CORC Technical Report TR-2001-03
Cuts for mixed 0-1 conic programming

M. T. Çezik * G. Iyengar†

July 25, 2005

Abstract

In this we paper we study techniques for generating valid convex constraints for mixed 0-1 conic programs. We show that many of the techniques developed for generating linear cuts for mixed 0-1 linear programs, such as the Gomory cuts, the lift-and-project cuts, and cuts from other hierarchies of tighter relaxations, extend in a straightforward manner to mixed 0-1 conic programs. We also show that simple extensions of these techniques lead to methods for generating convex quadratic cuts. Gomory cuts for mixed 0-1 conic programs have interesting implications for comparing the semidefinite programming and the linear programming relaxations of combinatorial optimization problems, e.g. we show that all the subtour elimination inequalities for the traveling salesman problem are rank-1 Gomory cuts with respect to a single semidefinite constraint. We also include results from our preliminary computational experiments with these cuts.

1 Introduction

This paper discusses cut generation for mixed 0-1 conic programs (MCP) defined as follows.

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^T\mathbf{x}, \\ \text{subject to} & \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}, \\ & x_j \in \{0,1\}, \quad j = 1,\ldots,p, \end{array} \tag{1}$$

where $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{b} \in \mathbf{R}^m$, $\mathbf{A} = [\mathbf{a}_1,\ldots,\mathbf{a}_n] \in \mathbf{R}^{m\times n}$, $\mathbf{a}_j \in \mathbf{R}^m$, $j = 1,\ldots,n$, and $\succeq_{\mathcal{K}}$ denotes the partial ordering with respect to the cone $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \ldots \times \mathcal{K}_r$, where each $\mathcal{K}_i$ is one of the following homogeneous, self-dual, proper, convex cones:

(i) linear cone $\mathcal{K}_l = \{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$. We will denote $\mathbf{x} \in \mathcal{K}_l$ by $\mathbf{x} \geq \mathbf{0}$.

(ii) second-order cone $\mathcal{K}_q = \{\mathbf{x} = (x_0, \bar{\mathbf{x}}) : x_0 \geq \|\bar{\mathbf{x}}\|\}$, where $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$ denotes the usual Euclidean norm. We will denote $\mathbf{x} \in \mathcal{K}_q$ by $\mathbf{x} \succeq_q \mathbf{0}$.

*GERAD and Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal, H3C 3J7, Canada. Email: `ceziktol@iro.umontreal.ca`. This research was conducted while this author was at the IEOR Dept. in Columbia Universtity.

†IEOR Department, Columbia University, New York, New York 10027 USA. Email: `garud@ieor.columbia.edu`.
Research partially supported by NSF grants CCR-00-09972, DMS-01-04282 and ONR grant N000140310514.

(iii) semidefinite cone $\mathcal{K}_s = \left\{\mathbf{x} : \mathbf{x} = \mathbf{vec}(\mathbf{X}), \mathbf{X} = \mathbf{X}^T \succeq \mathbf{0}\right\}$, where $\mathbf{vec}(\mathbf{X})$ is the matrix $\mathbf{X}$ in column-major form and $\mathbf{X} \succeq \mathbf{0}$ denotes that $\mathbf{X}$ is positive semidefinite.

We will assume that the linear inequalities, $0 \leq x_j \leq 1$, $j = 1, \ldots, p$, are present in the *conic constraint* $\mathbf{Ax} \succeq_{\mathcal{K}} \mathbf{b}$. We will denote the feasible set of the MCP (1) by $\mathcal{C}^\circ$ and denote its continuous relaxation by $\mathcal{C}$, i.e. $\mathcal{C} = \{\mathbf{x} \in \mathbf{R}^n : \mathbf{Ax} \succeq_{\mathcal{K}} \mathbf{b}\}$. Here is a simple example of an MCP.

Example 1: Consider the norm-constrained set

$$\mathcal{C}^\circ = \left\{\mathbf{x} \in \mathbf{R}^3 : \|\mathbf{x}\| \leq 1, x_j \in \{0,1\}, j = 1,2\right\},$$

where $\|\cdot\|$ denotes the Euclidean norm. Since $\|x\| \leq 1$ if, and only if, $(1, x_1, x_2, x_3)^T \in \mathcal{K}_q \subset \mathbf{R}^4$, it follows that

$$\mathcal{C}^\circ = \left\{ \begin{array}{c} \mathbf{x} \in \mathbf{R}^3 : \left[\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline -1 & 0 & 0 \\ 0 & -1 & 0 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \succeq_{\mathcal{K}} \left[\begin{array}{c} -1 \\ 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline -1 \\ -1 \end{array}\right] \\ x_j \in \{0,1\}, \quad j = 1,2 \end{array} \right\},$$

where $\mathcal{K} = \mathcal{K}_q^4 \times \mathcal{K}_l^4$, $\mathcal{K}_q^4$ is the second-order cone in $\mathbf{R}^4$ and $\mathcal{K}_l^4$ is the linear cone in $\mathbf{R}^4$. It is easy to check that $\mathbf{conv}\left(\mathcal{C}^0\right) = \{\mathbf{x} : x_j \geq 0, j = 1, 2, x_1 + x_2 + |x_3| \leq 1\}$. ■

Conic constraints include a large variety of convex constraints such as linear constraints, convex quadratic constraints, linear fractional constraints, eigenvalue constraints, etc. (see [37, 30, 2, 45] for details). Thus, conic programs include linear programs (LPs) and second-order cone programs (SOCPs) and semidefinite programs (SDPs) as special cases. In the early 1990s, Nemirovski and Nesterov [37] developed interior point algorithms for solving conic programs. Since then conic programming models have found applications in many disparate fields such as robust control [12], combinatorial optimization [1, 17, 18, 15], robust optimization [9, 16], and finance [29, 8, 21].

