text{MISOCP} \subset \mathcal{R}MIQP$, and OFV($\mathcal{R}$MISOCP) $\geq$ OFV($\mathcal{R}$MIQP). Therefore, we are able to obtain stronger lower bounds using MISOCP than MIQP under suitable choices for D_δ as described in Section 5.3.

5.2 Mixed-integer semi-infinite integer linear program

An alternative approach to reformulate PRef is via *perspective cuts* developed by [17, 18]. To apply perspective cuts, we use the reformulation idea first proposed in [17] by introducing dummy decision matrix D to distinguish the separable and non-separable part of the objective function; we also add the additional constraint $d = \beta$ where d_{jk} is (j,k) element of matrix D and β is the decision variable in the optimization problem. Following this approach, MIQP can be reformulated as an SIMILP:

$$\textbf{SIMILP} \quad \min_{B\in\mathbb{R}^{m\times m}, v\in\mathbb{R}^{|\overrightarrow{E}|}, g\in\{0,1\}^{|\overrightarrow{E}|}} \quad \operatorname{tr}\Big[(I - B - B^\top)\mathcal{X}^\top\mathcal{X} + DD^\top Q\Big] + \tag{21a}$$

$$\sum_{j=1}^{m} \sum_{(j,k)\in\overrightarrow{E}} \delta_j v_{jk} + \lambda_n \sum_{(j,k)\in\overrightarrow{E}} g_{jk},$$

$$d_{jk} = \beta_{jk} \quad (j,k)\in\overrightarrow{E}, \tag{21b}$$

$$v_{jk} \geq 2\bar{\beta}_{jk}\beta_{jk} - \bar{\beta}_{jk}^2 g_{jk} \quad \forall\bar{\beta}_{jk}\in[-M,M] \quad \forall(j,k)\in\overrightarrow{E}, \tag{21c}$$

$$(16b) - (16c), \tag{21d}$$

$$v_{jk} \geq 0, \quad (j,k)\in\overrightarrow{E}. \tag{21e}$$

The set of constraints in (21c) is known as perspective cuts. Note that there are infinitely many such constraints. Although this problem cannot be solved directly, it lends itself to a delayed cut generation approach whereby a (small) finite subset of constraints in (21c) is kept, the current solution $(\beta^\star, g^\star, v^\star)$ of the relaxation is obtained, and all the violated inequalities for the relaxation solution are added for $\bar{\beta}_{jk} = \frac{\beta^\star_{jk}}{g^\star_{jk}}$ (assuming $\frac{0}{0} = 0$). This process is repeated until termination criteria are met. This procedure can be implemented using the cut callback function available by off-the-shelf solvers such as Gurobi or CPLEX.

5.3 Selecting δ

In the MISOCP and SIMILP formulations, one important question is how to identify a valid δ. A natural choice is $\operatorname{diag}(\delta) = \lambda_{\min} e$, where $\lambda_{\min}$ is the minimum eigenvalue of $\mathcal{X}^\top\mathcal{X}$ and e is a column vector of ones. The issue with this approach is that if $\lambda_{\min} = 0$, then diag(δ) becomes a trivial 0 matrix. If diag(δ) turns out to be a zero matrix, then MISOCP formulation reduces to the big-M formulation. [18] present an effective approach for obtaining a valid δ by solving the following semidefinite program (SDP)

$$\max_{\delta \in \mathbb{R}^{|V|}} \left\{ \sum_{i \in V} \delta_i \; : \; \mathcal{X}^\top \mathcal{X} - \text{diag}(\delta) \succeq 0, \delta_i \geq 0 \right\}. \tag{22a}$$

This formulation can attain a non-zero D_δ even if $\lambda_{\min} = 0$. Numerical results by [18] show that this method compares favorably with the minimum eigenvalue approach. [60] propose an SDP approach, which obtains D_δ such that the continuous relaxation of MISOCP (20) is as tight as possible.

Similar to [15], our formulation does not require adding a Tikhonov regularization. In this case, PRef is effective when $\mathcal{X}^\top \mathcal{X}$ is sufficiently diagonally dominant. When $n \geq m$ and each row of $\mathcal{X}$ is independent, then $\mathcal{X}^\top \mathcal{X}$ is guaranteed to be a positive semi-definite matrix [15]. On the other hand, when $n < m$, $\mathcal{X}^\top \mathcal{X}$ is not full-rank. Therefore, a Tikhonov regularization term should be added with sufficiently large μ to make $\mathcal{X}^\top \mathcal{X} + \mu I \succeq 0$ [15] in order to benefit from the strengthening provided by PRef.

6 Experiments

In this section, we report the results of our numerical experiments that compare different formulations and evaluate the effect of different tuning parameters and estimation strategies. Our experiments are performed on a cluster operating on UNIX with Intel Xeon E5-2640v4 2.4GHz. All formulations are implemented in the Python programming language. Gurobi 8.1 is used as the solver. Unless otherwise stated, a time limit of $50m$ (in seconds), where m denotes the number of nodes, and an MIQP relative optimality gap of 0.01 are imposed across all experiments after which runs are aborted. The *relative* optimality gap is calculated by RGAP:= $\frac{UB(X)-LB(X)}{UB(X)}$ where UB(X) denotes the objective value associated with the best feasible integer solution (incumbent) and LB(X) represents the best obtained lower bound during the branch-and-bound process for the formulation $X \in \{\text{MIQP}, \text{SIMILP}, \text{MISOCP}\}$.

Unless otherwise stated, we assume $\lambda_n = \log(n)$ which corresponds to the Bayesian information criterion (BIC) score. To select the big-M parameter, M, in all formulations we use the proposal of Park and Klabjan [38]. Specifically, given λ_n, we solve each problem without cycle prevention constraints and obtain β^R. We then use the upper bound $M = 2 \max_{(j,k) \in \overrightarrow{E}} |\beta^R_{jk}|$. Although this value does not guarantee an upper bound for M, the results provided in [38] and [32] computationally confirm that this approach gives a large enough value of M.

The goals of our computational study are twofold. First, we compare the various mathematical formulations to determine which gives us the best performance in Subsection 6.1, compare the sensitivity to the model parameters in Subsection 6.2, and the choice of the regularization term in Subsection 6.3. Second, in Subsection 6.4 we use the best-performing formulation to investigate the implications of the early stopping condition on the quality of the solution with respect to the true graph. To be able to perform such a study, we use synthetic data so that the true graph is available. In Subsection 6.5, we compare our algorithm against two state-of-the-art benchmark algorithms on publicly available datasets.

We use the package `pcalg` in `R` to generate random graphs. First, we create a DAG by `randomDAG` function and assign random arc weights (i.e., β) from a uniform distribution, $\mathcal{U}[0.1, 1]$. Next, the resulting DAG and random coefficients are fed into the `rmvDAG` function to generate multivariate data based on linear SEMs (columns of matrix $\mathcal{X}$) with the standard normal error distribution. We consider $m \in \{10, 20, 30, 40\}$ nodes and $n = 100$ samples. The average outgoing degree of each node, denoted by d, is set to two. We generate 10 random Erdős-Rényi graphs for each setting (m, n, d). We observe that in our instances, the minimum eigenvalue of $\mathcal{X}^\top \mathcal{X}$ across all instances is 3.26 and the maximum eigenvalue is 14.21.

