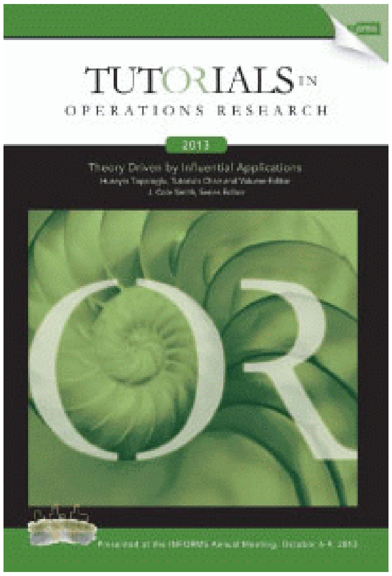

INFORMS TutORials in Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Mixed-Integer Second-Order Cone Programming: A Survey

Hande Y. Benson, Ümit Sağlam

TUTORIALS IN OPERATIONS RESEARCH
INFORMS 2013

 | ISBN 978-0-9843378-4-2
http://dx.doi.org/10.1287/educ.2013.0115

Mixed-Integer Second-Order Cone Programming: A Survey

Hande Y. Benson, Ümit Sağlam
Department of Decision Sciences, Bennett S. LeBow College of Business, Drexel University, Philadelphia, Pennsylvania 19104 {benson@drexel.edu, us26@drexel.edu}

Abstract Second-order cone programming problems (SOCPs) have been well studied in literature, and computationally efficient implementations of solution algorithms exist. Of relatively new interest to the research community is an extension: mixed-integer second-order cone programming problems (MISOCPs). In this tutorial, we present a thorough survey of the literature on applications of and algorithms for this class of problems. Applications include options pricing, portfolio optimization, network design and operations, and statistics, with some examples arising from the use of MISOCPs as relaxations or reformulations of mixed-integer nonlinear programming problems (MINLPs). The solution algorithms combine existing solution methods for SOCPs with extensions of mixed-integer linear programming or MINLP methods. Therefore, we present a brief overview of solution methods for SOCPs first, and then we describe cuts and relaxations that enable the application of branch-and-cut, branch-and-bound, and outer approximation methods to MISOCPs.

Keywords mixed-integer second-order cone programming; mixed-integer conic programming; second-order cone programming; algorithms; applications

1. Introduction

Mixed-integer second-order cone programming problems (MISOCPs) can be expressed as

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & c^T x \\ \text{s.t.} \quad & \|A_i x + b_i\| \le a_{0i}^T x + b_{0i}, \quad i = 1,\dots,m, \end{aligned} \tag{1}$$

where x is the n-vector of decision variables, $\mathcal{X} = \{(y,z)\colon y \in \mathcal{Z}^p, z \in \mathcal{R}^k, p+k=n\}$, and the data are $c \in \mathcal{R}^n$, $A_i \in \mathcal{R}^{m_i \times n}$, $b_i \in \mathcal{R}^{m_i}$, $a_{0i} \in \mathcal{R}^n$, and $b_{0i} \in \mathcal{R}$ for $i = 1,\dots,m$. The notation $\|\cdot\|$ denotes the Euclidean norm, and the constraints are said to define the *second-order cone*, also referred to as the *Lorentz cone*.

When $p = 0$, (1) reduces to a second-order cone programming problem (SOCP) in the so-called dual form. SOCPs have been widely studied in literature; see Lobo et al. [32] for a general introduction and an extensive list of applications and Alizadeh and Goldfarb [2] for an overview of the general properties, duality theory, and interior-point methods. In fact, interior-point methods have been the most popular solution methods for this class of problems, enjoying both good theoretical convergence properties (Nesterov and Nemirovsky [38]) and efficient computational performance in implementations such as SeDuMi (Sturm [48]), SDPT3 (Toh et al. [50]), MOSEK (Andersen and Andersen [4]), and CPLEX (IBM ILOG [27]). Besides interior-point methods designed specifically for SOCPs, there is also a sequential linear programming method (Krishnan and Mitchell [29]), a simplex method for conic programs (Goldfarb [24]), an interior-point method for nonlinear programming that treats SOCPs as a special case (Benson and Vanderbei [10]), as well as approaches based on polyhedral reformulations of the second-order cone (Ben-Tal and Nemirovski [12]).

Comparatively, MISOCP is a less mature field, but these problems arise in a variety of important application areas, and there have been significant advances in solution algorithms in the last decade. In the next section, we start the tutorial with a literature review on applications ranging from scheduling to electrical engineering, from finance to operations management. Several of the examples arise as reformulations or even relaxations of mixed-integer nonlinear programming problems (MINLPs), because MISOCPs can sometimes present advantages over mixed-integer linear programming problems (MILPs) in this regard. In §3, we provide a brief overview of relevant solution methods for SOCP, since the choice of method for the underlying continuous relaxation greatly influences the design and performance of the overall solution method for MISOCP. These solution methods include interior-point methods designed specifically for SOCP, adaptations of interior-point methods for nonlinear programming to the case of SOCP, and lifted polyhedral relaxations, and they are incorporated into the MISOCP algorithms presented in §4. In general, MISOCP algorithms fall into two groups: extensions of MILP approaches (since the second-order cone can be viewed as an extension of the linear cone) or special-purpose MINLP approaches (since SOCPs can also be viewed as nonlinear programming problems with a special structure). As such, we will present an overview of cuts, including extensions of Gomory and rounding cuts, and relaxations for MISOCP, while discussing the adaptation of branch-and-cut, branch-and-bound, and outer approximation methods to the case of this class of problems.

2. Applications of Mixed-Integer Second-Order Cone Programming

In this section, we give an overview of MISOCPs arising in a variety of application areas in business, engineering, and statistics. It should be noted that this is only a representative list and not an exhaustive one by any means. One of the challenges in gathering a literature review on MISOCP is that, many times, authors do not recognize the special structure of the problem and simply identify the model as a MINLP, solved using traditional MINLP methods. Therefore, we have included in this section only those models that have been recognized by the authors as MISOCPs.

We should also note that because of the wide variety of models, each subsection below will have a self-contained list of notation. We will return to the formulation (1) and related notation in the next section.

2.1. Portfolio Optimization

One of the most prominent application areas for MISOCP is that of portfolio optimization, so we start our description of applications by presenting a model that incorporates some of the most common components found in literature. We consider a single-period portfolio optimization problem that is based on the Markowitz mean-variance framework (Markowitz [36]), where there is a trade-off between expected return and risk (market volatility) that the investor may be willing to take on. Portfolio optimization literature has come quite far in the decades since the publication of Markowitz [35], and many modern models are formulated using second-order cone constraints and take discrete decisions into consideration. We now present a portfolio optimization model that is formulated as an MISOCP and is aggregated from current literature. In this model, we have included transaction costs, conditional value-at-risk (CVaR) constraints, diversification-by-sector constraints, and buy-in thresholds. These model components have been adapted from Adcock and Meade [1], Lobo et al. [31], Bonami and Lejeune [13], Garleanu and Pedersen [21], and Brown and Smith [15]. Numerical results on this aggregate model can be found in Benson and Sağlam [11].

The portfolio optimization model considered here can be written as

$$
\begin{aligned}
\max \quad & \left\{ \sum_{j=0}^{n} r_j (w_j + x_j^+ - x_j^-) - \frac{1}{2}(x^+ + x^-)^T \Lambda (x^+ + x^-) \right\} \\
\text{s.t.} \quad & \Phi^{-1}(\eta_k) \| \Sigma^{1/2}(w + x^+ - x^-) \| \le \sum_{j=0}^{n} r_j (w_j + x_j^+ - x_j^-) - W_k^{\text{low}}, \quad k = 1, \ldots, M, \\
& s_{\min} \zeta_k \le \sum_{j \in S_k} (w_j + x_j^+ - x_j^-) \le s_{\min} + (1 - s_{\min}) \zeta_k, \quad k = 1, \ldots, L, \\
& \sum_{k=1}^{L} \zeta_k \ge L_{\min}, \\
& w_{\min} \delta_j \le w_j + x_j^+ - x_j^- \le \delta_j, \quad j = 1, \ldots, n, \\
& \sum_{j=0}^{n} (w_j + x_j^+ - x_j^-) = 1, \\
& w_j + x_j^+ - x_j^- \ge -s_j, \quad j = 1, \ldots, n, \\
& x^+, x^- \ge 0, \\
& \zeta \in \{0, 1\}^L,
\end{aligned}
\tag{2}
$$

where we consider cash (index 0) and n risky assets from L different sectors for inclusion in our portfolio. The decision variables are $x^+ \in \mathcal{R}^{n+1}$ and $x^- \in \mathcal{R}^{n+1}$, which denote the buy and sell transactions, respectively; $\zeta \in \{0,1\}^L$, the elements of which denote whether there are sufficient investments in each sector; and $\delta \in \mathcal{R}^n$, which will be used for the buy-in threshold constraints. We describe the remaining model components in detail below.

The investor's objective is to choose the optimal trading strategies to maximize the end-of-period expected total return. Denoting the expected rates of return by $r \in \mathcal{R}^{n+1}$ and the current portfolio holdings by $w \in \mathcal{R}^{n+1}$, the expected total portfolio value at the end of the period is given by

$$\sum_{j=0}^{n} r_j (w_j + x_j^+ - x_j^-).$$

However, both the buy and sell transactions are penalized by transaction costs. According to recent dynamic portfolio choice literature (Garleanu and Pedersen [21], Brown and Smith [15], for example), transaction costs include a number of factors, such as price impacts of transactions, brokerage commissions, bid–ask spreads, and taxes. As such, there are a number of different ways to model transaction costs, including linear and convex or concave nonlinear cost functions. Here, we have decided to use the quadratic convex transaction cost formulation of Garleanu and Pedersen [21] because it provides the best fit to our framework. Therefore, the total transaction costs appear as a penalty term in the objective function:

$$\frac{1}{2}(x^+ + x^-)^T \Lambda (x^+ + x^-),$$

where $\Lambda \in \mathcal{R}^{(n+1)\times(n+1)}$ is the trading cost matrix and is obtained as a positive multiple of the covariance matrix of the expected returns. Because of this connection to a covariance matrix, Λ is symmetric and positive definite. Note that both buy and sell transactions receive the same transaction cost, but it would be straightforward to instead include two quadratic terms in the objective function with different trading cost matrices for each type of transaction.