Several applications of conic programming involve discrete variables. For instance, combinatorial optimization problems such as the TSP [15, 13] and the maxcut [19] can be formulated as mixed 0-1 SDPs. Another important source of mixed 0-1 SDPs is the semidefinite lifting of mixed 0-1 LPs [31, 25, 26]. Since this lifting yields tighter relaxations, it is likely that cuts generated from the mixed 0-1 SDP formulation may be deeper and, therefore, particularly useful in solving mixed 0-1 LPs that have a small number of binary variables but a large integrality gap. The mixed 0-1 SDP formulation allows valid conic constraints, such as norm constraints, to be used for cut generation, and this formulation can also be used to generate more general cuts. Portfolio selection problems with fixed transaction costs [29], and robust mixed 0-1 LPs, where the robustness is with respect to an uncertain constraint matrix, lead to mixed 0-1 SOCPs.

Currently MCPs are approximately solved by relaxing the binary constraints and suitably rounding the fractional solution. This is similar to the early attempts to approximately solve mixed 0-1 LPs. As the techniques for solving LPs matured, problem specific cutting planes [36] and branch-and-cut methods were developed; subsequently cutting plane methods for general mixed 0-1 LPs appeared [6]. We expect a similar trend for MCPs. In fact, Helmberg and Rendl [23, 22] have investigated the performance of problem specific cutting planes and Mitchell [35] has a proposed a branch-and-cut algorithm for the maxcut and related quadratic optimization problems.

In this paper, we propose cut generation strategies for general MCPs. The intent of this paper is to show that cut generation strategies for general mixed 0-1 LPs, such as the Gomory cuts [36] and cuts from tighter convex relaxations [6, 31, 41, 25], extend in a very natural manner to general MCPs. While many of our results follow from those in the literature, these extensions have never been carefully explored. Since second-order cone constraints and semidefinite constraints can be reformulated as smooth convex constraints [10], disjunctive cuts [5] for MCPs are a special case of disjunctive cuts for mixed 0-1 convex programs proposed in Stubbs and Mehrotra [42]. However, restricting oneself to conic programs, allows one to use conic programming duality to compactly characterize the set of valid cuts and choose an appropriate cut. Also, given the number of applications, cut generation for conic programs is worth studying in its own right.

The organization of the paper is as follows. In Section 2 we present the extension of the Chvátal-Gomory procedure to MCPs. Section 3 discusses hierarchies of relaxations motivated by the ones proposed by Balas, Ceria and Cornuéjols [6]; Sherali and Adams [40]; Lovász and Schrijver [31]; and Lasserre [25] (see also [26]). In Section 4 we define cut generation and cut lifting; and Section 5 discusses the results of our computational experiments. Section 6 includes some concluding remarks.

2 Gomory cuts

In this section, we extend the Chvátal-Gomory procedure for generating cuts for linear integer programs to MCPs. Since all the results in this section are valid for $x_i \in \mathbf{Z}_+$, $i = 1, \ldots, p$, where $\mathbf{Z}_+$ denotes the set of non-negative integers, we consider pure integer conic programs, i.e. $p = n$. Extensions to MCPs will be clear and are left to the reader to construct.

The extension discussed in this section is based on the following equivalence

$$\Big\{\mathbf{x} \in \mathbf{R}^n : \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}\Big\} \Leftrightarrow \Big\{\mathbf{x} \in \mathbf{R}^n : (\mathbf{A}^T\mathbf{u})^T\mathbf{x} \geq \mathbf{u}^T\mathbf{b}, \forall \mathbf{u} \succeq_{\mathcal{K}^*} \mathbf{0}\Big\}, \tag{2}$$

where $\mathcal{K}^* = \big\{\mathbf{u} : \mathbf{u}^T\mathbf{y} \geq 0, \forall \mathbf{y} \in \mathcal{K}\big\}$ denotes the dual cone of the cone $\mathcal{K}$. Since each of the cones $\mathcal{K}_l$, $\mathcal{K}_q$ and $\mathcal{K}_s$ are self-dual, it follows the $\mathcal{K} = \mathcal{K}^*$. Let $\mathcal{C}^\circ$ denote the feasible set of a pure integer conic program, i.e. $\mathcal{C}^\circ = \big\{\mathbf{x} \in \mathbf{Z}^n_+ : \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}\big\}$. Consider the following extension of the three step Chvátal-Gomory (C-G) procedure:

(i) Choose $\mathbf{u} \succeq_{\mathcal{K}^*} \mathbf{0}$. Then $(\mathbf{A}^T\mathbf{u})^T\mathbf{x} = \sum_{j=1}^n (\mathbf{a}_j^T\mathbf{u})x_j \geq \mathbf{u}^T\mathbf{b}$, where $\mathbf{a}_j$, $j = 1, \ldots, n$, denotes the j-th column of $\mathbf{A}$.

(ii) Since $\mathbf{x} \geq \mathbf{0}$, we have $\sum_{j=1}^n \lceil \mathbf{a}_j^T\mathbf{u}\rceil x_j \geq \mathbf{u}^T\mathbf{b}$.

(iii) Since $\mathbf{x} \in \mathbf{Z}^n_+$, it follows that $\sum_{j=1}^n \lceil \mathbf{a}_j^T\mathbf{u}\rceil x_j \geq \lceil \mathbf{u}^T\mathbf{b}\rceil$. Hence, this valid linear inequality can be added to the conic constraints $\mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}$.

Below we illustrate this C-G procedure on **Example 1**.

