A Unified Approach to Mixed-Integer Optimization: Nonlinear Formulations and Scalable Algorithms

Dimitris Bertsimas
Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA.
ORCID: 0000-0002-1985-1003
dbertsim@mit.edu

Ryan Cory-Wright
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA,
ORCID: 0000-0002-4485-0619
ryancw@mit.edu

Jean Pauphilet
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA, USA,
ORCID: 0000-0001-6352-0984
jpauph@mit.edu

Abstract: We propose a unified framework to address a family of classical mixed-integer optimization problems, including network design, facility location, unit commitment, sparse portfolio selection, binary quadratic optimization 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; nonlinear optimization

Subject classifications: programming: integer, applications; non-linear: quadratic; Facility Location; Network Design; Unit Commitment; Binary Quadratic Optimization

Area of review: Optimization

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 Glover (1975), 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 a numerically 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

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \in \mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0, \quad \forall i \in [n], \tag{1}$$

where $\mathcal{Z} \subseteq \{0,1\}^n$, $\mathbf{c} \in \mathbb{R}^n$ is a cost vector, $g(\cdot)$ is a generic convex function, and $\Omega(\cdot)$ is a convex regularization function; we formally state its structure in Assumption 1.

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). Second, we leverage the regularization term $\Omega(\boldsymbol{x})$ to derive a tractable reformulation of Problem (1). Finally, by invoking strong duality, we reformulate Problem (1) as a mixed-integer saddle-point problem, which is solvable via outer approximation.

Observe that the structure of Problem (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) encompasses a large number of problems from the operations research literature, such as the network design problem described in Example 1. In the past, those problems have typically been studied separately. However, the techniques developed for each problem are actually different facets of a single unified story, and, as we will demonstrate in this paper, can be applied to a much more general class of problems than is often appreciated.

Example 1. *Network design is an important example of problems of the form* (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:*

$$\min_{\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{x} \geq \boldsymbol{0}} \quad \boldsymbol{c}^\top \boldsymbol{z} + \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x} \quad \text{s.t.} \quad \begin{aligned} &\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \\ &x_e = 0 \text{ if } z_e = 0, \quad \forall e \in E, \end{aligned} \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ 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 semi-definite 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 (2) *is of the same form as Problem* (1) *with*

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

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 Glover (1975). This approach was first applied to mixed-integer non-linear optimization (MINLO) in the context of sparse portfolio selection by Bienstock (1996). 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 Beaumont 1990, Section 5).

This observation led Frangioni and Gentile (2006a) 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 Aktürk et al. (2009), who proved that MINLO problems with indicator variables can often be reformulated as mixed-integer second-order cone problems (see Günlük and Linderoth (2010) for a survey). More recently, a third approach for coupling the discrete and the continuous in MINLO was proposed independently for sparse regression by Pilanci et al. (2015) and Bertsimas and Van Parys (2019): 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 MINLO problems. The first such decomposition method is known as outer-approximation, and was proposed by Duran and Grossmann (1986), who proved its finite termination. The outer-approximation method was subsequently generalized to account for non-linear integral variables by Fletcher and Leyffer (1994). 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 greatly benefited from recent improvements in mixed-integer linear solvers in the past decades, beginning with the branch-and-cut approaches of (Padberg and Rinald 1991, Stubbs and Mehrotra 1999), 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 Fischetti et al. (2016a,b) 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 Fischetti et al. (2016a), 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 criterion of Magnanti and Wong (1981)).

In this paper, we argue that selecting a way to reformulate logical constraints constitutes a modeling choice, and connect this choice to the aforementioned key ingredients for modern decomposition schemes. We also argue that rather than making a modeling choice to fix a MINLO formulation and subsequently designing tractable algorithms for said formulation, optimizers should account for the tractability of their formulation when making their modeling choice.

1.3. Structure

We propose a single 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, 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)'s objective function.

In Section 3, we propose a conjunction of general-purpose numerical algorithms to solve Problem (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 recently proposed perspective cut approach (Frangioni and Gentile 2006a) 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, 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 $\varphi(\boldsymbol{x}, t) = t f(\boldsymbol{x}/t)$ if $t > 0$, $\varphi(\mathbf{0}, 0) = 0$, and ∞ elsewhere, is also convex (Boyd and Vandenberghe 2004, 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) 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) can be viewed as a discrete optimization problem:

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \quad \boldsymbol{c}^\top \boldsymbol{z} + f(\boldsymbol{z}), \tag{3}$$

where the inner minimization problem

$$f(\boldsymbol{z}) := \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad g(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \quad \text{s.t.} \quad x_i = 0 \text{ if } z_i = 0, \quad \forall i \in [n], \tag{4}$$

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

Example 2. *For the network design example* (2)*, we have*

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

2.1.1. Network Design Example 1 illustrates that the single-commodity network design problem is a special case of Problem (1). We now formulate the k-commodity network design problem with directed capacities:

$$\begin{aligned} f(\boldsymbol{z}) := \min_{\boldsymbol{f}^j\geq\boldsymbol{0},\boldsymbol{x}} \quad \boldsymbol{c}^\top \boldsymbol{z} + \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{x} \quad \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{f}^j = \boldsymbol{b}^j, \quad \forall j \in [k], \\ & \boldsymbol{x} = \sum_{j=1}^m \boldsymbol{f}^j, \\ & \boldsymbol{x} \leq \boldsymbol{u}, \\ & x_e = 0 \text{ if } z_e = 0, \quad \forall e \in E. \end{aligned} \tag{5}$$

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.,

$$f(\boldsymbol{z}) := \min_{\boldsymbol{w}\in\mathbb{R}^p} \quad \sum_{i=1}^n \ell\left(y_i, \boldsymbol{w}^\top \boldsymbol{x}_i\right) + \frac{1}{2\gamma}\|\boldsymbol{w}\|_2^2, \quad \text{s.t.} \quad w_j = 0 \text{ if } z_j = 0, \quad \forall j \in [p], \tag{6}$$

over $\mathcal{Z} = \{\boldsymbol{z} \in \{0,1\}^p : \boldsymbol{e}^\top \boldsymbol{z} \leq k\}$, 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 real-world interest.

Method	Loss function	Domain	Fenchel conjugate
Linear regression	$\frac{1}{2}(y-u)^2$	$y \in \mathbb{R}$	$\ell^\star(y,\alpha) = \frac{1}{2}\alpha^2 + \alpha y$
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 $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$, uncertainty budget parameter $\sigma > 0$, cardinality budget parameter $k \in \{2, ..., n-1\}$, 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 $\mathcal{Z} = \left\{\boldsymbol{z} \in \{0,1\}^n : \ \boldsymbol{e}^\top \boldsymbol{z} \leq k\right\}$ the function

$$f(\boldsymbol{z}) = \min_{\boldsymbol{x} \in \mathbb{R}^n_+} \quad \frac{\sigma}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{x} \quad \text{s.t.} \quad \begin{aligned} & \boldsymbol{l} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{u}, \\ & \boldsymbol{e}^\top \boldsymbol{x} = 1, \\ & x_i = 0 \text{ if } z_i = 0, \quad \forall i \in [n]. \end{aligned} \tag{7}$$

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 can be formulated as follows:

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

where:

$$f(\boldsymbol{z}) = \min_{\boldsymbol{x}} \quad \sum_{t \in \mathcal{T}} \sum_{i=1}^n \tfrac{1}{2} a_i x_{i,t}^2 + b_i x_{i,t} \quad \text{s.t.} \quad \begin{aligned} & \sum_{i=1}^n x_{i,t} \geq D_t, \quad \forall t \in \mathcal{T}, \\ & x_{i,t} \in [0, u_{i,t}], \quad \forall i \in [m], \forall t \in \mathcal{T}, \\ & x_{i,t} = 0 \text{ if } z_{i,t} = 0, \quad \forall i \in [m], \forall t \in \mathcal{T}. \end{aligned} \tag{8}$$

Observe that a necessary and sufficient condition for $f(\boldsymbol{z})$ to be feasible for a given choice of $\boldsymbol{z}$ is that $\sum_i z_{i,t} u_{i,t} \geq D_t, \forall t \in \mathcal{T}$. Therefore, we will impose this constraint in our first-stage problem apriori, in order to avoid the undesirable possibility of solving infeasible subproblems.