As we will see in the following discussion, the continuous relaxation of (2) includes only linear and second-order cone constraints. However, the quadratic term in the objective function prevents the overall problem from being formulated as an MISOCP. We introduce a new variable $\rho \in \mathcal{R}$ and rewrite the objective function of (2) as

$$\sum_{j=0}^{n} r_j(w_j + x_j^+ - x_j^-) - \rho,$$

with

$$\frac{1}{2}(x^+ + x^-)^T \Lambda (x^+ + x^-) \leq 2\rho.$$

Note that this constraint is equivalent to

$$(x^+ + x^-)^T \Lambda (x^+ + x^-) \leq (1+\rho)^2 - (1-\rho)^2,$$

and moving the last term to the left-hand side and taking the square root of both sides gives the following second-order cone constraint:

$$\left\| \begin{pmatrix} \Lambda^{1/2}(x^+ + x^-) \\ 1-\rho \end{pmatrix} \right\| \leq 1+\rho.$$

We know that the matrix $\Lambda^{1/2}$ exists since Λ is positive-definite. Additionally, this conversion does not increase the difficulty of solving the problem significantly—we add only one auxiliary variable, so the Newton system does not become significantly larger. Also, worsening the sparsity of the problem is not a concern here, since the original problem (2) has a quite dense matrix in the Newton system due to the covariance matrix and the related trading cost matrix both being dense.

As stated above, we are considering both return and risk in this model. In the objective function, we focus on maximizing the expected total return less transaction costs, so we will seek to limit our risk using the first set of constraints. To that end, we will use CVaR constraints, as was done by Lobo et al. [31]. For each CVaR constraint k, $k = 1, \ldots, M$, we will require that our expected wealth at the end of the period be above some threshold level W_k^{low} with a probability of at least η_k. Thus, letting

$$W = \sum_{j=0}^{n} \hat{r}_j(w_j + x_j^+ - x_j^-),$$

where $\hat{r}$ is the random vector of returns, we require that

$$\mathcal{P}(W \geq W_k^{\text{low}}) \geq \eta_k, \quad k = 1, \ldots, M.$$

We assume that the elements of r have jointly Gaussian distribution so that W is normally distributed with a mean of

$$\sum_{j=0}^{n} r_j(w_j + x_j^+ - x_j^-)$$

and a standard deviation of

$$\|\Sigma^{1/2}(w + x^+ - x^-)\|,$$

where Σ is the covariance matrix of the returns.

Therefore, the CVaR constraints can be formulated as

$$\mathcal{P}\left(\frac{W - \sum_{j=0}^{n} r_j(w_j + x_j^+ - x_j^-)}{\|\Sigma^{1/2}(w + x^+ - x^-)\|} \geq \frac{W_k^{\text{low}} - \sum_{j=0}^{n} r_j(w_j + x_j^+ - x_j^-)}{\|\Sigma^{1/2}(w + x^+ - x^-)\|}\right) \geq \eta_k,$$

for each $k = 1, \ldots, M$. This implies that

$$1 - \Phi\left(\frac{W_k^{\text{low}} - \sum_{j=0}^n r_j(w_j + x_j^+ - x_j^-)}{\|\Sigma^{1/2}(w + x^+ - x^-)\|}\right) \geq \eta_k, \quad k = 1, \ldots, M,$$

where Φ is the cumulative distribution function for a standard normal random variable, or

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2}\, dt.$$

Rearranging the terms and taking the inverse gives us

$$\frac{W_k^{\text{low}} - \sum_{j=0}^n r_j(w_j + x_j^+ - x_j^-)}{\|\Sigma^{1/2}(w + x^+ - x^-)\|} \leq \Phi^{-1}(1 - \eta_k), \quad k = 1, \ldots, M.$$

Using the symmetry of the standard normal distribution function, we can rewrite the constraint again as

$$-\frac{W_k^{\text{low}} - \sum_{j=0}^n r_j(w_j + x_j^+ - x_j^-)}{\|\Sigma^{1/2}(w + x^+ - x^-)\|} \geq \Phi^{-1}(\eta_k), \quad k = 1, \ldots, M.$$

Finally, rearranging the terms gives us the second-order cone constraint in (2):

$$\Phi^{-1}(\eta_k)\|\Sigma^{1/2}(w + x^+ - x^-)\| \leq \sum_{j=0}^{n} r_j(w_j + x_j^+ - x_j^-) - W_k^{\text{low}}, \quad k = 1, \ldots, M.$$

Diversification is another important instrument used to reduce the level of risk in the portfolio. In the second and third set of constraints, we impose a diversification requirement to the investor to allocate sufficiently large amounts in at least $L_{\min}$ of the L different economic sectors. This type of constraint was considered by Bonami and Lejeune [13]. To express this diversification requirement, we start by defining binary variables $\zeta_k \in \{0, 1\}$, $k = 1, \ldots, L$, for each economic sector k. If $\zeta_k = 1$, our total portfolio allocation in assets from sector k will be at least $s_{\min}$ (and, of course, no more than 1). Otherwise, it will mean that our total portfolio allocation in those assets fell short of the threshold level $s_{\min}$. Denoting the set of assets that belong to economic sector k, $k = 1, \ldots, L$, by S_k, the second and third sets of constraints are written to express this diversification requirement.

Since we have included transaction costs in our model (2), we will be mindful of the number of transactions as well. Therefore, with the fourth constraint, we can impose a requirement that the investors do not hold very small active positions (see Bonami and Lejeune [13]). Introducing new binary variables $\delta \in \{0, 1\}^n$, we can write this requirement using constraints of the following form:

$$w_{\min}\delta_j \leq w_j + x_j^+ - x_j^- \leq \delta_j, \quad j = 1, \ldots, n,$$

where $w_{\min}$ is a predetermined proportion of the available capital.

The remaining constraints in our problem are grouped into the general category of portfolio constraints. They require that we allocate the full portfolio and allow for short sales of the nonliquid assets by stating that we can take a limited short position s for each one.

2.2. Options Pricing

Pinar [42] describes a pricing problem for an American option in a financial market under uncertainty. The multiperiod, discrete time, and discrete state space structure is modeled using a scenario tree, and therefore the resulting problem is large scale. The set of nodes is denoted by N, and the nodes corresponding to time period t are denoted by N_t. The planning horizon is at time T. $\pi(n)$ denotes the parent node of n, and $A(n)$ denotes the ascendant nodes of n, including itself. The probability of each $n \in N_T$ is denoted by p_n.

We assume that there is a market consisting of $J+1$ securities, with prices at node n given by $z_n = (z_n^0, z_n^1, \ldots, z_n^J)^T$. The security with index 0 is assumed to be risk free. The decision variables $\theta_n \in \mathcal{R}^{J+1}$ denote the portfolio allocations at node n, and thus $z_n^T \theta_n$ denotes the value of the portfolio at the node. The binary decision variables e_n indicate whether an American option is exercised at node n, and, if exercised, the holder would have a payoff of h_n. Auxiliary variables x_n (free) and v_n (nonnegative) are also introduced to denote that the final wealth position can be unrestricted in sign. An additional auxiliary variable, v, is introduced as the initial wealth of the portfolio.

$$
\begin{aligned}
\max \quad & v \\
\text{s.t.} \quad & \sum_{n \in N_T} p_n x_n - \lambda \sqrt{\sum_{n \in N_T} p_n \left(x_n - \sum_{k \in N_T} p_k x_k \right)^2} \geq 0, \\
& \sum_{m \in A(n)} e_m \leq 1, \quad n \in N_T, \\
& z_0^T \theta_0 = h_0 e_0 - v, \\
& z_n^T (\theta_n - \theta_{\pi(n)}) = h_n e_n, \quad n \in N_t,\, t = 1, \ldots, T, \\
& z_n^T \theta_n - x_n - v_n = 0, \quad n \in N_T, \\
& v_n \geq 0, \quad n \in N_T, \\
& e_n \in \{0, 1\}, \quad n \in N.
\end{aligned}
\tag{3}
$$

The second-order cone constraints arise as risk constraints that provide a lower bound for the Sharpe ratio of the final wealth position of the buyer. The term

$$\sum_{n \in N_T} p_n x_n$$

is the expected value of the final wealth position,

$$\sum_{n \in N_T} p_n \left(x_n - \sum_{k \in N_T} p_k x_k \right)^2$$

is its variance, and λ is the lower bound on the Sharpe ratio.

The second set of constraints enforce that the option is exercised at no more than one node in each sample path, and the remaining linear constraints ensure flow balance through the scenario tree. The numerical results show that these problems, with over 20,000 continuous variables, 5,000 discrete variables, and 30,000 constraints to accommodate large enough scenario trees, are quite challenging for existing MISOCP software.

2.3. Network Design and Operations

We now present a group of problems that we have loosely termed under the heading of Network Design and Operations. They arise in telecommunications networks that model the flow of commodities, cellular networks that must assign base stations to mobile units, power systems, and highway networks with vehicular traffic. Despite the similarities in the structures of the systems, the applications all have different objectives and concerns, so there are a variety of different uses for the binary variables and the second-order cone constraints in the following four applications.

