A UNIFIED APPROACH TO MIXED-INTEGER OPTIMIZATION PROBLEMS WITH LOGICAL CONSTRAINTS

DIMITRIS BERTSIMAS*, RYAN CORY-WRIGHT†, AND JEAN PAUPHILET‡

Abstract. We propose a unified framework to address a family of classical mixed-integer optimization problems with logically constrained decision variables, including network design, facility location, unit commitment, sparse portfolio selection, binary quadratic optimization, sparse principal component analysis, and sparse learning problems. These problems exhibit logical relationships between continuous and discrete variables, which are usually reformulated linearly using a big-M formulation. In this work, we challenge this longstanding modeling practice and express the logical constraints in a non-linear way. By imposing a regularization condition, we reformulate these problems as convex binary optimization problems, which are solvable using an outer-approximation procedure. In numerical experiments, we establish that a general-purpose numerical strategy, which combines cutting-plane, first-order, and local search methods, solves these problems faster and at a larger scale than state-of-the-art mixed-integer linear or second-order cone methods. Our approach successfully solves network design problems with 100s of nodes and provides solutions up to 40% better than the state-of-the-art; sparse portfolio selection problems with up to 3,200 securities compared with 400 securities for previous attempts; and sparse regression problems with up to 100,000 covariates.

Key words. mixed-integer optimization; branch and cut; outer approximation

AMS subject classifications. 90C11, 90C57, 90C90

1. Introduction.

Many important problems from the Operations Research literature exhibit a logical relationship between continuous variables x and binary variables z of the form "$x = 0$ if $z = 0$". Among others, start-up costs in machine scheduling problems, financial transaction costs, cardinality constraints and fixed costs in facility location problems exhibit this relationship. Since the work of [34], this relationship is usually enforced through a "big-M" constraint of the form $-Mz \leq x \leq Mz$ for a sufficiently large constant $M > 0$. Glover's work has been so influential that big-M constraints are now considered as intrinsic components of the initial problem formulations themselves, to the extent that textbooks in the field introduce facility location, network design or sparse portfolio problems with big-M constraints *by default*, although they are actually *reformulations* of logical constraints.

In this work, we adopt a different perspective on the big-M paradigm, viewing it as a regularization term, rather than a modeling trick. Under this lens, we show that regularization drives the computational tractability of problems with logical constraints, explore alternatives to the big-M paradigm and propose an efficient algorithmic strategy which solves a broad class of problems with logical constraints.

1.1. Problem Formulation and Main Contributions.

We consider optimization problems which unfold over two stages. In the first stage, a decision-maker activates binary variables, while satisfying resource budget constraints and incurring activation costs. Subsequently, in the second stage, the decision-maker optimizes over the continuous variables. Formally, we consider the problem

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \in \mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n], \tag{1.1}$$

*Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, USA (dbertsim@mit.edu).

†Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA (ryancw@mit.edu).

‡London Business School, London, UK (jpauphilet@london.edu).

where $\mathcal{Z} \subseteq \{0,1\}^n$, $\boldsymbol{c} \in \mathbb{R}^n$ is a cost vector, $g(\cdot)$ is a generic convex function which possibly models convex constraints $\boldsymbol{x} \in \mathcal{X}$ for a convex set $\mathcal{X} \subseteq \mathbb{R}^n$ implicitly—by requiring that $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$, and $\Omega(\cdot)$ is a convex regularization function; we formally state its structure in Assumption 2.2.

In this paper, we provide three main contributions: First, we reformulate the logical constraint "$x_i = 0$ if $z_i = 0$" in a non-linear way, by substituting $z_i x_i$ for x_i in Problem (1.1). Second, we leverage the regularization term $\Omega(\boldsymbol{x})$ to derive a tractable reformulation of (1.1). Finally, by invoking strong duality, we reformulate (1.1) as a mixed-integer saddle-point problem, which is solvable via outer approximation.

Observe that the structure of Problem (1.1) is quite general, as the feasible set $\mathcal{Z}$ can capture known lower and upper bounds on $\boldsymbol{z}$, relationships between different z_i's, or a cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$. Moreover, constraints of the form $\boldsymbol{x} \in \mathcal{X}$, for some convex set $\mathcal{X}$, can be encoded within the domain of g, by defining $g(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin \mathcal{X}$. As a result, Problem (1.1) encompasses a large number of problems from the Operations Research literature, such as the network design problem described in Example 1.1. These problems are typically studied separately. However, the techniques developed for each problem are actually different facets of a single unified story, and, as we demonstrate in this paper, can be applied to a much more general class of problems than is often appreciated.

EXAMPLE 1.1. *Network design is an important example of problems of the form (1.1). Given a set of m nodes, the network design problem consists of constructing edges to minimize the construction plus flow transportation cost. Let E denote the set of all potential edges and let $n = |E|$. Then, the network design problem is given by:*

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \in \mathbb{R}^n_+} \quad \boldsymbol{c}^\top \boldsymbol{z} + \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad x_e = 0 \text{ if } z_e = 0 \quad \forall e \in E, \tag{1.2}$$

where $\mathcal{Z} = \{0,1\}^n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the flow conservation matrix, $\boldsymbol{b} \in \mathbb{R}^m$ is the vector of external demands and $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$, $\boldsymbol{d} \in \mathbb{R}^n$ define the quadratic and linear costs of flow circulation. We assume that $\boldsymbol{Q} \succeq \boldsymbol{0}$ is a positive semidefinite matrix. Inequalities of the form $\boldsymbol{\ell} \leq \boldsymbol{z} \leq \boldsymbol{u}$ can be incorporated within $\mathcal{Z}$ to account for existing/forbidden edges in the network. Problem (1.2) is of the same form as Problem (1.1) with

$$g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) := \begin{cases} \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x}, & \text{if } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}, \\ +\infty, & \text{otherwise.} \end{cases}$$

We present a generalized model with edge capacities and multiple commodities in Section 2.1.1.

1.2. Background and Literature Review. Our work falls into two areas of the mixed-integer optimization literature which are often considered in isolation: (a) modeling forcing constraints which encode whether continuous variables are active and can take non-zero values or are inactive and forced to 0, and (b) decomposition algorithms for mixed-integer optimization problems.

Formulations of forcing constraints. The most popular way to impose forcing constraints on continuous variables is to introduce auxiliary discrete variables which encode whether the continuous variables are active, and relate the discrete and continuous variables via the big-M approach of [34]. This approach was first applied to mixed-integer non-linear optimization (MINLO) in the context of sparse portfolio selection by [14]. With the big-M approach, the original MINLO admits bounded relaxations and can therefore be solved via branch-and-bound. Moreover, because the relationship between discrete and continuous variables is enforced via linear constraints, a big-M reformulation has a theoretically low impact on the tractability of the MINLOs continuous relaxations. However, in practice, high values of M lead to numerical instability and provide low-quality bounds [see 4, Section 5].

This observation led [28] to propose a class of cutting-planes for MINLO problems with indicator variables, called perspective cuts, which often provide a tighter reformulation of the logical constraints. Their approach was subsequently extended by [1], who, building upon the work of [5, pp. 88, item 5], proved that MINLO problems with indicator variables can often be reformulated as mixed-integer second-order cone problems (see [37] for a survey). More recently, a third approach for coupling the discrete and the continuous in MINLO was proposed independently for sparse regression by [47] and [13]: augmenting the objective with a strongly convex term of the form $\|\boldsymbol{x}\|_2^2$, called a ridge regularizer.

In the present paper, we synthesize the aforementioned and seemingly unrelated three lines of research under the unifying lens of regularization. Notably, our framework includes big-M and ridge regularization as special cases, and provides an elementary derivation of perspective cuts.

Numerical algorithms for mixed-integer optimization. A variety of "classical" general-purpose decomposition algorithms have been proposed for general MINLOs. The first such decomposition method is known as Generalized Benders Decomposition, and was proposed by [33] as an extension of [6]. A similar method, known as outer-approximation was proposed by [22], who proved its finite termination. The outer-approximation method was subsequently generalized to account for non-linear integral variables by [25]. These techniques decompose MINLOs into a discrete master problem and a sequence of continuous separation problems, which are iteratively solved to generate valid cuts for the master problem.

Though slow in their original implementation, decomposition schemes have benefited from recent improvements in mixed-integer linear solvers in the past decades, beginning with the branch-and-cut approaches of [45, 48], which embed the cut generation process within a single branch-and-bound tree, rather than building a branch-and-bound tree before generating each cut. We refer to [23, 24] for recent successful implementations of "modern" decomposition schemes. From a high-level perspective, these recent successes require three key ingredients: First, a fast cut generation strategy. Second, as advocated by [23], a rich cut generation process at the root node. Finally, a cut selection rule for degenerate cases where multiple valid inequalities exist (e.g., the Pareto optimality criteria of [43]).

In this paper, we connect the regularization used to reformulate logical constraints with the aforementioned key ingredients for modern decomposition schemes. Hence, instead of considering a MINLO formulation as a given and subsequently attempt to solve it at scale, our approach view big-M constraints as one of many alternatives. We argue that regularization is a modeling choice that impacts the tractability of the formulation and should be made accordingly.

1.3. Structure. We propose a unifying framework to address mixed-integer optimization problems, and jointly discuss modeling choice and numerical algorithms.

In Section 2, we identify a general class of mixed-integer optimization problems, which encompasses sparse regression, sparse portfolio selection, sparse principal component analysis, unit commitment, facility location, network design and binary quadratic optimization as special cases. For this class of problems, we discuss how imposing either big-M or ridge regularization accounts for non-linear relationships between continuous and binary variables in a tractable fashion. We also establish that regularization controls the convexity and smoothness of Problem (1.1)'s objective function.

In Section 3, we propose a conjunction of general-purpose numerical algorithms to solve Problem (1.1). The backbone of our approach is an outer approximation framework, enhanced with first-order methods to solve the Boolean relaxations and obtain improved lower bounds, certifiably near-optimal warm-starts via randomized rounding, and a discrete local search procedure. We also connect our approach to the perspective cut approach [28] from a theoretical and implementation standpoint.

Finally, in Section 4, we demonstrate empirically that algorithms derived from our framework can outperform state-of-the-art solvers. On network design problems with 100s of nodes and binary quadratic optimization problems with 100s of variables, we improve the objective value of the returned solution by 5 to 40% and 5 to 85% respectively, and our edge increases as the problem size increases. On empirical risk minimization problems, our method with ridge regularization is able to accurately select features among $100,000$s (resp. $10,000$s) of covariates for regression (resp. classification) problems, with higher accuracy than both Lasso and non-convex penalties from the statistics literature. For sparse portfolio selection, we solve to provable optimality problems one order of magnitude larger than previous attempts. We then analyze the benefits of the different ingredients in our numerical recipe on facility location problems, and discuss the relative merits of different regularization approaches on unit commitment instances.

Notation. We use nonbold face characters to denote scalars and components of matrices, lowercase bold faced characters such as $\boldsymbol{x}$ to denote vectors, uppercase bold faced characters such as $\boldsymbol{X}$ to denote matrices, and calligraphic characters such as $\mathcal{X}$ to denote sets. We let $\mathbf{e}$ denote a vector of all 1's, and $\mathbf{0}$ denote a vector of all 0's, with dimension implied by the context. If $\boldsymbol{x}$ is a n-dimensional vector then $\mathrm{Diag}(\boldsymbol{x})$ denotes the $n \times n$ diagonal matrix whose diagonal entries are given by $\boldsymbol{x}$. If $f(\boldsymbol{x})$ is a convex function then its perspective function $\varphi(\boldsymbol{x}, t)$, defined as $\varphi(\boldsymbol{x}, t) = tf(\boldsymbol{x}/t)$ if $t > 0$, $\varphi(\mathbf{0}, 0) = 0$, and ∞ elsewhere, is also convex [17, Chapter 3.2.6.]. Finally, we let $\mathbb{R}^n_+$ denote the n-dimensional nonnegative orthant.

2. Framework and Examples.

In this section, we present the family of problems to which our analysis applies, discuss the role played by regularization, and provide some examples from the Operations Research literature.

2.1. Examples. Problem (1.1) has a two-stage structure which comprises first "turning on" some indicator variables $\boldsymbol{z}$, and second solving a continuous optimization problem over the active components of $\boldsymbol{x}$. Precisely, Problem (1.1) can be viewed as a discrete optimization problem:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}), \tag{2.1}$$

where the inner minimization problem

$$f(\boldsymbol{z}) := \min_{\boldsymbol{x} \in \mathbb{R}^n} \quad g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n], \tag{2.2}$$

yields a best choice of $\boldsymbol{x}$ given $\boldsymbol{z}$. As we illustrate in this section, a number of problems of practical interest exhibit this structure.