Example 1 (contd): Set $\mathbf{u} = (\sqrt{3}, -1, -1, -1, 0, 0, 0, 0)^T$. Since $\|[-1, -1, -1]\| \leq \sqrt{3}$, it is clear that $\mathbf{u} \succeq_{\mathcal{K}^*} \mathbf{0}$. The Gomory cut corresponding to $\mathbf{u}$ is given by

$$\mathbf{u}^T\mathbf{A}\mathbf{x} = -x_1 - x_2 - x_3 \geq \lceil -\sqrt{3}\rceil \quad \Rightarrow \quad x_1 + x_2 + x_3 \leq \lfloor \sqrt{3} \rfloor = 1.$$

Choosing $\mathbf{u} = (\sqrt{3}, -1, -1, 1, 0, 0, 0, 0)^T$ yields the Gomory cut $x_1 + x_2 - x_3 \leq 1$. ∎

The C-G procedure when applied to mixed 0-1 SDPs has some interesting implications for comparing SDP and LP relaxations for combinatorial optimization problems. We illustrate this using the traveling salesman problem (TSP) and the maxcut problem. Cvetković et al [15] (see also [18, 13]) proposed the following mixed 0-1 formulation for the TSP.

$$\begin{array}{lrcl} \text{minimize} & & \mathbf{C} \bullet \mathbf{X}, & \\ \text{subject to} & \mathbf{X1} = \mathbf{X}^T\mathbf{1} & = & \mathbf{1}, \\ & \mathbf{diag}(\mathbf{X}) & = & \mathbf{0}, \\ & h_1\mathbf{I} + \left(\frac{2-h_1}{n}\right)\mathbf{1}\mathbf{1}^T - (\mathbf{X} + \mathbf{X}^T) & \succeq & 0, \\ & X_{ij} & \in & \{0, 1\}, \quad i, j = 1, \ldots, n. \end{array} \tag{3}$$

where $\mathbf{A} \bullet \mathbf{B} = \mathbf{Tr}(\mathbf{AB})$, for $\mathbf{A}, \mathbf{B} \in \mathbf{R}^{n \times n}$, $\mathbf{C} \in \mathbf{R}^{n \times n}$ is the distance matrix, $\mathbf{diag}(\mathbf{X}) = (X_{11}, \ldots, X_{nn})^T$ denote the main diagonal of the matrix $\mathbf{X}$, $\mathbf{1}$ is the vector of all ones, and $h_1 = 2\cos\left(\frac{2\pi}{n}\right)$. The subtour elimination inequalities for the TSP (see [27] for details) are given by

$$\sum_{i,j \in W} X_{ij} \leq |W| - 1, \quad \forall\, W \subset \{1, \ldots, n\}, \quad 3 \leq |W| \leq n - 1,$$

Lemma 1 *The sub-tour elimination inequalities for the TSP are C-G inequalities with respect to the semidefinite constraint in (3).*

Proof: Fix $W \subseteq \{1, \ldots, n\}$. Choose $\mathbf{Y} = \frac{1}{2}\mathbf{1}_W\mathbf{1}_W^T \succeq \mathbf{0}$ where $\mathbf{1}_W$ is the indicator vector for the indices in W. Since the semidefinite cone is self-dual, $\mathbf{Y} \succeq_{\mathcal{K}^*} \mathbf{0}$. On applying the C-G procedure with $\mathbf{Y}$, we get

$$\mathbf{Y} \bullet (\mathbf{X} + \mathbf{X}^T) = \sum_{i,j \in W} X_{ij} \leq \left\lfloor \mathbf{Y} \bullet \left(h_1\mathbf{I} + \left(\frac{2 - h_1}{n}\right)\mathbf{1}\mathbf{1}^T\right)\right\rfloor. \tag{4}$$

Since (4) is a valid inequality for the TSP, all that we have to show is that the constant term in (4) is at most $|W| - 1$. Since $\mathbf{Y} = \frac{1}{2}\mathbf{1}_W\mathbf{1}_W^T$ we have

$$\begin{aligned} \mathbf{Y} \bullet \left(h_1\mathbf{I} + \left(\frac{2 - h_1}{n}\right)\mathbf{1}\mathbf{1}^T\right) &= \frac{h_1}{2}|W| + \frac{2 - h_1}{2n}|W|^2, \\ &= |W|\left(\left(1 - \frac{h_1}{2}\right)\frac{|W|}{n} + \frac{h_1}{2}\right), \\ &< |W|, \end{aligned} \tag{5}$$

where (5) follows from $(1 - h_1/2) = 1 - \cos(2\pi/n) > 0$ and $|W| < n$. Taking the floor of the constant term we establish the claim. ∎

Lemma 1 asserts that one semidefinite constraint closely approximates all the exponentially many subtour elimination inequalities. Thus, if one allows for a small approximation error, one is able to represent all the exponentially many sub-tour elimination inequalities by a single $(n \times n)$ semidefinite constraint, which from the perspective of computational complexity, is equivalent to n inequalities.

The semidefinite formulation of the maximum cut problem is given by

$$\begin{array}{ll} \text{maximize} & \frac{1}{4}\mathbf{L} \bullet \mathbf{X}, \\ \text{subject to} & \mathbf{diag}(\mathbf{X}) = \mathbf{1}, \\ & \mathbf{X} \succeq \mathbf{0}, \\ & X_{ij} \in \{-1, +1\}, \end{array} \tag{6}$$

where $\mathbf{L}$ is the Laplacian matrix of weights [19]. The following result relates the semidefinite formulation to linear relaxations.