2.1.5. Facility Location Given a set of n potential facilities and m customers, the facility location problem consists in constructing facilities $i = 1, \dots, n$ at cost c_i in order to satisfy demand at minimal cost, i.e.,

$$\begin{aligned}
\min_{\boldsymbol{z} \in \{0,1\}^n} \min_{\boldsymbol{X} \in \mathbb{R}_+^{n \times m}} \boldsymbol{c}^\top \boldsymbol{z} + \sum_{j=1}^m \sum_{i=1}^n C_{ij} X_{ij} \quad \text{s.t.} \ & \sum_{j=1}^m X_{ij} \leq U_i, \quad \forall i \in [n], \\
& \sum_{i=1}^n X_{ij} = d_j, \quad \forall j \in [m], \\
& X_{ij} = 0 \ \text{ if } \ z_i = 0, \quad \forall i \in [n], j \in [m].
\end{aligned} \tag{9}$$

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. 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 solves:

$$\min_{\boldsymbol{z} \in \{0,1\}^n} \boldsymbol{z}^\top \boldsymbol{Q} \boldsymbol{z}. \tag{10}$$

This formulation is non-convex and does not include continuous variables. However, introducing auxiliary continuous variables yields an equivalent convex formulation (Glover and Woolsey 1974), which is given by:

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

We remark that the well-known triangle inequalities (see Deza and Laurent 2009, and references therein) substantially improve the quality of binary quadratic relaxations and can easily be added via lazy callbacks within numerical solvers, as well as within our numerical strategy. We will make use of these inequalities in our numerical experiments and state them for completeness:

$$\begin{aligned}
Y_{i,j} + Y_{i,k} + Y_{j,k} - z_i - z_j - z_k \geq -1, \quad & \forall i, j, k \in [n], \\
-Y_{i,j} - Y_{i,k} + Y_{j,k} + z_i \geq 0, \quad & \forall i, j, k \in [n], \\
-Y_{i,j} + Y_{i,k} - Y_{j,k} + z_j \geq 0, \quad & \forall i, j, k \in [n], \\
Y_{i,j} - Y_{i,k} - Y_{j,k} + z_k \geq 0, \quad & \forall i, j, k \in [n].
\end{aligned}$$

2.2. A Regularization Assumption

When we stated Problem (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 1. *In Problem* (1)*, the regularization term* $\Omega(\boldsymbol{x})$ *is one of:*

- *a big-M penalty function,* $\Omega(\boldsymbol{x}) = 0$ *if* $\|\boldsymbol{x}\|_\infty \le M$ *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 3. *In the network design example* (2)*, given the flow conservation structure* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$*, we have that* $\boldsymbol{x} \le M\boldsymbol{e}$*, 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 apriori*

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

and using q_i *as the ridge regularizer for each index* i *(Frangioni and Gentile 2007) .*

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.

2.3. Duality to the Rescue

In this section, we derive Problem (4)'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 1, 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 (4)'s dual, we require the following assumption:

ASSUMPTION 2. *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 six problems stated in Section 2.1 satisfy Assumption 2, as their inner problems are convex quadratics with linear constraints (see Boyd and Vandenberghe 2004, Section 5.2.3). Under Assumption 2, the following theorem reformulates Problem (3) as a saddle-point problem:

THEOREM 1. *Under Assumption 2, Problem* (3) *is equivalent to the following problem:*

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i \, \Omega^\star(\alpha_i), \tag{11}$$

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 Boyd and Vandenberghe 2004, Chap. 3.3), and

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

Proof of Theorem 1 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

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

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

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

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

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

where the last equality uses the fact that $z_i > 0$ 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]. □

Example 4. *For the network design problem* (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 (2) *is equivalent to minimizing over* $\boldsymbol{z} \in \mathcal{Z}$ *the function*

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

Theorem 1 reformulates the function $f(\boldsymbol{z})$ in Problem (3) as an inner maximization problem

$$f(\boldsymbol{z}) = \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i\, \Omega^\star(\alpha_i), \tag{12}$$

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 convexity and smoothness f.

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 (12), we have the following lower-approximation

$$f(\tilde{\boldsymbol{z}}) \geq f(\boldsymbol{z}) + \nabla f(\boldsymbol{z})^\top (\tilde{\boldsymbol{z}} - \boldsymbol{z}), \quad \forall \tilde{\boldsymbol{z}} \in \mathcal{Z}, \tag{13}$$

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 Bender's decomposition. Since the work of Magnanti and Wong (1981) on Pareto optimal cuts, many strategies have been proposed to improve the cut selection process in the presence of degeneracy (see Fischetti et al. 2016a, Section 4.4 for a review). However, the use of ridge regularization ensures that the objective function in (11) 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 1:

Proposition 1. *For any feasible (not necessarily binary) $\boldsymbol{z}$ and $\boldsymbol{z}'$,*

(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 of Proposition 1 By Equation (11),

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

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

Proposition 1 demonstrates that, because the coordinates of $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ are uniformly bounded[3] with respect to $\boldsymbol{z}$, $f(\boldsymbol{z})$ is Lipschitz-continuous, with a constant 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. Examples - Continued

We now derive the dual reformulation (11) for the examples from Section 2.1.

2.4.1. Network Design For the k-commodity problem network design problem (5) with capacity constraints $\boldsymbol{x} \leq \boldsymbol{u}$, the dual formulation (11) is given by:

$$h(\boldsymbol{\alpha}) = \max_{\boldsymbol{\beta}_u \geq \boldsymbol{0}, \boldsymbol{\xi}, \boldsymbol{p}^j} \quad \sum_{j=1}^{k} (\boldsymbol{b}^j)^\top \boldsymbol{p}^j - \boldsymbol{u}^\top \boldsymbol{\beta}_u - \tfrac{1}{2} \|\boldsymbol{\xi}\|_2^2 \quad \text{s.t } \boldsymbol{Q}^{1/2} \boldsymbol{\xi} \geq \boldsymbol{A}^\top \boldsymbol{p}^j - \boldsymbol{\beta}_u - \boldsymbol{d} + \boldsymbol{\alpha}, \quad \forall j \in [m].$$

When the transportation costs are linear and $\boldsymbol{Q} = \boldsymbol{0}$, we can eliminate the variables $\boldsymbol{\xi}$, by setting $\boldsymbol{\xi} = \boldsymbol{0}$. Modeling capacities through a hard constraint on $\boldsymbol{x}$ can be numerically challenging, as capacities are more likely to define infeasible first stage variables $\boldsymbol{z}$. To circumvent this difficulty, we can instead penalize $\max(\boldsymbol{x} - \boldsymbol{u}, \boldsymbol{0})$, by augmenting the primal objective with the term $\lambda \boldsymbol{e}^\top \max(\boldsymbol{x} - \boldsymbol{u}, \boldsymbol{0})$ for some large constant λ, or equivalently impose the constraint $\boldsymbol{\beta}_u \leq \lambda$ in the dual formulation.

2.4.2. Sparse Empirical Risk Minimization Let $\ell^\star$ denote the Fenchel conjugate of the loss function ℓ with respect to its second argument (see Table 1 for examples). Then, we can rewrite Problem (6) as (see Bertsimas et al. 2019b, for a proof):

$$f(\boldsymbol{z}) = \max_{\boldsymbol{\beta} \in \mathbb{R}^p} \quad h(\boldsymbol{\beta}) - \sum_{j=1}^{p} z_j \Omega^\star(\beta_j), \qquad \text{with} \qquad h(\boldsymbol{\beta}) := \max_{\boldsymbol{\alpha} \in \mathbb{R}^n : \boldsymbol{\beta} = \boldsymbol{X}^\top \boldsymbol{\alpha}} \quad - \sum_{i=1}^{n} \ell^\star(y_i, \alpha_i).$$

2.4.3. Sparse Portfolio Selection After either imposing an additional ridge regularization penalty term $1/2\gamma \|\boldsymbol{x}\|_2^2$ or decomposing $\boldsymbol{\Sigma}$ into a diagonal matrix $\boldsymbol{D}$ plus a low-rank matrix $\boldsymbol{V}^\top \boldsymbol{F} \boldsymbol{V}$ and using the term $\boldsymbol{x}^\top \boldsymbol{D} \boldsymbol{x}$ as a ridge regularization with a different value of γ for each coordinate x_i^2, we can rewrite Problem (6) as:

$$\begin{aligned} f(\boldsymbol{z}) = \max_{\substack{\boldsymbol{w} \in \mathbb{R}^n,\ \boldsymbol{\alpha} \in \mathbb{R}^r, \\ \boldsymbol{\beta}_l,\ \boldsymbol{\beta}_u \in \mathbb{R}^m_+,\ \lambda \in \mathbb{R}}} \quad & \boldsymbol{y}^\top \boldsymbol{\alpha} - \tfrac{1}{2}\|\boldsymbol{\alpha}\|_2^2 + \boldsymbol{\beta}_l^\top \boldsymbol{l} - \boldsymbol{\beta}_u^\top \boldsymbol{u} + \lambda - \tfrac{\gamma}{2} \sum_i z_i w_i^2 \\ \text{s.t.} \quad & \boldsymbol{w} \geq \boldsymbol{X}^\top \boldsymbol{\alpha} + \lambda \boldsymbol{e} + \boldsymbol{A}^\top (\boldsymbol{\beta}_l - \boldsymbol{\beta}_u) - \boldsymbol{d}. \end{aligned}$$

where $\boldsymbol{X} := \sqrt{\boldsymbol{\Sigma}}$ denotes the square root of the covariance matrix and $\boldsymbol{y}$, $\boldsymbol{d}$ are the projections[4] of $\boldsymbol{\mu}$ onto the span and nullspace of $\boldsymbol{X}$ (see Bertsimas and Cory-Wright 2018, for a proof).