EXAMPLE 2.1. *For the network design example* (1.2)*, we have*

$$f(\boldsymbol{z}) := \min_{\boldsymbol{x}\in\mathbb{R}^n_+:\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \quad \tfrac{1}{2}\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top\boldsymbol{x} \quad \text{s.t.} \quad x_e = 0 \text{ if } z_e = 0 \quad \forall e \in E.$$

2.1.1. Network Design. Example 1.1 illustrates that the single-commodity network design problem is a special case of Problem (1.1). We now formulate the k-commodity network design problem with directed capacities as minimizing over $\mathcal{Z} = \{0,1\}^n$ the function:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{f}^j,\boldsymbol{x}\in\mathbb{R}^n_+} \quad & \tfrac{1}{2}\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{f}^j = \boldsymbol{b}^j \quad \forall j \in [k], \\ & \boldsymbol{x} = \sum_{j=1}^k \boldsymbol{f}^j, \ \boldsymbol{x} \le \boldsymbol{u}, \\ & x_e = 0 \text{ if } z_e = 0 \quad \forall e \in E. \end{aligned} \tag{2.3}$$

2.1.2. Sparse Empirical Risk Minimization. Given a matrix of covariates $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and a response vector $\boldsymbol{y} \in \mathbb{R}^n$, the sparse empirical risk minimization problem seeks a vector $\boldsymbol{w}$ which explains the response in a compelling manner, i.e., minimizes over $\mathcal{Z} := \{\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top\boldsymbol{z} \le k\}$ the function:

$$f(\boldsymbol{z}) := \min_{\boldsymbol{w}\in\mathbb{R}^p} \sum_{i=1}^n \ell\left(y_i, \boldsymbol{w}^\top\boldsymbol{x}_i\right) + \frac{1}{2\gamma}\|\boldsymbol{w}\|_2^2 \quad \text{s.t.} \quad w_j = 0 \text{ if } z_j = 0 \quad \forall j \in [p], \tag{2.4}$$

where ℓ is an appropriate convex loss function; we provide examples of suitable loss functions in Table 1.

Table 1: Loss functions and Fenchel conjugates for ERM problems of interest.

Method	Loss function	Domain	Fenchel conjugate
OLS	$\frac{1}{2}(y-u)^2$	$y \in \mathbb{R}$	$\ell^\star(y,\alpha) = \frac{1}{2}\alpha^2 + \alpha y$
SVM	$\max(1-yu, 0)$	$y \in \{\pm 1\}$	$\ell^\star(y,\alpha) = \begin{cases} \alpha y, & \text{if } \alpha y \in [-1,0], \\ \infty, & \text{otherwise.} \end{cases}$

2.1.3. Sparse Portfolio Selection. Given an expected marginal return vector $\boldsymbol{\mu} \in \mathbb{R}^n$, estimated covariance matrix $\boldsymbol{\Sigma} \in \mathcal{S}^n_+$, uncertainty budget parameter $\sigma > 0$, cardinality budget parameter $k \in \{2, \dots, n-1\}$, linear constraint matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, and right-hand-side bounds $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^m$, investors determine an optimal allocation of capital between assets by minimizing over $\mathcal{Z} = \left\{\boldsymbol{z} \in \{0,1\}^n : \ \boldsymbol{e}^\top\boldsymbol{z} \le k\right\}$ the function

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{x}\in\mathbb{R}^n_+} \quad & \frac{\sigma}{2}\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x} - \boldsymbol{\mu}^\top\boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{l} \le \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{u}, \ \boldsymbol{e}^\top\boldsymbol{x} = 1, \ x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [n]. \end{aligned} \tag{2.5}$$

2.1.4. Unit Commitment. In the DC-load-flow unit commitment problem, each generation unit i incurs a cost given by a quadratic cost function $f^i(x) = a_i x^2 + b_i x + c_i$ for its power generation output $x \in [0, u_i]$. Let $\mathcal{T}$ denote a finite set of time periods covering a time horizon (e.g., 24 hours). At each time period $t \in \mathcal{T}$, there is an estimated demand d_t. The objective is to generate sufficient power to satisfy demand at minimum cost, while respecting minimum time on/time off constraints.

By introducing binary variables $z_{i,t}$, which denote whether generation unit i is active in time period t, requiring that $\boldsymbol{z} \in \mathcal{Z}$, i.e., $\boldsymbol{z}$ obeys physical constraints such as minimum time on/off, the unit commitment problem admits the formulation:

$$\min_{\boldsymbol{z}} \quad f(\boldsymbol{z}) + \sum_{t \in \mathcal{T}} \sum_{i=1}^{n} c_i z_{i,t} \quad \text{s.t.} \quad \boldsymbol{z} \in \mathcal{Z} \subseteq \{0,1\}^{n \times |\mathcal{T}|}, \tag{2.6}$$

$$\begin{aligned} \text{where:} \quad f(\boldsymbol{z}) := \min_{\boldsymbol{x}} \quad & \sum_{t \in \mathcal{T}} \left(\sum_{i=1}^{n} \tfrac{1}{2} a_i x_{i,t}^2 + b_i x_{i,t} \right) \text{ s.t. } \sum_{i=1}^{n} x_{i,t} \geq D_t \quad \forall t \in \mathcal{T}, \\ & x_{i,t} \in [0, u_{i,t}] \quad \forall i \in [n], \forall t \in \mathcal{T}, \\ & x_{i,t} = 0 \text{ if } z_{i,t} = 0 \quad \forall i \in [n], \forall t \in \mathcal{T}. \end{aligned} \tag{2.7}$$

2.1.5. Facility Location. Given a set of n facilities and m customers, the facility location problem consists of constructing facilities $i \in [n]$ at cost c_i to satisfy demand at minimal cost, i.e., minimizing over $\mathcal{Z} = \{0,1\}^n$ the function:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{X} \in \mathbb{R}_+^{n \times m}} \; & \boldsymbol{c}^\top \boldsymbol{z} + \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij} \text{ s.t. } \sum_{j=1}^{m} x_{ij} \leq u_i \quad \forall i \in [n], \\ & \sum_{i=1}^{n} x_{ij} = d_j \quad \forall j \in [m], \; x_{ij} = 0 \text{ if } z_i = 0 \quad \forall i \in [n], j \in [m]. \end{aligned} \tag{2.8}$$

In this formulation, x_{ij} corresponds to the quantity produced in facility i and shipped to customer j at a marginal cost of c_{ij}. Moreover, each facility i has a maximum output capacity of u_i and each customer j has a demand of d_j. In the uncapacitated case where $u_i = \infty$, the inner minimization problems decouple into independent knapsack problems for each customer j.

2.1.6. Sparse Principal Component Analysis (PCA). Given a $p \times p$ positive semidefinite covariance matrix $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma} \in S_+^p$ in short, the sparse PCA problem is to select a vector $\boldsymbol{z}$ which maximizes over $\mathcal{Z} = \left\{ \boldsymbol{z} \in \{0,1\}^p : \; \boldsymbol{e}^\top \boldsymbol{z} \leq k \right\}$ the function

$$f(\boldsymbol{z}) = \max_{\boldsymbol{x} \in \mathbb{R}^p} \quad \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} \text{ s.t. } \|\boldsymbol{x}\|_2^2 = 1, x_i = 0 \text{ if } z_i = 0 \quad \forall i \in [p]. \tag{2.9}$$

This function is apparently non-concave in $\boldsymbol{z}$, because $f(\boldsymbol{z})$ is the optimal value of a non-convex quadratic optimization problem. Fortuitously however, this problem admits an exact mixed-integer semidefinite reformulation, namely

$$f(\boldsymbol{z}) = \max_{\boldsymbol{X} \in S_+^p} \; \langle \boldsymbol{\Sigma}, \boldsymbol{X} \rangle \text{ s.t. } \operatorname{tr}(\boldsymbol{X}) = 1, x_{i,j} = 0 \text{ if } z_i = 0 \text{ or } z_j = 0 \quad \forall i, j \in [p]. \tag{2.10}$$

Indeed, for any fixed $\boldsymbol{z}$, Problem (2.10) maximizes a linear function in $\boldsymbol{X}$ and therefore admits a rank-one optimal solution. Thus, we prove that sparse PCA admits an exact mixed-integer semidefinite optimization reformulation.

2.1.7. Binary Quadratic Optimization. Given a symmetric cost matrix $\boldsymbol{Q}$, the binary quadratic optimization problem consists of selecting a vector of binary variables $\boldsymbol{z}$ which minimizes over $\mathcal{Z} = \{0,1\}^n$ the function:

$$f(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{Q} \boldsymbol{z}. \tag{2.11}$$

This formulation is non-convex and does not include continuous variables. However, introducing auxiliary continuous variables yields the equivalent formulation [26] of minimizing over $\mathcal{Z} = \{0,1\}^n$ the function:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{Y} \in \mathbb{R}_+^{n \times n}} \quad \langle \boldsymbol{Q}, \boldsymbol{Y} \rangle \quad \text{s.t.} \quad & y_{i,j} \leq 1 && \forall i,j \in [n], \\ & y_{i,j} \geq z_i + z_j - 1 && \forall i \in [n], \forall j \in [n] \backslash \{i\}, \\ & y_{i,i} \geq z_i && \forall i \in [n], \\ & y_{i,j} = 0 \text{ if } z_i = 0 && \forall i,j \in [n], \\ & y_{i,j} = 0 \text{ if } z_j = 0 && \forall i,j \in [n]. \end{aligned} \tag{2.12}$$

2.1.8. Union of Ellipsoidal Constraints. We now demonstrate that an even broader class of problems than MIOs with logical constraints can be cast within our framework. Concretely, we demonstrate that constraints $\boldsymbol{x} \in \mathcal{S} := \bigcup_{i=1}^k (Q_i \cap P_i)$, where $Q_i := \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^\top \boldsymbol{Q}_i \boldsymbol{x} + \boldsymbol{h}_i^\top \boldsymbol{x} + g_i \leq 0\}$, with $\boldsymbol{Q}_i \succeq \boldsymbol{0}$, is an ellipsoid and $P_i := \{\boldsymbol{x} : \boldsymbol{A}_i \boldsymbol{x} \leq \boldsymbol{b}_i\}$ is a polytope, can be reformulated as a special case of our framework. We remark that the constraint $\boldsymbol{x} \in \mathcal{S}$ is very general. Indeed, if we were to omit the quadratic constraints then we obtain a so-called ideal union of polyhedra formulation, which essentially all mixed-binary linear feasible regions admit [see 52].

To derive a mixed-integer formulation with logical constraints of $\mathcal{S}$ that fits within our framework, we introduce $\boldsymbol{x}_i \in \mathbb{R}^n$ and $\delta_i \in \{0,1\}^n$, such that $\boldsymbol{x}_i \in Q_i \cap P_i$ if $\delta_i = 1$, $\boldsymbol{x}_i = \boldsymbol{0}$ otherwise, and $\boldsymbol{x} = \sum_i \boldsymbol{x}_i$. We enforce $\boldsymbol{x}_i \in Q_i \cap P_i$ by introducing slack variables $\boldsymbol{\xi_i}$, ρ_i for the linear and quadratic constraints respectively, and forcing them to be zero whenever $\delta_i = 1$. Formally, $\mathcal{S}$ admits the following formulation

$$\begin{aligned} \boldsymbol{x} = \sum_{i=1}^k \boldsymbol{x}_i, \ \sum_{i=1}^k \delta_i &= 1, \\ \boldsymbol{A}_i \boldsymbol{x}_i \leq \boldsymbol{b}_i + \boldsymbol{\xi}_i & \quad \forall i \in [k], \\ \boldsymbol{x}_i^\top \boldsymbol{Q}_i \boldsymbol{x}_i + \boldsymbol{h}_i^\top \boldsymbol{x}_i + g_i \leq \rho_i & \quad \forall i \in [k], \\ \boldsymbol{x}_i = \boldsymbol{0} \text{ if } \delta_i = 0 & \quad \forall i \in [k], \\ \boldsymbol{\xi}_i = 0 \text{ if } (1-\delta_i) = 0 & \quad \forall i \in [k], \\ \rho_i = 0 \text{ if } (1-\delta_i) = 0 & \quad \forall i \in [k]. \end{aligned} \tag{2.13}$$

2.2. A Regularization Assumption. When we stated Problem (1.1), we assumed that its objective function consists of a convex function $g(\boldsymbol{x})$ plus a regularization term $\Omega(\boldsymbol{x})$. We now formalize this assumption:

ASSUMPTION 2.2. *In Problem* (1.1)*, the regularization term* $\Omega(\boldsymbol{x})$ *is one of:*
- *a big-M penalty function,* $\Omega(\boldsymbol{x}) = 0$ *if* $\|\boldsymbol{x}\|_\infty \leq M$ *and* ∞ *otherwise,*
- *a ridge penalty,* $\Omega(\boldsymbol{x}) = \dfrac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$.

This decomposition often constitutes a modeling choice in itself. We now illustrate this idea via the network design example.