Two types of problem instances are considered: (i) a set of instances with known moral graph corresponding to the true DAG; (ii) a set of instances with a complete undirected graph, i.e., assuming no prior knowledge. We refer to the first class of instances as *moral* instances and to the second class as *complete* instances. The observational data, $\mathcal{X}$, for both classes of instances are the same. The function `moralize(graph)` in the `pcalg` `R`-package is used to generated the moral graph from the true DAG. Although the moral graph can be consistently estimated from data using penalized estimation procedures with polynomial complexity [e.g., 31], the quality of moral graph affects all optimization models. Therefore, we use the true moral graph in our experiments, unless otherwise stated.

6.1 Comparison of Mathematical Formulations

We use the following MIQP-based metrics to measure the quality of a solution: relative optimality gap (RGAP), computation time in seconds (Time), Upper Bound (UB), Lower Bound (LB), objective function value (OFV) of the initial continuous relaxation, and the number of explored nodes in the branch-and-bound tree (# BB). An in-depth analysis comparing the existing mathematical formulations that rely on linear encodings of the constraints in (13c) for MIQP formulations is conducted by [32]. The authors conclude that MIQP+LN formulation outperforms the other MIQP formulations, and the promising performance of MIQP+LN can be attributed to its size: (1) MIQP+LN has fewer binary variables and constraints than MIQP+TO, (2) MIQP+LN is a compact (polynomial-sized) formulation in contrast to MIQP+CP which has an exponential number of constraints. Therefore, in this paper, we analyze the formulations based on the convex encodings of the constraints in (13c).

Table 1: Optimality gaps for MISOCP+TO and MISOCP+LN formulations

	Moral		Complete	
m	MISOCP+TO	MISOCP+LN	MISOCP+TO	MISOCP+LN
10	0.000	0.000	0.009	0.008
20	0.021	0.006	0.272	0.195
30	-	0.010	-	0.195
40	-	0.042	-	0.436

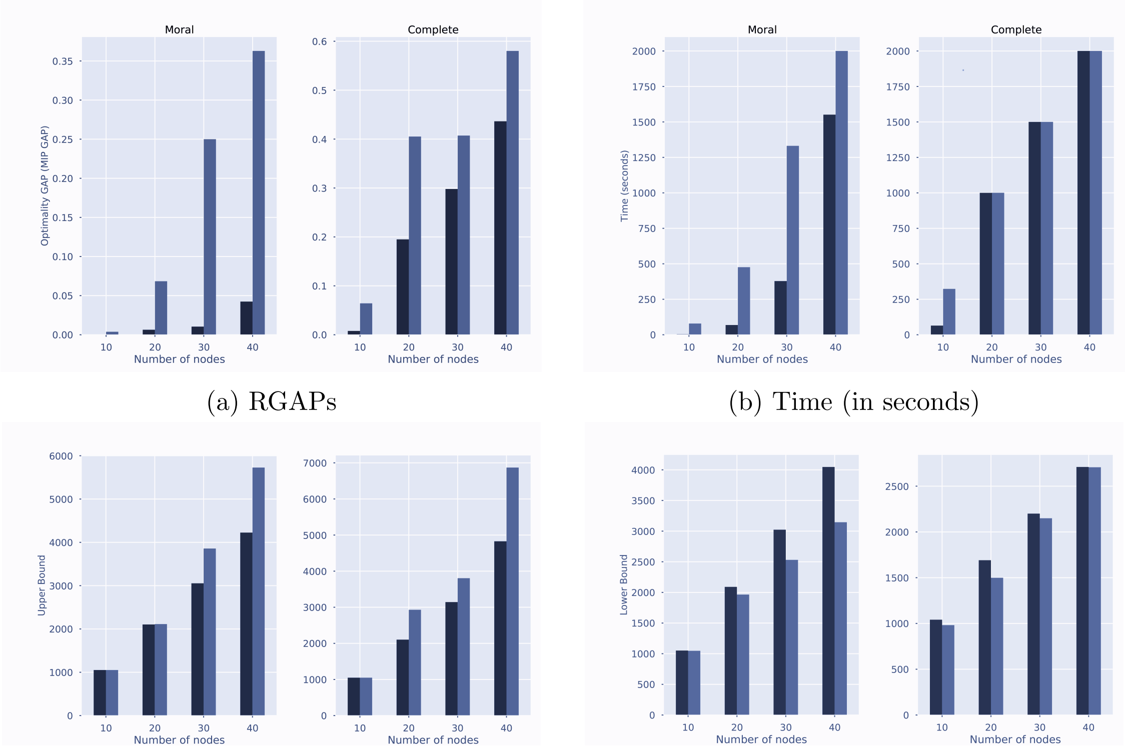
"-" denotes that no feasible solution, i.e., UB, is obtained within the time limit, so optimality gap cannot be computed.

6.1.1 Comparison of MISOCP formulations

We next experiment with MISOCP formulations. For the set of constraints in (13b), we use LN, TO, and CP constraints discussed in Section 4.1 resulting in three formulations denoted as MISOCP+LN, MISOCP+TO, MISOCP+CP, respectively. The MISOCP+TO formulation fails to find a feasible solution for instances with 30 and 40 nodes, see Table 1. For moral instances, the optimality gaps for MISOCP+TO are 0.000 and 0.021 for instances with 10 and 20 nodes, respectively; for complete instances, the optimality gap for MISOCP+TO formulation are 0.009 and 0.272 for instances with 10 and 20 nodes, respectively. Moreover, Table 1 illustrates that MISOCP+LN performs better than MISOCP+TO for even small instances (i.e., 10 and 20 nodes).

For MISOCP+CP, instead of incorporating all constraints given by (23), we begin with no constraint of type (23). Given an integer solution with cycles, we detect a cycle and impose a new cycle prevention constraint to remove the detected cycle. Depth First Search (DFS) can detect a cycle in a directed graph with complexity $O(|V| + |E|)$. Gurobi `LazyCallback` function is used, which allows adding cycle prevention constraints in the branch-and-bound algorithm, whenever an integer solution with cycles is found. The same approach is used by [38] to solve the corresponding MIQP+CP. Note that Gurobi solver follows a branch-and-cut implementation and adds many general-purpose and special-purpose cutting planes.

Figures 1a and 1b show that MISOCP+LN outperforms MISOCP+CP in terms of relative optimality gap and computational time. In addition, MISOCP+LN attains better upper and lower bounds than MISOCP+CP (see, Figures 1c and 1d). MISOCP+CP requires the solution of a second-order cone program (SOCP) after each cut, which reduces its computational efficiency and results in higher optimality gaps than MISOCP+LN. MISOCP+TO requires many binary variables which makes the problem very inefficient when the network becomes denser and larger as shown in Table 1. Therefore, we do not illustrate the MISOCP+TO results in Figure 1.