2.3.1. Delays in Telecommunication Networks. Hijazi et al. [26] investigate a telecommunications network problem that seeks to minimize the network response time to a user request. The network is represented by vertices V and edges E, and vectors of capacities c and routing costs w for the edges are given. We assume that the network can handle multiple commodities grouped by the set K, there is an amount $\bar{v}_k$ of commodity k, and that each commodity k has a set of candidate paths $P(k)$, leading from its source to its destination. The decision variables in the problem are continuous variables x_e, representing the flow along edge e, and ϕ_{ik}, representing the fraction of commodity k routed along path P_{ik}, as well as binary variables z_{ik}, which indicate whether the path P_{ik} is open.

The initial model has the following form:

$$\begin{aligned}
\min\ & \sum_{e\in E} w_e x_e \\
\text{s.t.}\ & \sum_{e\in P_{ik}} \frac{1}{c_e - x_e} \le \alpha_k, \quad k\in K,\ P_{ik}\in P(k) \text{ if } z_{ik}=1, \\
& \sum_{i:\, P_{ik}\in P(k)} \phi_{ik} = 1, \quad k\in K, \\
& \sum_{k\in K}\left(\bar{v}_k \sum_{P_{ik}:\, e\in P_{ik}} \phi_{ik}\right) = x_e, \quad e\in E, \\
& x_e \le c_e, \quad e\in E, \\
& \sum_{P_{ik}\in P(k)} z_{ik} \le N, \quad k\in K, \\
& \phi_{ik} \le z_{ik}, \quad k\in K,\ P_{ik}\in P(k), \\
& z_{ik}\in\{0,1\}, \quad k\in K,\ P_{ik}\in P(k), \\
& \phi_{ik} \ge 0, \quad k\in K,\ P_{ik}\in P(k).
\end{aligned} \tag{4}$$

The objective function minimizes the total cost of all the flows along the edges of the network. With c_e denoting the capacity along edge e, the average queueing plus transmission delay using an $M/M/1$ model is computed to be $1/(c_e - x_e)$. Thus, the first constraint ensures that the total end-to-end delay on any active path through which a commodity k must travel is no greater than some parameter α_k, and it is this constraint that will require further examination. The second, third, and fourth constraints ensure that all parts of a commodity are routed, that the flow along each edge is the total flow over all the commodities that use the edge as a part of one or more associated paths, and that the total flow along an edge does not exceed the capacity of the edge. The fifth constraint states that the commodity cannot be partitioned to more than N paths, and the sixth constraint ensures that only the paths that will be opened for the commodity are allowed to have flow of that commodity along them.

The authors reexamine the first constraint, and note that since the delay along each open path is uncertain, they can also model it using a robust constraint. These constraints are also disjunctive since they are only used if the path is open. To handle both the uncertainty and the disjunction, the authors propose an extended formulation and a perspective function approach. The additional details and notation required to introduce these MISOCPs is beyond the scope of this paper, and the interested reader is referred to Hijazi et al. [26]. The numerical testing shows that CPLEX has trouble solving large MISOCP instances, whereas related MINLPs are solved in within reasonable time requirements by Bonmin.

2.3.2. Coordinated Multipoint Transmission in Cellular Networks. Cheng et al. [17] model and solve a coordinated multipoint transmission problem for cellular networks. For a network with L multiple-antenna base stations and K single-antenna mobile stations, the problem is to find w_{kl} as the beamforming vector used at base station l to transmit to mobile station k using the following model:

$$\begin{aligned}
\min\quad & \sum_{k=1}^{K}\sum_{l=1}^{L}(\|w_{kl}\|^2+\lambda_{kl}U(\|w_{kl}\|)) \\
\text{s.t.}\quad & \sum_{l=1}^{L}U(\|w_{kl}\|)\le c_k, \quad k=1,\dots,K, \\
& \mathrm{SINR}_k\ge\gamma_k, \quad k=1,\dots,K, \\
& \sum_{k=1}^{K}\|w_{kl}\|^2\le P_l, \quad l=1,\dots,L,
\end{aligned} \tag{5}$$

where λ_{kl} denotes the penalty of serving mobile station l by base station k, the function $U(x)=0$ if $x=0$ and 1 otherwise, c_k is the maximum number of base stations that can be assigned to mobile station k, SINR_k is the received signal-to-interference-plus-noise ratio (SINR) at mobile station k, γ_k is the minimum SINR level required to provide sufficient quality of service at k, and P_l is the maximum available transmit power at base station l.

The SINR constraints can be reformulated as second-order cone constraints of the form

$$\|(h_k^H W,\sigma_k)^T\|\le\sqrt{1+1/\gamma_k}\,\mathrm{Re}\{h_k^H w_k\}, \quad \mathrm{Im}\{h_k^H w_k\}=0, \ \ k=1,\dots,K,$$

where h_k represent the matrix of frequency-flat vectors to mobile station k; W is the matrix whose columns are w_k, $k=1,\dots,K$; and σ_k is the standard deviation of the white noise at mobile station k. In addition, binary variables a_{kl} and auxiliary continuous variables t_{kl} are introduced to convert (5) to the following MISOCP:

$$\begin{aligned}
\min\quad & \sum_{k=1}^{K}\sum_{l=1}^{L}(t_{kl}+\lambda_{kl}a_{kl}), \\
\text{s.t.}\quad & \|(2w_{kl}^T,a_{kl}-t_{kl})^T\|\le a_{kl}+t_{kl}, \quad k=1,\dots,K,\ l=1,\dots,L, \\
& \sum_{k=1}^{K}t_{kl}\le P_l, \quad l=1,\dots,L, \\
& \|(h_k^H W,\sigma_k)^T\|\le\sqrt{1+1/\gamma_k}\,\mathrm{Re}\{h_k^H w_k\}, \quad k=1,\dots,K, \\
& \mathrm{Im}\{h_k^H w_k\}=0, \quad k=1,\dots,K, \\
& \sum_{l=1}^{L}a_{kl}\le c_k, \quad k=1,\dots,K, \\
& a_{kl}\in\{0,1\}, \quad k=1,\dots,K,\ l=1,\dots,L, \\
& t_{kl}\ge 0, \quad k=1,\dots,K,\ l=1,\dots,L,
\end{aligned} \tag{6}$$

where the first set of second-order cone constraints are reformulations of the quadratic constraints

$$\|w_{kl}\|^2\le a_{kl}t_{kl}, \quad k=1,\dots,K,\ l=1,\dots,L$$

as described in §2.1 for the portfolio optimization problem.

Because of the large size of the problem instances, the authors propose a heuristic, which is able to obtain slightly worse solutions in significantly less CPU times than CPLEX.

2.3.3. Power Distribution Systems. Taylor and Hover [49] present several problems from power distribution system reconfiguration, one of which is formulated as an MISOCP. Given a set of lines W and a set of buses B, along with subsets $W^S \subseteq W$ with switches and $B^F \subseteq B$ with substations, the goal is to minimize the loss by choosing the right combination of open and closed switches along the system. The problem data include the real and reactive powers from each substation i, p_i^F and q_i^F; the real and reactive loads at a bus i without a substation, p_i^L and q_i^L; and resistance of the line from bus i to j, r_{ij}. The model using the *DistFlow* equations of Baran and Wu [9] has the following form:

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in W} r_{ij}(p_{ij}^2 + q_{ij}^2) \\
\text{s.t.} \quad & \sum_{k:\,(i,k)\in W} p_{ik} = p_{ji} - r_{ij}\frac{p_{ji}^2 + q_{ji}^2}{v_j^2} - p_i^L, \quad i \in B\backslash B^F, \\
& \sum_{k:\,(i,k)\in W} q_{ik} = q_{ji} - x_{ij}\frac{p_{ji}^2 + q_{ji}^2}{v_j^2} - q_i^L, \quad i \in B\backslash B^F, \\
& v_i^2 = v_j^2 - 2(r_{ij}p_{ji} + x_{ij}q_{ji}) + (r_{ij}^2 + x_{ij}^2)\frac{p_{ji}^2 + q_{ji}^2}{v_j^2}, \quad (i,j)\in W, \\
& \sum_{j:\,(i,j)\in W} p_{ij} = p_i^F, \quad i \in B^F, \\
& \sum_{j:\,(i,j)\in W} q_{ij} = q_i^F, \quad i \in B^F, \\
& 0 \le p_{ij} \le M z_{ij}, \quad (i,j)\in W, \\
& 0 \le q_{ij} \le M z_{ij}, \quad (i,j)\in W, \\
& z_{ij} \ge 0, \quad (i,j)\in W, \\
& z_{if} = 0, \quad (i,f)\in W\colon f \in B^F, \\
& z_{ij} + z_{ji} = 1, \quad (i,j)\in W\backslash W^S, \\
& z_{ij} + z_{ji} = y_{ij}, \quad (i,j)\in W^S, \\
& \sum_{j:\,(i,j)\in W} z_{ij} = 1, \quad i \in B^F, \\
& y_{ij} \in \{0,1\}, \quad (i,j)\in W^S.
\end{aligned}
\tag{7}
$$

The decision variables are the continuous p_{ij} and q_{ij} denoting the real power flow from bus i to j, continuous z_{ij} denoting the orientation of the line (i,j), and the discrete y_{ij} denoting whether the switch on the line (i,j) will be open or closed. In addition, the squared variables v_i^2 are the voltage magnitude. The first three constraints represent the *DistFlow* equations, followed by two flow balance constraints. The sixth and seventh constraints ensure that the power flow only occurs along edges with open switches, and the remaining constraints seek to define that power will flow only in one direction and in a manner consistent with the network configuration.