2.4.4. Unit Commitment The DC-load-flow unit commitment problem (8) fits in our framework without any modification, as the quadratic cost function provides an implicit ridge regularization term, and there is a natural big-M constraint as well. Therefore, after letting $d_i := \frac{b_i}{\sqrt{a_i}}$, $\boldsymbol{A} := \text{Diag}(\boldsymbol{a})$, the minimization problem is equivalent to the following saddle-point problem:

$$\begin{aligned} \min_{\boldsymbol{z} \in \mathcal{Z}} \quad \sum_{t \in \mathcal{T}} \Bigg(\max_{\substack{\boldsymbol{\alpha}_t \in \mathbb{R}^n, \lambda_t \in \mathbb{R}_+ \\ \boldsymbol{\beta}_{\ell,t}, \boldsymbol{\beta}_{u,t} \in \mathbb{R}^n_+}} \quad & D_t \lambda_t - \boldsymbol{u}^\top \boldsymbol{\beta}_{u,t} + \sum_{i=1}^n z_{i,t} c_i - \tfrac{1}{2} \sum_{i=1}^n z_{i,t} \left(\alpha_{i,t} - d_i\right)^2 \Bigg) \\ \text{s.t.} \quad & \boldsymbol{A}^\top \boldsymbol{\alpha}_t = \lambda_t \boldsymbol{e} + \boldsymbol{\beta}_{\ell,t} - \boldsymbol{\beta}_{u,t},\ \forall t \in \mathcal{T}. \end{aligned}$$

Here, λ_t, $\boldsymbol{\beta}_{\ell,t}$ and $\boldsymbol{\beta}_{u,t}$ are the dual variables associated with the constraints $\boldsymbol{e}^\top \boldsymbol{x}_t \geq D_t$, $\boldsymbol{x}_t \geq 0$ and $\boldsymbol{x}_t \leq \boldsymbol{u}$ respectively.

2.4.5. Facility Location The facility location problem (9) admits a natural big-M regularization by taking $M = \min(U_i, d_j)$ for each X_{ij}, and its dual formulation (11) involves:

$$\begin{aligned} h(\boldsymbol{\alpha}) &= \min_{\boldsymbol{X} \in \mathbb{R}^{n \times m}_+} \quad (\boldsymbol{C} - \boldsymbol{\alpha})^\top \boldsymbol{X} \ \text{ s.t. } \boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{u},\ \boldsymbol{X}^\top \boldsymbol{e} = \boldsymbol{d} \\ &= \max_{\boldsymbol{p} \in \mathbb{R}^m, \boldsymbol{\beta}_u \in \mathbb{R}^n_+} \quad \boldsymbol{d}^\top \boldsymbol{p} - \boldsymbol{u}^\top \boldsymbol{\beta}_u \ \text{ s.t. } \boldsymbol{e}\boldsymbol{p}^\top - \boldsymbol{\beta}_u \boldsymbol{e}^\top \leq \boldsymbol{C} - \boldsymbol{\alpha}. \end{aligned}$$

2.4.6. Binary Quadratic Optimization In the binary quadratic optimization problem, there are two families of logical constraints. Therefore, we require two regularization terms. After imposing them, the dual formulation of $f(\boldsymbol{z})$ is:

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\theta} \in \mathbb{R}^{n \times n}} \quad & -\sum_{i,j=1}^n \rho_{i,j} + \sum_{i,j=1: j \neq i}^n \theta_{i,j}(z_i + z_j - 1) + \sum_{i=1}^n \theta_{i,i} z_i - \sum_{i,j=1}^n \left[z_i \Omega^\star(\alpha_{i,j}) + z_j \Omega^\star(\beta_{i,j}) \right] \\ \text{s.t.} \quad & \boldsymbol{Q} \geq \boldsymbol{\theta} + \boldsymbol{\alpha} + \boldsymbol{\beta} - \boldsymbol{\rho}, \quad \boldsymbol{\rho}, \boldsymbol{\theta} \geq \boldsymbol{0}. \end{aligned}$$

2.5. Relative Merits of Ridge, Big-M Regularization: A Theoretical Perspective

In this section, we proposed 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 regularization paradigm 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 will be the topic of the following two sections.

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

Regularization	Characteristics
Big-M	(+) Linear constraints (+) Leads to the same objective value 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 value for any $\gamma > 0$ (+) Preserves the feasible set

3. An Efficient Numerical Approach

In this section, we present an efficient numerical approach to solve Problem (11). 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 problem's Boolean relaxation to improve the duality gap.

3.1. Overall Outer-Approximation Scheme

Theorem 1 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 $f(\boldsymbol{z})$, as outlined in Equation (13). Consequently, a valid numerical strategy for minimizing $f(\boldsymbol{z})$ comprises iteratively minimizing 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 1. This scheme was originally proposed for continuous decision variables by Kelley (1960), and later extended to binary decision variables by Duran and Grossmann (1986), who provide a proof of termination in a finite, yet exponential in the worst case, number of iterations.

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. Lazy callbacks are now standard tools in commercial solvers such as Gurobi and CPLEX

Algorithm 1 Outer-approximation scheme

```
Require: Initial solution $\boldsymbol{z}^1$
  $t \leftarrow 1$
  repeat
    Compute $\boldsymbol{z}^{t+1}, \eta^{t+1}$ solution of
      $$\min_{\boldsymbol{z} \in \mathcal{Z}, \eta} \boldsymbol{c}^\top \boldsymbol{z} + \eta \quad \text{s.t. } \forall s \in \{1, \dots, t\}, \, \eta \geq f(\boldsymbol{z}^s) + \nabla f(\boldsymbol{z}^s)^\top (\boldsymbol{z} - \boldsymbol{z}^s)$$
    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 $\boldsymbol{z}^t$
```

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 when 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[5].

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 (4) (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). 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 (6), 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 (Fischetti et al. 2016b). We also recommend adding a cardinality or budget constraint on $\boldsymbol{z}$, to ensure the sparsity of each incumbent solution, and to restrict the number of feasible choices of $\boldsymbol{z}$.

2. *Cut selection rule in presence of degeneracy:* In the presence of degeneracy, selection criteria, such as Pareto optimality (Magnanti and Wong 1981), have been proposed to accelerate converge. However, these criteria are numerous, computationally expensive and all in all, can do more harm than good (Papadakos 2008). In an opposite direction, we recommend alleviate the burden of degeneracy by design, by imposing a ridge regularizer whenever degeneracy appears to hinder convergence.

3. *Rich root node analysis:* As suggested in Fischetti et al. (2016a), 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 Fischetti et al. (2016a, Section 5.2), could also be useful, although we do not implement them in the present paper.

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 $\boldsymbol{z} \in \{0,1\}^n$ to $\boldsymbol{z} \in [0,1]^n$. In this case, the Boolean relaxation of (3) is:

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

which can be solved using Kelley's algorithm (Kelley 1960), which is a continuous analog of Algorithm 1. Stabilization strategies have been empirically successful to accelerate the convergence of Kelley's algorithm, as recently demonstrated on uncapacitated facility location problems by Fischetti et al. (2016b). However, for Boolean relaxations, Kelleys'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 Fischetti et al. (2016b).

Alternatively, the continuous minimization problem admits a saddle-point reformulation

$$\min_{\boldsymbol{z} \in \mathrm{Conv}(\mathcal{Z})} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m} \quad \boldsymbol{c}^\top \boldsymbol{z} + h(\boldsymbol{\alpha}) - \sum_{i=1}^n z_i \, \Omega^\star(\alpha_j). \tag{14}$$

analogous to Problem (11). Under Assumption 2, we can further write the min-max relaxation formulation (14) as a non-smooth maximization problem

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

and apply a projected sub-gradient ascent method as in Bertsimas et al. (2019b). We refer to (Bertsekas 2016, Chapter 7.5.) for a discussion on implementation choices regarding step-size schedule and stopping criterion, and Renegar and Grimmer (2018) 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 (3). 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 2. *Let $\boldsymbol{z}^\star$ denote a solution to the Boolean relaxation* (14)*, $\mathcal{R}$ 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., $z_j \sim Bernouilli(z_j^\star)$, 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*

$$\kappa := 2M^2L^2|\mathcal{R}|^2 \quad \textit{for the big-M penalty,}$$

$$\kappa := \tfrac{1}{2}\gamma^2L^4|\mathcal{R}|^2 \quad \textit{for the ridge penalty.}$$

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 (3) over coordinates in $\mathcal{R}$. In general, this approach provides sub-optimal solutions. However, Theorem 2 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 Raghavan and Tompson (1987).
- If $\mathcal{R}$ is empty, by the probabilistic method, the relaxation is tight and $\boldsymbol{z}^\star$ solves Problem (11).

Under specific problem structure, other strategies might be more efficient than Kelley's method or the subgradient algorithm. For instance, if $\mathcal{Z}$ 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 Lasserre (2001) hierarchy (see Laurent 2003, 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 2, 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 Atamturk and Gomez (2019).

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

In order 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 2. Given $\boldsymbol{z}_0 \in \mathrm{Conv}(\mathcal{Z})$, we generate randomly rounded vectors $\boldsymbol{z}$ such that $z_i \sim \mathrm{Bernoulli}(z_{0i})$ and $\boldsymbol{z} \in \mathcal{Z}$.

Our second strategy is a sequential rounding procedure, which is informed by the lower-approximation on $f(\boldsymbol{z})$, as laid out in Equation (13). 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 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. Sensitivity Analysis and Warm-Starts

The regularization term $\Omega(\boldsymbol{x})$ in the objective function is sometimes added artificially. In this case, it may be necessary to solve the problem for various values of M or γ, to either identify the regularization value which performs best out-of-sample, or approximate the non-regularized problem by setting $M \to +\infty$ or $\gamma \to +\infty$. This situation appears notably for the sparse ERM problem (6) and the sparse portfolio selection problem (7), where hyper-parameters are tuned using a computationally expensive cross-validation procedure.