(a) RGAPs

(b) Time (in seconds)

(c) Best upper bounds

(d) Best lower bounds

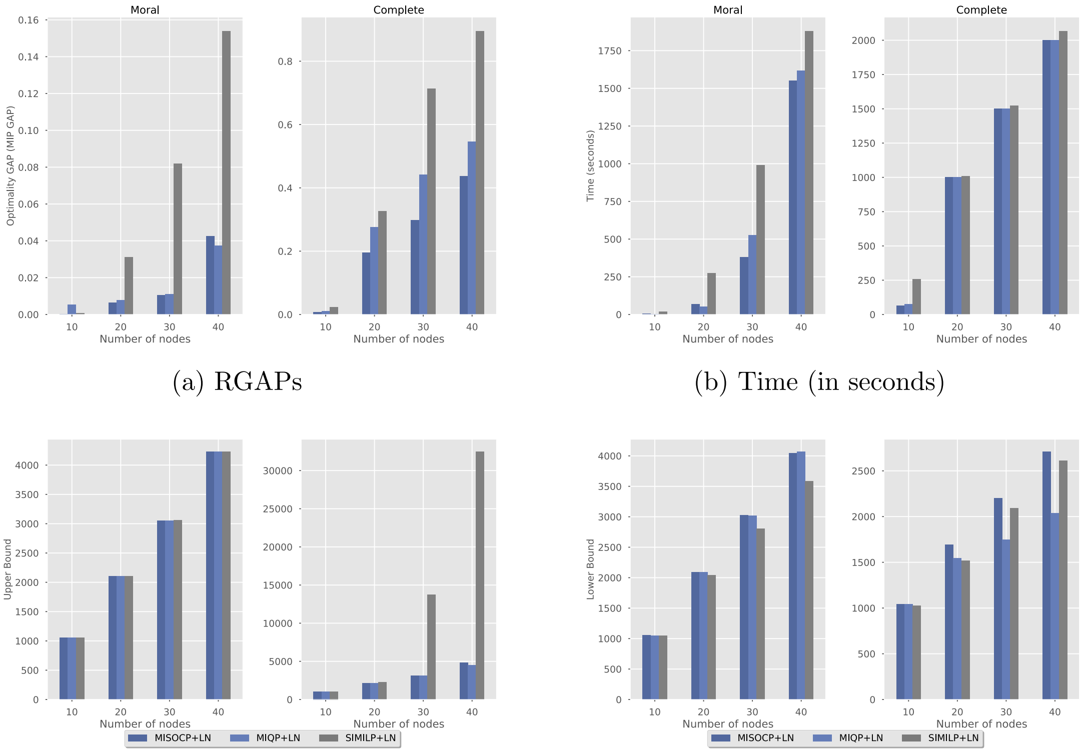
Figure 1: Optimization-based measures for MISOCP+LN (left bar) and MISOCP+CP (right bar) formulations for $n = 100$.

6.1.2 Comparison of MISOCP versus SIMILP

Our computational experiments show that the SIMILP formulation generally performs poorly when compared to MISOCP+LN and MIQP+LN in terms of optimality gap, upper bound, and computational time. We report the results for SIMILP+LN, MISOCP+LN, and MIQP+LN formulations in Figure 2. We only consider the LN formulation because that is the best performing model among the alternatives both for MISOCP and MIQP formulations.

Figures 2a and 2b show the relative optimality gaps and computational times for these three formulations. Figures 2c and 2d demonstrate that SIMILP+LN attains lower bounds that are comparable with other two formulations. In particular, for complete instances with large number of nodes, SIMILP+LN attains better lower bounds than MIQP+LN. Nonetheless, SIMILP+LN fails to obtain good upper bounds. Therefore, the relative optimality gap is considerably larger for SIMILP+LN.

The poor performance of SIMILP+LN might be because state-of-the-art optimization



(a) RGAPs

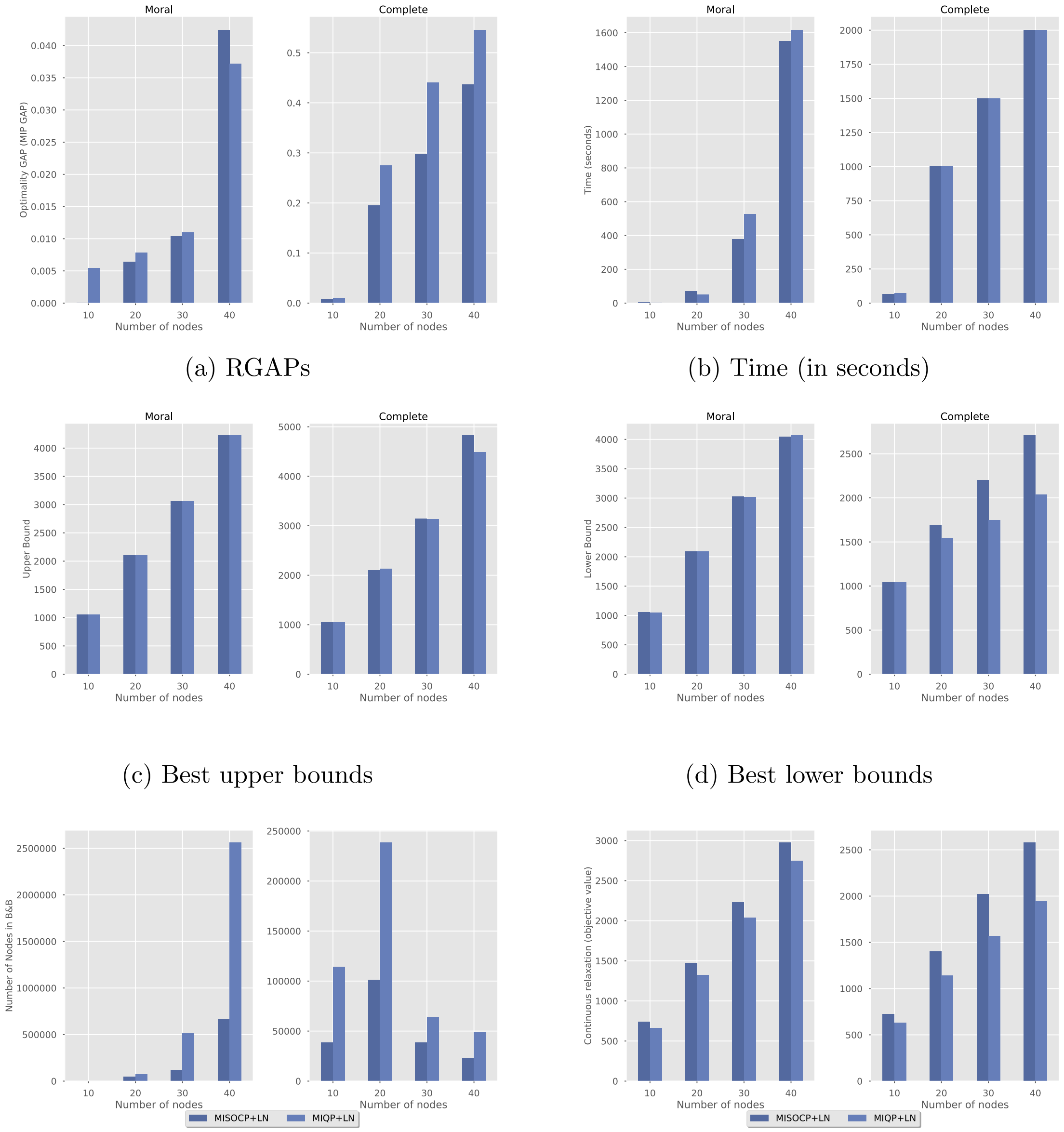
(b) Time (in seconds)

(c) Best upper bounds

(d) Best lower bounds

Figure 2: Optimization-based measures for MISOCP+LN, MIQP+LN, and SIMILP+LN formulations for $n = 100$.

packages (e.g., Gurobi, CPLEX) use many heuristics to obtain a good feasible solution (i.e., upper bound) for a compact formulation. In contrast, SIMILP is not a compact formulation, and we build the SIMILP gradually by adding violated constraints iteratively. Therefore, a feasible solution to the original formulation is not available while solving the relaxations with a subset of the constraints (21c). Moreover, the optimization solvers capable of solving MISOCP formulations have witnessed noticeable improvement due to theoretical developments in this field. In particular, Gurobi reports 20% and 38% improvement in solution time for versions 8 and 8.1, respectively. In addition, Gurobi v8.1 reports over four times faster solution times than CPLEX for solving MISOCP on their benchmark instances.