Lemma 2 *All the triangle inequalities $-X_{pq}+X_{pr}+X_{qr} \leq 1$, $p \neq q \neq r$, are C-G cuts with respect to semidefinite constraint in (6).*

Proof: Fix $i \neq j \neq k$. Then the corresponding triangle inequality is the C-G cut generated by the matrix $\mathbf{Y}$ where

$$Y_{ij} = \begin{cases} 0.5, & (i,j) = (p,q), \\ -0.5, & (i,j) \in \{(p,r),(r,p),(q,r),(r,q)\} \\ 0.5, & (i,j) \in \{(p,p),(q,q),(r,r)\}, \\ 0, & \text{otherwise.} \end{cases}$$

The eigenvalues of $\mathbf{Y}$ are $\{0,0,3\}$, thus $\mathbf{Y} \succeq \mathbf{0}$. ■

This result provides some theoretical support for the folk knowledge that triangular inequalities do not improve the SDP relaxation of the maxcut [3, 24].

Just as in the case of integer LPs, the C-G procedure can be applied iteratively to the generated inequalities and/or the original conic constraints to generate a hierarchy of C-G cuts. The following lemma extends Chvátal's result to sets defined by conic constraints.

Lemma 3 *Suppose $\mathcal{C}^\circ = \big\{\mathbf{x} \in \mathbf{Z}^n_+ : \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}\big\}$ and bounded. Then every valid inequality for* $\mathbf{conv}\,(\mathcal{C}^\circ)$ *can be obtained by repeating the C-G procedure a finite number of times.*

Proof: Let $\mathcal{C} = \big\{\mathbf{x} \in \mathbf{R}^n_+ : \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}\big\}$ denote the continuous relaxation and let

$$\widehat{\mathcal{C}} = \Big\{\mathbf{x} \in \mathbf{R}^n_+ : \sum_{j=1}^{n} \lceil \mathbf{a}_j^T \mathbf{u} \rceil x_j \geq \lceil \mathbf{b}^T \mathbf{u} \rceil, \forall \mathbf{u} \succeq_{\mathcal{K}^*} \mathbf{0} \Big\}$$

denote the convex set obtained by generating all possible Gomory cuts from the conic constraint $\mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}$. Note that the conic constraints, $\mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}$, are *not* present in the definition of $\widehat{\mathcal{C}}$; or equivalently, the constraints of the form $\sum_{j=1}^n (\mathbf{a}_j^T\mathbf{u})x_j \geq \mathbf{b}^T\mathbf{u}$, $\mathbf{u} \succeq_{\mathcal{K}^*} \mathbf{0}$, are *not* present in $\widehat{\mathcal{C}}$.

First we establish that $\mathbf{Z}^n_+ \cap \widehat{\mathcal{C}} = \mathcal{C}^\circ$. (Since $\widehat{\mathcal{C}}$ does not contain inequalities of the form $\sum_{j=1}^n (\mathbf{a}_j^T\mathbf{u})x_j \geq \mathbf{b}^T\mathbf{u}$, $\mathbf{u} \succeq_{\mathcal{K}^*} \mathbf{0}$, this equality is not immediately obvious.) Suppose this is not the case, i.e. there exists $\widehat{\mathbf{x}} \in \mathbf{Z}^n_+$ such that $\widehat{\mathbf{x}} \in \widehat{\mathcal{C}}$ but $\widehat{\mathbf{x}} \notin \mathcal{C}$. Let $d = \min_{\mathbf{x} \in \mathcal{C}} \big\|\widehat{\mathbf{x}} - \mathbf{x}\big\|$. Since $\mathcal{C}$ is closed, $d > 0$. From duality for minimum norm problems it follows that there exists $\widehat{\mathbf{u}} \succeq_{\mathcal{K}^*} \mathbf{0}$ such that

$$d = \frac{\mathbf{b}^T\widehat{\mathbf{u}} - \sum_{j=1}^n (\mathbf{a}_j^T\widehat{\mathbf{u}})\widehat{x}_j}{\|\mathbf{A}^T\mathbf{u}\|}, \tag{7}$$

where $\big\|\mathbf{A}^T\mathbf{u}\big\| = \sqrt{\sum_{j=1}^n (\mathbf{a}_j^T\mathbf{u})^2}$. Since $\widehat{\mathbf{x}} \in \widehat{\mathcal{C}}$, we have

$$\sum_{j=1}^{n} \lceil \mathbf{a}_j^T \widehat{\mathbf{u}} \rceil \widehat{x}_j \geq \lceil \mathbf{b}^T \widehat{\mathbf{u}} \rceil \tag{8}$$

Since $k\widehat{\mathbf{u}} \succeq_{\mathcal{K}^*} 0$ for all $k \geq 0$, (8) implies that

$$\begin{aligned} \lceil k\mathbf{b}^T\widehat{\mathbf{u}} \rceil - \sum_{j=1}^{n} \lceil k\mathbf{a}_j^T\widehat{\mathbf{u}} \rceil \widehat{x}_j &= k\Big(\mathbf{b}^T\widehat{\mathbf{u}} - \sum_{j=1}^{n} \mathbf{a}_j^T\widehat{\mathbf{u}}\widehat{x}_j\Big) + \Big((k\mathbf{b}^T\widehat{\mathbf{u}})_f - \sum_{j=1}^{n} (k\mathbf{a}_j^T\widehat{\mathbf{u}})_f \widehat{x}_j\Big), \\ &\geq kd\,\big\|\mathbf{A}^T\mathbf{u}\big\| - n\widehat{x}_{max}, \end{aligned}$$

where $0 \leq (a)_f = a - \lfloor a \rfloor \leq 1$ and $\widehat{x}_{max} = \max_i\{x_i\}$. Thus, by choosing a large enough k, we get

$$\lceil k\mathbf{b}^T\widehat{\mathbf{u}}\rceil - \sum_{j=1}^{n}\lceil k\mathbf{a}_j^T\widehat{\mathbf{u}}\rceil\widehat{x}_j > 0.$$

A contradiction. Thus, $\mathbf{Z}_+^n \cap \widehat{\mathcal{C}} = \mathcal{C}^\circ$.