When converting the problem into an MISOCP, the authors drop the last term in the third constraint and replace the first three constraints with the following system, which includes auxiliary variables $\tilde{p}$, $\tilde{q}$, and $\tilde{v}^2$:

$$\sum_{j:\,(i,j)\in W} (p_{ij} - p_{ji}) - p_i^L = \tilde{p}_i, \quad i \in B\backslash B^F,$$

$$
\begin{aligned}
\sum_{j:\,(i,j)\in W} (q_{ij} - q_{ji}) - q_i^L = \tilde{q}_i, \quad i \in B \backslash B^F,\\
\tilde{v}_i^2 \le v_j^2 + M(1 - z_{ji}), \quad (i,j) \in W,\\
\tilde{v}_i^2 \ge v_j^2 - M(1 - z_{ji}), \quad (i,j) \in W,\\
r_{ij}(p_{ji}^2 + q_{ji}^2) \le \tilde{v}_i^2 \tilde{p}_i, \quad (i,j) \in W,\\
x_{ij}(p_{ji}^2 + q_{ji}^2) \le \tilde{v}_i^2 \tilde{q}_i, \quad (i,j) \in W,\\
v_i^2 \le v_j^2 - 2(r_{ij}p_{ji} + x_{ij}q_{ji}) + M(1 - z_{ij}), \quad (i,j) \in W,\\
v_i^2 \ge v_j^2 - 2(r_{ij}p_{ji} + x_{ij}q_{ji}) - M(1 - z_{ij}), \quad (i,j) \in W.
\end{aligned}
$$

Although the authors do not mention doing so, we would also need to introduce an auxiliary variable to move the quadratic objective function into a constraint and then replace the constraint with an equivalent second-order cone constraint. Numerical results are presented on 32 to 880 bus systems using CPLEX.

2.3.4. Battery Swapping Stations on a Freeway Network. Mak et al. [34] consider the problem of creating a network infrastructure and providing coverage for battery swapping stations to service electric vehicles. Given an existing freeway network, they consider candidate locations, J, and use a binary variable, x_j, for each candidate j to denote whether or not a swapping station is located there. Additional binary variables, y_{jp} and z_{jq}, denote whether vehicles traveling along a path $p \in P$ or a portion $q \in Q$ of a path along the network will visit swapping station j. The number of electric vehicles that travel along each portion of a path is random, so demand at each swapping station is uncertain. The model seeks to minimize the total cost, which consists of the fixed costs associated with opening and operating the swapping stations and the expected holding costs at each station:

$$
\begin{aligned}
\min \quad & \sum_{j\in J} (f_j x_j + hG_j(y))\\
\text{s.t.} \quad & \sum_{j\in J} a_{jq} z_{jq} \ge 1, \quad q \in Q,\\
& y_{jp} \ge b_{pq} z_{jq}, \quad j \in J, p \in P, q \in Q,\\
& y_{jp} \le x_j, \quad j \in J, p \in P,\\
& H_j(y) \ge 1 - \epsilon, \quad j \in J,\\
& x_j \in \{0,1\}, \quad j \in J,\\
& y_{jp} \in \{0,1\}, \quad j \in J, p \in P,\\
& z_{jq} \in \{0,1\}, \quad j \in J, q \in Q.
\end{aligned}
\tag{8}
$$

In the objective function, f_j is the annualized fixed cost incurred if a station is located at $j \in J$, and h is the annualized holding cost per battery and $G_j(y)$ denotes the expected largest total demand at swapping station j given the assignments of stations to paths. If Q only contains those portions that are longer than a maximum length dictated by battery life, and denote by a_{jq} a binary parameter that indicates whether station j is on portion q, then the first constraint states that there needs to be at least one swapping station along the portion q. In addition, the second constraint states that if portion q, with a station, is a part of multiple paths as indicated by the binary parameter b_{pq} with $p \in P$, then each of those paths inherit the swapping station at q. The third constraint ensures that vehicles are assigned only to stations that are open. In the fourth constraint, $H_j(y)$ is the worst-case probability of the demand at station j being less than the number of simultaneous recharges permitted by the grid, and a worst-case service level of at least $1 - \epsilon$ is guaranteed, where $\epsilon > 0$ is a small constant.

There are two parts of the problem, the nonlinear term in the objective function and the chance constraint, that have to be dealt with before obtaining an MISOCP. To handle the objective function term, introduce auxiliary variables $v_j \geq 0$ for each station j, modify the objective function to

$$\sum_{j\in J}(f_j x_j + h v_j),$$

and let

$$v_j \geq G_j(y), \quad j \in J.$$

It is shown in Mak et al. [34] that the worst-case scenario demand at j, $G_j(y)$, has an upper bound that consists of the sum of a Euclidean norm of a linear vector involving y and another linear term also involving y. Because of the multitude of additional notation in the calculation of this upper bound, we have not included the exact formulation here and invite the interested reader to read the details in Mak et al. [34]. We simply note that such a construct leads to a second-order cone constraint. The numerical studies conducted by the authors indicate that the upper bound is accurate, and they also show that it is asymptotically tight if the underlying uncertainties share the same descriptive statistics.

To handle the chance constraint indicating a robust service level requirement, the authors introduce a conditional value-at-risk constraint as was used in our earlier discussion on portfolio optimization problems. The resulting problem is an MISOCP, and they solve it with data from the San Francisco freeway network using CPLEX.

2.4. Euclidean k-Center

Brandenberg and Roth [14] introduce a new algorithm for the Euclidean k-center problem, which deals with the clustering of a group of points among k balls and arises in facility location and data classification applications. Without loss of generality, assume that sets $S_1, \ldots, S_k$ exist of points that are to be clustered together and that there are still remaining points in S_0 that have not yet been assigned a cluster. There are a total of m points in $\mathcal{R}^n$. The clusters, as stated, will be enclosed in balls, and the continuous variables in the problem are the coordinates of the centers, $c \in \mathcal{R}^n$, for each ball. The binary variable, λ_{ij}, denotes the assignment of the points $p_j \in S_0$ to ball i. The model can be formulated as follows:

$$\begin{aligned}
\min \quad & \rho \\
\text{s.t.} \quad & \|p_j - c_i\| \leq \rho, \quad p_j \in S_i,\ i = 1, \ldots, k, \\
& \|\lambda_{ij} p_j - c_i + (1-\lambda_{ij}) q_{ij}\| \leq \rho, \quad p_j \in S_0,\ i = 1, \ldots, k, \\
& \sum_{i=1}^{k} \lambda_{ij} = 1, \quad p_j \in S_0, \\
& \lambda_{ij} \in \{0, 1\}, \quad p_j \in S_0,\ i = 1, \ldots, k.
\end{aligned} \tag{9}$$

The first set of second-order cone constraints is a reformulation of the requirement to minimize the maximum Euclidean distance between a point and the center of a cluster, and it is obtained by introducing an auxiliary variable ρ to denote the maximum distance. The second set of second-order cone constraints are used to denote that if a point is assigned to a ball, then the Euclidean distance between the point and the center of the ball must be no more than the radius of the ball. If the assignment is not made, then the constraint reduces to a given reference point q_{ij}, already in the ball, being within the radius. This reference point is usually chosen as the point in S_i closest to p_j.

We should note here that the authors consider norms other than the Euclidean norm in the paper, and the MISOCP is a special case of their basic model. Numerical results are obtained using a branch-and-bound method calling SeDuMi.

2.5. Operations Management

Atamturk et al. [7] explore a joint facility location and inventory management model under stochastic retailer demand. The binary variables arise in the choice of candidate locations at which to open distribution centers and the assignment of retailers to the distribution centers. The second-order cone constraints appear in the reformulation of the uncapacitated problem to move the nonlinear objective function terms denoting the fixed costs of placing and shipping orders and the expected safety stock cost into the constraints. The complete model has the following form:

$$\begin{aligned}
\min\ & \sum_{j\in J}\Bigl(f_j x_j + \sum_{i\in I} d_{ij} y_{ij} + K_j s_j + q_j t_j\Bigr)\\
\text{s.t.}\ & \sqrt{\sum_{i\in I}\mu_i y_{ij}^2} \le s_j, \quad j\in J,\\
& \sqrt{\sum_{i\in I}\sigma_i^2 y_{ij}^2} \le t_j, \quad j\in J,\\
& \sum_{j\in J} y_{ij} = 1, \quad i\in I,\\
& y_{ij} \le x_j, \quad i\in I, j\in J,\\
& x_j\in\{0,1\}, \quad s_j, t_j \ge 0, \quad j\in J,\\
& y_{ij}\in\{0,1\}, \quad i\in I, j\in J,
\end{aligned} \tag{10}$$

where I is the set of existing retailers, J is the set of candidate locations for opening distribution centers, and the variables $x\in\mathcal{R}^{|J|}$ represent choices among the candidates, with $y\in\mathcal{R}^{|I|\times|J|}$ assigning existing retailers to the new distribution centers. Auxiliary variables are introduced to denote cost terms that are computed nonlinearly,

$$s_j = \sqrt{\sum_{i\in I}\mu_i y_{ij}}, \quad t_j = \sqrt{\sum_{i\in I}\sigma_i^2 y_{ij}},$$

and the fact that $y_{ij} = y_{ij}^2$ is used to obtain the first two constraints in (10), which are second-order cone constraints. In the otherwise linear problem, f is the vector of fixed costs for opening a distribution center at each candidate location; d is the matrix of unit shipping costs between retailers and distribution centers; K and q aid in the calculation of costs for shipping, safety stocks, and any related inventory costs for the assignments made to each distribution center; and μ and σ denote the mean and standard deviation, respectively, of the daily demand at each retailer. The third and fourth constraints in (10) ensure that each retailer is assigned to only one distribution center and that assignment is only made to those distribution centers that are open.