(a) RGAPs

(b) Time (in seconds)

(c) Best upper bounds

(d) Best lower bounds

(e) Number of Branch and Bound nodes

(f) Continuous relaxation objective function

Figure 3: Optimization-based measures for MISOCP+LN, MIQP+LN formulations for $n = 100$.

6.1.3 Comparison of MISOCP versus MIQP formulations

In this section, we demonstrate the benefit of using the second-order conic formulation MISOCP+LN instead of the linear big-M formulation MIQP+LN. As before, we only consider the LN formulation for this purpose. Figures 3a and 3b show that MISOCP+LN performs better than MIQP+LN in terms of the average relative optimality gap across all number of nodes $m \in \{10, 20, 30, 40\}$. The only exception is $m = 40$ for moral instances, for which MIQP+LN performs better than MISOCP+LN. Nonetheless, we observe that MISOCP+LN clearly outperforms MIQP+LN for complete instances which are more difficult to solve.

Figures 3c and 3d show the performance of both formulations in terms of the resulting upper and lower bounds on the objective function. We observe that MISOCP+LN attains better lower bounds especially for complete instances. However, MISOCP+LN cannot always obtain a better upper bound. In other words, MISOCP+LN is more effective in improving the lower bound instead of the upper bound as expected.

Figures 3e and 3f show that MISOCP+LN uses fewer branch-and-bound nodes and achieves better continuous relaxation values than MIQP+LN.

6.2 Analyzing the Choices of λ_n and M

We now experiment on different values for λ_n and M to assess the effects of these parameters on the performance of MISOCP+LN and MIQP+LN. First, we consider multiple λ values, $\lambda_n \in \{\log(n), 2\log(n), 4\log(n)\}$, while keeping the value of M the same (i.e., $M = 2 \max_{(j,k)\in\overrightarrow{E}} |\beta^{\star}_{jk}|$). Table 2 shows that as λ_n increases, MISOCP+LN consistently performs better than MIQP+LN in terms of the relative optimality gap, computational time, the number of branch-and-bound nodes, and continuous relaxation objective function value. Indeed, the difference becomes even more pronounced for more difficult cases (i.e., complete instances). For instance, for $\lambda_n = 4\log(n) = 18.4$, the relative optimality gap reduces from 0.465 to 0.374, an over 24% improvement. In addition, MISOCP+LN allows more instances to be solved to optimality within the time limit. For example, for moral instances with $m = 40$, $\lambda_n = 18.4$, eight out of ten instances are solved to optimality using MISOCP+LN while only two instances are solved to optimality by MIQP+LN.

Finally, we study the influence of the big-M parameter. Instead of a coefficient $\gamma = 2$ in [38], we experiment with $M = \gamma \max_{(j,k)\in\overrightarrow{E}} |\beta^{R}_{jk}|$ for $\gamma \in \{2, 5, 10\}$ in Table 3, where $|\beta^{R}_{jk}|$ denotes the optimal solution of each optimization problem without the constraints to remove cycles. The larger the big-M parameter, the worse the effectiveness of both models. However, comparing the continuous relaxation objective function values, we observe that MISOCP+LN tightens the formulation using the conic constraints whereas MIQP+LN does not have any means to tighten the formulation instead of big-M constraints which have poor relaxation. In most cases, the MISOCP+LN formulation allows more instances

Table 2: Computational results for different values of $\lambda_n = t\log(n)$ for $t \in \{1,2,4\}$, * indicates that the problem is solved to the optimality tolerance. Superscript ^{i} indicates that out of ten runs, i instances finish before hitting the time limit. Time is averaged over instances that solve within the time limit, RGAP is averaged over instances that reach the time limit. Better RGAPs are in bold.

		Moral								Complete							
Instances		RGAP		Time		# nodes		Relaxation OFV		RGAP		Time		# nodes		Relaxation OFV	
m	λ_n	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP
10	4.6	*	*	3	2	1306	3715	738.7	664.9	*	*	65	74	38850	114433	724.4	629.3
10	9.2	*	*	4	3	1116	2936	784.6	693.5	*	*	31	39	15736	55543	772.5	662.2
10	18.4	*	*	3	2	1269	2457	857.0	747.5	*	*	26	29	18223	41197	844.5	720.2
20	4.6	*	*	69	51	46513	76261	1474.2	1325.8	**.195**	.275	1000	1000	101509	238765	1404.9	1144.5
20	9.2	*	*	26	27	10695	31458	1589.6	1406.8	**.152**	.250	1000	1000	152206	274514	1526.9	1238.6
20	18.4	*	*	24	36	9574	33788	1763.7	1552.7	**.113**^{2}	.208	944	1000	159789	277687	1697.1	1395.0
30	4.6	**.010**^{8}	0.011^{8}	378	527	121358	514979	2230.1	2037.7	**.298**	.441	1500	1500	38474	64240	2024.0	1569.7
30	9.2	*	*	104	291	33371	248190	2392.4	2168.5	**.239**	.395	1500	1500	59034	71475	2217.5	1741.5
30	18.4	*	*	48	74	15649	57909	2608.3	2383.8	**.215**	.318	1500	1500	74952	96586	2449.2	2006.9
40	4.6	.042^{6}	**.037**^{4}	1551	1615	664496	2565247	2979.3	2748.6	**.436**	.545	2000	2000	23083	49050	2582.0	1946.3
40	9.2	**.024**^{8}	.036^{4}	1125	1336	353256	1347702	3200.7	2923.5	**.397**	.473	2000	2000	29279	73917	2869.9	2216.9
40	18.4	**.024**^{8}	.035^{2}	1099	1375	434648	1137666	3521.8	3225.4	**.374**	.465	2000	2000	31298	60697	3240.1	2633.1

Table 3: Computational results for different values of γ, * indicates that the problem is solved to the optimality tolerance. Superscript ^{i} indicates that out of ten runs, i instances finish before hitting the time limit. Time is averaged over instances that solve within the time limit, RGAP is averaged over instances that reach the time limit. Better RGAPs are in bold.