EXAMPLE 2.3. *In the network design example (1.2), given the flow conservation structure $\boldsymbol{Ax} = \boldsymbol{b}$, we have that $\boldsymbol{x} \leq M\boldsymbol{e}$, where $M = \sum_{i:b_i>0} b_i$. In addition, if $\boldsymbol{Q} \succ \boldsymbol{0}$ then the objective function naturally contains a ridge regularization term with $1/\gamma$ equal to the smallest eigenvalue of $\boldsymbol{Q}$. Moreover, it is possible to obtain a tighter natural ridge regularization term by solving the following auxiliary semidefinite optimization problem a priori*

$$\max_{\boldsymbol{q} \geq \boldsymbol{0}} \ \boldsymbol{e}^\top \boldsymbol{q} \quad s.t. \quad \boldsymbol{Q} - \mathrm{Diag}(\boldsymbol{q}) \succeq \boldsymbol{0},$$

and using q_i as the ridge regularizer for each index i [30].

Big-M constraints are often considered to be a modeling trick. However, our framework demonstrates that imposing either big-M constraints or a ridge penalty is a regularization method, rather than a modeling trick. Interestingly, ridge regularization accounts for the relationship between the binary and continuous variables just as well as big-M regularization, without performing an algebraic reformulation of the logical constraints[1].

Conceptually, both regularization functions are equivalent to a soft or hard constraint on the continuous variables $\boldsymbol{x}$. However, they admit practical differences: For big-M regularization, there usually exists a finite value M_0, typically unknown a priori, such that if $M < M_0$, the regularized problem is infeasible. Alternatively, for every value of the ridge regularization parameter γ, if the original problem is feasible then the regularized problem is also feasible. Consequently, if there is no natural choice of M then imposing ridge regularization may be less restrictive than imposing big-M regularization. However, for any $\gamma > 0$, the objective of the optimization problem with ridge regularization is different from its unregularized limit as $\gamma \to \infty$, while for big-M regularization, there usually exists a finite value M_1 above which the two objective values match. We illustrate this discussion numerically in Section 4.3.

2.3. Duality to the Rescue. In this section, we derive Problem (2.2)'s dual and reformulate $f(\boldsymbol{z})$ as a maximization problem. This reformulation is significant for two reasons: First, as shown in the proof of Theorem 2.5, it leverages a non-linear reformulation of the logical constraints "$x_i = 0$ if $z_i = 0$" by introducing additional variables v_i such that $v_i = z_i x_i$. Second, it proves that the regularization term $\Omega(\boldsymbol{x})$ drives the convexity and smoothness of $f(\boldsymbol{z})$, and thereby drives the computational tractability of the problem. To derive Problem (2.2)'s dual, we require:

ASSUMPTION 2.4. *For each subproblem generated by $f(\boldsymbol{z})$, where $\boldsymbol{z} \in \mathcal{Z}$, either the optimization problem is infeasible, or strong duality holds.*

Note that all seven problems stated in Section 2.1 satisfy Assumption 2.4, as their inner problems are convex quadratics with linear or semidefinite constraints [17, Section 5.2.3]. Under Assumption 2.4, the following theorem reformulates Problem (2.1) as a saddle-point problem:

[1]Specifically, ridge regularization enforces logical constraints through perspective functions, as is made clear in Section 3.4.

THEOREM 2.5. *Under Assumption 2.4, Problem (2.1) is equivalent to:*

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i\, \Omega^\star(\alpha_i), \tag{2.14}$$

where $h(\boldsymbol{\alpha}) := \inf_{\boldsymbol{v}}\ g(\boldsymbol{v}) - \boldsymbol{v}^\top \boldsymbol{\alpha}$ is, up to a sign, the Fenchel conjugate of g [see 17, Chap. 3.3], and

$$\begin{aligned} \Omega^\star(\beta) &:= M|\beta| \quad \textit{for the big-}M\textit{ penalty},\\ \Omega^\star(\beta) &:= \tfrac{\gamma}{2}\beta^2 \quad \textit{for the ridge penalty}. \end{aligned}$$

Proof. Let us fix some $\boldsymbol{z} \in \{0,1\}^n$, and suppose that strong duality holds for the inner minimization problem which defines $f(\boldsymbol{z})$. Then, after introducing additional variables $\boldsymbol{v} \in \mathbb{R}^n$ such that $v_i = z_i x_i$, we have

$$f(\boldsymbol{z}) = \min_{\boldsymbol{x},\boldsymbol{v}}\ g(\boldsymbol{v}) + \Omega(\boldsymbol{x}) \quad \text{s.t. } \boldsymbol{v} = \text{Diag}(\boldsymbol{z})\boldsymbol{x}.$$

Let $\boldsymbol{\alpha}$ denote the dual variables associated with the coupling constraint $\boldsymbol{v} = \text{Diag}(\boldsymbol{z})\boldsymbol{x}$. The minimization problem is then equivalent to its dual problem, which is given by:

$$f(\boldsymbol{z}) = \max_{\boldsymbol{\alpha}}\ h(\boldsymbol{\alpha}) + \min_{\boldsymbol{x}} \left[\Omega(\boldsymbol{x}) + \boldsymbol{\alpha}^\top \text{Diag}(\boldsymbol{z})\boldsymbol{x}\right],$$

Since $\Omega(\cdot)$ is decomposable, i.e., $\Omega(\boldsymbol{x}) = \sum_i \Omega_i(x_i)$, we obtain:

$$\begin{aligned} \min_{\boldsymbol{x}} \left[\Omega(\boldsymbol{x}) + \boldsymbol{\alpha}^\top \text{Diag}(\boldsymbol{z})\boldsymbol{x}\right] &= \sum_{i=1}^n \min_{x_i} \left[\Omega_i(x_i) + z_i x_i \alpha_i\right] \\ &= \sum_{i=1}^n -\Omega^\star(-z_i\alpha_i) = -\sum_{i=1}^n z_i \Omega^\star(\alpha_i), \end{aligned}$$

where the last equality holds as $z_i > 0$ for the big-M and $z_i^2 = z_i$ for the ridge penalty.

Alternatively, if the inner minimization problem defining $f(\boldsymbol{z})$ is infeasible, then its dual problem is unbounded by weak duality[2]. □

Remark 2.6. Without regularization, i.e., $\Omega(\boldsymbol{x}) = 0$, a similar proof shows that Problem (2.1) admits an interesting saddle-point formulation:

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) \text{ s.t. } \alpha_i = 0, \text{ if } z_i = 1 \quad \forall i \in [n],$$

since $\Omega^\star(\alpha) = \min_x [x\alpha - \Omega(x)] = 0$ if $\alpha = 0$, and $+\infty$ otherwise. Consequently, the regularized formulation can be regarded as a relaxation of the original problem where the hard constraint $\alpha_i = 0$ if $z_i = 1$ is replaced with a soft penalty term $-z_i\Omega^\star(\alpha_i)$.

Remark 2.7. The proof of Theorem 2.5 exploits three attributes of the regularizer $\Omega(\boldsymbol{x})$. Namely, (1) decomposability, i.e., $\Omega(\boldsymbol{x}) = \sum_i \Omega_i(x_i)$, for appropriate scalar functions Ω_i, (2) the convexity of $\Omega(\boldsymbol{x})$ in $\boldsymbol{x}$, and (3) the fact that $\Omega(\cdot)$ regularizes[3] towards 0, i.e., $\boldsymbol{0} \in \arg\min_{\boldsymbol{x}} \Omega(\boldsymbol{x})$. However, the proof does not explicitly require that $\Omega(\boldsymbol{x})$ is either a big-M or a ridge regularizer. This suggests that our framework could be extended to other regularization functions.

[2]Weak duality implies that the dual problem is either unfeasible or unbounded. Since the feasible set of the maximization problem does not depend on $\boldsymbol{z}$, it is always feasible, unless the original problem (1.1) is itself infeasible. Therefore, we assume without loss of generality that it is unbounded.

[3]Importantly, the third attribute allows us to strengthen the formulation by not associating $\boldsymbol{z}$ with $\boldsymbol{x}$ in $\Omega(\boldsymbol{x})$, since $\boldsymbol{x}_i = 0$ is a feasible, indeed optimal choice of $\boldsymbol{x}$ for minimizing the regularizer when $z_i = 0$; this issue is explored in more detail in [8, Lemma 1]; see also [11, Appendix A.1].

EXAMPLE 2.8. *For the network design problem* (1.2), *we have*

$$h(\boldsymbol{\alpha}) = \min_{\boldsymbol{x}\geq \boldsymbol{0}: \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}} \quad \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{d}-\boldsymbol{\alpha})^\top \boldsymbol{x},$$
$$= \max_{\boldsymbol{\beta}_0 \geq \boldsymbol{0}, \boldsymbol{p}} \quad \boldsymbol{b}^\top \boldsymbol{p} - \tfrac{1}{2}\left(\boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha} + \boldsymbol{\beta}_0\right)^\top \boldsymbol{Q}^{-1}\left(\boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha} + \boldsymbol{\beta}_0\right).$$

Introducing $\boldsymbol{\xi} = \boldsymbol{Q}^{-1/2}\left(\boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha} + \boldsymbol{\beta}_0\right)$, *we can further write*

$$h(\boldsymbol{\alpha}) = \max_{\boldsymbol{\xi}, \boldsymbol{p}} \quad \boldsymbol{b}^\top \boldsymbol{p} - \tfrac{1}{2}\|\boldsymbol{\xi}\|_2^2 \;\; s.t \;\; \boldsymbol{Q}^{1/2}\boldsymbol{\xi} \geq \boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha}.$$

Hence, Problem (1.2) *is equivalent to minimizing over* $\boldsymbol{z} \in \mathcal{Z}$ *the function*

$$\boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}) = \max_{\alpha, \boldsymbol{\xi}, \boldsymbol{p}} \quad \boldsymbol{c}^\top \boldsymbol{z} + \boldsymbol{b}^\top \boldsymbol{p} - \tfrac{1}{2}\|\boldsymbol{\xi}\|_2^2 - \sum_{j=1}^n z_j\, \Omega^\star(\alpha_j)$$
$$s.t \quad \boldsymbol{Q}^{1/2}\boldsymbol{\xi} \geq \boldsymbol{A}^\top \boldsymbol{p} - \boldsymbol{d} + \boldsymbol{\alpha}.$$

Theorem 2.5 reformulates $f(\boldsymbol{z})$ as an inner maximization problem, namely

$$f(\boldsymbol{z}) = \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i\, \Omega^\star(\alpha_i), \tag{2.15}$$

for any feasible binary $\boldsymbol{z} \in \mathcal{Z}$. The regularization term Ω will be instrumental in our numerical strategy for it directly controls both the convexity and smoothness of f. Note that (2.15) extends the definition of $f(\boldsymbol{z})$ to the convex set $\mathrm{Bool}(\mathcal{Z})$, obtained by relaxing the constraints $\boldsymbol{z} \in \{0,1\}^p$ to $\boldsymbol{z} \in [0,1]^p$ in the definition of $\mathcal{Z}$.

Convexity. $f(\boldsymbol{z})$ is convex in $\boldsymbol{z}$ as a point-wise maximum of linear function of $\boldsymbol{z}$. In addition, denoting $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ a solution of (2.15), we have the lower-approximation:

$$f(\tilde{\boldsymbol{z}}) \geq f(\boldsymbol{z}) + \nabla f(\boldsymbol{z})^\top (\tilde{\boldsymbol{z}} - \boldsymbol{z}) \quad \forall \tilde{\boldsymbol{z}} \in \mathcal{Z}, \tag{2.16}$$

where $[\nabla f(\boldsymbol{z})]_i := -\Omega^\star(\alpha^\star(\boldsymbol{z})_i)$ is a sub-gradient of f at $\boldsymbol{z}$.

We remark that if the maximization problem in $\boldsymbol{\alpha}$ defined by $f(\boldsymbol{z})$ admits multiple optimal solutions then the corresponding lower-approximation of f at $\boldsymbol{z}$ may not be unique. This behavior can severely hinder the convergence of outer-approximation schemes such as Benders' decomposition. Since the work of [43] on Pareto optimal cuts, many strategies have been proposed to improve the cut selection process in the presence of degeneracy [see 23, Section 4.4 for a review]. However, the use of ridge regularization ensures that the objective function in (2.14) is strongly concave in α_i such that $z_i > 0$, and therefore guarantees that there is a unique optimal choice of $\alpha_i^\star(\boldsymbol{z})$. In other words, ridge regularization naturally inhibits degeneracy.

Smoothness. $f(\boldsymbol{z})$ is smooth, in the sense of Lipschitz continuity, which is a crucial property for deriving bounds on the integrality gap of the Boolean relaxation, and designing local search heuristics in Section 3. Formally, the following proposition follows from Theorem 2.5:

PROPOSITION 2.9. *For any* $\boldsymbol{z}, \boldsymbol{z}' \in \mathrm{Bool}(\mathcal{Z})$,

(a) With big-M regularization, $f(\boldsymbol{z}') - f(\boldsymbol{z}) \leq M \sum_{i=1}^n (z_i - z_i')|\alpha^\star(\boldsymbol{z}')_i|$.

(b) With ridge regularization, $f(\boldsymbol{z}') - f(\boldsymbol{z}) \leq \frac{\gamma}{2}\sum_{i=1}^n (z_i - z_i')\alpha^\star(\boldsymbol{z}')_i^2$.