Since $\mathbf{Z}_+^n \cap \widehat{\mathcal{C}} = \mathcal{C}^\circ$ and the form of the set $\widehat{\mathcal{C}}$ is the same as one that would be obtained after one round of Gomory cuts on an integer linear program, the result can be established using the techniques developed for the integer linear programs (see Section II.1.2 in [36] for details) ■

From the above analysis it is clear that any technique for generating fractional Gomory cuts for mixed integer linear programs can be suitably extended to MCPs. The extension is left to the reader.

3 Cuts from hierarchies of tighter relaxations for $\mathcal{C}^\circ$

In this section, we construct hierarchies of relaxations $\tilde{\mathcal{C}}$ for $\mathcal{C}^\circ$ that are tighter than the continuous relaxation $\mathcal{C} = \{\mathbf{x} : \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}\}$. Next, we use conic duality to construct the set of valid linear inequalities for the tighter relaxation $\tilde{\mathcal{C}}$ and show how to interpret these inequalities as linear and convex quadratic valid inequalities for $\mathcal{C}^\circ$. The tighter relaxations are generated by suitably extending the sequential convexification approaches proposed by Balas, Ceria and Cornuéjols [6]; Sherali and Adams [40, 41]; Lovász and Schrijver [31]; and Lasserre [25] (see also [26]). While some of these hierarchies were proposed for pure 0-1 LPs, they can all be extended to mixed 0-1 LPs.

3.1 Lovász-Schrijver and Balas-Ceria-Cornuéjols hierarchy

In this section we discuss relaxations that are motivated by the hierarchies of relaxations introduced by Lovász and Schrijver [31] and Balas, Ceria and Cornuéjols [6].

Fix a subset $B \subseteq P = \{1, \ldots, p\}$ with size $|B| = l$. By permuting the variables, if necessary, we assume that $B = \{1, \ldots, l\}$. Let $\mathbf{x} \in \mathcal{C}^\circ$ and let $\mathbf{Y}^0 = [\mathbf{y}_1^0, \ldots, \mathbf{y}_l^0]$, $\mathbf{Y}^1 = [\mathbf{y}_1^1, \ldots, \mathbf{y}_l^1] \in \mathbf{R}^{(n+1)\times l}$ where

$$\mathbf{y}_k^0 = (1 - x_{j_k})\begin{bmatrix}1\\ \mathbf{x}\end{bmatrix}, \quad \mathbf{y}_k^1 = x_{j_k}\begin{bmatrix}1\\ \mathbf{x}\end{bmatrix}, \quad k = 1, \ldots, l.$$

Then it is clear that $(\mathbf{x}, \mathbf{Y}_0, \mathbf{Y}_1)$ belong to the sets $\mathcal{M}_B(\mathcal{C})$ and $\mathcal{M}_B^+(\mathcal{C})$ defined as follows.

$$\mathcal{M}_B(\mathcal{C}) = \left\{ (\mathbf{x}, \mathbf{Y}^0, \mathbf{Y}^1) : \begin{array}{rcll} \mathbf{y}_k^0 + \mathbf{y}_k^1 & = & \begin{bmatrix}1\\ \mathbf{x}\end{bmatrix}, & k = 1, \ldots, l, \\ \sum_{i=1}^n y_{ik}^0 \mathbf{a}_i & \succeq_{\mathcal{K}} & y_{0k}^0 \mathbf{b}, & k = 1, \ldots, l, \\ \sum_{i=1}^n y_{ik}^1 \mathbf{a}_i & \succeq_{\mathcal{K}} & y_{0k}^1 \mathbf{b}, & k = 1, \ldots, l, \\ y_{kk}^0 & = & 0, & k = 1, \ldots, l, \\ y_{kk}^1 & = & y_{0k}^1, & k = 1, \ldots, l, \\ \mathbf{Y}_B^1 & = & (\mathbf{Y}_B^1)^T & \\ \mathbf{Y}^0, \mathbf{Y}^1 & \in & \mathbf{R}^{(n+1)\times l} & \end{array} \right\}, \tag{9}$$

and

$$\mathcal{M}_B^+(\mathcal{C}) = \Big\{ (\mathbf{x}, \mathbf{Y}^0, \mathbf{Y}^1) : (\mathbf{x}, \mathbf{Y}^0, \mathbf{Y}^1) \in \mathcal{M}_B(\mathcal{C}), \mathbf{Y}_B^1 \succeq \mathbf{x}_B\mathbf{x}_B^T \Big\}, \tag{10}$$

where $\mathbf{x}_B = [x_j]_{j \in B}$ denotes the subvector and $\mathbf{Y}_B^1 = [y_{jk}^1]_{j,k \in B}$ denotes the submatrix corresponding to indices in $B = \{1, \ldots, l\}$. Since the linear inequalities $0 \le x_j \le 1$, $j = 1, \ldots, p$, are assumed to be present in the conic constraint $\mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}$, it follows that $y_{0k}^0, y_{0k}^1 \ge 0$ for all $k = 1, \ldots, l$. Tighter relaxations for $\mathcal{C}^\circ$ are obtained by projecting $\mathcal{M}_B(\mathcal{C})$ and $\mathcal{M}_B^+(\mathcal{C})$ onto the $\mathbf{x}$-variables as follows.