		Moral								Complete							
Instances		RGAP		Time		# nodes		Relaxation OFV		RGAP		Time		# nodes		Relaxation OFV	
m	γ	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP
10	2	*	*	3	2	1306	3715	738.7	664.9	*	*	65	74	38850	114433	724.4	629.3
10	5	*	*	5	2	1433	3026	717.9	647.1	*	*	81	82	42675	130112	705.1	607.8
10	10	*	*	5	2	1523	2564	712.5	641.1	*	*	74	100	35576	174085	699.8	600.3
20	2	*	*	69	51	46513	76261	1474.2	1325.8	**.195**	.275	1000	1000	101509	238765	1404.9	1144.5
20	5	*	*	103	156	65951	209595	1438.2	1274.2	**.211**	.308	1000	1000	97940	225050	1375.3	1080.9
20	10	*	*	215	207	150250	349335	1427.7	1256.6	**.230**	.310	1000	1000	90864	257998	1366.3	1058.2
30	2	**.010**^{8}	.011^{8}	378	527	121358	514979	2230.1	2037.7	**.298**	.441	1500	1500	38474	64240	2024.0	1569.7
30	5	**.011**^{8}	.014^{8}	571	620	164852	527847	2173.9	1950.3	**.336**	.474	1501	1500	33120	64339	1969.4	1448.4
30	10	.024^{8}	**.014**^{8}	630	638	202635	585234	2156.5	1919.6	**.349**	.480	1500	1500	30579	77100	1951.2	1404.0
40	2	.042^{6}	**.037**^{4}	1551	1615	664496	2565247	2979.3	2748.6	**.436**	.545	2000	2000	23083	49050	2582.0	1946.3
40	5	**.045**^{6}	.047^{2}	1643	1634	638323	1347868	2895.6	2635.0	**.579**	.580	2000	2000	12076	30858	2488.0	1751.7
40	10	**.056**^{4}	.057^{2}	1639	1632	599281	1584187	2869.2	2595.6	**.585**	.594	2000	2000	11847	30222	2456.1	1679.6

to be solved to optimality than MIQP+LN. For larger m, because Gurobi solves larger SOCP relaxations in each branch-and-bound node, the MISOCP+LN formulation explores much fewer branch-and-bound nodes and stops with a similar RGAP at termination. For $M > 2 \max_{(j,k)\in \overrightarrow{E}} |\beta^R_{jk}|$, MISOCP+LN outperforms MIQP+LN in all measures, in most cases.

Table 4: Computational results for different values of μ, * indicates that the problem is solved to the optimality tolerance. Superscript ^{i} indicates that out of ten runs, i instances finish before hitting the time limit. Time is averaged over instances that solve within the time limit, RGAP is averaged over instances that reach the time limit. Better RGAPs are in bold.

		Moral								Complete							
Instances		RGAP		Time		# nodes		Relaxation OFV		RGAP		Time		# nodes		Relaxation OFV	
m	μ	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP	MISOCP	MIQP
10	0	*	*	3	2	1306	3715	738.7	664.9	*	*	65	74	38850	114433	724.4	629.3
10	4.6	*	*	4	2	1043	2758	802.0	708.5	*	*	69	72	38778	119825	789.3	675.7
10	9.2	*	*	4	2	1067	2231	858.0	748.1	*	*	72	74	36326	114383	843.2	712.3
20	0	*	*	69	51	46513	76261	1474.2	1325.8	**.195**	.275	1000	1000	101509	238765	1404.9	1144.5
20	4.6	*	*	45	45	15111	55302	1604.1	1426.5	**.167**	.242	1000	1000	102467	249490	1551.7	1267.1
20	9.2	*	*	43	55	15384	62297	1716.8	1515.7	**.142**	.223	1000	1000	94360	258194	1668.3	1355.1
30	0	**.010**^{8}	.011^8	378	527	121358	514979	2230.1	2037.7	**.298**	.441	1500	1500	38474	64240	2024.0	1569.7
30	4.6	**.008**^{9}	.011^8	310	392	76668	358544	2432.5	2187.7	**.237**	.387	1500	1500	45473	69258	2286.4	1788.5
30	9.2	**.009**^{9}	.010^8	67	377	12410	320632	2612.6	2311.4	**.209**	.367	1500	1500	41241	68661	2484.3	1915.7
40	0	.042^6	**.037**^{4}	1551	1615	664496	2565247	2979.3	2748.6	**.436**	.545	2000	2000	23083	49050	2582.0	1946.3
40	4.6	**.027**^{8}	.029^4	1331	1620	422654	1303301	3281.6	2972.8	**.354**	.471	2000	2000	13209	30995	2985.4	2261.3
40	9.2	**.020**^{8}	.028^6	870	1507	239214	1762210	3575.4	3165.3	**.367**	.445	2000	2000	13884	54638	3321.7	2468.7

6.3 The Effect of Tikhonov Regularization

In this subsection, we consider the effect of adding a Tikhonov regularization term to the objective (see Remark 1) by considering $\mu \in \{0, \log(n), 2\log(n)\}$ while keeping the values of $\lambda_n = \log(n)$ and M the same as before. Table 4 demonstrates that for all instances with $\mu > 0$, MISOCP+LN outperforms MIQP+LN. For complete instances with $m = 40$ and $\mu = 9.2$, MISOCP+LN improves the optimality gap from 0.445 to 0.367, an improvement over 21%. The reason for this improvement is that $\mu > 0$ makes the matrix more diagonally dominant; therefore, it makes the conic constraints more effective in tightening the formulation and obtaining a better optimality gap. Also, MISOCP+LN allows more instances to be solved to optimality than MIQP+LN.

6.4 Practical Implications of Early Stopping

In this subsection, we evaluate the quality of the estimated DAGs obtained from MISOCP+LN by comparing them with the ground truth DAG. To this end, we use three measures: the average structural Hamming distance (SHD) which counts the number of arc differences (additions, deletions, or reversals) required to transform the estimated DAG to the true DAG, the average false positive rate (FPR) which is the proportion of edges appearing in the estimated DAG but not the true DAG and the average true positive rate (TPR) which is the proportion of edges appearing in both the true DAG and the estimated DAG. Since Gurobi sets a minimum relative gap RGAP= $1e^{-4}$, the solution obtained within this relative gap is considered optimal. Finally, because the convergence

Table 5: Structural Hamming distances (SHD), False Positive Rate (FPR) and True Positive Rate (TPR) for early stopping with $n = 100, \lambda_n = \log(m)$, GAP $\leq \tau$ for moral instances. The superscripts ^{i} indicate that out of ten runs, i instances finish before hitting the time limit. Time is averaged over instances that solve within the time limit, GAP and RGAP are averaged over instances that terminate early.

		$\tau = 0$						$\tau = \frac{\log(m)}{n} s_m$					
m	s_m	Time	GAP	RGAP	SHD	FPR	TPR	Time	GAP	RGAP	SHD	FPR	TPR
10	19	1.28^{10}	0.00	0.000	0.75	0.04	1.00	1.28^{10}	0.06	0.000	0.77	0.02	1.00
20	58	6.15^{9}	0.71	0.000	1.50	0.01	1.00	6.04^{9}	1.44	0.001	2.00	0.01	1.00
30	109	37.40^{7}	13.96	0.004	1.67	0.00	1.00	27.63^{7}	15.80	0.005	1.66	0.00	1.00
40	138	935^{2}	63.06	0.131	5.00	0.01	1.00	640.15^{2}	66.34	0.132	5.00	0.01	1.00

of the branch-and-bound process may be slow in some cases, we set a time limit of $100m$.

To test the quality of the solution obtained with an early stopping criterion, we set the absolute optimality gap parameter as $GAP = \frac{\log(m)}{n} s_m$ and the ℓ_0 regularization parameter as $\lambda_n = \log m$ as suggested by the discussion following Proposition 2 for achieving a consistent estimate. We compare the resulting suboptimal solution to the solution obtained by setting GAP $=$ UB $-$ LB $= 0$ to obtain the truly optimal solution.