Proof. By Equation (2.14),

$$f(\boldsymbol{z}') - f(\boldsymbol{z}) = \max_{\boldsymbol{\alpha}' \in \mathbb{R}^n} \left(h(\boldsymbol{\alpha}') - \sum_{i=1}^n z_i' \Omega^\star(\alpha_i') \right) - \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left(h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i \Omega^\star(\alpha_i) \right),$$

$$= h(\boldsymbol{\alpha}^\star(\boldsymbol{z}')) - \sum_{i=1}^n z_i' \Omega^\star(\alpha^\star(\boldsymbol{z}')_i) - h(\boldsymbol{\alpha}^\star(\boldsymbol{z}')) + \sum_{i=1}^n z_i \Omega^\star(\alpha^\star(\boldsymbol{z}')_i),$$

$$\leq \sum_{i=1}^n (z_i - z_i') \Omega^\star(\alpha^\star(\boldsymbol{z}')_i),$$

where the inequality holds because an optimal choice of $\boldsymbol{\alpha}'$ is a feasible choice of $\boldsymbol{\alpha}$. ∎

Proposition 2.9 demonstrates that, when the coordinates of $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ are uniformly bounded[4] with respect to $\boldsymbol{z}$, $f(\boldsymbol{z})$ is Lipschitz-continuous, with a constant L proportional to M (resp. γ) in the big-M (resp. ridge) case. We provide explicit bounds on the magnitude of L in Appendix B.

2.4. Merits of Ridge, Big-M Regularization: Theoretical Perspective. In this section, we propose a framework to reformulate MINLOs with logical constraints, which comprises regularizing MINLOs via either the widely used big-M modeling paradigm or the less popular ridge regularization paradigm. We summarize the advantages and disadvantages of each regularizer in Table 2. However, note that we have not yet established how these characteristics impact the numerical tractability and quality of the returned solution; this is the topic of the next two sections.

Table 2: Summary of the advantages (+) /disadvantages (−) of both techniques.

Regularization	Characteristics
Big-M	(+) Linear constraints (+) Supplies the same objective if $M > M_1$, for some $M_1 < \infty$ (−) Leads to infeasible problem if $M < M_0$, for some $M_0 < \infty$
Ridge	(+) Strongly convex objective (−) Systematically leads to a different objective for any $\gamma > 0$ (+) Preserves the feasible set

3. An Efficient Numerical Approach.

We now present an efficient numerical approach to solve Problem (2.14). The backbone is an outer-approximation strategy, embedded within a branch-and-bound procedure to solve the problem exactly. We also propose local search and rounding heuristics to find good feasible solutions, and use information from the Boolean relaxation to improve the duality gap.

3.1. Overall Outer-Approximation Scheme. Theorem 2.5 reformulates the function $f(\boldsymbol{z})$ as an inner maximization problem, and demonstrates that $f(\boldsymbol{z})$ is convex in $\boldsymbol{z}$, meaning a linear outer approximation provides a valid underestimator of

[4] Such a uniform bound always exists, as $f(\boldsymbol{z})$ is only supported on a finite number of binary points. Moreover, the strong concavity of h can yield stronger bounds (see Appendix B).

$f(\boldsymbol{z})$, as outlined in Equation (2.16). Consequently, a valid numerical strategy for minimizing $f(\boldsymbol{z})$ is to iteratively minimize a piecewise linear lower-approximation of f and refining this approximation at each step until some approximation error ε is reached, as described in Algorithm 3.1. This scheme was originally proposed for continuous decision variables by [40], and later extended to binary decision variables by [22], who provide a proof of termination in a finite, yet exponential in the worst case, number of iterations.

Algorithm 3.1 Outer-approximation scheme

Require: Initial solution $\boldsymbol{z}^1$
  $t \leftarrow 1$
  repeat
    Compute $\boldsymbol{z}^{t+1}, \eta^{t+1}$ solution of

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \eta} \boldsymbol{c}^\top \boldsymbol{z} + \eta \quad \text{s.t. } \forall s \in \{1, \ldots, t\},\ \eta \geq f(\boldsymbol{z}^s) + \nabla f(\boldsymbol{z}^s)^\top (\boldsymbol{z} - \boldsymbol{z}^s)$$

    Compute $f(\boldsymbol{z}^{t+1})$ and $\nabla f(\boldsymbol{z}^{t+1})$
    $t \leftarrow t+1$
  until $f(\boldsymbol{z}^{t+1}) - \eta^{t+1} \leq \varepsilon$
  return $\boldsymbol{z}^t$

To avoid solving a mixed-integer linear optimization problem at each iteration, as suggested in the pseudo-code, this strategy can be integrated within a single branch-and-bound procedure using lazy callbacks, as originally proposed by [48]. Lazy callbacks are now standard tools in commercial solvers such as Gurobi and CPLEX and provide significant speed-ups for outer-approximation algorithms. With this implementation, the commercial solver constructs a single branch-and-bound tree and generates a new cut at a feasible solution $\boldsymbol{z}$.

We remark that the second-stage minimization problem may be infeasible at some $\boldsymbol{z}^t$. In this case, we generate a feasibility cut rather than outer-approximation cut. In particular, the constraint $\sum_i z_i^t(1-z_i) + \sum_i (1-z_i^t) z_i \geq 1$ excludes the iterate $\boldsymbol{z}^t$ from the feasible set. Stronger feasibility cuts can be obtained by leveraging problem specific structure. For instance, when the feasible set satisfies $\boldsymbol{z}^t \notin \mathcal{Z} \implies \forall \boldsymbol{z} \leq \boldsymbol{z}^t,\ \boldsymbol{z} \notin \mathcal{Z}$, $\sum_i (1-z_i^t) z_i \geq 1$ is a valid feasibility cut. Alternatively, one can invoke conic duality if $g(\boldsymbol{x})$ generates a conic feasibility problem. Formally, assume

$$g(\boldsymbol{x}) = \begin{cases} \langle \boldsymbol{c}, \boldsymbol{x} \rangle, & \text{if } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \in \mathcal{K}, \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\mathcal{K}$ is a closed convex cone. This assumption gives rise to some loss of generality. Note, however, that all the examples in the previous section admit conic reformulations by taking appropriate Cartesian products of the linear, second-order and semidefinite cones [5]. Assuming that $g(\boldsymbol{x})$ is of the prescribed form, we have the dual conjugate

$$h(\boldsymbol{\alpha}) = \inf_{\boldsymbol{x}} \langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle - g(\boldsymbol{x}) = \max_{\boldsymbol{\pi}} \langle \boldsymbol{b}, \boldsymbol{\pi} \rangle + \begin{cases} 0, & \text{if } \boldsymbol{c} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\pi} \in \mathcal{K}^\star, \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\mathcal{K}^\star$ is the dual cone to $\mathcal{K}$. In this case, if some binary vector $\boldsymbol{z}$ gives rise to an infeasible subproblem, i.e., $f(\boldsymbol{z}) = +\infty$, then the conic duality theorem implies[5] that there is a *certificate* of infeasibility $(\boldsymbol{\alpha}, \boldsymbol{\pi})$ such that

$$\boldsymbol{c} - \boldsymbol{\alpha} - \boldsymbol{A}^\top \boldsymbol{\pi} \in \mathcal{K}^\star, \langle \boldsymbol{b}, \boldsymbol{\pi} \rangle > \sum_{i=1}^n z_i \Omega^\star(\alpha_i).$$

Therefore, to restore feasibility, we can simply impose the cut $\langle \boldsymbol{b}, \boldsymbol{\pi} \rangle \leq \sum_{i=1}^n z_i \Omega^\star(\alpha_i)$.

As mentioned in Section 1.2, the rate of convergence of outer-approximation schemes depends heavily on three criterion. We now provide practical guidelines on how to meet these criterion:

1. *Fast cut generation strategy:* To generate a cut, one solves the second-stage minimization problem (2.2) (or its dual) in $\boldsymbol{x}$, which contains no discrete variables and is usually orders of magnitude faster to solve than the original mixed-integer problem (1.1). Moreover, the minimization problem in $\boldsymbol{x}$ needs to be solved only for the coordinates x_i such that $z_i = 1$. In practice, this approach yields a sequence of subproblems of much smaller size than the original problem, especially if $\mathcal{Z}$ contains a cardinality constraint. For instance, for the sparse empirical risk minimization problem (2.4), each cut is generated by solving a subproblem with n observations and k features, where $k \ll p$. For this reason, we recommend generating cuts at binary $\boldsymbol{z}$'s, which are often sparser than continuous $\boldsymbol{z}$'s. This recommendation can be relaxed in cases where the separation problem can be solved efficiently even for dense $\boldsymbol{z}$'s; for instance, in uncapacitated facility location problems, each subproblem is a knapsack problem which can be solved by sorting [24]. If possible, we recommend theoretically analyzing the sparsity of the optimal solution a priori, to derive an explicit cardinality or budget constraint on $\boldsymbol{z}$ and ensure the sparsity of each incumbent solution.
2. *Cut selection rule in presence of degeneracy:* In the presence of degeneracy, selection criteria, such as Pareto optimality [43], have been proposed to accelerate convergence. However, these criteria are numerous, computationally expensive and all in all, can do more harm than good [46]. In an opposite direction, we recommend alleviating the burden of degeneracy by design, by imposing a ridge regularizer whenever degeneracy hinders convergence.
3. *Rich root node analysis:* As suggested in [23], providing the solver with as much information as possible at the root node can drastically improve convergence of outer-approximation methods. This is the topic of the next two sections. Restarting mechanisms, as described in [23, Section 5.2], could also be useful, although we do not implement them in the present paper.

These ingredients, and especially the ability to generate cuts efficiently, dictate which types of problems could benefit the most from our approach and which regularizer to use. Problems with an explicit cardinality constraint, for instance, would require a small subproblem to be solved at each iteration. For network design problems,

[5] We should note that this statement is, strictly speaking, not true unless we impose regularization. Indeed, the full conic duality theorem [5, Theorem 2.4.1] allows for the possibility that a problem is infeasible but asymptotically feasible, i.e.,

$$\nexists \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \mathcal{K} \text{ but } \exists \{\boldsymbol{x}_t\}_{t=1}^\infty : \boldsymbol{x}_t \in \mathcal{K}\ \forall t \text{ with } \|\boldsymbol{A}\boldsymbol{x}_t - \boldsymbol{b}\| \to 0.$$

Fortunately, the regularizer $\Omega(\boldsymbol{x})$ alleviates this issue, because it is coercive (i.e., "blows up" to $+\infty$ as $\|\boldsymbol{x}\| \to \infty$) and therefore renders all unbounded solutions infeasible and ensures the compactness of the level sets of $g(\boldsymbol{x}) + \Omega(\boldsymbol{x})$.

the network flow structure of the feasible set is a key numerical asset so we intuit that ridge regularization, which leaves the feasible set unchanged, would be very efficient. On the other hand, for uncapacitated facility location, sub-problems with big-M regularization boils down to a knapsack problem and can be solved efficiently via sorting, as discussed in [24, Section 3.1].

3.2. Improving the Lower-Bound: A Boolean Relaxation. To certify optimality, high-quality lower bounds are of interest and can be obtained by relaxing the integrality constraint $\boldsymbol{z} \in \{0,1\}^n$ in the definition of $\mathcal{Z}$ to $\boldsymbol{z} \in [0,1]^n$. In this case, the Boolean relaxation of (2.1) is:

$$\min_{\boldsymbol{z} \in \mathrm{Bool}(\mathcal{Z})} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}),$$

which can be solved using Kelley's algorithm [40], which is a continuous analog of Algorithm 3.1. Stabilization strategies have been empirically successful to accelerate the convergence of Kelley's algorithm, as recently demonstrated on uncapacitated facility location problems by [24]. However, for Boolean relaxations, Kelley's algorithm computes $f(\boldsymbol{z})$ and $\nabla f(\boldsymbol{z})$ at dense vectors $\boldsymbol{z}$, which is (sometimes substantially) more expensive than for sparse binary vectors $\boldsymbol{z}$'s, unless each subproblem can be solved efficiently as in [24].

Alternatively, the continuous minimization problem admits a reformulation

$$\min_{\boldsymbol{z} \in \mathrm{Bool}(\mathcal{Z})} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i \, \Omega^\star(\alpha_j). \tag{3.1}$$

analogous to Problem (2.14). Under Assumption 2.4, we can further write the min-max relaxation formulation (3.1) as a non-smooth maximization problem

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \ q(\boldsymbol{\alpha}), \quad \text{with} \quad q(\boldsymbol{\alpha}) := h(\boldsymbol{\alpha}) + \min_{\boldsymbol{z} \in \mathrm{Bool}(\mathcal{Z})} \sum_{i=1}^n (c_i - \Omega^\star(\alpha_i))\, z_i$$

and apply a projected sub-gradient ascent method as in [12]. We refer to [7, Chapter 7.5.] for a discussion on implementation choices regarding step-size schedule and stopping criteria, and [50] for recent enhancements using restarting.