For both of the aforementioned problems, the optimal support indices $\boldsymbol{z}$ obtained for lower values of M or γ can serve as a high-quality warm-start for problem with higher regularization values. We bound the quality of these warm-starts in the following proposition:

PROPOSITION 2. *Let $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ be an optimal choice of $\boldsymbol{\alpha}$ for a fixed choice of $\boldsymbol{z}$ in Problem* (11)*, and suppose that for any feasible $\boldsymbol{z} \in \mathcal{Z}$, $|\alpha^\star(\boldsymbol{z})_i| \leq L$ and $\|\boldsymbol{z}\|_1 \leq k$. Then, it follows that the regularized problem with parameter $M + \Delta$ (resp. $\gamma + \Delta$) has an optimal objective value within ΔLk (resp. $\frac{1}{2}\Delta L^2 k$) of the objective with regularization parameter M (resp. γ).*

Proof of Proposition 2 We detail the proof for ridge regularization, but it can be adapted to big-M regularization in a straightforward manner. Suppose that $\hat{\boldsymbol{z}} \in \mathcal{Z}$ is an optimal choice of $\boldsymbol{z}$ with regularization parameter γ, and let $f_\gamma(\boldsymbol{z})$ denote the optimal objective value of Problem (11) for a fixed choice of $\boldsymbol{z}$ and fixed regularizer γ. Then, we have that:

$$
\begin{aligned}
f_\gamma(\hat{\boldsymbol{z}}) - \min_{\boldsymbol{z}' \in \mathcal{Z}} f_{\gamma+\Delta}(\boldsymbol{z}') &= \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \left(h(\boldsymbol{\alpha}) - \frac{\gamma}{2} \sum_{i=1}^n \hat{z}_i \alpha_i^2 \right) - \min_{\boldsymbol{z}' \in \mathcal{Z}} \max_{\boldsymbol{\alpha}' \in \mathbb{R}^n} \left(h(\boldsymbol{\alpha}') - \frac{\gamma}{2} \sum_{i=1}^n \hat{z}_i \alpha_i'^2 \right), \\
&\leq \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \min_{\boldsymbol{\alpha}' \in \mathbb{R}^n} h(\boldsymbol{\alpha}) - h(\boldsymbol{\alpha}') - \frac{\gamma}{2} \sum_{i=1}^n \hat{z}_i (\alpha_i^2 - \alpha_i'^2) + \frac{\Delta}{2} \sum_{i=1}^n \hat{z}_i \alpha_i'^2, \\
&\leq \max_{\boldsymbol{\alpha}' \in \mathbb{R}^n} \frac{\Delta}{2} \sum_{i=1}^n \hat{z}_i \alpha_i'^2,
\end{aligned}
$$

where the first inequality follows from setting $\hat{\boldsymbol{z}} = \boldsymbol{z}'$, the second inequality follows from setting $\boldsymbol{\alpha} = \boldsymbol{\alpha}'$, and the result follows by invoking the inequalities $|\alpha^\star(\boldsymbol{z})_i| \leq L$, $\|\boldsymbol{z}\|_1 \leq k$. □

From the proof, we note that the bound is attained by setting $\hat{\boldsymbol{z}} = \boldsymbol{z}'$. Therefore, the proof also bounds the suboptimality of the support indices $\hat{\boldsymbol{z}}$ as a warm-start with regularization $\gamma + \Delta$.

Similarly, if a cardinality constraint right-hand-side is perturbed slightly, the solution obtained for a lower value of k often serves as a high-quality warm-start.

PROPOSITION 3. *Suppose that the set of feasible $\boldsymbol{z} \in \mathcal{Z}$ for Problem* (11) *contains a cardinality constraint, i.e., $\mathcal{Z} = \mathcal{Z}' \cap \{\boldsymbol{z} \in \{0,1\}^n : \boldsymbol{e}^\top \boldsymbol{z} \leq k\}$, and let $\boldsymbol{\alpha}^\star(\boldsymbol{z})$ be an optimal choice of $\boldsymbol{\alpha}$ for a fixed choice of $\boldsymbol{z}$ in Problem* (11)*. Suppose that the cardinality constraint $\boldsymbol{e}^\top \boldsymbol{z} \leq k$ is relaxed to $\boldsymbol{e}^\top \boldsymbol{z} \leq k + \Delta$. Then, the regularized problem with parameter M (resp. γ) and cardinality budget $k + \Delta$ has an objective value within ΔLM (resp. $\frac{1}{2}\gamma\Delta L^2$) of the previous objective value.*

Proof This result follows from the same argument as Proposition 2, *mutatis mutandis.* □

The above bound is attained by only setting z_i's equal to 1 in a more restricted subset if they feature in a less restricted subset. This result suggests that injecting an optimal solution with cardinality budget $k - t$ (padded out with t randomly selected z_i's) provides a high-quality warm-start when t is small.

3.5. Relationship With Perspective Cuts

In this section, we connect the perspective cuts introduced by Frangioni and Gentile (2006a) 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 Xie and Deng (2018). That is, the general connection we make here between the discrete optimization 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 Frangioni and Gentile (2006a):

Theorem 3. *Suppose that $\Omega(\boldsymbol{x}) = \frac{1}{2\gamma}\|\boldsymbol{x}\|_2^2$ and that Assumption 2 holds. Then, Problem* (11) *is equivalent to the following optimization problem:*

$$\min_{\boldsymbol{z}\in\mathcal{Z}} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \frac{1}{2\gamma}\sum_{i=1}^n \begin{cases} \dfrac{x_i^2}{z_i}, & \textit{if } z_i > 0, \\ 0, & \textit{if } z_i = 0 \textit{ and } x_i = 0, \\ \infty, & \textit{otherwise.} \end{cases} \tag{15}$$

Theorem 3 follows from taking the dual of the inner-maximization problem in Problem (12); see Appendix A.2 for a formal proof. Note that the equivalence stated in Theorem 3 also holds for $\boldsymbol{z} \in \mathrm{Conv}(\mathcal{Z})$. As previously observed in Aktürk et al. (2009), Problem (15) can be formulated as a second-order cone problem (SOCP)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}\in\mathcal{Z}, \boldsymbol{\theta}\in\mathbb{R}^n} \quad \boldsymbol{c}^\top \boldsymbol{z} + g(\boldsymbol{x}) + \sum_{i=1}^n \theta_i \quad \text{s.t.} \quad \left\| \begin{pmatrix} \sqrt{\frac{2}{\gamma}} x_i \\ \theta_i - z_i \end{pmatrix} \right\|_2 \le \theta_i + z_i, \quad \forall i \in [n]. \tag{16}$$

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 (Frangioni and Gentile 2006a, Günlük and Linderoth 2010, Frangioni et al. 2016). Observe that by separating Problem (15) into master and subproblems, an outer approximation algorithm would yield the same cut (13) as in our scheme. However, while theoretically similar, there are subtle differences which make the computational performance of our proposal more attractive:

- Our outer-approximation approach only includes cuts corresponding to optimal choices of $\boldsymbol{x}$ for a given incumbent solution $\boldsymbol{z}$. Alternatively, the perspective cut approach is a generalized Benders decomposition scheme (see Geoffrion 1972) which generates the optimal cut with respect to $(\boldsymbol{z}, \boldsymbol{x})$. Consequently, the perspective cut approach possesses weaker convergence properties.
- Our outer-approximation scheme can easily be implemented within a standard integer optimization solver such as CPLEX or Gurobi using callbacks. Unfortunately, a generalized Benders approach for the perspective formulation requires a tailored branch-and-bound procedure (see Frangioni and Gentile 2006a, Section 3.1 for details). In this regard, our approach is more practical.

3.6. Relative Merits of Ridge, Big-M Regularization: An 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 numerically evaluate our cutting-plane algorithm, implemented in Julia 1.0 using CPLEX 12.8.0 and the Julia package `JuMP.jl` version 0.18.4 (Dunning et al. 2017). 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. The big-M and MISOCP formulations are allocated 8 threads, unless explicitly stated otherwise. However, `JuMP.jl` version 0.18.4 does not allow our outer-approximation scheme to benefit from multi-threading.

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 (5). We adapt the methodology of Günlük and Linderoth (2010) 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

$$b^j_{j'} = \lfloor \mathcal{U}(5,25) \rceil, \ \forall j' \neq j \text{ and } b^j_j = -\sum_{j' \neq j} b^j_{j'},$$

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 sparse[8] and connected. 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 $\lambda = 1,000$.

We apply our approach to large networks with 100s nodes, i.e., $10,000$s edges, which is ten times larger than the state-of-the-art (Holmberg and Hellstrand 1998, Günlük and Linderoth 2010), and compare the quality of the incumbent solutions after an hour. 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, and usually outperforms big-M regularization, as reported in Table 3. 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 Wiegele (2007). Specifically, the bqp-$\{50, 100, 250, 500, 1000\}$ instances generated by Beasley (1990), which have a cost matrix density of 0.1, and the be-100 and be-120.8 instances generated by Billionnet and Elloumi (2007), which respectively have cost matrix densities of 1.0 and 0.8. We warm-start the cutting-plane approach with the best solution found after $10,000$ iterations of Goemans-Williamson rounding (see Goemans and Williamson 1995). We also consider imposing triangle inequalities (Deza and Laurent 2009) 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) into master and sub-problems.

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).