Table 5 shows the numerical results for the average solution time (in seconds) for instances that are solved within the time limit, the number of instances that were not solved within the time limit, the actual absolute optimality gap at termination, the average FPR, the average TPR, the average SHD of the resulting DAGs, across 10 runs for moral instances. Table 5 indicates that the average SHD for $GAP = \frac{\log(m)}{n} s_m$ is close to that of the truly optimal solution, and the average (FPR) and (TPR) are the same between setting GAP $= \frac{\log(m)}{n} s_m$ and GAP $= 0$ except for $m = 10$. Note that a lower GAP generally leads to a better SHD score. From a computational standpoint, we observe that by using the early stopping criterion, we are able to obtain consistent solutions faster. In particular, the average solution time reduces by 25% for $m = 30$. The number of instances which are solved before hitting the $100m$ time limit are the same for $GAP = 0$ and GAP $= \frac{\log(m)}{n} s_m$. Furthermore, stopping early does not sacrifice too much from the quality of the resulting DAG as can be seen from the SHD scores.

6.5 Comparison to Other Benchmarks

In this section, we compare the performance of MISOCP against the state-of-the-art benchmarks. These experiments are executed on a laptop with a Windows 10 operating system, an Intel Core i7-8750H 2.2-GHz CPU, 8-GB DRAM using Python 3.8 with Gurobi 9.1.1 Optimizer.

The benchmarks considered in this section include the top-down approach (EqVarDAG-TD) and the high-dimensional top-down approach (EqVarDAG-HD-TD) of [9], as well as the high-dimensional bottom-up approach (EqVarDAG-HD-BU) of [24]. By taking advantage of the conditions for identifiability in linear SEM models, these benchmark procedures offer polynomial-time algorithms for learning DAGs by iteratively identifying a source (top-down) or sink (bottom-up) node based on solving a series of covariance selection problems.

We compare the performance of the methods on twelve publicly available networks from [32] and Bayesian Network Repository (bnlearn). The number of nodes in these networks ranges from $m = 6$ to $m = 70$. We generate data from both identifiable and non-identifiable error distributions. In the case of identifiable distributions (ID), we generate the data by using random arc weights β from $\mathcal{U}[-1, -0.1] \cup \mathcal{U}[0.1, 1]$ and $n = 500$ samples standard normal errors. The data for the non-identifiable (NID) error distributions was generated similarly, but from normal errors with non-equal error variances chosen randomly from $\{0.5, 1, 1.5\}$.

As an input superstructure graph to MISOCP, other than the true moral graphs, we also consider a superstructure estimate based on the empirical correlation matrix (CorEst). This estimate—which is guaranteed to be a super set of the DAG skeleton under the faithfulness assumption—was obtained by testing whether each correlation coefficient is nonzero at 0.05 significance level; the p-values were obtained using the Fisher's Z-transformation for correlation coefficients. The MISOCP with true and correlation matrix superstructures are denoted as MISOCP-True and MISOCP-CorEst, respectively, in Table 6. A time limit of $50m$ (seconds), $\lambda = 2\log(n)$ and the Gurobi RGAP of 0.01 are imposed across the experiments.

Measures of performance of the benchmark algorithms are summarized in columns EqVarDAG-TD, EqVarDAG-HD-TD, and EqVarDAG-HD-BU of Table 6. The column Time reports the solution time in seconds. For all datasets, the true networks can be used to evaluate the quality of the estimated networks. We report SHD, TPR, and FPR for all the estimated networks. Given that the true causal network cannot be recovered in the setting of non-identifiable data (NID), we also report the structural SHD between the undirected skeleton of the true DAG and the corresponding skeleton of estimated network; this is denoted as SHDs in Table 6.

We observe that most of the EqVarDAG methods solve the problem within a second. With respect to the quality of the estimation, EqVarDAG-TD provides better performance in SHD compared to EqVarDAG-HD-TD and EqVarDAG-HD-BU. The column RGAP reports the relative gap at early termination. The symbol (*) denotes that the problem is solved to the optimality tolerance. Compared with the benchmarks, MISOCP with a CorEst or true superstructure requires longer solution times; however, MISOCP consistently provides high SHD and SHDs scores in every network. Moreover, MISOCP is able to provide the best estimation among all methods in most of the networks.

Finally, we highlight that in the non-identifiable datasets (NID), MISOCP clearly outperforms the benchmarks. This is, of course, not surprising, as the benchmark algorithms rely on the identifiability assumption and are not guaranteed to work if this assumption is violated. In contrast, in this case, MISOCP is guaranteed to find a member of the Markov equivalence class.

7 Conclusion

In this paper, we study the problem of learning an optimal directed acyclic graph (DAG) from continuous observational data, where the causal effect among the random variables is linear. The central problem is a quadratic optimization problem with regularization. We present a mixed-integer second order conic program (MISOCP) which entails a tighter relaxation than existing formulations with linear constraints. Our numerical results show that MISOCP can successfully improve the lower bound and results in better optimality gap when compared with other formulations based on big-M constraints, especially for dense and large instances. Moreover, we establish an early stopping criterion under which we can terminate branch-and-bound and achieve a solution which is asymptotically optimal. In addition, we show that our method outperforms two state-of-the-art algorithms, especially on non-identifiable datasets.

Acknowledgments

We thank the AE and three anonymous reviewers for their detailed comments that improved the paper. Simge Küçükyavuz and Linchuan Wei were supported, in part, by ONR grant N00014-19-1-2321 and NSF grant CIF-2007814. Ali Shojaie was supported by NSF grant DMS-1561814 and NIH grant R01GM114029. Hao-Hsiang Wu is supported, in part, by MOST Taiwan grant 109-2222-E-009-005-MY2.

A Alternative linear encodings of constraints (13b)

There are several ways to ensure that the estimated graph does not contain any cycles. The first approach is to add a constraint for each cycle in the graph, so that at least one arc in this cycle must not exist in $\mathcal{G}(B)$. A *cutting plane* (CP) method is used to solve such a formulation which may require generating an exponential number of constraints. In particular, let $\mathcal{C}$ be the set of all possible directed cycles and $\mathcal{C}_A \in \mathcal{C}$ be the set of arcs defining a cycle. The CP formulation removes cycles by imposing the following constraints for (13b)

$$\textbf{CP} \qquad \sum_{(j,k)\in \mathcal{C}_A} g_{jk} \leq |\mathcal{C}_A| - 1, \quad \forall \mathcal{C}_A \in \mathcal{C}. \tag{23}$$

This formulation has exponentially many constraints.

Another way to rule out cycles is by imposing constraints such that the nodes follow a topological order [38]. A topological ordering is a unique ordering of the nodes of a graph from 1 to m such that the graph contains an arc (j, k) if node j appears before node k

in the order. We refer to this formulation as *topological ordering* (TO). Define decision variables $z_{jk} \in \{0,1\}$ for all $(j,k) \in \overrightarrow{E}$ and $o_{rs} \in \{0,1\}$ for all $r, s \in \{1,\ldots,m\}$. The variable z_{jk} takes value 1 if there is an arc (j,k) in the network, and o_{rs} takes value 1 if the topological order of node r equals s. The TO formulation rules out cycles in the graph by the following constraints

$$\begin{aligned}
\textbf{TO} \quad & g_{jk} \leq z_{jk}, && \forall (j,k) \in \overrightarrow{E}, && \text{(24a)}\\
& z_{jk} - m z_{kj} \leq \sum_{s \in V} s\,(o_{ks} - o_{js}), && \forall (j,k) \in \overrightarrow{E}, && \text{(24b)}\\
& \sum_{s \in V} o_{rs} = 1 && \forall r \in V, && \text{(24c)}\\
& \sum_{r \in V} o_{rs} = 1 && \forall s \in V. && \text{(24d)}
\end{aligned}$$

This formulation has $\mathcal{O}(m^2)$ variables and $\mathcal{O}(|\overrightarrow{E}|)$ constraints.