The benefit from solving the Boolean relaxation with these algorithms is threefold. First, it provides a lower bound on the objective value of the discrete optimization problem (2.1). Second, it generates valid linear lower approximations of $f(\boldsymbol{z})$ to initiate the cutting-plane algorithm with. Finally, it supplies a sequence of continuous solutions that can be rounded and polished to obtain good binary solutions. Indeed, the Lipschitz continuity of $f(\boldsymbol{z})$ suggests that high-quality feasible binary solutions can be found in the neighborhood of a solution to the Boolean relaxation. We formalize this observation in the following theorem:

THEOREM 3.1. *Let $\boldsymbol{z}^\star$ denote a solution to the Boolean relaxation (3.1), $\mathcal{R}$ denote the indices of $\boldsymbol{z}^\star$ with fractional entries, and $\alpha^\star(\boldsymbol{z})$ denote a best choice of $\boldsymbol{\alpha}$ for a given $\boldsymbol{z}$. Suppose that for any $\boldsymbol{z} \in \mathcal{Z}$, $|\alpha^\star(\boldsymbol{z})_j| \leq L$. Then, a random rounding $\boldsymbol{z}$ of $\boldsymbol{z}^\star$, i.e., $z_j \sim$ Bernoulli$(z_j^\star)$, satisfies $0 \leq f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \leq \epsilon$ with probability at least $p = 1 - |\mathcal{R}| \exp\left(-\frac{\epsilon^2}{\kappa}\right)$, where*

$$\begin{aligned} &\kappa := 2M^2L^2|\mathcal{R}|^2 && \textit{for the big-}M \textit{ penalty,} \\ &\kappa := \tfrac{1}{2}\gamma^2 L^4 |\mathcal{R}|^2 && \textit{for the ridge penalty.} \end{aligned}$$

We provide a formal proof of this result in Appendix A.1. This result calls for multiple remarks:

- For $\varepsilon > \sqrt{\kappa \ln(|\mathcal{R}|)}$, we have that $p > 0$, which implies the existence of a binary ε-optimal solution in the neighborhood of $\boldsymbol{z}^\star$, which in turn bounds the integrality gap by ε. As a result, lower values of M or γ typically make the discrete optimization problem easier.
- A solution to the Boolean relaxation often includes some binary coordinates, i.e., $|\mathcal{R}| < n$. In this situation, it is tempting to fix $z_i = z_i^\star$ for $i \notin \mathcal{R}$ and solve the master problem (2.1) over coordinates in $\mathcal{R}$. In general, this approach provides sub-optimal solutions. However, Theorem 3.1 quantifies the price of fixing variables and bounds the optimality gap by $\sqrt{\kappa \ln(|\mathcal{R}|)}$.
- In the above high-probability bound, we do not account for the feasibility of the randomly rounded solution $\boldsymbol{z}$. Accounting for $\boldsymbol{z}$'s feasibility marginally reduces the probability given above, as shown for general discrete optimization problems by [49].

Under specific problem structure, other strategies might be more efficient than Kelley's method or the subgradient algorithm. For instance, if $\mathrm{Bool}(\mathcal{Z})$ is a polyhedron, then the inner minimization problem defining $q(\boldsymbol{\alpha})$ is a linear optimization problem that can be rewritten as a maximization problem by invoking strong duality. Although we only consider linear relaxations here, tighter bounds could be attained by taking a higher-level relaxation from a relaxation hierarchy, such as the [41] hierarchy [see 42, for a comparison]. The main benefit of such a relaxation is that while the aforementioned Boolean relaxation only controls the first moment of the probability measure studied in Theorem 3.1, higher level relaxations control an increasing sequence of moments of the probability measure and thereby provide non-worsening probabilistic guarantees for randomized rounding methods. However, the additional tightness of these bounds comes at the expense of solving relaxations with additional variables and constraints[6]; yielding a sequence of ever-larger semidefinite optimization problems. Indeed, even the SDP relaxation which controls the first two moments of a randomized rounding method is usually intractable when $n > 300$, with current technology. For an analysis of higher-level relaxations in sparse regression problems, we refer the reader to [2].

3.3. Improving the Upper-Bound: Local Search and Rounding.

To improve the quality of the upper-bound, i.e., the cost associated with the best feasible solution found so far, we implement two rounding and local-search strategies.

Our first strategy is a randomized rounding strategy, which is inspired by Theorem 3.1. Given $\boldsymbol{z}_0 \in \mathrm{Bool}(\mathcal{Z})$, we generate randomly rounded vectors $\boldsymbol{z}$ by sampling $\boldsymbol{z}$ according to $z_i \sim \mathrm{Bernoulli}(z_{0i})$ until $\boldsymbol{z} \in \mathcal{Z}$, which happens with high probability since $\mathbb{E}[\boldsymbol{z}] = \boldsymbol{z}_0$ satisfies all the constraints which describe $\mathcal{Z}$, besides integrality [49].

Our second strategy is a sequential rounding procedure, which is informed by the lower-approximation on $f(\boldsymbol{z})$, as laid out in Equation (2.16). Observing that the ith coordinate $\nabla f(\boldsymbol{z}_0)_i$ provides a first-order indication of how a change in z_i might impact the overall cost, we proceed in two steps. We first round down all coordinates such that $\nabla f(\boldsymbol{z}_0)_i(0 - \boldsymbol{z}_{0i}) < 0$. Once the linear approximation of f only suggests rounding up, we round all coordinates of $\boldsymbol{z}$ to 1 and iteratively bring some coordinates to 0 to restore feasibility.

If $\boldsymbol{z}_0$ is binary, we implement a comparable local search strategy. If $\boldsymbol{z}_{0i} = 0$, then

[6] n^2 additional variables and n^2 additional constraints for empirical risk minimization, versus $n+1$ additional variables and n additional constraints for the linear relaxation.

switching the ith coordinate to one increases the cost by at least $\nabla f(\boldsymbol{z}_0)_i$. Alternatively, if $\boldsymbol{z}_{0i} = 1$, then switching it to zero increases the cost by at least $-\nabla f(\boldsymbol{z}_0)_i$. We therefore compute the one-coordinate change which provides the largest potential cost improvement. However, as we only have access to a lower approximation of f, we are not guaranteed to generate a cost-decreasing sequence. Therefore, we terminate the procedure as soon as it cycles. A second complication is that, due to the constraints defining $\mathcal{Z}$, the best change sometimes yields an infeasible $\boldsymbol{z}$. In practice, for simple constraints such as $\boldsymbol{\ell} \leq \boldsymbol{z} \leq \boldsymbol{u}$, we forbid switches which break feasibility; for cardinality constraints, we perform the best switch and then restore feasibility at minimal cost when necessary.

3.4. Relationship With Perspective Cuts. In this section, we connect the perspective cuts introduced by [28] with our framework and discuss the merits of both approaches, in theory and in practice. To the best of our knowledge, a connection between Boolean relaxations of the two approaches has only been made in the context of sparse regression, by [54]. That is, the general connection we make here between the discrete problems, as well as their respective cut generating procedures, is novel.

We first demonstrate that imposing the ridge regularization term $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ naturally leads to the perspective formulation of [28]:

THEOREM 3.2. *Suppose that* $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ *and that Assumption 2.4 holds. Then, Problem* (2.14) *is equivalent to the following optimization problem:*

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \frac{1}{2\gamma}\sum_{i=1}^n \begin{cases} \dfrac{x_i^2}{z_i}, & \text{if } z_i > 0, \\ 0, & \text{if } z_i = 0 \text{ and } x_i = 0, \\ \infty, & \text{otherwise.} \end{cases} \tag{3.2}$$

Theorem 3.2 follows from taking the dual of the inner-maximization problem in Problem (2.15); see Appendix A.2 for a formal proof. Note that the equivalence stated in Theorem 3.2 also holds for $\boldsymbol{z} \in \mathrm{Bool}(\mathcal{Z})$. As previously observed in [5, 1], Problem (3.2) can be formulated as a second-order cone problem (SOCP)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}\in\mathcal{Z}, \boldsymbol{\theta}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \sum_{i=1}^n \theta_i \quad \text{s.t.} \quad \left\| \begin{pmatrix} \sqrt{\frac{2}{\gamma}} x_i \\ \theta_i - z_i \end{pmatrix} \right\|_2 \leq \theta_i + z_i \quad \forall i \in [n]. \tag{3.3}$$

and solved by linearizing the SOCP constraints into so-called perspective cuts, i.e., $\theta_i \geq \frac{1}{2\gamma}\bar{x}_i(2x_i - \bar{x}_i z_i), \forall \bar{x} \in \bar{\mathcal{X}}$, which have been extensively studied in the literature in the past fifteen years [28, 36, 20, 27, 2]. Observe that by separating Problem (3.2) into master and subproblems, an outer approximation algorithm yields the same cut (2.16) as in our scheme. In this regard, our approach supplies a new and insightful derivation of the perspective cut approach. It is worth noting that our proposal can easily be implemented within a standard integer optimization solver such as CPLEX or Gurobi using callbacks, while existing implementations of the perspective cut approach have required tailored branch-and-bound procedures [see, e.g., 28, Section 3.1].

3.5. Merits of Ridge, Big-M Regularization: Algorithmic Perspective. We now summarize the relative merits of applying either ridge or big-M regularization from an algorithmic perspective:

- As noted in our randomized rounding guarantees in Section 3.2, the two regularization methods provide comparable bound gaps when $2M \approx \gamma L$, while if $2M \ll \gamma L$, big-M regularization provides smaller gaps, and if $2M \gg \gamma L$, ridge regularization provides smaller gaps.

- For linear problems, ridge regularization limits dual degeneracy, while big-M regularization does not. This benefit, however, has to be put in balance with the extra runtime and memory requirements needed for solving a quadratic, instead of linear, separation problem.

In summary, the benefits of applying either big-M or ridge regularization are largely even and depend on the specific instance to be solved. In the next section, we perform a sequence of numerical experiments on the problems studied in Section 2.1, to provide empirical guidance on which regularization approach works best when.

4. Numerical Experiments.

In this section, we evaluate our single-tree cutting-plane algorithm, implemented in Julia 1.0 using CPLEX 12.8.0 and the Julia package `JuMP.jl` version 0.18.4 [21]. We compare our method against solving the natural big-M or MISOCP formulations directly, using CPLEX 12.8.0. All experiments were performed on one Intel Xeon E5 − 2690 v4 2.6GHz CPU core and using 32 GB RAM.

4.1. Overall Empirical Performance Versus State-of-the-Art.

In this section, we compare our approach to state-of-the-art methods, and demonstrate that our approach outperforms the state-of-the-art for several relevant problems.

4.1.1. Network Design.

We begin by evaluating the performance of our approach for the multi-commodity network design problem (2.3). We adapt the methodology of [36] and generate instances where each node $i \in [m]$ is the unique source of exactly one commodity ($k = m$). For each commodity $j \in [m]$, we generate demands according to $b^j_{j'} = \lfloor \mathcal{U}(5, 25) \rceil$ for $j' \neq j$ and $b^j_j = -\sum_{j' \neq j} b^j_{j'}$, where $\lfloor x \rceil$ is the closest integer to x and $\mathcal{U}(a, b)$ is a uniform random variable on $[a, b]$. We generate edge construction costs, c_e, uniformly on $\mathcal{U}(1, 4)$, and marginal flow circulation costs proportionally to each edge length[7]. The discrete set $\mathcal{Z}$ contains constraints of the form $\boldsymbol{z}_0 \leq \boldsymbol{z}$, where $\boldsymbol{z}_0$ is a binary vector which encodes existing edges. We generate graphs which contain a spanning tree plus pm additional randomly picked edges, with $p \in [4]$, so that the initial network is connected with $O(m)$ edges. We also impose a cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq (1 + 5\%)\boldsymbol{z}_0^\top \boldsymbol{e}$, which ensures that the network size increases by no more than 5%. For each edge, we impose a capacity $u_e \sim \lfloor \mathcal{U}(0.2, 1)B/A \rceil$, where $B = -\sum_{j=1}^m b^j_j$ is the total demand and $A = (1 + p)m$. We penalize the constraint $\boldsymbol{x} \leq \boldsymbol{u}$ with a penalty parameter $\lambda = 1,000$[8]. For big-M regularization, we set $M = \sum_j |b^j_j|$, and take $\gamma = \frac{2}{m(m-1)}$ for ridge regularization.

We apply our approach to large networks with 100s nodes, i.e., 10,000s edges, which is ten times larger than the state-of-the-art [38, 36], and compare the quality of the incumbent solutions after an hour, since no approach could terminate up to a satisfiable optimality gap within this time limit. Note that we define the quality of a solution as its cost in absence of regularization, although we might have augmented the original formulation with a regularization term to compute the solution. As a result, we can compare the performance big-M and ridge regularization directly, despite the fact that the optimization problems they solve are actually different. On the other hand, performance metrics that depend on the function being minimized,

[7] Nodes are uniformly distributed over the unit square $[0, 1]^2$. We fix the cost to be ten times the Euclidean distance.