			Big-M			Ridge			Overall
m	p	unit	CPLEX	Cuts	Improvement	CPLEX	Cuts	Improvement	Improvement
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%

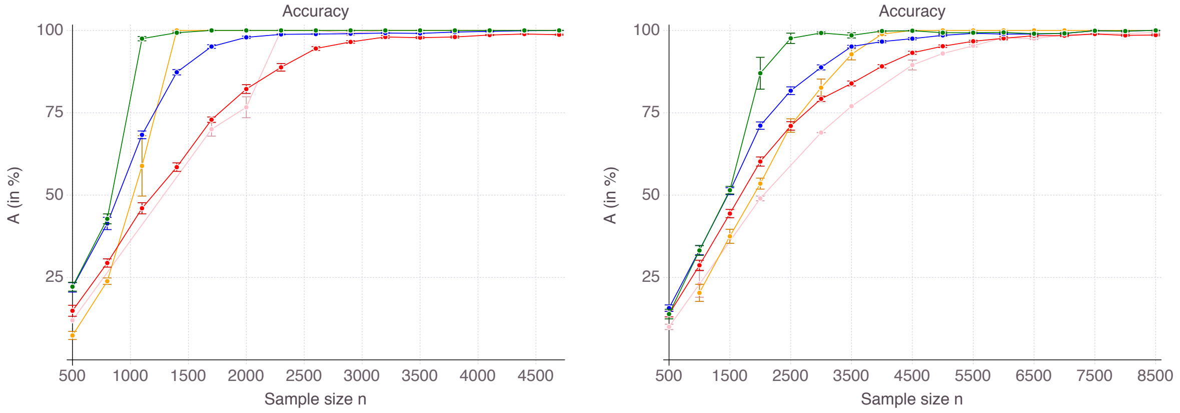
Table 4 Average runtime in seconds on binary quadratic optimization problems from the Biq-Mac library Wiegele (2007), Billionnet and Elloumi (2007). Values are averaged over 10 instances. A "–" denotes an instance which was not solved because the approach did not respect the 32GB peak memory budget.

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%

Table 5 Average incumbent objective value (higher is better) after 1 hour for medium-scale binary quadratic optimization problems from the Biq-Mac library Wiegele (2007), Billionnet and Elloumi (2007). "–" 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**

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 (Bertsimas et al. 2019b). This constitutes a three-order-of-magnitude improvement over previous attempts using big-M regularization (Bertsimas et al. 2016). 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.



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

Figure 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 Bertsimas et al. 2019b, 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).

4.1.4. Sparse Portfolio Selection We applied our approach to sparse portfolio selection problems in Bertsimas and Cory-Wright (2018) and, by introducing a ridge regularization term, successfully solved instances to optimality at a scale of one order of magnitude larger than previous attempts as summarized in Table 6.

Table 6 Largest sparse portfolio instances reliably solved by each approach

Reference	Solution method	Largest instance size solved (no. securities)
Frangioni and Gentile (2009)	Perspective cut+SDP	400
Bonami and Lejeune (2009)	Nonlinear B&B	200
Gao and Li (2013)	Lagrangian relaxation B&B	300
Cui et al. (2013)	Lagrangian relaxation B&B	300
Zheng et al. (2014)	SDP B&B	400
Frangioni et al. (2016)	Approx. Proj. Perspective Cut	400
Bertsimas and Cory-Wright (2018)	Algorithm 1 with ridge regularization	$3,200$

4.2. Evaluation of Different Ingredients in Our Numerical Recipe

We now consider the capacitated facility problem (9) on 112 real-world instances available from the OR-Library (Beasley 1990, Holmberg et al. 1999), 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 7. Although each ingredient contributes independently, jointly improving the lower and upper bounds provides the greatest improvement.

4.3. Big-M Versus Ridge Regularization

In this section, our primary interest is in ascertaining the 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 function is sufficiently strongly convex.

To illustrate this point, we consider large instances of the thermal unit commitment problem originally generated by Frangioni and Gentile (2006b), and multiply the quadratic coefficient a_i for each generator i by a constant factor $\alpha \in \{0.1, 1, 2, 5, 10\}$. Table 8 depicts the average runtime

Table 7 Proportion of wins and relative improvement over CPLEX in terms of computational time on the 112 instances from the OR-library (Beasley 1990, Holmberg et al. 1999) 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%

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 favourably whenever the quadratic term is sufficiently strong.

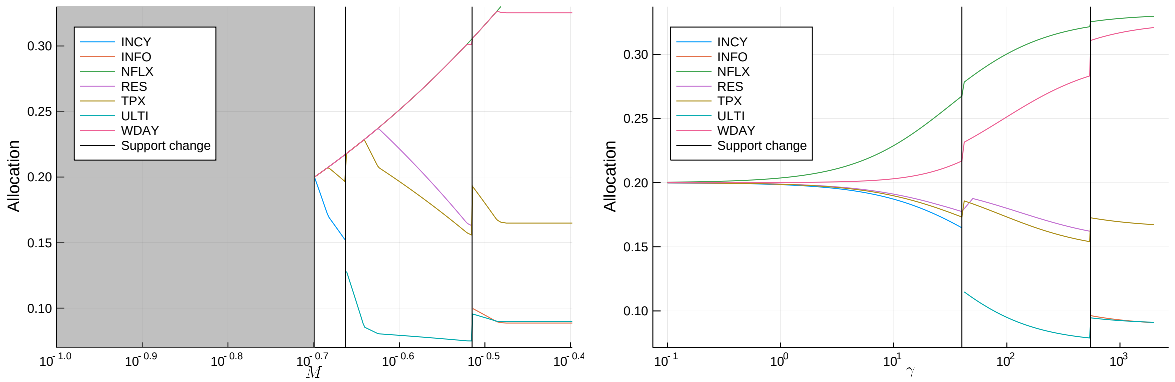
Table 8 Average runtime in seconds per approach, on data from Frangioni and Gentile (2006b) where the quadratic cost are inflated by a factor of α. If the method did not terminate in one hour, we report the bound gap. n denotes the number of generators, each instances has 24 trade periods.

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 (7). 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 apriori. Second, the profile for ridge regularization seems smoother than its equivalent with big-M.

4.4. Relative Merits of Big-M, Ridge Regularization: An Experimental Perspective

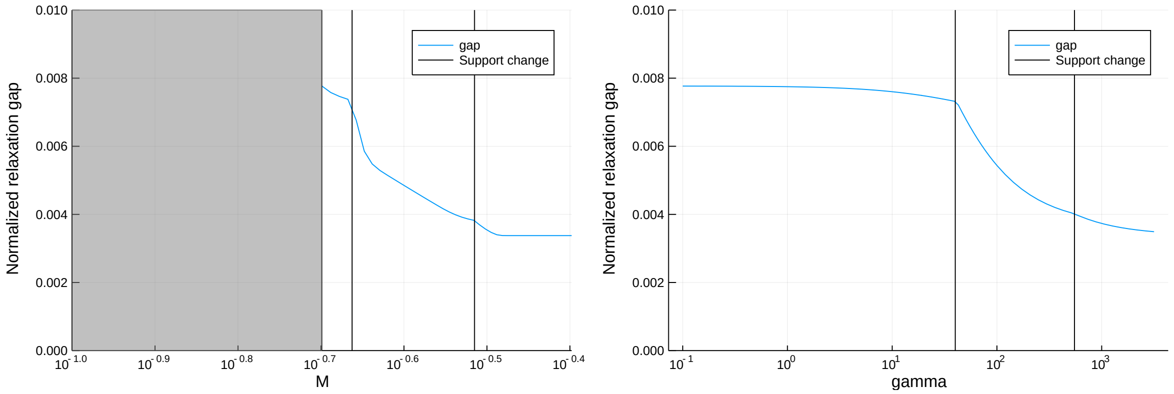
We now conclude our comparison of big-M and ridge regularization, as initiated in Sections 2.5 and 3.6, by indicating the benefits of big-M and ridge regularization, from an experimental perspective:



(a) Big-M regularization

(b) Ridge regularization

Figure 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 Bertsimas and Cory-Wright (2018). Setting $M < \frac{1}{k}$ renders the entire problem infeasible.



(a) Big-M regularization

(b) Ridge regularization

Figure 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

- 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 of degeneracy issues typically 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 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 conic optimization, such as mixed-integer semidefinite optimization, as developed in Bertsimas et al. (2019a).

Endnotes

1. Specifically, ridge regularization enforces logical constraints through perspective functions, as is made clear in Section 3.5.

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) is itself infeasible. Therefore, we assume without loss of generality that it is unbounded.

3. 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 the function h can provide stronger bounds (see Appendix B).

4. Formally, $\boldsymbol{y} := (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X}\boldsymbol{\mu}$, and $\boldsymbol{d} := (\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X} - \mathbb{I})\boldsymbol{\mu}$.

5. Stronger feasibility cuts can be obtained by leveraging problem specific structure, for instance (a) when the feasible set satisfies $\boldsymbol{z}^t \notin \mathcal{Z} \implies \forall \boldsymbol{z} \leq \boldsymbol{z}^t,\ \boldsymbol{z} \notin \mathcal{Z},$ as occurs for all six problems in the preceding section, $\sum_i (1 - z_i^t) z_i \geq 1$ is a valid feasibility cut; (b) if the second-stage problem is a linear optimization problem, an extreme ray with positive marginal cost defines a feasibility cut.

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.

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

8. The number of initial edges is $O(m)$.

References

Aktürk MS, Atamtürk A, Gürel S (2009) A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Oper. Res. Lett.* 37(3):187–191.

Atamturk A, Gomez A (2019) Rank-one convexification for sparse regression. *arXiv:1901.10334* .

Beasley JE (1990) Or-library: distributing test problems by electronic mail. *J. Oper. Res. Soc.* 41(11):1069–1072.