References

[1] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(Jan):171–234, 2010.

[2] Bryon Aragam and Qing Zhou. Concave penalized estimation of sparse Gaussian Bayesian networks. *Journal of Machine Learning Research*, 16:2273–2328, 2015.

[3] Alper Atamtürk and Andres Gómez. Rank-one convexification for sparse regression. *arXiv preprint arXiv:1901.10334*, 2019.

[4] Mark Barlett and James Cussens. Advances in bayesian network learning using integer programming. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pages 182–191, Arlington, Virginia, USA, 2013. AUAI Press.

[5] Mark Bartlett and James Cussens. Integer linear programming for the Bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017.

[6] Dimitris Bertsimas and Bart Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, 48(1):300–323, 2020.

[7] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

[8] Peter Bühlmann and Sara A van de Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.

[9] Wenyu Chen, Mathias Drton, and Y Samuel Wang. On causal discovery with an equal-variance assumption. *Biometrika*, 106(4):973–980, 09 2019.

[10] XT Cui, XJ Zheng, SS Zhu, and XL Sun. Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems. *Journal of Global Optimization*, 56(4):1409–1423, 2013.

[11] James Cussens. Maximum likelihood pedigree reconstruction using integer programming. In *WCB@ ICLP*, pages 8–19, 2010.

[12] James Cussens. Bayesian network learning with cutting planes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, pages 153–160, Arlington, Virginia, USA, 2011. AUAI Press.

[13] James Cussens, David Haws, and Milan Studenỳ. Polyhedral aspects of score equivalence in Bayesian network structure learning. *Mathematical Programming*, 164(1-2): 285–324, 2017.

[14] James Cussens, Matti Järvisalo, Janne H Korhonen, and Mark Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *J. Artif. Intell. Res.(JAIR)*, 58:185–229, 2017.

[15] Hongbo Dong, Kun Chen, and Jeff Linderoth. Regularization vs. relaxation: A conic optimization perspective of statistical variable selection. *arXiv preprint arXiv:1510.06083*, 2015.

[16] Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

[17] Antonio Frangioni and Claudio Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming*, 106(2):225–236, 2006.

[18] Antonio Frangioni and Claudio Gentile. SDP diagonalizations and perspective cuts for a class of nonseparable MIQP. *Operations Research Letters*, 35(2):181–185, 2007.

[19] Antonio Frangioni and Claudio Gentile. A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes. *Operations Research Letters*, 37(3):206–210, 2009.

[20] Antonio Frangioni, Claudio Gentile, Enrico Grande, and Andrea Pacifici. Projected perspective reformulations with applications in design problems. *Operations Research*, 59(5):1225–1232, 2011.

[21] Fei Fu and Qing Zhou. Learning sparse causal Gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

[22] Tian Gao, Ziheng Wang, and Qiang Ji. Structured feature selection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4256–4264, 2015.

[23] Asish Ghoshal and Jean Honorio. Information-theoretic limits of Bayesian network structure learning. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 767–775, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[24] Asish Ghoshal and Jean Honorio. Learning linear structural equation models in polynomial time and sample complexity. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1466–1475. PMLR, 09–11 Apr 2018.

[25] Andrés Gómez and O Prokopyev. A mixed-integer fractional optimization approach to best subset selection. *INFORMS Journal on Computing*, 33(2):551–565, 2021.

[26] Sung Won Han, Gong Chen, Myun-Seok Cheon, and Hua Zhong. Estimation of directed acyclic graphs through two-stage adaptive lasso for gene network inference. *Journal of the American Statistical Association*, 111(515):1004–1019, 2016.

[27] Raymond Hemmecke, Silvia Lindner, and Milan Studenỳ. Characteristic imsets for learning Bayesian network structure. *International Journal of Approximate Reasoning*, 53(9):1336–1349, 2012.

[28] Mikko Koivisto. Advances in exact bayesian structure discovery in bayesian networks. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'06, pages 241–248, Arlington, Virginia, USA, 2006. AUAI Press.

[29] Nevena Lazic, Christopher Bishop, and John Winn. Structural expectation propagation (SEP): Bayesian structure learning for networks with latent variables. In *Artificial Intelligence and Statistics*, pages 379–387, 2013.

[30] Peijing Liu, Salar Fattahi, Andrés Gómez, and Simge Küçükyavuz. A graph-based decomposition method for convex quadratic optimization with indicators. *arXiv preprint arXiv:2110.12547*, 2021.

[31] Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15(1):3065–3105, 2014.

[32] Hasan Manzour, Simge Küçükyavuz, Hao-Hsiang Wu, and Ali Shojaie. Integer programming for learning directed acyclic graphs from continuous data. *INFORMS Journal on Optimization*, 3(1):46–73, 2021.

[33] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[34] Chris. J. Oates, Jim Q. Smith, and Sach Mukherjee. Estimating causal structure using conditional DAG models. *Journal of Machine Learning Research*, 17(54):1–23, 2016.

[35] Chris J Oates, Jim Q Smith, Sach Mukherjee, and James Cussens. Exact estimation of multiple directed acyclic graphs. *Statistics and Computing*, 26(4):797–811, 2016.

[36] Temel Öncan, İ Kuban Altınel, and Gilbert Laporte. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654, 2009.

[37] Sascha Ott, Seiya Imoto, and Satoru Miyano. Finding optimal models for small gene networks. In *Pacific Symposium on Biocomputing*, volume 9, pages 557–567. World Scientific, 2004.

[38] Young Woong Park and Diego Klabjan. Bayesian network learning via topological order. *Journal of Machine Learning Research*, 18(99):1–32, 2017.

[39] Young Woong Park and Diego Klabjan. Subset selection for multiple linear regression via optimization. *Journal of Global Optimization*, 77(3):543–574, Jul 2020.

[40] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009.

[41] Jonas Peters and Peter Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2013.

[42] Mert Pilanci, Martin J Wainwright, and Laurent El Ghaoui. Sparse learning via Boolean relaxations. *Mathematical Programming*, 151(1):63–87, 2015.

[43] Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018.

[44] Takumi Saegusa and Ali Shojaie. Joint estimation of precision matrices in heterogeneous populations. *Electronic Journal of Statistics*, 10(1):1341–1392, 2016.

[45] Ali Shojaie and George Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

[46] Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal bayesian network structure. In *Conference on Uncertainty in Artificial Intelligence*, pages 445–452, 2006.

[47] Liam Solus, Yuhao Wang, and Caroline Uhler. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814.

[48] Arjun Sondhi and Ali Shojaie. The reduced PC-algorithm: Improved causal structure learning in large random networks. *Journal of Machine Learning Research*, 20(164): 1–31, 2019.