The model with capacities additionally has a second-order cone constraint arising from moving an objective function term for the average inventory holding cost into a constraint, and another one arising from the reformulation of a capacity constraint. Other related models with similar features are provided in the paper. Numerical results are conducted using algorithms studied in Shen et al. [43], Ozsen et al. [41], and CPLEX.

2.6. Scheduling and Logistics

Du et al. [19] present an MISOCP as a relaxation of the MINLP that arises in the problem of determining the berthing positions and order for a group of vessels, V, waiting at a container terminal to minimize the total fuel cost and waiting time of the vessels. The MINLP is formulated as follows:

$$\begin{aligned}
\min\quad & \sum_{i\in V}(c_i^0 a_i + c_i^1 m_i^{\mu_i} a_i^{1-\mu_i}) + \lambda\sum_{i\in V}(y_i + h_i - d_i)^+ \\
\text{s.t.}\quad & x_i + l_i \le L, \quad i\in V, \\
& x_i + l_i \le x_j + L(1-\sigma_{ij}), \quad i,j\in V, i\ne j, \\
& y_i + h_i \le y_j + M(1-\delta_{ij}), \quad i,j\in V, i\ne j, \\
& 1 \le \sigma_{ij} + \sigma_{ji} + \delta_{ij} + \delta_{ji} \le 2, \quad i,j\in V, i<j, \\
& \underline{a}_i \le a_i \le \bar{a}_i, \quad i\in V, \\
& a_i \le y_i, \quad i\in V, \\
& a_i, x_i \ge 0, \quad i\in V, \\
& \sigma_{ij}, \delta_{ij} \quad i,j\in V, i\ne j.
\end{aligned} \tag{11}$$

In (11), binary variables σ and δ are used to denote the relative positions of pairs of vessels (whether one vessel is to the left of another and whether one vessel is earlier than another). Additional continuous variables x, a, and y denote the leftmost berthing positions, the terminal arrivals, and the start of the berthing times for each vessel, respectively. In the problem data, L denotes the wharf length at the terminal, and l, h, and d denote the length, handling time, and requested departure time of each vessel, respectively. As is customary, M denotes an arbitrary large constant. The first four constraints of the problem are linear and serve to set the rules on wharf length and the positions and handling time of each vessel. Figure 1 from Du et al. [19] depicts an example which clarify these relationships. The fifth and sixth constraints allow the vehicle to adjust its sailing speed to save fuel—the actual arrival time at the terminal is allowed to be in an interval $[\underline{a}_i, \bar{a}_i]$, while still remaining before the berthing time y_i.

The MINLP model (11) incorporates two objective functions, fuel consumption and total departure delay, both of which are minimized. We have introduced a weight of λ to combine these two objective functions and simplify the problem. Let us first discuss the fuel consumption objective function. This function is obtained using regression analysis for each vessel, and c^0 and c^1 denote the regression coefficients, m denotes the distance of the vessels from the terminal, and $\mu_i \in \{3.5, 4, 4.5\}$ for each $i\in V$. Introducing auxiliary variables $q\in\mathcal{R}^{|V|}$, this function can be rewritten as

$$\sum_{i\in V}(c_i^0 a_i + c_i^1 m_i^{\mu_i} q_i)$$

if the constraints

$$a_i^{1-\mu_i} \le q_i, \quad q_i \ge 0, \quad i\in V,$$

are introduced. These constraints can then be transformed into hyperbolic inequalities and then rewritten as second-order cone constraints. When $\mu_i = 3.5$, for example, additional variables $u_{i1}, u_{i2} \ge 0$ and the additional constraints

$$\|(2u_{i1}, a_i - 1)\| \le a_i + 1, \quad \|(2u_{i2}, u_{i1} - q_i)\| \le u_{i1} + q_i, \quad \|(2, a_i - u_{i2})\| \le a_i + u_{i2}$$

can be introduced. Similar transformations for $\mu_i = 4$ and 4.5 are given in Du et al. [19].

The second objective function is handled by introducing auxiliary variables $t \in \mathcal{R}^{|V|}$, rewriting it as

$$\sum_{i \in V} t_i,$$

and introducing the linear constraints

$$y_i + h_i - d_i \le t_i, \quad t_i \ge 0, \quad i \in V.$$

With these transformations, the resulting problem is an MISOCP. Numerical results are conducted for instances up to 28 vessels using CPLEX, which has runtime and memory problems as the problem size grows.

3. Algorithms for Second-Order Cone Programming

In this section, we give a brief overview of several algorithms for solving the underlying SOCPs. These algorithms will have a significant impact on the design and efficiency of the overall MISOCP methods that will be discussed in the next section. For a thorough overview of SOCP, including theory, applications, and solution algorithms, we refer the reader to Alizadeh and Goldfarb [2].

3.1. Interior-Point Methods for SOCP

The continuous relaxation of (1) is given by a problem of the same form as (1), but with $x \in \mathcal{R}^n$. To write the dual problem, let us first introduce auxiliary variables $(t_{0i}, t_i) \in \mathcal{R}^{m_i+1}$ for $i = 1, \ldots, m$ and rewrite the continuous relaxation of (1) as

$$\begin{aligned} \min_{x,\, t_0,\, t} \quad & c^T x \\ \text{s.t.} \quad & t_{0i} = a_{0i}^T x + b_{0i}, \quad i = 1, \ldots, m, \\ & t_i = A_i x + b_i, \quad i = 1, \ldots, m, \\ & \|t_i\| \le t_{0i}, \quad i = 1, \ldots, m. \end{aligned} \tag{12}$$

The dual problem can now be written as

$$\begin{aligned} \max_{\lambda_0,\, \lambda} \quad & \sum_{i=1}^{m} (b_i^T \lambda_i + b_{0i}^T \lambda_{0i}) \\ \text{s.t.} \quad & \sum_{i=1}^{m} (A_i^T \lambda_i + a_{0i} \lambda_{0i}) = c, \\ & \|\lambda_i\| \le \lambda_{0i}, \quad i = 1, \ldots, m, \end{aligned} \tag{13}$$

where $(\lambda_{0i}, \lambda_i) \in \mathcal{R}^{m_i+1}$ for $i = 1, \ldots, m$ are the dual variables. Assuming that we have strict interiors for (12) and (13), strong duality holds and the optimality conditions for (12) are

$$\begin{gathered} t_{0i} = a_{0i}^T x + b_{0i}, \quad i = 1, \ldots, m, \\ t_i = A_i x + b_i, \quad i = 1, \ldots, m, \\ \sum_{i=1}^{m} (A_i^T \lambda_i + a_{0i} \lambda_{0i}) = c, \\ \|t_i\| \le t_{0i}, \quad i = 1, \ldots, m, \\ \|\lambda_i\| \le \lambda_{0i}, \quad i = 1, \ldots, m, \\ (t_{0i}, t_i) \circ (\lambda_{0i}, \lambda_i) = 0, \quad i = 1, \ldots, m, \end{gathered} \tag{14}$$

where

$$(t_{0i}, t_i) \circ (\lambda_{0i}, \lambda_i) = (t_i^T \lambda_i, t_{0i}\lambda_i + \lambda_{0i} t_i)^T.$$

As with linear programming, an interior-point method starts by introducing a barrier parameter $\mu > 0$, perturbing the last (complementarity) condition in (14) as

$$(t_{0i}, t_i) \circ (\lambda_{0i}, \lambda_i) = 2\mu e_i, \quad i = 1, \ldots, m,$$

with $e_i = \binom{1}{0^{m_i}}$, and initializing t and λ on the strict interior of the second-order cone. The Newton system associated with the perturbed conditions

$$\begin{aligned} t_{0i} &= a_{0i}^T x + b_{0i}, \quad i = 1, \ldots, m, \\ t_i &= A_i x + b_i, \quad i = 1, \ldots, m, \\ \sum_{i=1}^{m} &(A_i^T \lambda_i + a_{0i}\lambda_{0i}) = c, \\ (t_{0i}, t_i) \circ (\lambda_{0i}, \lambda_i) &= 2\mu e_i, \quad i = 1, \ldots, m, \end{aligned}$$

is solved at each iteration, but a scaling, such as HRVW/KSH/M (Helmberg et al. [25], Kojima et al. [28], Monteiro [37]), AHO (Alizadeh et al. [3]), or NT (Nesterov and Todd [39, 40]), may be needed to do obtain iterates on the interior of the second-order cone. The barrier parameter is also reduced at each iteration. Optimality is declared when (14) are satisfied to within a small tolerance.

Interior-point methods for SOCP are theoretically robust and computationally efficient. However, there are drawbacks when they are used within an MISOCP framework, including the need to start from a strictly feasible primal–dual pair of solutions and the accuracy level of the optimal solution obtained. The former makes it hard to warm start the algorithm from a previously obtained solution, whereas the latter may create issues with declaring feasibility with respect to the integer variables and adding cuts to the underlying SOCP.

3.2. Extensions of Interior-Point Methods for NLP to SOCP

Whereas SOCP can be seen as an extension of linear programming, it can also be seen as a special case of nonlinear programming. The formulation (12) is already in the form of a structured, convex NLP. Therefore, another possibility for a solution algorithm is to use interior-point methods that have been developed for NLP. However, an interior-point method for NLP requires that all objective and constraint functions be twice continuously differentiable, so the main challenge in using such a method is the nondifferentiability of the second-order cone constraint functions due to the use of the Euclidean norm.

Benson and Vanderbei [10] investigated the nondifferentiability of an SOCP and proposed several reformulations of the second-order cone constraint to overcome this issue. Note that the nondifferentiability is only an issue if it occurs at the optimal solution. Since an initial solution can be randomized, especially when using an infeasible interior-point method to solve the SOCP, the probability of encountering a point of nondifferentiability is 0.