Beaumont N (1990) An algorithm for disjunctive programs. *Euro. J. Oper. Res.* 48(3):362–371.

Bertsekas DP (2016) *Nonlinear programming: 3rd Edition* (Athena Scientific Belmont).

Bertsimas D, Cory-Wright R (2018) A scalable algorithm for sparse and robust portfolios. *arXiv:1811.00138* .

Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. *Ann. Stat.* 44(2):813–852.

Bertsimas D, Lamperski J, Pauphilet J (2019a) Certifiably optimal sparse inverse covariance estimation. *under rev.* .

Bertsimas D, Pauphilet J, Van Parys B (2019b) Sparse regression: Scalable algorithms and empirical performance. *arXiv:1902.06547* .

Bertsimas D, Van Parys B (2019) Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *Ann. Stat., Accepted* .

Bienstock D (1996) Computational study of a family of mixed-integer quadratic programming problems. *Math. Prog.* 74(2):121–140.

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

Bonami P, Lejeune MA (2009) An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Oper. Res.* 57(3):650–670.

Boyd S, Vandenberghe L (2004) *Convex optimization* (Cambridge university press).

Cui X, Zheng X, Zhu S, Sun X (2013) Convex relaxations and miqcqp reformulations for a class of cardinality-constrained portfolio selection problems. *J. Glob. Opt.* 56(4):1409–1423.

Deza MM, Laurent M (2009) *Geometry of cuts and metrics*, volume 15 (Springer).

Dunning I, Huchette J, Lubin M (2017) Jump: A modeling language for mathematical optimization. *SIAM Rev.* 59(2):295–320.

Duran MA, Grossmann IE (1986) An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Prog.* 36(3):307–339.

Fischetti M, Ljubić I, Sinnl M (2016a) Benders decomposition without separability: A computational study for capacitated facility location problems. *Euro. J. Oper. Res.* 253(3):557–569.

Fischetti M, Ljubić I, Sinnl M (2016b) Redesigning benders decomposition for large-scale facility location. *Mang. Sci.* 63(7):2146–2162.

Fletcher R, Leyffer S (1994) Solving mixed integer nonlinear programs by outer approximation. *Math. Prog.* 66(1-3):327–349.

Frangioni A, Furini F, Gentile C (2016) Approximated perspective relaxations: a project and lift approach. *Comp. Opt. Appl.* 63(3):705–735.

Frangioni A, Gentile C (2006a) Perspective cuts for a class of convex 0–1 mixed integer programs. *Math. Prog.* 106(2):225–236.

Frangioni A, Gentile C (2006b) Solving nonlinear single-unit commitment problems with ramping constraints. *Oper. Res.* 54(4):767–775.

Frangioni A, Gentile C (2007) Sdp diagonalizations and perspective cuts for a class of nonseparable miqp. *Oper. Res. Lett.* 35(2):181–185.

Frangioni A, Gentile C (2009) A computational comparison of reformulations of the perspective relaxation: Socp vs. cutting planes. *Oper. Res. Lett.* 37(3):206–210.

Gao J, Li D (2013) Optimal cardinality constrained portfolio selection. *Oper. Res.* 61(3):745–761.

Geoffrion AM (1972) Generalized benders decomposition. *J. Opt. Theory Appl.* 10(4):237–260.

Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. *Mang. Sci.* 22(4):455–460.

Glover F, Woolsey E (1974) Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Oper. Res.* 22(1):180–182.

Goemans MX, Williamson DP (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* 42(6):1115–1145.

Günlük O, Linderoth J (2010) Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Math. Prog.* 124(1-2):183–205.

Holmberg K, Hellstrand J (1998) Solving the uncapacitated network design problem by a lagrangean heuristic and branch-and-bound. *Oper. Res.* 46(2):247–259.

Holmberg K, Rönnqvist M, Yuan D (1999) An exact algorithm for the capacitated facility location problems with single sourcing. *Euro. J. Oper. Res.* 113(3):544–559.

Kelley JE Jr (1960) The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.* 8(4):703–712.

Lasserre JB (2001) An explicit exact sdp relaxation for nonlinear 0-1 programs. *International Conference on Integer Programming and Combinatorial Optimization*, 293–303 (Springer).

Laurent M (2003) A comparison of the sherali-adams, lovász-schrijver, and lasserre relaxations for 0–1 programming. *Math. Oper. Res.* 28(3):470–496.

Magnanti TL, Wong RT (1981) Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.* 29(3):464–484.

Padberg M, Rinald G (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.* 33(1):60–100.

Papadakos N (2008) Practical enhancements to the magnanti–wong method. *Oper. Res. Lett.* 36(4):444–449.

Pilanci M, Wainwright MJ, El Ghaoui L (2015) Sparse learning via boolean relaxations. *Math. Prog.* 151(1):63–87.

Raghavan P, Tompson CD (1987) Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica* 7(4):365–374.

Renegar J, Grimmer B (2018) A simple nearly-optimal restart scheme for speeding-up first order methods. *arXiv:1803.00151* .

Rigollet P, Hütter JC (2015) High dimensional statistics. *Lecture notes for course 18S997* .

Stubbs RA, Mehrotra S (1999) A branch-and-cut method for 0-1 mixed convex programming. *Math. Prog.* 86(3):515–532.

Wiegele A (2007) Biq mac librarya collection of max-cut and quadratic 0-1 programming instances of medium size. Technical report, Alpen-Adria-Universitt Klagenfurt, Austria.

Xie W, Deng X (2018) The ccp selector: Scalable algorithms for sparse ridge regression from chance-constrained programming. *arXiv:1806.03756* .

Zheng X, Sun X, Li D (2014) Improving the performance of miqp solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach. *INFORMS J. Comp.* 26(4):690–703.

Appendix A: Omitted Proofs

A.1. Proof of Theorem 2: Quality of the Random Rounding Strategy

Proof of Theorem 2 We only detail the proof for the big-M regularization case, as the ridge regularization case follows *mutatis mutandis.* From Proposition 1,

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

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 Rigollet and Hütter 2015, 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

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

A.2. Proof of Theorem 3: Relationship With Perspective Cuts

Proof of Theorem 3 Let us fix $\boldsymbol{z} \in \mathcal{Z}$. Then, we have that:

$$\max_{\boldsymbol{\alpha}} h(\boldsymbol{\alpha}) - \frac{\gamma}{2} \sum_{j=1}^n z_j \alpha_j^2 = \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} h(\boldsymbol{\alpha}) - \frac{\gamma}{2} \sum_{j=1}^n z_j \beta_j^2 \text{ s.t. } \boldsymbol{\beta} = \boldsymbol{\alpha},$$

$$= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \min_{\boldsymbol{x}} h(\boldsymbol{\alpha}) - \frac{\gamma}{2}\sum_{j=1}^{n} z_j \beta_j^2 - \boldsymbol{x}^\top(\boldsymbol{\beta}-\boldsymbol{\alpha}),$$

$$= \min_{\boldsymbol{x}} \underbrace{\max_{\boldsymbol{\alpha}} \left[h(\boldsymbol{\alpha}) + \boldsymbol{x}^\top \boldsymbol{\alpha}\right]}_{(-h)^\star(\boldsymbol{x})=g(\boldsymbol{x})} + \sum_{i=1}^{n} \max_{\beta_i} \left[-\frac{\gamma}{2} z_i \beta_i^2 - x_i\beta_i\right].$$

Finally, observing that

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

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 $\boldsymbol{z} \in \mathcal{Z}$, $\|\boldsymbol{\alpha}^\star(\boldsymbol{z})\| \leq L$. 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 $\boldsymbol{\beta}$ for the sparse ERM problem (6), and with respect to $\boldsymbol{w}$ for the portfolio selection problem (7), 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 (11) is strongly concave in $\boldsymbol{\alpha}$, which occurs for the sparse ERM problem (6) with ordinary least-squares loss, the unit commitment problem (8), and the portfolio selection (7) and network design problems whenever $\boldsymbol{\Sigma}$ (resp. $\boldsymbol{Q}$) is full-rank:

Lemma 1. *Let $h(\cdot)$ be a strongly concave function with parameter $\mu > 0$ (see Boyd and Vandenberghe 2004, Chapter 9.1.2 for a general theory of strong convexity), and suppose that $\mathbf{0} \in dom(g)$ and $\boldsymbol{\alpha}^\star := \arg\max_{\boldsymbol{\alpha}} h(\boldsymbol{\alpha})$. Then, for any choice of $\boldsymbol{z}$, we have*

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

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

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

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

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

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

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 1:

Corollary 1. *For the sparse ERM problem* (6) *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*

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

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

Proof Applying Lemma 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: Numerical Results

In this Appendix, we provide additional material pertaining to the numerical results introduced in Section 4.

C.1. Facility Location

We first present the instance-wise runtimes (in seconds) for all instances from the OR-library Beasley (1990), in Table 9.

Table 9 Runtime (in seconds) on 49 capacitated facility location instances from OR-library (Beasley 1990). n and m respectively denote the number of facilities and customers.