$$\mathcal{N}_B(\mathcal{C}) = \Big\{ \mathbf{x} : \exists \mathbf{Y}^0, \mathbf{Y}^1 \text{ s.t. } (\mathbf{x}, \mathbf{Y}^0, \mathbf{Y}^1) \in \mathcal{M}_B(\mathcal{C}) \Big\}, \tag{11}$$

$$\mathcal{N}_B^+(\mathcal{C}) = \Big\{ \mathbf{x} : \exists \mathbf{Y}^0, \mathbf{Y}^1 \text{ s.t. } (\mathbf{x}, \mathbf{Y}^0, \mathbf{Y}^1) \in \mathcal{M}_B^+(\mathcal{C}) \Big\}. \tag{12}$$

Balas, Ceria and Cornuéjols [6] discuss the special case $|B| = 1$ in the context of mixed 0-1 LPs and Lovász and Schrijver [31] consider the case $|B| = p = n$ in the context of pure integer LPs. The relaxations $\mathcal{N}_B(\mathcal{C})$ and $\mathcal{N}_B^+(\mathcal{C})$ are described by $\mathcal{O}(n\,|B|)$ variables and $\mathcal{O}(|B|)$ m-dimensional conic constraints, i.e. the number of variables and constraints grow *linearly* with the $|B|$. Below we illustrate this "lifting" technique on **Example 1**.

Example 1 (contd): Set $B = \{1\}$. The constraints satisfied by $(\mathbf{x}, \mathbf{y}^0, \mathbf{y}^1) \in \mathcal{M}_B(\mathcal{C})$ are as follows:

$$\mathbf{y}^0 + \mathbf{y}^1 = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$$

and

$$y_1^0 = 0, \tag{13}$$

$$y_1^1 = y_0^1. \tag{14}$$

The second-order cone constraint $\|\mathbf{x}\| \le 1$ implies the following constraints on $\mathbf{y}^i$, $i = 0, 1$:

$$x_1(1 - \|\mathbf{x}\|) \ge 0 \;\Rightarrow\; y_0^1 \ge \left\| [y_1^1,\ y_2^1,\ y_3^1] \right\|, \tag{15}$$

$$(1 - x_1)(1 - \|\mathbf{x}\|) \ge 0 \;\Rightarrow\; y_0^0 \ge \left\| [y_1^0,\ y_2^0,\ y_3^0] \right\|. \tag{16}$$

On multiplying the inequalities $0 \le x_j \le 1$, $j = 1, 2$, by x_1 and $(1 - x_1)$; and linearizing, we get

$$0 \le y_j^0 \le y_0^0, \quad j = 1, 2, \tag{17}$$

$$0 \le y_j^1 \le y_0^1, \quad j = 1, 2, \tag{18}$$

Note that $y_0^i \ge 0$, $i = 0, 1$. The constraints (16) and (14) imply that $y_2^1 = y_3^1 = 0$. Thus, $x_1 = y_1^1 = y_0^1$, $x_2 = y_2^0$, $x_3 = y_3^0$, and the constraint (16) implies that

$$\left\| [y_1^0,\ y_2^0,\ y_3^0] \right\| = \sqrt{x_2^2 + x_3^2} \le y_0^0 = 1 - y_0^1 = 1 - x_1.$$

Thus, the relaxation $\mathcal{N}_B(\mathcal{C})$ is given by
$\mathcal{N}_B(\mathcal{C}) = \big\{ \mathbf{x} \in \mathbf{R}^n : 0 \le x_j \le 1, j = 1, 2, \sqrt{x_2^2 + x_3^2} \;+\; x_1 \;\le\; 1 \big\}$. ■

Theorem 1

(i) For all $B \subseteq \{1, \ldots, p\}$, $\mathcal{C}^\circ \subseteq \mathcal{N}_B^+(\mathcal{C}) \subseteq \mathcal{N}_B(\mathcal{C}) \subseteq \cap_{(j \in B)} \mathcal{C}_j$, where $\mathcal{C}_j = \mathbf{conv}\,(\mathcal{C} \cap \{\mathbf{x} : x_j \in \{0, 1\}\})$, $j = 1, \ldots, p$.

(ii) *Let* $(\mathcal{N}_P)^0(\mathcal{C}) = (\mathcal{N}_P^+)^0(\mathcal{C}) = \mathcal{C}$*), and* $(\mathcal{N}_P)^k(\mathcal{C}) = \mathcal{N}_P\big((\mathcal{N}_P)^{k-1}(\mathcal{C})\big)$*,* $(\mathcal{N}_P^+)^k(\mathcal{C}) = \mathcal{N}_P^+\big((\mathcal{N}_P^+)^{k-1}(\mathcal{C})\big)$*, for* $k \geq 1$*. Then* $(\mathcal{N}_P)^p(\mathcal{C}) = (\mathcal{N}_P^+)^p(\mathcal{C}) = \mathbf{conv}\,(\mathcal{C}^\circ)$*.*

(iii) *Let* $\mathcal{N}_j(\mathcal{C})$ *denote the projection corresponding to* $B = \{j\} \subset P$*. Then* $\mathcal{N}_j(\mathcal{C}) = \mathcal{C}_j$*, for all* $j = 1, \ldots, p$*. Let* $(\mathcal{N})^0(\mathcal{C}) = \mathcal{C}$ *and* $(\mathcal{N})^k(\mathcal{C}) = \mathcal{N}_k\big((\mathcal{N})^{k-1}(\mathcal{C})\big)$*. Then* $(\mathcal{N})^k(\mathcal{C}) = \mathbf{conv}\,(\mathcal{C} \cap \{\mathbf{x} : x_j \in \{0,1\}, j = 1, \ldots, k\})$*,* $k = 1, \ldots, p$*.*

Theorem 1 follows from simple extensions of the results in [31] and [6]. Vanderbei and Benson [10] show that the conic constraints

$$\mathbf{Ax} \succeq_{\mathcal{K}} \mathbf{b} \quad \Leftrightarrow \quad g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, r,$$

for appropriately chosen smooth convex functions $g_i(\mathbf{x})$, $i = 1, \ldots, r$. Therefore, Theorem 1 also follows from the results in [42]. Since the recession cone of a conic program is well defined, we do not need to impose the constraint qualification $\max_{\mathbf{x} \in \mathcal{C}} \max_{1 \leq i \leq n} |x_i| \leq L$ imposed in [42]. Next, we characterize a restricted class of linear inequalities for the "lifted set" $\mathcal{M}_B^+$.