[49] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

[50] Milan Studenỳ and David C Haws. On polyhedral approximations of polytopes for learning Bayesian networks. *Journal of Algebraic Statistics*, 4(1), 2013.

[51] Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, 41(2):436–463, 2013.

[52] Sara van de Geer and Peter Bühlmann. ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

[53] Roman Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012.

[54] Linchuan Wei, Andrés Gómez, and Simge Küçükyavuz. On the convexification of constrained quadratic optimization problems with indicator variables. In Daniel Bienstock and Giacomo Zambelli, editors, *Integer Programming and Combinatorial Optimization*, pages 433–447, Cham, 2020. Springer International Publishing.

[55] Linchuan Wei, Alper Atamtürk, Andrés Gómez, and Simge Küçükyavuz. On the convex hull of convex quadratic optimization problems with indicators. *arXiv preprint arXiv:2201.00387*, 2021.

[56] Linchuan Wei, Andrés Gómez, and Simge Küçükyavuz. Ideal formulations for constrained convex optimization problems with indicator variables. *Mathematical Programming*, 192(1):57–88, 2022.

[57] Jing Xiang and Seyoung Kim. A* lasso for learning a sparse Bayesian network structure for continuous variables. In *Advances in Neural Information Processing Systems*, pages 2418–2426, 2013.

[58] Weijun Xie and Xinwei Deng. Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization*, 30(4):3359–3386, 2020.

[59] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[60] Xiaojin Zheng, Xiaoling Sun, and Duan Li. Improving the performance of MIQP solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach. *INFORMS Journal on Computing*, 26(4):690–703, 2014.

[61] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 9492–9503, Red Hook, NY, USA, 2018. Curran Associates Inc.

Table 6: The comparison between MISOCP and the state-of-the-art EqVarDAG methods of [9] and [24].

		EqVarDAG-HD-BU					EqVarDAG-HD-TD					EqVarDAG-TD					MISOCP-CorEst						MISOCP-True					
Data	Network(m)	Time	SHD	SHDs	TPR	FPR	Time	SHD	SHDs	TPR	FPR	Time	SHD	SHDs	TPR	FPR	Time	RGAP	SHD	SHDs	TPR	FPR	Time	RGAP	SHD	SHDs	TPR	FPR
ID	Dsep(6)	≤ 1	3	3	1	.001	≤ 1	3	3	1	.001	≤ 1	0	0	1	0	≤ 1	*	0	0	1	0	≤ 1	*	0	0	1	0
	Asia(8)	≤ 1	0	0	1	0	≤ 1	0	0	1	0	≤ 1	0	0	1	0	≤ 1	*	0	0	1	0	≤ 1	*	0	0	1	0
	Bowling(9)	≤ 1	5	4	1	.002	≤ 1	1	1	1	.001	≤ 1	0	0	1	0	≤ 1	*	0	0	1	0	≤ 1	*	0	0	1	0
	InsSmall(15)	≤ 1	11	8	1	.005	≤ 1	2	2	1	.001	≤ 1	2	2	1	.001	≥ 750	.05	0	0	1	0	4	*	0	0	1	0
	Rain(14)	≤ 1	5	4	1	.002	≤ 1	2	2	1	.001	≤ 1	0	0	1	0	155	*	0	0	1	0	1	*	0	0	1	0
	Cloud(16)	≤ 1	15	12	1	.006	≤ 1	6	6	1	.003	≤ 1	0	0	1	0	≥ 800	.101	0	0	1	0	≤ 1	*	0	0	1	0
	Funnel(18)	≤ 1	10	9	1	.004	≤ 1	4	4	1	.002	≤ 1	0	0	1	0	5	*	0	0	1	0	≤ 1	*	0	0	1	0
	Galaxy(20)	≤ 1	16	13	1	.007	≤ 1	9	9	1	.004	≤ 1	1	1	1	.001	≥ 1000	.048	0	0	1	0	2	*	0	0	1	0
	Insurance(27)	2	27	23	.962	.011	2	11	11	1	.005	≤ 1	0	0	1	0	≥ 1350	.216	0	0	1	0	176	*	0	0	1	0
	Factors(27)	2	25	25	.977	.01	2	17	17	.985	.007	≤ 1	3	3	.956	0	≥ 1350	.157	0	0	1	0	≥ 1350	.036	0	0	1	0
	Hailfinder(56)	10	78	70	.985	.033	11	19	19	1	.008	≤ 1	1	1	.985	0	≥ 2800	.21	13	12	.955	.004	≥ 2800	.052	0	0	1	0
	Hepar2(70)	17	147	139	.959	.062	19	70	70	.968	.029	2	1	1	.992	0	≥ 3500	.371	134	122	.878	.052	≥ 3500	.096	0	0	1	0
NID	Dsep(6)	≤ 1	4	3	1	.002	≤ 1	2	1	1	.001	≤ 1	1	1	.833	0	≤ 1	*	1	1	.833	0	≤ 1	*	1	1	.833	0
	Asia(8)	≤ 1	16	13	.875	.006	≤ 1	6	4	.875	.002	≤ 1	6	4	.875	.002	≤ 1	*	4	2	.875	.001	≤ 1	*	6	3	.875	.002
	Bowling(9)	≤ 1	7	6	.909	.003	≤ 1	7	5	.909	.003	≤ 1	4	2	.909	.001	≤ 1	*	2	1	.909	.001	≤ 1	*	4	2	.909	.001
	InsSmall(15)	≤ 1	24	19	.96	.01	≤ 1	8	7	.96	.003	≤ 1	8	7	.96	.003	≥ 750	.029	7	5	.88	.002	12	*	2	1	.96	.001
	Rain(14)	≤ 1	17	12	.944	.007	≤ 1	10	7	.944	.004	≤ 1	4	1	.944	.001	43	*	7	4	.833	.002	2	*	4	1	.944	.001
	Cloud(16)	≤ 1	19	14	.895	.007	≤ 1	20	14	.895	.007	≤ 1	12	6	.895	.004	≥ 800	.062	8	5	.947	.003	≤ 1	*	8	2	.947	.003
	Funnel(18)	≤ 1	12	11	.944	.005	≤ 1	6	6	.944	.002	≤ 1	1	1	.944	0	8	*	1	1	.944	0	≤ 1	*	1	1	.944	0
	Galaxy(20)	≤ 1	36	28	.955	.015	≤ 1	27	20	.955	.011	≤ 1	17	10	.955	.007	≥ 1000	.02	8	5	.909	.003	2	*	6	3	.955	.002
	Insurance(27)	2	40	32	1	.017	2	23	19	.981	.009	≤ 1	13	10	.9615	.005	≥ 1350	.191	11	9	.923	.003	≥ 1350	.055	6	4	.962	.002
	Factors(27)	2	12	11	.971	.004	2	32	24	.882	.010	≤ 1	32	24	.765	.007	≥ 1350	.111	19	15	.853	.004	≥ 1350	.062	22	16	.868	.006
	Hailfinder(56)	8	103	88	.97	.043	8	90	70	.97	.038	2	59	40	.894	.022	≥ 2800	.156	20	11	.985	.008	≥ 2800	.096	15	8	.985	.006
	Hepar2(70)	14	110	95	.9919	.048	16	73	61	.992	.031	2	52	40	.862	.015	≥ 3500	.199	33	23	.935	.011	≥ 3500	.112	36	18	.96	.014