For a constraint of the form

$$\|t_i\| \le t_{0i}, \tag{15}$$

Benson and Vanderbei [11] proposed the following.

- Exponential reformulation: Replacing (15) with $e^{(t_i^T t_i - t_{0i}^2)/2} \le 1$ and $t_{0i} \ge 0$ gives a smooth and convex reformulation of the problem, but numerical issues frequently arise because of the exponential.
- Smoothing by perturbation: Introducing a scalar variable v into the norm gives a constraint of the form $\sqrt{v^2 + t_i^T t_i} \le t_{0i}$, but for the formulation to be smooth, we need $v > 0$. This is ensured by setting $v \ge \epsilon$ for a small constant ϵ, usually taken around $10^{-6} - 10^{-4}$.

• Ratio reformulation: Replacing (15) with $(t_i^T t_i)/t_{0i} \le t_{0i}$ and $t_{0i} \ge 0$ yields a convex reformulation of the problem, but the constraint function may still not be smooth. Nevertheless, in many applications, such as the portfolio optimization problems to be studied in the next section, the right-hand side of the second-order cone constraint in (1) is either a scalar or bounded away from zero at the optimal solution.

Benson and Sağlam [11] used the ratio reformulation for solving portfolio optimization problems, as they were able to show that the right-hand side of the cone constraints were sufficiently bounded away from 0 at the optimal solution. For a general problem, depending on the scale of the numbers, the level of accuracy required, and any such bounds on the right-hand sides of the constraints, any one of the above three reformulations may be more preferable over the others.

Once the reformulation is complete, the problem can be solved using any variant of an interior-point method. Benson and Vanderbei [10] used the infeasible primal–dual interior-point method that was implemented in LOQO (Vanderbei [51]). Introducing a barrier parameter $\mu > 0$ and slack variables w, $s \ge 0$, the perturbed optimality conditions for an SOCP that has undergone the ratio reformulation can be expressed as

$$\begin{aligned} t_{0i} &= a_{0i}^T x + b_{0i}, \quad i = 1, \ldots, m, \\ t_i &= A_i x + b_i, \quad i = 1, \ldots, m, \\ \sum_{i=1}^{m} (A_i^T \lambda_i &+ a_{0i} \lambda_{0i}) = c, \\ \frac{t_i^T t_i}{t_{0i}} + w &= t_{0i}, \quad i = 1, \ldots, m, \\ \frac{\lambda_i^T \lambda_i}{\lambda_{0i}} + s &= \lambda_{0i}, \quad i = 1, \ldots, m, \\ w_i s_i &= \mu, \quad i = 1, \ldots, m. \end{aligned} \tag{16}$$

Starting at an initial solution with w, $s > 0$, we solve the Newton system associated with (16) at each iteration, find an appropriate steplength using a merit function or a filter, and update the barrier parameter as needed. The algorithm stops when it satisfies (16), with μ sufficiently close to 0, to a desired level of accuracy.

Using this approach, the accuracy of the solution to the SOCP still remains a concern. However, due to recent work in the area, the warm-start issue is starting to get resolved. We refer the reader to Benson and Sağlam [11] for details on a primal–dual penalty method that enables warm starts when solving SOCPs.

3.3. Lifted Polyhedral Relaxation

A very different perspective on (approximately) solving SOCPs is to employ a polyhedral relaxation of the convex feasible region and to solve a related linear programming problem instead. However, in doing so, it is important to ensure that the size of the linear programming problem remains tractable. Ben-Tal and Nemirovski [12] have presented a lifted polyhedral relaxation that uses a polynomial number of constraints and auxiliary variables, and this relaxation method has been further refined by Glineur [23].

Starting with an SOCP of the form (12), let us focus on the constraint $\|t_i\| \le t_{0i}$ for some i. The goal is to construct a polyhedron that is ϵ-tight, i.e., satisfies $\|t_i\| \le (1+\epsilon)t_{0i}$ for a small $\epsilon > 0$. For ease of presentation, we assume that m_i is an integer power of 2. (We refer the reader to the details provided in Ben-Tal and Nemirovski [12] and Glineur [23] for the case when m_i is not an integer power of 2.) If the variables are grouped into $r/2$ pairs and an auxiliary variable ρ_j is associated with the jth pair, then the set of points satisfying the original cone constraint can be rewritten as

$$\{(t_{0i},t_i)\in\mathcal{R}^{m_i+1}\colon \exists\rho\in\mathcal{R}^{m_i/2} \text{ s.t. } \rho^T\rho\le t_{0i}^2, t_{i(2j-1)}^2+t_{i(2j)}^2\le\rho_j^2,\, j=1,\ldots,m_i\}.$$

This new definition uses one cone of dimension $m_i/2+1$ and $m_i/2$ cones of dimension 3. This process is recursively applied to cone of dimension $m_i/2+1$ until there are only three-dimensional cones left. Then, each three-dimensional cone can be replaced with a polyhedral relaxation of the form

$$\begin{aligned}\Bigg\{(r_0,r_1,r_2)\in\mathcal{R}^3\colon r_0\ge 0 \quad \text{and} \quad \exists(\alpha,\beta)\in\mathcal{R}^{2s} \text{ s.t.}&\\ r_0=\alpha_s\cos\left(\frac{\pi}{2^s}\right)+\beta_s\sin\left(\frac{\pi}{2^s}\right),&\\ \alpha_1=r_1\cos(\pi)+r_2\sin(\pi),&\\ \beta_1\ge|r_2\cos(\pi)-r_1\sin(\pi)|,&\\ \alpha_{i+1}=\alpha_i\cos\left(\frac{\pi}{2^i}\right)+\beta_i\sin\left(\frac{\pi}{2^i}\right),\quad i=1,\ldots,s-1,&\\ \beta_{i+1}\ge\left|\beta_i\cos\left(\frac{\pi}{2^i}\right)-\alpha_i\sin\left(\frac{\pi}{2^i}\right)\right|,\quad i=1,\ldots,s-1\Bigg\},&\end{aligned} \tag{17}$$

for some $s\in\mathcal{Z}$.

Given that the resulting problem is a linear program, it can be solved using any number of suitable methods, including the simplex method or a crossover approach, both of which would yield very efficient warm starts.

4. Algorithms for Mixed-Integer Second-Order Cone Programming

One very straightforward way to devise a method for solving MISOCPs is to use a branch-and-bound algorithm that calls an interior-point method designed specifically for SOCPs at each node. However, if such a method is to be competitive on large-scale MISOCPs, it is important to reduce the number of nodes in the tree using cuts and relaxations designed specifically for MISOCPs and to reduce the runtime at each node using an SOCP solver that is capable of warmstarting and infeasibility detection.

There are other approaches for MINLP besides branch-and-bound, which can similarly be adopted for the case of MISOCP, using the fact that the underlying SOCPs are essentially convex NLPs. These approaches include outer approximation (Duran and Grossmann [20]), extended cutting-plane methods (Westerlund and Pettersson [53]), and generalized Benders decomposition (Geoffrion [22]). However, the nondifferentiability of the constraint functions in (1) is of particular concern when generating the gradient-based cuts required by these methods, and their application to MISOCP should be done carefully and by considering this special case.

Additionally, any method that can convert the underlying SOCPs into linear programming problems can take advantage of the efficient algorithms designed for MILP.

A number of studies appear in literature dealing with algorithms specifically for MISOCP, and we will now present them here.

4.1. Gomory Cuts and Tight Relaxations

Cezik and Iyengar [16] study mixed-integer conic programming problems, of which both mixed-integer linear programming problems and MISOCPs are subsets. Their approach is to extend some well-known techniques for mixed-integer linear programming to mixed-integer programs involving second-order cone and/or semidefinite constraints. Since the problem setup in Cezik and Iyengar [16] includes a more general cone than ours, we have adapted their discussion to the case of the second-order cone.

Their first extension is that of Gomory cuts to integer conic programs. For the case of integer SOCPs, they note that

$$
\begin{aligned}
&\{x \in \mathcal{R}^n \colon \|A_i x + b_i\| \le a_{0i}^T x + b_{0i}, i = 1, \ldots, m\} \\
\Leftrightarrow\ &\Bigg\{x \in \mathcal{R}^n \colon \Bigg(\sum_{i=1}^{m}\Bigg(a_{0i}u_{0i} + \sum_{j=1}^{n} a_{ij}^T u_i\Bigg)\Bigg)^T x \ge \sum_{i=1}^{m}(b_{0i}u_{0i} + b_i^T u_i), (u_{0i}, u_i)^T \in K_i^*, \\
&\qquad i = 1, \ldots, m\Bigg\},
\end{aligned} \tag{18}
$$

where $A_i = [a_{i1}, a_{i2}, \ldots, a_{in}]$, and K_i^* is the dual cone of the ith second-order cone. This equivalence leads to the following natural extension of the Chvátal–Gomory procedure for integer SOCPs:

(1) Choose $(u_{0i}, u_i)^T \in K_i^*, i = 1, \ldots, m$. Then,

$$\Bigg(\sum_{i=1}^{m}\Bigg(a_{0i}u_{0i} + \sum_{j=1}^{n} a_{ij}^T u_i\Bigg)\Bigg)^T x \ge \sum_{i=1}^{m}(b_{0i}u_{0i} + b_i^T u_i).$$

(2) Without loss of generality, $x \ge 0$, so

$$\Bigg(\sum_{i=1}^{m}\Bigg(\lceil a_{0i}\rceil u_{0i} + \sum_{j=1}^{n} \lceil a_{ij}\rceil^T u_i\Bigg)\Bigg)^T x \ge \sum_{i=1}^{m}(b_{0i}u_{0i} + b_i^T u_i).$$

(3) By the integrality of x, it holds that

$$\Bigg(\sum_{i=1}^{m}\Bigg(\lceil a_{0i}\rceil u_{0i} + \sum_{j=1}^{n} \lceil a_{ij}\rceil^T u_i\Bigg)\Bigg)^T x \ge \sum_{i=1}^{m}(\lceil b_{0i}\rceil u_{0i} + \lceil b_i\rceil^T u_i)$$

is a valid linear inequality that can be added to the cone constraints.