Theorem 2 *Suppose* $\mathbf{int}(\mathbf{conv}\,(\mathcal{C}^\circ)) \neq \emptyset$*. Fix* $B = \{1, \ldots, l\} \subseteq P$*, and let* $Y_B^1 = [y_{jk}^1]_{j,k \in B}$*. Then* $\mathbf{Q} \bullet \mathbf{Y}_B^1 + \boldsymbol{\alpha}^T \mathbf{x} \geq \beta$*,* $\mathbf{Q} = \mathbf{Q}^T \in \mathbf{R}^{|B| \times |B|}$*, for all* $(\mathbf{x}, \mathbf{Y}^0, \mathbf{Y}^1) \in \mathcal{M}_B^+(\mathcal{C})$ *if, and only if, there exist* $\mathbf{u}^k \in \mathbf{R}^{n+1}$*,* $\mathbf{v}^{0k}, \mathbf{v}^{1k} \succeq_{\mathcal{K}^*} \mathbf{0}$*,* $k = 1, \ldots, l$*,* $\boldsymbol{\gamma}^0, \boldsymbol{\gamma}^1 \in \mathbf{R}^l$*, and a positive semidefinite matrix* $\bar{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & \mathbf{w} \\ \mathbf{w}^T & w_0 \end{bmatrix} \in \mathbf{R}^{(l+1) \times (l+1)}$ *satisfying*

$$\begin{array}{rrrcll} \alpha_j & + \sum_{k=1}^l u_j^k & - 2w_j & = & 0, & j \in B, \\ \alpha_j & + \sum_{k=1}^l u_j^k & & = & 0, & j \notin B, \\ \beta & - \sum_{k=1}^l u_0^k & + w_0 & = & 0, & \end{array} \tag{19}$$

and

$$\begin{array}{rrrrrcll} u_0^k & & & - \mathbf{b}^T \mathbf{v}^{0k} & & = & 0, & k \in B, \\ u_0^k & - \gamma_k^1 & & - \mathbf{b}^T \mathbf{v}^{1k} & & = & 0, & k \in B, \\ u_j^k & + \gamma_j^0 \delta_{jk} & & + \mathbf{a}_j^T \mathbf{v}^{0k} & & = & 0, & j = 1, \ldots, n, k \in B, \\ u_j^k + u_k^j & + \gamma_k^1 \delta_{jk} & + \mathbf{a}_k^T \mathbf{v}^{1j} & + \mathbf{a}_j^T \mathbf{v}^{1k} & -2W_{jk} & = & 2Q_{jk}, & j, k \in B, \\ u_j^k & & & \mathbf{a}_j^T \mathbf{v}^{1k} & & = & 0, & j \notin B, k \in B, \end{array} \tag{20}$$

where $\mathbf{a}_i$*,* $i = 1, \ldots, n$ *are the columns of* $\mathbf{A}$*, and* $\delta_{jk} = 1$ *when* $j = k$ *and* 0 *otherwise.*

This result follows from a straightforward application of conic duality. In order to obtain the set of linear inequalities valid for $\mathcal{M}_B(\mathcal{C})$ set $\bar{\mathbf{W}} = \mathbf{0}$. The condition $\mathbf{int}(\mathbf{conv}\,(\mathcal{C}^\circ)) \neq \emptyset$ is a mild constraint qualification and holds for most problems. Moreover, since we only need the constraint qualification to ensure strong duality, it can be replaced by any other condition that ensures the same. Since $\mathcal{N}_B^+$ has $\mathcal{O}(|B|)$ conic constraints, the dual program as $\mathcal{O}(|B|)$ variables that are constrained to lie in the dual cone.

One can generate linear and convex quadratic valid inequalities for the relaxations $\mathcal{N}_B^+(\mathcal{C})$ by suitably restricting the choices for $\mathbf{Q}$ and $\boldsymbol{\alpha}$.

Lemma 4 *Fix* $B = \{1, \ldots, l\} \subseteq P = \{1, \ldots, p\}$ *and suppose* $\mathbf{int}(\mathbf{conv}\,(\mathcal{C}^\circ)) \neq \emptyset$*.*

(i) *The inequality* $\boldsymbol{\alpha}^T \mathbf{x} \geq \beta$ *is valid for* $\mathcal{N}_B^+(\mathcal{C})$ *if, and only if, there exists* $(\mathbf{0}, \boldsymbol{\alpha}, \beta)$ *satisfy (19) and (20).*

(ii) The inequality $\boldsymbol{\alpha}^T\mathbf{x} \geq \beta$ is valid for $\mathcal{N}_{\{l\}}(\mathcal{C})$, $l \in P$, if, and only if, there exist γ^0,$\gamma^1 \in \mathbf{R}$, and $\mathbf{v}^0$, $\mathbf{v}^1 \succeq_{\mathcal{K}^*} \mathbf{0}$, such that

$$\begin{array}{rrrrcl} \boldsymbol{\alpha} & -\gamma^0\mathbf{e}_l & -\mathbf{A}^T\mathbf{v}^0 & & = & 0, \\ \boldsymbol{\alpha} & -\gamma^1\mathbf{e}_l & -\mathbf{A}^T\mathbf{v}^1 & & = & 0, \\ & & \mathbf{b}^T\mathbf{v}^0 & -\beta & = & 0, \\ & \gamma^1 & +\mathbf{b}^T\mathbf{v}^1 & -\beta & = & 0, \end{array} \tag{21}$$

where e_l denotes the l-th standard basis vector in $\mathbf{R}^n$.