[8] We do so to allow for a fair comparison between big-M and ridge regularization. By penalizing the capacity constraint, we remove a natural big-M regularization term and no regularization can be considered as more natural than the other.

Table 3: Best solution found after one hour on network design instances with m nodes and $(1+p)m$ initial edges. We report improvement, i.e., the relative difference between the solutions returned by CPLEX and the cutting-plane. Values are averaged over five randomly generated instances. For ridge regularization, we report the "unregularized" objective value, that is we fix $\boldsymbol{z}$ to the best solution found and resolve the corresponding sub-problem with big-M regularization. A "−" indicates that the solver could not finish the root node inspection within the time limit (one hour), and "Imp." is an abbreviation of improvement.

				Big-M			Ridge		Overall
m	p	unit	CPLEX	Cuts	Imp.	CPLEX	Cuts	Imp.	Imp.
40	0	$\times 10^9$	1.17	**1.16**	0.86%	1.55	**1.16**	24.38%	1.74%
80	0	$\times 10^9$	8.13	7.52	6.99%	9.95	**7.19**	26.74%	10.85%
120	0	$\times 10^{10}$	3.03	2.10	29.94%	−	**1.94**	−%	35.30%
160	0	$\times 10^{10}$	5.90	4.32	26.69%	−	**4.07**	−%	30.91%
200	0	$\times 10^{10}$	11.45	**7.78**	31.45%	−	**7.50**	−%	32.32%
40	1	$\times 10^8$	5.53	5.47	1.07%	5.97	**5.45**	8.74%	1.41%
80	1	$\times 10^9$	2.99	**2.94**	1.81%	3.16	2.95	6.78%	1.89%
120	1	$\times 10^9$	8.38	**7.82**	6.69%	−	**7.82**	−%	6.86%
160	1	$\times 10^{10}$	1.64	**1.54**	5.98%	−	**1.54**	−%	6.03%
200	1	$\times 10^{10}$	2.60	2.54	2.33%	−	**2.26**	−%	12.98%
40	2	$\times 10^8$	4.45	4.38	1.62%	4.76	**4.36**	8.27%	2.06%
80	2	$\times 10^9$	2.44	**2.31**	5.39%	2.46	**2.31**	5.97%	5.40%
120	2	$\times 10^9$	6.23	**5.89**	5.55%	−	**5.89**	−%	5.75%
160	2	$\times 10^{11}$	1.22	1.16	4.74%	−	**0.71**	−%	19.33%
200	2	$\times 10^{10}$	2.06	1.43	30.46%	−	**1.01**	−%	73.43%
40	3	$\times 10^8$	3.91	**3.85**	1.58%	4.13	**3.85**	6.73%	1.78%
80	3	$\times 10^9$	2.06	**1.94**	5.76%	2.04	**1.94**	5.44%	5.85%
120	3	$\times 10^9$	5.43	5.15	5.31%	−	**4.2**	−%	12.35%
40	4	$\times 10^8$	3.32	3.28	1.35%	3.53	**3.26**	7.71%	1.85%
80	4	$\times 10^9$	1.88	**1.77**	5.59%	−	**1.77**	−%	5.64%

such as the optimality gap, would not permit such a comparison. In 100 instances, our cutting plane algorithm with big-M regularization provides a better solution 94% of the time, by 9.9% on average, and by up to 40% for the largest networks. For ridge regularization, the cutting plane algorithm scales to higher dimensions than plain mixed-integer SOCP, returns solutions systematically better than those found by CPLEX (in terms of unregularized cost), by 11% on average. Also, ridge regularization usually outperforms big-M regularization, as reported in Table 3. Given how numerically challenging these optimization problems are, the optimality gaps returned by all methods are often uninformative ($> 100\%$) - see Section C Table 10. Still, we observe that, with big-M regularization, CPLEX systematically returns tighter optimality gaps that the cutting-plane approach, while with ridge regularization, the gaps obtained by the cutting-plane algorithm are tighter 86% of the times. All in all, even artificially added, ridge regularization improves the tractability of outer approximation.

4.1.2. Binary Quadratic Optimization. We study some of the binary quadratic optimization problems collated in the BQP library by [53]. Specifically, the bqp-$\{50, 100, 250, 500, 1000\}$ instances generated by [3], which have a cost matrix density of 0.1, and the be-100 and be-120.8 instances generated by [15], which re-

Table 4: Average runtime in seconds on binary quadratic optimization problems from the Biq-Mac library [53, 15]. Values are averaged over 10 instances. A "−" denotes an instance which was not solved because the approach did not respect the 32GB peak memory budget.

Instance	n	Average runtime (s)/Average optimality gap (%)			
		CPLEX-M	CPLEX-M-Triangle	Cuts-M	Cuts-M-Triangle
bqp-50	50	29.4	0.6	30.6	**0.4**
bqp-100	100	122.3	51.7	25.3%	**38.6**
bqp-250	250	1108.1%	83.5%	87.0%	**46.1%**
bqp-500	500	2055.8%	1783.3%	**157.3%**	410.7%
bqp-1000	1000	−	−	**260.9%**	−
be100	100	**79.7%**	208.0%	249.4%	201.2%
be120.8	120	**146.4%**	225.8%	264.1%	220.3%

spectively have cost matrix densities of 1.0 and 0.8. Note that these instances were generated as maximization problems, and therefore we consider a higher objective value to be better. We warm-start the cutting-plane approach with the best solution found after 10,000 iterations of Goemans-Williamson rounding [see 35]. We also consider imposing triangle inequalities [19] via lazy callbacks, for they substantially tighten the continuous relaxations.

Within an hour, only the bqp-50 and bqp-100 instances could be solved by any approach considered here, in which case cutting-planes with big-M regularization is faster than CPLEX (see Table 4). For instances which cannot be solved to optimality, although CPLEX has an edge in producing tighter optimality gaps for denser cost matrices, as depicted in Table 4, the cutting-plane method provides tighter optimality gaps for sparser cost matrices, and provides higher-quality solutions than CPLEX for all instances, especially as n increases (see Table 5).

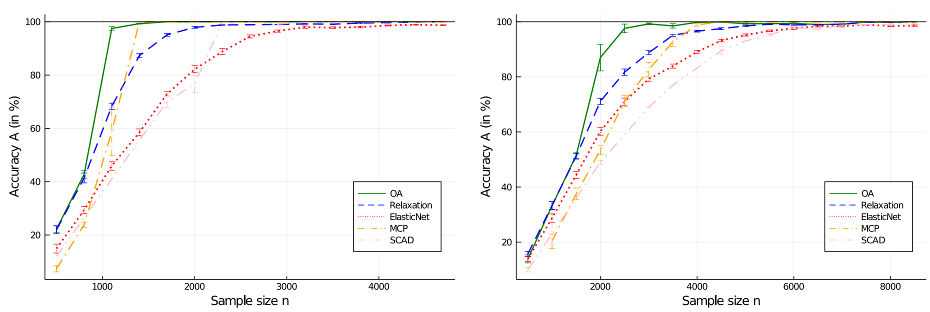
We remark that the cutting plane approach has low peak memory usage compared with the other methods: For the bqp-1000 instances, cutting-planes without triangle inequalities was the only method which respected the 32GB memory budget. This is another benefit of decomposing Problem (1.1) into master and sub-problems.

4.1.3. Sparse Empirical Risk Minimization. For sparse empirical risk minimization, our method with ridge regularization scales to regression problems with up $p = 100,000$s features and classification problems with $p = 10,000$s of features [12]. This constitutes a three-order-of-magnitude improvement over previous attempts using big-M regularization [10]. We also select features more accurately, as shown in Figure 1, which compares the accuracy of the features selected by the outer-approximation algorithm (in green) with those obtained from the Boolean relaxation (in blue) and other methods.

4.1.4. Sparse Principal Component Analysis. We applied our approach to sparse principal component analysis problems in [9], and by (a) introducing either big-M or ridge regularization and (b) introducing additional valid inequalities into the master problem, which we derived from the Gershgorin Circle Theorem [see 9, Section 2.3, for details] successfully solved problems where $p = 100$s to certifiable optimality, and problems where $p = 1000$s to certifiable near optimality, as reported

Table 5: Average incumbent objective value (higher is better) after 1 hour for medium-scale binary quadratic optimization problems from the Biq-Mac library [53, 15]. "–" denotes an instance which was not solved because the approach did not respect the 32GB peak memory budget. Values are averaged over 10 instances. Cuts-Triangle includes an extended formulation in the master problem.

Instance	n	Average objective value			
		CPLEX-M	CPLEX-M-Triangle	Cuts-M	Cuts-M-Triangle
bqp-250	250	9920.8	41843.4	**43774.9**	43701.5
bqp-500	500	19417.1	19659.0	**122879.3**	122642.4
bqp-1000	1000	–	–	**351450.7**	–
be100	100	16403.0	16985.0	17152.1	**17178.5**
be120.8	120	17943.2	19270.3	19307.7	**19371.2**



(a) Regression, $p = 20,000$ (b) Classification, $p = 10,000$

Fig. 1: Accuracy (A) of the feature selection method as the number of samples n increases, for the outer-approximation algorithm (in green), the solution found by the subgradient algorithm (in blue), ElasticNet (in red), MCP (in orange), SCAD (in pink) [see 12, for definitions]. Results are averaged over 10 instances of synthetic data with $(SNR, p, k) = (6, 20000, 100)$ for regression (left) and $(5, 10000, 100)$ for classification (right).

in Table 6; we refer to [9] for descriptions of the datasets studied and more extensive numerical experiments. This constitutes an order-of-magnitude improvement over existing certifiably near-optimal approaches, which rely on semidefinite techniques and therefore cannot scale to $p = 1000$s.

4.1.5. Sparse Portfolio Selection. We applied our approach to sparse portfolio selection problems in [8]. By introducing a ridge regularization term, we successfully solved instances to optimality at a scale of one order of magnitude larger than previous attempts as summarized in Table 7. Specifically, we optimized over the securities in the Wilshire 5000, which contains around 3,200 securities, an improvement upon existing techniques, which cannot currently scale beyond the securities in the $S\&P$ 500. Moreover, at smaller scales which existing techniques have been benchmarked on—including the set of synthetic instances generated by [28] with $200 - 400$

Table 6: Runtime in seconds per approach. We run all approaches on one thread, and impose a time limit of 600s. If a solver fails to converge, we report the relative bound gap at termination in brackets, and the no. explored nodes and cuts at the time limit. For ridge regularization, we set $\gamma = 100/k$.

Dataset	p	k	Big-M regularization			Ridge regularization		
			Time(s)	Nodes	Cuts	Time(s)	Nodes	Cuts
Pitprops	13	5	**0.09**	45	22	0.42	42	16
		10	**0.08**	223	223	0.68	615	244
Wine	13	5	**0.04**	143	69	0.10	73	36
		10	**0.09**	364	232	0.61	394	230
Miniboone	50	5	0.03	3	6	**0.01**	0	2
		10	**0.04**	4	6	0.07	10	13
Communities	101	5	**0.15**	109	2	0.54	272	55
		10	**0.44**	373	76	2.20	$1,800$	328
Arrhythmia	274	5	**5.27**	$1,080$	192	6.75	$1,242$	282
		10	**(4.21%)**	$61,000$	$11,600$	(4.63%)	$77,200$	$11,360$
Micromass	1300	5	**131.3**	$4,580$	4	163.2	4	$3,809$
		10	**378.6**	321	$16,090$	510.3	$21,700$	566

securities—our approach is as fast as and often faster than existing state-of-the-art approaches including [55, 27] among others [see 8, Section 5.2, for details].

Table 7: Largest sparse portfolio instances reliably solved by each approach

Reference	Solution method	Largest instance size solved (no. securities)
[31]	Perspective cut+SDP	400
[16]	Nonlinear B&B	200
[32]	Lagrangian relaxation B&B	300
[18]	Lagrangian relaxation B&B	300
[55]	SDP B&B	400
[27]	Approx. Proj. Perspective Cut	400
[8]	Algorithm 3.1 with ridge regularization	$3,200$

4.2. Evaluation of Different Ingredients in Our Numerical Recipe. We now consider the capacitated facility problem (2.8) on 112 real-world instances available from the OR-Library [3, 39], with the natural big-M and the ridge regularization with $\gamma = 1$. In both cases, the algorithms return the true optimal solution. Compared to CPLEX with big-M regularization, our cutting plane algorithm with big-M regularization is faster in 12.7% of instances (by 53.6% on average), and in 23.85% of instances (by 54.5% on average) when using a ridge penalty. This observation suggests that ridge regularization is better suited for outer-approximation, most likely because, as discussed in Section 3.1, a strongly convex ridge regularizer breaks the degeneracy of the separation problems. Note that our approach could benefit from multi-threading and restarting.

We take advantage of these instances to breakdown the independent contribution of each ingredient in our numerical recipe in Table 8. Although each ingredient contributes independently, jointly improving the lower and upper bounds provides the greatest improvement.