The authors also prove that every valid inequality for the convex hull of the feasible region of an integer SOCP can be obtained by repeating the above procedure a finite number of times.

The second extension is that of sequential convexification to the case of integer conic programs. This approach, which was studied in Balas et al. [8], Sherali and Adams [44, 45], Lovasz and Schrijver [33], and Lasserre [30] for pure and mixed-integer linear programming problems, can provide tighter relaxations than the continuous relaxation of the integer SOCP. To extend the Lovasz–Schrijver and Balas–Ceria–Cornuejols hierarchies, the authors start by picking a subset of size l of the variables and introduce $Y^0 = [y_1^0 \ldots y_l^0]$ and $Y^1 = [y_1^1 \ldots y_l^1]$ with

$$y_k^0 = (1 - x_{j_k})\begin{pmatrix}1\\ x\end{pmatrix}, \quad y_k^1 = x_{j_k}\begin{pmatrix}1\\ x\end{pmatrix}, \quad k = 1, \ldots, l.$$

Then, the following is a relaxation for (1) with all binary variables:

$$\begin{aligned}
\min\ & c^T x \\
\text{s.t.}\ & y_k^0 + y_k^1 = \begin{pmatrix} 1 \\ x \end{pmatrix}, \quad k = 1, \ldots, l, \\
& \left\| \sum_{j=1}^{n} y_{jk}^0 a_{ij} + y_{0k}^0 b_i \right\| \le \sum_{j=1}^{n} y_{jk}^0 a_{0j} + y_{0k}^0 b_0, \quad k = 1, \ldots, l, \\
& \left\| \sum_{j=1}^{n} y_{jk}^1 a_{ij} + y_{0k}^1 b_i \right\| \le \sum_{j=1}^{n} y_{jk}^1 a_{0j} + y_{0k}^1 b_0, \quad k = 1, \ldots, l, \\
& y_{kk}^0 = 0, \quad k = 1, \ldots, l, \\
& y_{kk}^1 = y_{0k}^1, \quad k = 1, \ldots, l, \\
& Y^1 = (Y^1)^T.
\end{aligned}$$

To extend the Sherali–Adams and Laserre hierarchies, the authors also start by picking a subset of size l of the variables and call this subset B. Let y be a vector that is indexed by the empty set, subsets $H \subseteq B$, and sets of the form $H \cup \{j\}$ for j not picked for B, and define y as follows:

$$y_I = \begin{cases} 1 & I = \varnothing, \\ \prod_{j \in I} x_j & \text{otherwise.} \end{cases}$$

Then, define $z_0^I \in \mathcal{R}$ and $z^I \in \mathcal{R}^n$ for $I \subseteq B$:

$$\begin{aligned}
z_0^I &= \prod_{j \in I} x_j \prod_{j \in B \setminus I} (1 - x_j) = \sum_{I \subseteq H \subseteq B} (-1)^{|B \setminus H|} y_H \ge 0, \\
z_k^I &= x_k \prod_{j \in I} x_j \prod_{j \in B \setminus I} (1 - x_j) = \sum_{I \subseteq H \subseteq B} (-1)^{|B \setminus H|} y_{H \cup \{k\}}, \quad k = 1, \ldots, n.
\end{aligned}$$

Thus, the following problem is a relaxation of the binary SOCP:

$$\begin{aligned}
\min\ & c^T x \\
\text{s.t.}\ & x_j = y_{\{j\}}, \quad k = 1, \ldots, n, \\
& \|A_i z^I + b_i z_0^I\| \le a_{0i}^T z^I + b_{0i} z_0^I, \quad I \subseteq B.
\end{aligned}$$

Additional hierarchies based on these principles are also discussed in the paper.

The authors propose a cut algorithm can use the Chvátal–Gomory procedure and the tighter relaxations. However, the success of this algorithm is rather limited since the authors consider only interior-point methods for SOCP as the solution algorithm for the underlying SOCPs and implement it using SeDuMi. As they note, the use of interior-point methods results in a solution that is feasible subject to a tolerance and may need rounding prior to applying the cut generation procedure. In addition, warm starts from feasible dual solutions are not available within SeDuMi, as is the case for most other codes for mixed-integer conic programs. Noting these limitations, the authors present preliminary numerical results and pointers for future improvement.

4.2. Rounding Cuts

Atamturk and Narayanan [5] focus on MISOCPs and their solution using a branch-and-bound framework. They introduce rounding cuts obtained by first decomposing each second-order cone constraint into polyhedral sets. To introduce this approach, we first note that according to the definition of (1), the variable x can be decomposed into (y,z): $y \in \mathcal{Z}^p, z \in \mathcal{R}^k$, $p+k=n$. For each second-order cone $i=1,\ldots,m$, and partitioning the columns of A_i into A_i^y and A_i^z and the vector a_{0i} into a_{0i}^y and a_{0i}^z, the constraints of (12) can be rewritten as follows:

$$\begin{aligned} t_{0i} &\le (a_{0i}^y)^T y + (a_{0i}^z)^T z + b_{0i}, \\ t_i &\ge |A_i^y y + A_i^z z - b_i|, \\ \|t_i\| &\le t_{0i}. \end{aligned}$$

We assume that the absolute value in the second constraint is element-wise and focus on one such constraint, which we will write as

$$|a^y y + a^z z + b| \le t, \tag{19}$$

where a^y is a row of A_i^y for some i, a^z is the corresponding row in A_i^z, and b and t are the corresponding elements of b_i and t_i, respectively. This form is both for ease of notation and to better match the exposition in Atamturk and Narayanan [5]. The set S is defined as $\{y \in \mathcal{Z}^p, z \in \mathcal{R}^k, t \in \mathcal{R}\colon |a^y y + a^z z + b| \le t, y \ge 0, z \ge 0\}$, where the nonnegativities of y and z can be imposed without loss of generality (i.e., if a variable is free, we can always split it into two nonnegative ones). Grouping the terms of $a^z z$ with positive and negative coefficients into z^+ and z^-, respectively, (19) is rewritten as

$$|a^y y + z^+ - z^- + b| \le t. \tag{20}$$

The authors first define a rounding function φ_f for $0 \le f < 1$ as

$$\varphi_f(v) = \begin{cases} (1-2f)n - (v-n) & \text{if } n \le v < n+f, \\ (1-2f)n + (v-n) - 2f & n+f \le v < n+1, \end{cases}$$

where $n \in \mathcal{Z}$. Then, they show that the following is a valid inequality for S:

$$\sum_{j=1}^{n} \varphi_f(a_j/\alpha) y_j - \varphi_f(b/\alpha) \le (t + z^+ + z^-)/|\alpha|$$

for any $\alpha \ne 0$ and $f = b/\alpha - \lfloor b/\alpha \rfloor$. In addition, if $b/a_i > 0$ for some i and $\alpha = a_i$, then the above inequality is shown to be facet defining for the convex hull of S.

These rounding cuts are added at the root node of the branch-and-bound tree, and the preliminary results in Atamturk and Narayanan [5] show that the cuts can significantly reduce the number of nodes in the tree. The authors provide a more thorough analysis and further examples showing the success of their approach in Atamturk and Narayanan [6].

4.3. MILP Methods Applied to Lifted Polyhedral Relaxation

Vielma et al. [52] propose using a lifted polyhedral relaxation (Ben-Tal and Nemirovski [12], Glineur [23]; described in §3.3) of the underlying SOCPs, thereby solving the MISOCPs using a linear programming-based branch-and-bound framework. Their approach can be generalized to any convex MINLP, does not use gradients to generate the cuts, and benefits from the linear programming structure that can use a simplex-based method with warm-starting capabilities within the solution process.

The *lifted LP branch-and-bound algorithm* presented in Vielma et al. [52] is too detailed to present in its entirety here, so we will give a brief outline, and the interested reader is referred to Vielma et al. [52], particularly Figure 1 of that paper. In general, the algorithm proceeds as the usual branch-and-bound method for MILP by branching on discrete variables with noninteger values, except solving the lifted polyhedral relaxation of the associated SOCP at each node. If a feasible solution is found at any node, the continuous relaxation of the MISOCP is solved at that node to see if the exact solution (rather than an ϵ-tight relaxation) still yields a feasible solution. If so, the node is fathomed by integrality. Otherwise, we continue branching on a discrete variable with a noninteger value. This approach ensures that only linear programming problems are solved at most nodes of the tree and limits the solution of the underlying SOCPs to a much smaller number of nodes.

Numerical studies on portfolio optimization problems show that the method outperforms CPLEX and Bonmin. A similar method is used in Soberanis [46] to solve the MISOCP reformulations of risk optimization problems with p-order conic constraints.

4.4. Extensions of Convex MINLP Methods to MISOCP

Drewes [18] proposes both a branch-and-cut method and a hybrid branch-and-bound/outer approximation method for solving MISOCPs. The branch-and-cut method uses techniques similar to Cezik and Iyengar [16] and those developed in Stubbs and Mehrotra [47] for mixed-integer convex optimization problems with binary variables. Therefore, given its similarity to the method presented in §4.1, we will not present this approach, but instead provide details on the hybrid branch-and-bound/outer approximation method. Numerical results for both methods are provided in Drewes [18] for a number of test problems.