Instance	m	n	CPLEX Big-M	CPLEX Ridge with $\gamma = 1$	Cuts Ridge with $\gamma = 1$
cap41	50	16	0.07	0.93	**0.06**
cap42	50	16	0.06	0.94	**0.06**
cap43	50	16	0.07	0.96	**0.05**
cap44	50	16	0.07	3.18	**0.06**
cap51	50	16	**0.01**	6.05	0.03
cap61	50	16	**0.01**	4.78	0.04
cap62	50	16	**0.01**	5.26	0.03
cap63	50	16	**0.01**	4.82	0.03
cap64	50	16	**0.01**	4.81	0.03
cap71	50	16	**0.01**	6.03	0.03
cap72	50	16	**0.01**	5.42	0.03
cap73	50	16	**0.01**	5.37	0.03
cap74	50	16	**0.01**	5.42	0.03
cap81	50	25	0.22	7.7	**0.05**
cap82	50	25	0.17	7.39	**0.05**
cap83	50	25	0.24	6.76	**0.05**
cap84	50	25	0.29	5.52	**0.05**
cap91	50	25	0.09	6.91	**0.05**
cap92	50	25	0.14	6.84	**0.06**
cap93	50	25	0.16	7.09	**0.05**
cap94	50	25	0.09	8.22	**0.05**
capa1	1000	100	**8.61**	3600	3600
capa2	1000	100	**8.53**	3600	3600
capa3	1000	100	**7.02**	3600	3600
capa4	1000	100	**8.73**	3600	3600
capb1	1000	100	**5.22**	3600	3600
capb2	1000	100	**4.53**	3600	3600
capb3	1000	100	**4.9**	3600	4293.84
capb4	1000	100	**4.91**	3600	4520.88
capc1	1000	100	**4.97**	3600	3600
capc2	1000	100	**5.16**	3600	3600
capc3	1000	100	**4.99**	3600	3600
capc4	1000	100	**5.01**	3600	3600
cap101	50	25	0.06	7.31	**0.05**
cap102	50	25	0.14	7.19	**0.05**
cap103	50	25	0.1	6.71	**0.05**
cap104	50	25	0.15	6.15	**0.05**
cap111	50	50	0.13	23.75	**0.11**
cap112	50	50	**0.11**	26.07	0.56
cap113	50	50	**0.13**	17.08	0.53
cap114	50	50	**0.13**	29.68	0.61
cap121	50	50	**0.08**	11.6	0.11
cap122	50	50	**0.07**	86.13	0.58
cap123	50	50	**0.08**	17.94	0.54
cap124	50	50	**0.08**	89.95	0.62
cap131	50	50	**0.07**	57.34	0.13
cap132	50	50	**0.07**	171.43	0.53
cap133	50	50	**0.07**	19.79	0.51
cap134	50	50	**0.06**	81.9	0.58

Table 10 Runtime (in seconds) on 63 capacitated facility location instances from Holmberg et al. (1999). n and m respectively denote the number of facilities and customers. Part 1 of 2.

Instance	m	n	CPLEX Big-M	CPLEX Ridge $\gamma = 1$	Cuts Ridge $\gamma = 1$
p1	50	10	**0.01**	0.38	0.03
p2	50	10	**0.01**	0.49	0.03
p3	50	10	**0.01**	0.57	0.03
p4	50	10	**0.01**	0.59	0.03
p5	50	10	**0.01**	0.51	0.03
p6	50	10	**0.01**	0.44	0.03
p10	50	10	**0.01**	0.35	0.03
p11	50	10	**0.01**	0.37	0.03
p12	50	10	**0.01**	0.38	0.03
p13	50	20	**0.06**	1.09	0.3
p14	50	20	**0.06**	0.59	0.08
p15	50	20	**0.06**	0.76	0.3
p16	50	20	**0.05**	6.31	0.3
p17	50	20	**0.06**	6.54	0.29
p18	50	20	**0.06**	0.82	0.08
p19	50	20	**0.06**	2.08	0.29
p20	50	20	**0.06**	7.47	0.31
p21	50	20	**0.06**	6.82	0.29
p22	50	20	**0.06**	1.23	0.08
p23	50	20	**0.05**	2.09	0.29
p24	50	20	**0.06**	7.42	0.3
p25	150	30	0.85	27.84	**0.49**
p26	150	30	**0.68**	24.2	6.55
p27	150	30	**0.88**	26.55	24.89
p28	150	30	**0.74**	19.31	38.19
p29	150	30	2.64	25.67	**2.54**
p30	150	30	2.9	20.22	**2.61**
p31	150	30	**2.64**	28.57	18.62
p32	150	30	**1.76**	64.43	35.5
p33	150	30	**0.81**	26.31	5.22

Table 11 Runtime (in seconds) on 63 capacitated facility location instances from Holmberg et al. (1999). n and m respectively denote the number of facilities and customers. Part 2 of 2.

Instance	m	n	CPLEX Big-M	CPLEX Ridge $\gamma = 1$	Cuts Ridge $\gamma = 1$
p34	150	30	**0.7**	23.42	5.85
p35	150	30	**0.78**	36.17	27.14
p36	150	30	**0.83**	35.15	39.88
p37	150	30	0.69	44.1	**0.49**
p38	150	30	**0.71**	23.11	6.06
p39	150	30	**0.76**	36.63	26.1
p40	150	30	**0.68**	33.89	48.75
p41	90	10	0.05	0.99	**0.03**
p42	80	20	**0.08**	16.06	10.85
p43	70	30	**0.09**	23.63	254.16
p44	90	10	0.05	0.92	**0.03**
p45	80	20	**0.08**	13.11	2.39
p46	70	30	**0.09**	17.75	65.52
p47	90	10	**0.19**	7.92	0.29
p48	80	20	**0.17**	18.11	0.31
p49	70	30	**0.09**	16.56	3.32
p50	100	10	**0.04**	8.18	0.41
p51	100	20	**0.11**	18.2	3.39
p52	100	10	**0.04**	1.11	0.04
p53	100	20	**0.12**	19.7	2.79
p54	100	10	**0.04**	1.08	0.08
p55	100	20	3.17	15.17	**0.32**
p56	200	30	0.85	9.83	**0.18**
p57	200	30	0.67	9.85	**0.18**
p58	200	30	**0.76**	23.54	3.26
p59	200	30	0.8	7.65	**0.19**
p60	200	30	0.87	10.05	**0.19**
p61	200	30	0.81	9.81	**0.19**
p62	200	30	**0.85**	12.01	3.92
p63	200	30	0.78	8.64	**0.18**
p64	200	30	0.72	9.19	**0.27**
p65	200	30	0.84	9.06	**0.25**
p66	200	30	**0.8**	21.43	2.95

C.2. Unit Commitment

We now present the instance-wise runtimes per approach for the largest-scale instances generated by Frangioni and Gentile (2006a), for varying inflation factors α, in Tables 12-16.

Table 12 Runtime in seconds per approach, on data from Frangioni and Gentile (2006b) where the quadratic cost are inflated by a factor of 0.1. n, t respectively denote the number of generators and trade periods.

Instance Name	n	t	Runtime (s)	
			CPLEX Big-M	CPLEX MISOCP
100_0_1_w	100	24	142.8	405.0
100_0_2_w			145.7	400.3
100_0_3_w			3.9	142.3
100_0_4_w			171.9	369.4
100_0_5_w			3.5	178.0
150_0_1_w	150	24	3.9	285.6
150_0_2_w			5.1	627.9
150_0_3_w			2.5	139.3
150_0_4_w			131.0	355.5
200_0_1_w	200	24	262.4	11.2
200_0_2_w			4.5	87.7
200_0_3_w			4.4	15.4
200_0_4_w			6.3	265.2
200_0_5_w			8.4	10.8
200_0_6_w			213.4	14.3
200_0_7_w			2.4	209.3
200_0_8_w			4.5	436.8
200_0_9_w			3.8	10.7
200_0_10_w			3.1	227.4
200_0_11_w			3.9	13.6
200_0_12_w			157.8	356.0

Table 13 Runtime in seconds per approach, on data from Frangioni and Gentile (2006b) where the quadratic cost are inflated by a factor of 1. n, t respectively denote the number of generators and trade periods.

Instance Name	n	t	Runtime (s)	
			CPLEX Big-M	CPLEX MISOCP
100_0_1_w	100	24	1.6	165.2
100_0_2_w			15.8	191.1
100_0_3_w			13.6	210.1
100_0_4_w			48.1	300.6
100_0_5_w			1.9	280.2
150_0_1_w	150	24	0.8	6.2
150_0_2_w			1.6	7.1
150_0_3_w			1.3	10.9
150_0_4_w			21.1	88.8
200_0_1_w	200	24	1.5	360.2
200_0_2_w			2.2	318.3
200_0_3_w			2.1	398.1
200_0_4_w			2.2	26.6
200_0_5_w			1.4	279.8
200_0_6_w			14.7	26.2
200_0_7_w			1.4	264.6
200_0_8_w			2.4	22.2
200_0_9_w			1.6	297.5
200_0_10_w			6.2	352.7
200_0_11_w			1.7	282.9
200_0_12_w			2.0	247.7

Table 14 Runtime in seconds per approach, on data from Frangioni and Gentile (2006b) where the quadratic cost are inflated by a factor of 2. n, t respectively denote the number of generators and trade periods.