Table 8: Proportion of wins and relative improvement over CPLEX in terms of computational time on the 112 instances from the OR-library [3, 39] for different implementations of our method: an outer-approximation (OA) scheme with cuts generated at the root node using Kelley's method (OA + Kelley), OA with the local search procedure (OA + Local search) and OA with a strategy for both the lower and upper bound (OA + Both). Relative improvement is averaged over all "win" instances.

	Big-M		Ridge	
Algorithm	% wins	Relative improvement	% wins	Relative improvement
OA + Kelley	1.8%	36.6%	30.1%	91.6%
OA + Local search	1.9%	49.5%	19.4%	73.8%
OA + Both	12.7%	53.6%	92.5%	91.7%

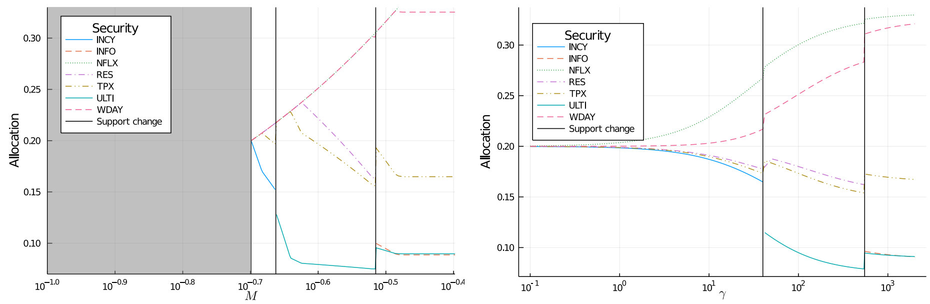
4.3. Big-M Versus Ridge Regularization. In this section, our primary interest is in ascertaining conditions under which it is advantageous to solve a problem using big-M or ridge regularization, and argue that ridge regularization is preferable over big-M regularization as soon as the objective is sufficiently strongly convex.

To illustrate this point, we consider large instances of the thermal unit commitment problem originally generated by [29], and multiply the quadratic coefficient a_i for each generator i by a constant factor $\alpha \in \{0.1, 1, 2, 5, 10\}$. Table 9 depicts the average runtime for CPLEX to solve both formulations to certifiable optimality, or provides the average bound-gap whenever CPLEX exceeds a time limit of 1 hour. Observe that when $\alpha \leq 1$, the big-M regularization is faster, but, when $\alpha > 1$ the MISOCP approach converges fast while the big-M approach does not converge within an hour. Consequently, ridge regularization performs more favorably whenever the quadratic term is sufficiently strong.

Table 9: Average runtime in seconds per approach, on data from [29] where the quadratic cost are multiplied by a factor of α. If the method did not terminate in one hour, we report the bound gap. n denotes the number of generators, each instances has 24 trade periods.

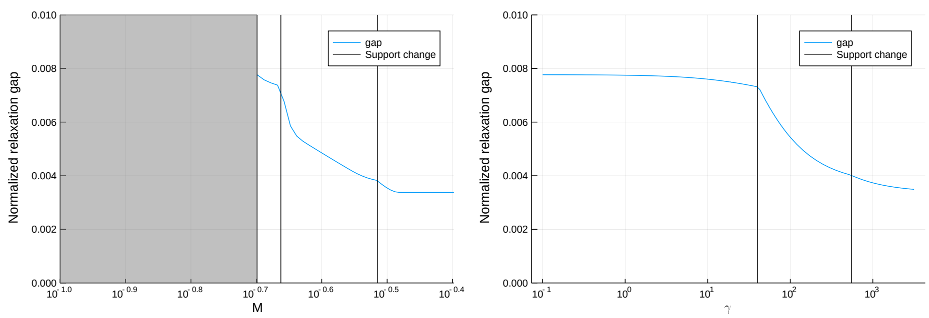
α	0.1		1		2		5		10	
n	Big-M	Ridge	Big-M	Ridge	Big-M	Ridge	Big-M	Ridge	Big-M	Ridge
100	**93.6**	299.0	**16.2**	229.4	0.32%	**47.9**	1.68%	**4.6**	2.76%	**6.0**
150	**35.6**	352.1	**6.2**	28.3	0.25%	**33.4**	1.69%	**6.4**	2.82%	**8.0**
200	**56.3**	138.1	**3.3**	239.7	0.24%	**112.9**	1.62%	**16.7**	2.81%	**21.2**

We also compare big-M and ridge regularization for the sparse portfolio selection problem (2.5). Figure 2 depicts the relationship between the optimal allocation of funds $\boldsymbol{x}^\star$ and the regularization parameter M (left) and γ (right), and Figure 3 depicts the magnitude of the gap between the optimal objective and the Boolean relaxation's objective, normalized by the unregularized objective. The two investment profiles are comparable, selecting the same stocks. Yet, we observe two main differences: First, setting $M < \frac{1}{k}$ renders the entire problem infeasible, while the problem remains feasible for any $\gamma > 0$. This is a serious practical concern in cases where a lower bound on the value of M is not known a priori. Second, the profile for ridge regularization seems smoother than its equivalent with big-M.



(a) Big-M regularization (b) Ridge regularization

Fig. 2: Optimal allocation of funds between securities as the regularization parameter (M or γ) increases. Data is obtained from the Russell 1000, with a cardinality budget of 5, a rank-200 approximation of the covariance matrix, a one-month holding period and an Arrow-Pratt coefficient of 1, as in [8]. Setting $M < \frac{1}{k}$ renders the entire problem infeasible.



(a) Big-M regularization (b) Ridge regularization

Fig. 3: Magnitude of the normalized absolute bound gap as the regularization parameter (M or γ) increases, for the portfolio selection problem studied in Figure 2

4.4. Relative Merits of Big-M, Ridge Regularization: An Experimental Perspective. We now conclude our comparison of big-M and ridge regularization, as initiated in Sections 2.4 and 3.5, by indicating the benefits of big-M and ridge regularization, from an experimental perspective:

- As observed in Section 4.3, big-M and ridge regularization play fundamentally the same role in reformulating logical constraints. This observation echoes our theoretical analysis in Section 2.
- As observed in the unit commitment and sparse portfolio selection problems studied in Section 4.3, ridge regularization should be the method of choice

whenever the objective function contains a naturally occurring strongly convex term, which is sufficiently large.

- As observed for network design and capacitated facility location problems in sections 4.1.1-4.2, ridge regularization is usually more amenable to outer-approximation than big-M regularization, because it eliminates most degeneracy issues associated with outer-approximating MINLOs.
- The efficiency of outer-approximation schemes relies on the speed at which separation problems are solved. In this regard, special problem-structure or cardinality constraints on the discrete variable $\boldsymbol{z}$ drastically help. This has been the case in network design, sparse empirical risk minimization and sparse portfolio selection problems in Section 4.1.1.

5. Conclusion. In this paper, we proposed a new interpretation of the big-M method, as a regularization term rather than a modeling trick. By expanding this regularization interpretation to include ridge regularization, we considered a wide family of relevant problems from the Operations Research literature and derived equivalent reformulations as mixed-integer saddle-point problems, which naturally give rise to theoretical analysis and computational algorithms. Our framework provides provably near-optimal solutions in polynomial time via solving Boolean relaxations and performing randomized rounding[9] as well as certifiably optimal solutions through an efficient branch-and-bound procedure, and indeed frequently outperforms the state-of-the-art in numerical experiments.

We believe our framework, which decomposes the problem into a discrete master problem and continuous subproblems, could be extended more generally to mixed-integer semidefinite optimization, as developed in [11, 9].

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

References.

[1] M. S. Aktürk, A. Atamtürk, and S. Gürel, *A strong conic quadratic reformulation for machine-job assignment with controllable processing times*, Oper. Res. Letters, 37 (2009), pp. 187–191.
[2] A. Atamtürk and A. Gomez, *Rank-one convexification for sparse regression*, arXiv:1901.10334, (2019).
[3] J. E. Beasley, *OR-library: distributing test problems by electronic mail*, J. Oper. Res. Soc., 41 (1990), pp. 1069–1072.
[4] N. Beaumont, *An algorithm for disjunctive programs*, Euro. J. Oper. Res., 48 (1990), pp. 362–371.
[5] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*, vol. 2, SIAM Philadelphia, PA, 2001.
[6] J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Num. Math., 4 (1962), pp. 238–252.
[7] D. P. Bertsekas, *Nonlinear programming: 3rd Edition*, Athena Scientific Belmont, 2016.
[8] D. Bertsimas and R. Cory-Wright, *A scalable algorithm for sparse portfolio selection*, arXiv:1811.00138, (2018).
[9] D. Bertsimas, R. Cory-Wright, and J. Pauphilet, *Solving large-scale sparse PCA to certifiable (near) optimality*, arXiv 2005.05195, (2020).
[10] D. Bertsimas, A. King, and R. Mazumder, *Best subset selection via a modern optimization lens*, Ann. Stat., 44 (2016), pp. 813–852.
[11] D. Bertsimas, J. Lamperski, and J. Pauphilet, *Certifiably optimal sparse inverse covariance estimation*, Math. Prog., 184 (2020), pp. 491–530.
[12] D. Bertsimas, J. Pauphilet, and B. Van Parys, *Sparse regression: Scalable algorithms and empirical performance*, Stat. Sci., 35 (2020), pp. 555–578.

[9] By "polynomial time", we mean with respect to the dimensionality of the relaxation, assuming the relaxation is solved to a fixed and finite precision and can be described using the symmetric cones described by [44], as occurs for all examples discussed in this paper.

[13] D. Bertsimas and B. Van Parys, *Sparse high-dimensional regression: Exact scalable algorithms and phase transitions*, Ann. Stat., 48 (2020), pp. 300–323.

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

[15] A. Billionnet and S. Elloumi, *Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem*, Math. Prog., 109 (2007), pp. 55–68.

[16] P. Bonami and M. A. Lejeune, *An exact solution approach for portfolio optimization problems under stochastic and integer constraints*, Oper. Res., 57 (2009), pp. 650–670.

[17] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, UK, 2004.

[18] X. Cui, X. Zheng, S. Zhu, and X. Sun, *Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems*, J. Glob. Opt., 56 (2013), pp. 1409–1423.

[19] M. M. Deza and M. Laurent, *Geometry of cuts and metrics*, vol. 15, Springer, 2009.

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

[21] I. Dunning, J. Huchette, and M. Lubin, *JuMP: A modeling language for mathematical optimization*, SIAM Rev., 59 (2017), pp. 295–320.

[22] M. A. Duran and I. E. Grossmann, *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Math. Prog., 36 (1986), pp. 307–339.

[23] M. Fischetti, I. Ljubić, and M. Sinnl, *Benders decomposition without separability: A computational study for capacitated facility location problems*, Euro. J. Oper. Res., 253 (2016), pp. 557–569.

[24] M. Fischetti, I. Ljubić, and M. Sinnl, *Redesigning Benders decomposition for large-scale facility location*, Mang. Sci., 63 (2016), pp. 2146–2162.

[25] R. Fletcher and S. Leyffer, *Solving mixed integer nonlinear programs by outer approximation*, Math. Prog., 66 (1994), pp. 327–349.

[26] R. Fortet, *Applications de l'algebre de boole en recherche opérationelle*, Revue Française de Recherche Opérationelle, 4 (1960), pp. 17–26.

[27] A. Frangioni, F. Furini, and C. Gentile, *Approximated perspective relaxations: a project and lift approach*, Comp. Opt. Appl., 63 (2016), pp. 705–735.

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

[29] A. Frangioni and C. Gentile, *Solving nonlinear single-unit commitment problems with ramping constraints*, Oper. Res., 54 (2006), pp. 767–775.

[30] A. Frangioni and C. Gentile, *SDP diagonalizations and perspective cuts for a class of nonseparable MIQP*, Oper. Res. Letters, 35 (2007), pp. 181–185.

[31] A. Frangioni and C. Gentile, *A computational comparison of reformulations of the perspective relaxation: Socp vs. cutting planes*, Oper. Res. Letters, 37 (2009), pp. 206–210.

[32] J. Gao and D. Li, *Optimal cardinality constrained portfolio selection*, Oper. Res., 61 (2013), pp. 745–761.

[33] A. M. Geoffrion, *Generalized Benders decomposition*, J. Opt. Theory Appl., 10 (1972), pp. 237–260.

[34] F. Glover, *Improved linear integer programming formulations of nonlinear integer problems*, Mang. Sci., 22 (1975), pp. 455–460.

[35] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[36] O. Günlük and J. Linderoth, *Perspective reformulations of mixed integer nonlinear programs with indicator variables*, Math. Prog., 124 (2010), pp. 183–205.

[37] O. Günlük and J. Linderoth, *Perspective reformulation and applications*, in Mixed Integer Nonlinear Programming, Springer, 2012, pp. 61–89.

[38] K. Holmberg and J. Hellstrand, *Solving the uncapacitated network design problem by a lagrangean heuristic and branch-and-bound*, Oper. Res., 46 (1998), pp. 247–259.