The hybrid approach extends outer approximation methods, which use gradient-based techniques to generate cuts, to the case of MISOCPs using subgradients. As in outer approximation, constraints of the form $\|t_i\| \le t_{0i}$ are replaced by

$$(\|\bar{t}_i\| - \bar{t}_{0i}\|) + \xi_i^T(t_i - \bar{t}_i) + \xi_{0i}(t_{0i} - \bar{t}_{0i}) \le 0,$$

where $(\bar{t}_{0i}, \bar{t}_i)$ is part of the solution of a continuous relaxation of the MISOCP, and (ξ_0, ξ) is a subgradient of the second-order cone constraint function $\|t_i\| - t_{0i}$ at $(\bar{t}_{0i}, \bar{t}_i)$. If $\bar{t}_i \ne 0$, the gradient can be used, and set

$$\xi_{0i} = -1 \quad \text{and} \quad \xi_i = \frac{\bar{t}_i}{\|\bar{t}_i\|}.$$

Otherwise, the dual variables $(\bar{\lambda}_{0i}, \bar{\lambda}_i)$ can be used to get an appropriate subgradient. Drewes [18] proposes that

$$\xi_{0i} = -1 \quad \text{and} \quad \xi_i = \begin{cases} -\dfrac{\bar{\lambda}_i}{\bar{\lambda}_{0i}} & \text{if } \bar{\lambda}_{0i} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Additional cuts are generated from infeasible instances and are described in Drewes [18].

Benson and Sağlam [11] propose two MINLP methods, branch-and-bound and outer approximation, for solving MISOCPs. Since the underlying problems are smoothed using the ratio reformulation, as described in §3.2, and a primal–dual penalty method is applied to the interior-point algorithm to enable warm starts and infeasibility detection, both MINLP methods can be applied directly and efficiently to solve an MISOCP. Preliminary numerical results on problems arising in portfolio optimization are encouraging.

5. Summary

In this tutorial, we gave an overview of the state of the art in mixed-integer second-order cone programming problems. We described numerous applications and a handful of solution algorithms. Given the wide range of fields from which the applications arise, we anticipate that this problem class will continue to flourish. The solution methods for MISOCP are still at their infancies, however, so for the growth of this problem class, it is important to continue to address issues of warm starts and levels of accuracy in methods for solving the continuous relaxations and to add to the types of cuts available to improve the efficiency of overall solution approaches. The lifted LP branch-and-bound algorithm presents another opportunity for algorithmic improvement, and it may be useful to investigate other approaches for solving SOCPs using an LP-based approach within the MISOCP framework.

Acknowledgments

The authors would like to thank Huseyin Topaloglu for inviting them to contribute this tutorial to the INFORMS 2013 Annual Meeting and for providing feedback on an earlier version of this paper.

References

[1] C. J. Adcock and N. Meade. A simple algorithm to incorporate transactions costs in quadratic optimisation. *European Journal of Operational Research* 79(1):85–94, 1994.

[2] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming* 95(1):3–51, 2003.

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM Journal on Optimization* 8(3):746–768, 1998.

[4] E. Andersen and K. Andersen. The MOSEK optimization software. EKA Consulting ApS, Copenhagen, Denmark.

[5] A. Atamturk and V. Narayanan. Cuts for conic mixed-integer programming. M. Fischetti and D. P. Williamson, eds. *Proceedings of the 2007 International Integer Programming and Combinatorial Optimization Conference*, Springer, Berlin, Heidelberg, 16–29, 2007.

[6] A. Atamturk and V. Narayanan. Conic mixed-integer rounding cuts. *Mathematical Programming Series A* 122:1–20, 2010.

[7] A. Atamturk, G. Berenguer, and Z.-J. M. Shen. A conic integer programming approach to stochastic joint location-inventory problems. *Operations Research* 60(2):366–381, 2012.

[8] E. Balas, S. Ceria, and G. Cornuejols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming* 58(1–3):295–324, 1993.

[9] M. Baran and F. Wu. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery* 4(2):1401–1407, 1989.

[10] H. Y. Benson and R. J. Vanderbei. Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming. *Mathematical Programming, Series B* 95(2):279–302, 2003.

[11] H. Y. Benson and U. Sağlam. Mixed-integer second-order cone programming: Algorithms and applications. Technical report, Drexel University, Philadelphia, 2012.

[12] A. Ben-Tal and A. Nemirovski. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research* 26(2):193–205, 2001.

[13] P. Bonami and M. A. Lejeune. An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations Research* 57(3):650–670, 2009.

[14] R. Brandenberg and L. Roth. New algorithms for k-center and extensions. B. Yang, D.-Z. Du, and C. A. Wang, eds. *Proceedings of the 2008 International Conference on Combinatorial Optimization and Applications*, Springer, Berlin, Heidelberg, 64–78, 2008.

[15] D. B. Brown and J. E. Smith. Dynamic portfolio optimization with transaction costs: Heuristics and dual bounds. *Management Science* 57(10):1752–1770, 2011.

[16] M. T. Cezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming, Series A* 104(1):179–202, 2005.

[17] Y. Cheng, S. Drewes, A. Philipp, and M. Pesavento. Joint network optimization and beamforming for coordinated multi-point transmission using mixed integer programming. *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, Piscataway, NJ, 3217–3220, 2012.

[18] S. Drewes. Mixed integer second order cone programming. Ph.D. thesis, Technischen Universitat Darmstadt, Munich, Germany, 2009.

[19] Y. Du, Q. Chen, X. Quan, L. Long, and R. Y. K. Fung. Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E* 47:1021–1037, 2011.

[20] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* 36(3):307–339, 1986.

[21] N. B. Garleanu and L. H. Pedersen. Dynamic trading with predictable returns and transaction costs. Technical report, National Bureau of Economic Research, Cambridge, MA, 2009.

[22] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications* 10(4):237–260, 1972.

[23] F. Glineur. Computational experiments with a linear approximation of second-order cone optimization. Technical Report 0001, Faculte Polytechnique de Mons, Mons, Belgium, 2000.

[24] D. Goldfarb. The simplex method for conic programming. Technical Report TR-2002-05, Computational Optimization Research Center, Department of Industrial Engineering and Operations Research, Columbia University, New York, 2002.

[25] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior point method for semidefinite programming. *SIAM Journal on Optimization* 6:342–361, 1996.

[26] H. Hijazi, P. Bonami, and A. Ouorou. Robust delay-constrained routing in telecommunications. Technical report, Centre National de la Recherche Scientifique, Aix-Marseille Universite, Marseille, France, 2012.

[27] IBM ILOG. CPLEX 11.0 reference manual. Accessed July 28, 2013, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[28] M. Kojima, S. Shindoh, and S. Hara. Interior-point methods for the finite semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization* 7(1):86–125, 1997.

[29] K. Krishnan and J. Mitchell. Linear programming approaches to semidefinite programming problems. Technical report, Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY, 2001.

[30] J. B. Lasserre. Global optimization with polynomials and the methods of moments. *SIAM Journal on Optimization* 1:796–817, 2001.

[31] M. S. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research* 152(1):341–365, 2007.

[32] M. S. Lobo, L. Vandenberghe, H. Lebret, and S. Boyd. Applications of second-order cone programming. *Linear Algebra and Applications* 284(1–3):13–228, 1998.

[33] L. Lovasz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization* 1(2):166–190, 1991.

[34] H.-Y. Mak, Y. Rong, and Z.-J. M. Shen. Infrastructure planning for electric vehicles with battery swapping. Technical report, Hong Kong University of Science and Technology, Kowloon, Hong Kong, 2012.

[35] H. Markowitz. Portfolio selection. *Journal of Finance* 7(1):77–91, 1952.

[36] H. M. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. John Wiley & Sons, New York, 1959.

[37] R. D. C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optimization* 7(3):663–678, 1997.

[38] Y. E. Nesterov and A. S. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM Publications, Philadelphia, 1993.

[39] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research* 22(1):1–42, 1997.

[40] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization* 8(2):324–364, 1998.

[41] L. Ozsen, C. R. Coullard, and M. S. Daskin. Capacitated warehouse location model with risk pooling. *Naval Research Logistics* 55:295–312, 2008.

[42] M. C. Pinar. Mixed-integer second-order cone programming for lower hedging of American contingent claims in incomplete markets. *Optimization Letters* 7(1):63–78, 2013.

[43] Z. J. M. Shen, C. R. Coullard, and M. S. Daskin. A joint location-inventory model. *Transportation Science* 37(1):40–55, 2003.

[44] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations of zero-one programming problems. *SIAM Journal of Discrete Mathematics* 3:411–430, 1990.

[45] H. D. Sherali and W. P. Adams. A hierarchy of relaxations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics* 52:83–106, 1994.

[46] P. A. Soberanis. Risk optimization with p-order cone constraints. Ph.D. dissertation, University of Iowa, Iowa City, 2009.

[47] R. A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming, Series A* 86:515–532, 1999.

[48] J. F. Sturm. Using SeDuMi 1.02: A MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[49] J. A. Taylor and F. S. Hover. Convex models of distribution and system reconfiguration. *IEEE Transactions on Power Systems* 27(3):1407–1413, 2012.

[50] K. C. Toh, M. J. Todd, and R. H. Tutuncu. SDPT3—A MATLAB software package for semidefinite programming. *Optimization Methods and Software* 11:545–581, 1999.

[51] R. J. Vanderbei. LOQO user's manual—Version 3.10. *Optimization Methods and Software* 12:485–514, 1999.

[52] J. P. Vielma, S. Ahmed, and G. L. Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing* 20(3):438–450, 2008.

[53] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering* 19:131–136, 1995.