Instance Name	n	t	Runtime (s)	
			CPLEX Big-M	CPLEX MISOCP
100_0_1_w	100	24	0.21%	78.4
100_0_2_w			0.40%	72.0
100_0_3_w			0.37%	64.7
100_0_4_w			0.29%	19.7
100_0_5_w			0.35%	4.5
150_0_1_w	150	24	0.17%	15.7
150_0_2_w			0.26%	16.9
150_0_3_w			0.24%	93.2
150_0_4_w			0.34%	7.8
200_0_1_w	200	24	0.19%	45.8
200_0_2_w			0.29%	25.5
200_0_3_w			0.20%	457.0
200_0_4_w			0.25%	26.4
200_0_5_w			0.29%	30.2
200_0_6_w			0.32%	25.1
200_0_7_w			0.20%	284.9
200_0_8_w			0.28%	23.7
200_0_9_w			0.16%	26.8
200_0_10_w			0.17%	359.4
200_0_11_w			0.24%	24.4
200_0_12_w			0.29%	26.1

Table 15 Runtime in seconds per approach, on data from Frangioni and Gentile (2006b), where the quadratic cost are inflated by a factor of 5. n, t respectively denote the number of generators and trade periods.

Instance Name	n	t	Runtime (s)	
			CPLEX Big-M	CPLEX MISOCP
100_0_1_w	100	24	1.36%	5.3
100_0_2_w			2.01%	4.5
100_0_3_w			1.67%	5.0
100_0_4_w			1.69%	4.0
100_0_5_w			1.69%	4.1
150_0_1_w	150	24	1.40%	6.3
150_0_2_w			1.99%	6.2
150_0_3_w			1.73%	6.2
150_0_4_w			1.65%	6.8
200_0_1_w	200	24	1.51%	13.2
200_0_2_w			1.84%	13.2
200_0_3_w			1.84%	13.8
200_0_4_w			1.73%	15.7
200_0_5_w			1.54%	11.3
200_0_6_w			1.83%	13.4
200_0_7_w			1.53%	12.6
200_0_8_w			1.70%	12.3
200_0_9_w			1.47%	16.8
200_0_10_w			1.33%	24.6
200_0_11_w			1.61%	36.0
200_0_12_w			1.51%	17.7

Table 16 Runtime in seconds per approach, on data from Frangioni and Gentile (2006b), where the quadratic cost are inflated by a factor of 10. n, t respectively denote the number of generators and trade periods.

Instance Name	n	t	Runtime (s)	
			CPLEX Big-M	CPLEX MISOCP
100_0_1_w	100	24	2.47%	6.1
100_0_2_w			2.80%	6.9
100_0_3_w			2.92%	6.1
100_0_4_w			2.96%	5.5
100_0_5_w			2.67%	5.5
150_0_1_w	150	24	2.42%	5.9
150_0_2_w			2.94%	7.0
150_0_3_w			3.10%	12.8
150_0_4_w			2.84%	6.4
200_0_1_w	200	24	2.55%	15.1
200_0_2_w			3.01%	25.7
200_0_3_w			3.02%	31.0
200_0_4_w			3.24%	15.0
200_0_5_w			2.65%	15.1
200_0_6_w			3.34%	15.2
200_0_7_w			2.63%	28.7
200_0_8_w			2.96%	26.5
200_0_9_w			2.62%	12.3
200_0_10_w			2.30%	27.8
200_0_11_w			2.84%	25.0
200_0_12_w			2.51%	16.5

C.3. Binary Quadratic Optimization

We now present the runtime in seconds for a subset of the binary quadratic problems in the Biq-Mac library Wiegele (2007), in Table 17, and the best solution identified by each approach, for the instances which were unsolved by all approaches after one hour, in Table 18. Note that the objective is to **maximize** $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$ for the bqp instances, and to **minimize** $\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$ for the be instances.

Table 17 Runtime in seconds on binary quadratic optimization problems from the Biq-Mac library Wiegele (2007), Billionnet and Elloumi (2007). Cuts-Triangle includes an extended formulation in the master problem.

Instance	n	Runtime (s)			
		CPLEX-M	CPLEX-M-Triangle	Cuts-M	Cuts-M-Triangle
bqp50-1	50	3.4	0.6	34.9	**0.4**
bqp50-2	50	1.6	0.7	10.1	**0.4**
bqp50-3	50	1.9	0.6	8.4	**0.4**
bqp50-4	50	1.5	0.6	9.9	**0.4**
bqp50-5	50	1.4	0.6	16.9	**0.4**
bqp50-6	50	1.1	0.6	3.7	**0.3**
bqp50-7	50	1.2	0.7	13.7	**0.4**
bqp50-8	50	2.4	0.7	111.2	**0.4**
bqp50-9	50	123.2	0.7	48.6	**0.6**
bqp50-10	50	155.9	0.6	49.2	**0.4**
bqp100-1	100	236.0	101.5	35.8%	**54.0**
bqp100-2	100	46.1	24.5	23.6%	**21.9**
bqp100-3	100	23.7	10.3	18.5%	**8.0**
bqp100-4	100	198.9	34.4	24.5%	**32.2**
bqp100-5	100	146.3	30.2	25.5%	**27.2**
bqp100-6	100	314.3	218.1	35.3%	**168.1**
bqp100-7	100	179.3	59.8	30.6%	**39.4**
bqp100-8	100	25.6	20.9	22.1%	**19.2**
bqp100-9	100	26.7	4.5	17.9%	**4.2**
bqp100-10	100	26.2	12.5	18.8%	**11.9**
be100-1	100	57.3%	171.8%	211.5%	163.2%
be100-2	100	52.4%	202.4%	251.6%	198.1%
be100-3	100	78.7%	188.3%	191.0%	192.0%
be100-4	100	41.0%	174.8%	220.5%	171.8%
be100-5	100	92.4%	223.8%	279.2%	221.2%
be100-6	100	54.7%	196.2%	247.8%	192.8%
be100-7	100	69.8%	189.7%	229.4%	183.3%
be100-8	100	77.3%	191.5%	229.6%	181.7%
be100-9	100	153.6%	295.2%	341.8%	273.1%
be100-10	100	119.6%	246.0%	291.6%	234.4%
be120.8-1	100	159.9%	254.6%	280.0%	236.3%
be120.8-2	100	158.3%	231.0%	280.2%	236.7%
be120.8-3	100	175.5%	229.7%	271.9%	226.4%
be120.8-4	100	135.5%	215.4%	245.4%	202.7%
be120.8-5	100	92.1%	201.6%	242.2%	197.4%
be120.8-6	100	210.3%	237.7%	283.6%	237.4%
be120.8-7	100	107.6%	195.0%	230.3%	191.3%
be120.8-8	100	135.4%	234.6%	267.2%	224.0%
be120.8-9	100	152.2%	237.7%	281.0%	234.8%
be120.8-10	100	137.4%	220.6%	259.6%	216.3%

Table 18 Best objective value (higher is better) found after 1 hour for the problems which were unsolved by all approaches after 1 hour. "–" denotes that the problem could not be solved within the 32GB memory budget.

Instance	n	Runtime (s)			
		CPLEX-M	CPLEX-M-Triangle	Cuts-M	Cuts-M-Triangle
be100-1	100	18703	19014	**19412**	**19412**
be100-2	100	**17252**	17191	17232	17232
be100-3	100	16954	**17532**	17423	17423
be100-4	100	**19125**	**19125**	19065	19110
be100-5	100	15189	15765	**15783**	15747
be100-6	100	15679	17316	**17368**	**17368**
be100-7	100	17795	18250	**18629**	**18629**
be100-8	100	17673	18232	**18632**	18582
be100-9	100	11692	12505	**13150**	**13150**
be100-10	100	13968	14920	15097	**15132**
be120.8-1	120	16970	17940	**18538**	**18538**
be120.8-2	120	16957	**18609**	18425	18397
be120.8-3	120	15911	**19208**	19152	19195
be120.8-4	120	18563	19982	20503	**20504**
be120.8-5	120	19648	**20381**	20303	20303
be120.8-6	120	17171	**18446**	18264	18264
be120.8-7	120	**22079**	22031	22017	22017
be120.8-8	120	18423	18985	**19379**	19361
be120.8-9	120	16060	**18181**	18091	18099
be120.8-10	120	17650	18940	**19035**	19034
bqp250-1	250	10321	42688	**45376**	45187
bqp250-2	250	10722	41271	**44455**	44451
bqp250-3	250	18238	48485	**48834**	48779
bqp250-4	250	2224	38856	**41057**	41023
bqp250-5	250	10597	46008	47607	**47619**
bqp250-6	250	8034	37975	**40321**	40285
bqp250-7	250	14775	45469	46521	**46601**
bqp250-8	250	2654	32635	**35018**	34984
bqp250-9	250	16157	46495	**48622**	48248
bqp250-10	250	5486	38552	**39938**	39838
bqp500-1	500	24381	15948	**113757**	113273
bqp500-2	500	14443	16023	126707	**127034**
bqp500-3	500	22959	24893	128921	**129058**
bqp500-4	500	38138	38308	**127904**	127346
bqp500-5	500	18430	11503	**123391**	123319
bqp500-6	500	10414	9201	**119303**	119277
bqp500-7	500	5682	15513	119840	**120054**
bqp500-8	500	30177	31365	**122101**	121079
bqp500-9	500	9388	13677	**117990**	117243
bqp500-10	500	20159	20159	**128879**	128741
bqp1000-1	1000	–	–	363905	–
bqp1000-2	1000	–	–	345579	–
bqp1000-3	1000	–	–	362184	–
bqp1000-4	1000	–	–	363421	–
bqp1000-5	1000	–	–	342762	–
bqp1000-6	1000	–	–	353348	–
bqp1000-7	1000	–	–	361544	–
bqp1000-8	1000	–	–	341070	–
bqp1000-9	1000	–	–	338506	–
bqp1000-10	1000	–	–	342205	–