[39] K. Holmberg, M. Rönnqvist, and D. Yuan, *An exact algorithm for the capacitated facility location problems with single sourcing*, Euro. J. Oper. Res., 113 (1999), pp. 544–559.

[40] J. E. Kelley, Jr, *The cutting-plane method for solving convex programs*, J. Soc. Ind. Appl. Math., 8 (1960), pp. 703–712.

[41] J. B. Lasserre, *An explicit exact SDP relaxation for nonlinear 0-1 programs*, in International Conference on Integer Programming and Combinatorial Optimization, Springer, 2001, pp. 293–303.

[42] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming*, Math. Oper. Res., 28 (2003), pp. 470–496.

[43] T. L. Magnanti and R. T. Wong, *Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria*, Oper. Res., 29 (1981), pp. 464–484.
[44] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM, 1994.
[45] M. Padberg and G. Rinaldi, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Rev., 33 (1991), pp. 60–100.
[46] N. Papadakos, *Practical enhancements to the Magnanti–Wong method*, Oper. Res. Letters, 36 (2008), pp. 444–449.
[47] M. Pilanci, M. J. Wainwright, and L. El Ghaoui, *Sparse learning via Boolean relaxations*, Math. Prog., 151 (2015), pp. 63–87.
[48] I. Quesada and I. E. Grossmann, *An LP/NLP based branch and bound algorithm for convex MINLP optimization problems*, Comp. & Chem. Eng., 16 (1992), pp. 937–947.
[49] P. Raghavan and C. D. Tompson, *Randomized rounding: a technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7 (1987), pp. 365–374.
[50] J. Renegar and B. Grimmer, *A simple nearly-optimal restart scheme for speeding-up first order methods*, arXiv:1803.00151, (2018).
[51] P. Rigollet and J.-C. Hütter, *High dimensional statistics*, Lecture notes for course 18S997, (2015).
[52] J. P. Vielma, *Mixed integer linear programming formulation techniques*, SIAM Rev., 57 (2015), pp. 3–57.
[53] A. Wiegele, *Biq Mac library—a collection of Max-Cut and quadratic 0-1 programming instances of medium size*, tech. report, Alpen-Adria-Universität Klagenfurt, Austria, 2007.
[54] W. Xie and X. Deng, *Scalable algorithms for the sparse ridge regression*, SIAM J. Opt., 30 (2020), pp. 3359–3386.
[55] X. Zheng, X. Sun, and D. Li, *Improving the performance of MIQP solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach*, INFORMS J. Comp., 26 (2014), pp. 690–703.

Appendix A. Omitted Proofs.

A.1. Proof of Theorem 3.1: Quality of the Random Rounding Strategy.

Proof. We only detail the proof for the big-M regularization case, as the ridge regularization case follows *mutatis mutandis*. From Proposition 2.9,

$$0 \leq f(\boldsymbol{z}) - f(\boldsymbol{z}^\star) \leq ML|\mathcal{R}| \max_{\boldsymbol{\alpha} \geq \boldsymbol{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i.$$

The polyhedron $\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \geq \boldsymbol{0}, \|\boldsymbol{\alpha}\|_1 \leq 1\}$ admits $|\mathcal{R}| + 1$ extreme points. However, if

$$\max_{\boldsymbol{\alpha} \geq \boldsymbol{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i > t,$$

for some $t > 0$, then the maximum can only occur at some $\alpha > \boldsymbol{0}$ so that we can restrict our attention to the $|\mathcal{R}|$ positive extreme points. Applying tail bounds on the maximum of sub-Gaussian random variables over a polytope [see 51, Theorem 1.16], since $\|\boldsymbol{\alpha}\|_2 \leq \|\boldsymbol{\alpha}\|_1 \leq 1$, we have for any $t > 0$,

$$\mathbb{P}\left(\max_{\boldsymbol{\alpha} \geq \boldsymbol{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i > t \right) \leq |\mathcal{R}| \exp\left(-\frac{t^2}{2}\right),$$

so that

$$\mathbb{P}\left(ML|\mathcal{R}| \max_{\boldsymbol{\alpha} \geq \boldsymbol{0}: \|\boldsymbol{\alpha}\|_1 \leq 1} \sum_{i \in \mathcal{R}} (z_i^\star - z_i)\alpha_i > \varepsilon \right) \leq |\mathcal{R}| \exp\left(-\frac{\varepsilon^2}{2M^2L^2|\mathcal{R}|^2}\right). \qquad \square$$

A.2. Proof of Theorem 3.2: Relationship With Perspective Cuts.

Proof. Let us fix $\boldsymbol{z} \in \mathcal{Z}$. Then, we have that:

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \ h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\sum_{j=1}^n z_j \alpha_j^2 &= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \ h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\sum_{j=1}^n z_j\, \beta_j^2 \text{ s.t. } \boldsymbol{\beta} = \boldsymbol{\alpha}, \\
&= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \ \min_{\boldsymbol{x}} \ h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\sum_{j=1}^n z_j\, \beta_j^2 - \boldsymbol{x}^\top(\boldsymbol{\beta} - \boldsymbol{\alpha}), \\
&= \min_{\boldsymbol{x}} \ \underbrace{\max_{\boldsymbol{\alpha}} \ \left[h(\boldsymbol{\alpha}) + \boldsymbol{x}^\top \boldsymbol{\alpha}\right]}_{(-h)^\star(\boldsymbol{x}) = g(\boldsymbol{x})} + \sum_{i=1}^n \max_{\beta_i} \ \left[-\frac{\gamma}{2} z_i\, \beta_i^2 - x_i \beta_i\right].
\end{aligned}$$

Finally, observing that

$$\max_{\beta_i} \ \left[-\frac{\gamma}{2} z_i\, \beta_i^2 - x_i\beta_i\right] = \begin{cases} \dfrac{x_i^2}{2\gamma z_i} & \text{if } z_i > 0, \\ \max_{\beta_i} \ x_i\beta_i & \text{if } z_j = 0, \end{cases}$$

concludes the proof. ∎

Appendix B. Bounding the Lipschitz Constant. In our results, we relied on the observation that there exists some constant $L > 0$ such that, for any $\boldsymbol{z} \in \mathcal{Z}$, $\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\| \le L$. Such an L always exists, since $\mathcal{Z}$ is a finite set. However, as our randomized rounding results depend on L, explicit bounds on L are desirable.

We remark that while our interest is in the Lipschitz constant with respect to "$\boldsymbol{\alpha}$" in a generic setting, we have used different notation for some of the problems which fit in our framework, in order to remain consistent with the literature. In this sense, we are also interested in obtaining a Lipschitz constant with respect to $\boldsymbol{w}$ for the portfolio selection problem (2.5), among others.

In this appendix, we bound the magnitude of L in a less conservative manner. Our first result provides a bound on L which holds whenever the function $h(\boldsymbol{\alpha})$ in Equation (2.14) is strongly concave in $\boldsymbol{\alpha}$, which occurs for the sparse ERM problem (2.4) with ordinary least-squares loss, the unit commitment problem (2.7), the portfolio selection (2.5), and network design problems whenever $\boldsymbol{\Sigma}$ (resp. $\boldsymbol{Q}$) is full-rank:

LEMMA B.1. *Let $h(\cdot)$ be a strongly concave function with parameter $\mu > 0$ [see 17, Chapter 9.1.2 for a general theory of strong convexity], and suppose that $\boldsymbol{0} \in dom(g)$ and $\boldsymbol{\alpha}^\star := \arg\max_{\boldsymbol{\alpha}} \ h(\boldsymbol{\alpha})$. Then, for any choice of $\boldsymbol{z}$, we have*

$$\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\|_2^2 \le 8\,\frac{h(\boldsymbol{\alpha}^\star) - h(\boldsymbol{0})}{\mu},$$

i.e., $\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\|_\infty \le L$, where $L := 2\sqrt{2\frac{h(\boldsymbol{\alpha}^\star) - h(\boldsymbol{0})}{\mu}}$.

Proof. By the definition of strong concavity, for any $\boldsymbol{\alpha}$ we have

$$h(\boldsymbol{\alpha}) \le h(\boldsymbol{\alpha}^\star) + \nabla h(\boldsymbol{\alpha}^\star)^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star) - \frac{\mu}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_2^2,$$

where $\nabla h(\boldsymbol{\alpha}^\star)^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star) \le 0$ by the first-order necessary conditions for optimality, leading to

$$\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star\|_2^2 \le 2\,\frac{h(\boldsymbol{\alpha}^\star) - h(\boldsymbol{\alpha})}{\mu}.$$

In particular for $\boldsymbol{\alpha} = \mathbf{0}$, we have

$$\|\boldsymbol{\alpha}^\star\|_2^2 \leq 2\,\frac{h(\boldsymbol{\alpha}^\star) - h(\mathbf{0})}{\mu},$$

and for $\boldsymbol{\alpha} = \boldsymbol{\alpha}^\star(\boldsymbol{z})$,

$$\|\boldsymbol{\alpha}^\star(\boldsymbol{z}) - \boldsymbol{\alpha}^\star\|_2^2 \leq 2\,\frac{h(\boldsymbol{\alpha}^\star) - h(\mathbf{0})}{\mu},$$

since

$$h(\boldsymbol{\alpha}^\star(\boldsymbol{z})) \geq h(\boldsymbol{\alpha}^\star(\boldsymbol{z})) - \sum_{j=1}^n z_j \Omega_j^\star(\alpha^\star(\boldsymbol{z})_j) \geq h(\mathbf{0}).$$

The result then follows by the triangle inequality. ∎

An important special case of the above result arises for the sparse ERM problem, as we demonstrate in the following corollary to Lemma B.1:

Corollary B.2. *For the sparse ERM problem (2.4) with an ordinary least squares loss function and a cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$, a valid bound on the Lipschitz constant is given by*

$$\begin{aligned}\|\boldsymbol{\beta}^\star(\boldsymbol{z})\|_\infty = \|\mathrm{Diag}(\boldsymbol{Z})\boldsymbol{X}^\top \boldsymbol{\alpha}^\star(\boldsymbol{z})\|_\infty &\leq \|\mathrm{Diag}(\boldsymbol{Z})\boldsymbol{X}^\top\|_\infty \|\boldsymbol{\alpha}^\star(\boldsymbol{z})\|_\infty \\ &\leq \max_i \boldsymbol{X}_{i,[k]} \|\boldsymbol{\alpha}\|_2 \leq 2 \max_i \boldsymbol{X}_{i,[k]} \|\boldsymbol{y}\|_2,\end{aligned}$$

where $\boldsymbol{X}_{i,[k]}$ is the sum of the k largest entries in the column $\boldsymbol{X}_{i,[k]}$.

Proof. Applying Lemma B.1 yields the bound

$$\|\boldsymbol{\alpha}\|_2 \leq 2\|\boldsymbol{y}\|_2,$$

after observing that we can parameterize this problem in terms of $\boldsymbol{\alpha}$, and for this problem:

1. Setting $\boldsymbol{\alpha} = 0$ yields $h(\boldsymbol{\alpha}) = 0$.
2. $0 \leq h(\boldsymbol{\alpha}^\star) \leq \boldsymbol{y}^\top \boldsymbol{\alpha}^\star - \frac{1}{2}\boldsymbol{\alpha}^{\star\top}\boldsymbol{\alpha}^\star \leq \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{y}$.
3. $h(\cdot)$ is strongly concave in $\boldsymbol{\alpha}$, with concavity constant $\mu \geq 1$.

The result follows by applying the definition of the operator norm, and pessimizing over $\boldsymbol{z}$. ∎

Appendix C. Supplementary material for the numerical experiments.

In this section, we report additional performance metrics for the network design experiments presented in Section 4.1.1. There, we reported the quality of the solution returned by all methods within one hour, and compared two regularization strategies (big-M vs. ridge) and two algorithms (CPLEX vs. Cuts). Indeed, given the sizes of problems considered, the network design instances are computationally very challenging to solve. At such scales, finding a good feasible solution is already a very difficult task. In practice, this translates into optimality gaps that are often irrelevant (i.e., higher than 100%) in most of the instances. Table 10 reports the optimality gaps returned by each method after one hour, on instances where at least one of the four gaps was less than 100%.

Table 10: Optimality gap after one hour on network design instances with m nodes and $(1+p)m$ initial edges. We only report results for instances where the resulting gap was less than 100% for at least one of the four approaches. A "–" indicates that the solver could not finish the root node inspection within the time limit (one hour).

		Big-M		Ridge	
m	p	CPLEX	Cuts	CPLEX	Cuts
40	0	69.8%	100%	98.9%	96.7%
80	0	100%	100%	100%	100%
40	1	38.6%	100%	99.8%	97.1%
80	1	100%	100%	100%	95.6%
120	1	100%	100%	–	96.6%
40	2	23.3%	100%	$>$ 100%	97.7%
80	2	100%	100%	100%	96.3%
40	3	74.6%	$>$ 100%	97.5%	98.1%
40	4	100%	100%	99.2%	98.2%
80	4	100%	100%	100%	80.2%