(iii) The convex quadratic inequality $\mathbf{x}_B^T\mathbf{Q}\mathbf{x}_B + \boldsymbol{\alpha}^T\mathbf{x} \geq \beta$, $-\mathbf{Q} \succeq \mathbf{0}$, is valid for $\mathcal{N}_B^+(\mathcal{C})$ if $(\mathbf{Q}, \boldsymbol{\alpha}, \beta)$ satisfy (19) and (20).

Proof: Parts (i) and (ii) immediately follow from Theorem 2. Part (iii) follows from the observation that $\mathbf{Y}_B^1 \succeq \mathbf{x}_B\mathbf{x}_B^T$ and $-\mathbf{Q} \succeq \mathbf{0}$ imply that $\mathbf{Q} \bullet \mathbf{Y}_B \leq \mathbf{x}_B^T\mathbf{Q}\mathbf{x}_B$. ■

No compact representation is known for the set of all valid inequalities of the set obtained by partially convexifying the feasible set of a mixed 0-1 convex program. This makes the cut generation proposed in [42] hard to use in practice. For MCPs, Theorem 2 establishes that the set of valid inequalities of the set $\mathcal{N}_{\mathcal{B}}^+(\mathcal{C})$ is itself described by conic constraints. Consequently, cut selection reduces to solving a conic program. For a more detailed discussion of convex quadratic cuts for MCPs see [14]. Stubbs and Mehrotra [43] investigate convex quadratic inequalities in the context of mixed 0-1 convex programs.

3.2 Sherali-Adams and Lasserre hierarchies

In this section we present two sets of tighter relaxations for $\mathcal{C}^\circ$ that are motivated by the hierarchies of relaxations introduced by Sherali and Adams [40, 41] and Laserre [25] (see also [26]).

Let $\mathbf{x} \in \mathcal{C}^\circ$, and, without loss of generality, fix $B = \{1, \dots, l\} \subseteq P$. Let $\mathbf{y} \in \mathbf{R}^{(n-l+1)2^l}$ denote a vector indexed by the empty set $\emptyset$, subsets $H \subseteq B$, and sets of the form $H \cup \{j\}$, $H \subseteq B$, $j \notin B$. Define $\mathbf{y}$ as follows:

$$y_I = \begin{cases} 1, & I = \emptyset, \\ \prod_{j \in I} x_j, & I = H \subseteq B, \text{ or } I = H \cup \{j\}, H \subseteq B, j \notin B. \end{cases} \tag{22}$$

For all subsets $I \subseteq B$, define

$$z_0^I = \prod_{j \in I} x_j \prod_{j \in B \setminus I} (1 - x_j) = \sum_{I \subseteq H \subseteq B} (-1)^{|B \setminus H|} y_H \geq 0, \tag{23}$$

and $\mathbf{z}^I \in \mathbf{R}^n$ as follows,

$$z_k^I = x_k \prod_{j \in I} x_j \prod_{j \in B \setminus I} (1 - x_j) = \sum_{I \subseteq H \subseteq B} (-1)^{|B \setminus H|} y_{H \cup k}, \quad k = 1, \dots, n. \tag{24}$$

Then multiplying the conic constraint $\mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}$ by the product $\prod_{j \in I} x_j \prod_{j \in B \setminus I}(1 - x_j)$ we get

$$\mathbf{A}\mathbf{z}^I - \mathbf{b}z_0^I \succeq_{\mathcal{K}} \mathbf{0}, \quad I \subseteq B.$$

Thus, it follows that the feasible set $\mathcal{C}^\circ$ of the MCP (1) is contained in the relaxation

$$\mathcal{R}_B(\mathcal{C}) = \left\{ \mathbf{x} : x_j = y_{\{j\}},\ j = 1, \ldots, n, \quad \mathbf{A}\mathbf{z}^I - \mathbf{b} z_0^I \succeq_{\mathcal{K}} \mathbf{0},\ I \subseteq B \right\} \tag{25}$$

where $(z_0^I, \mathbf{z}^I)$, $I \subseteq B$, are defined in (23) and (24). In contrast to the relaxation $\mathcal{N}_B(\mathcal{C})$ introduced in Section 3.1, the relaxation $\mathcal{R}_B(\mathcal{C})$ has $\mathcal{O}(n2^{|B|})$ variables and $\mathcal{O}(2^{|B|})$ conic constraints, i.e. the number of variables and constraints grow *exponentially* as a function of $|B|$.

Let $\mathbf{M}_B(\mathbf{y}) \in \mathbf{R}^{2^l \times 2^l}$ denote a matrix with entries indexed by subsets of B. Define

$$\mathbf{M}_B(\mathbf{y})_{I,J} = y_{I \cup J}, \quad I, J \subseteq B.$$

Then (22) implies that $\mathbf{M}(\mathbf{y}) = \mathbf{y}\mathbf{y}^T \succeq \mathbf{0}$. Thus,

$$\mathcal{R}_B^+(\mathcal{C}) = \left\{ \mathbf{x} : x_j = y_{\{j\}},\ j = 1, \ldots, n, \quad \mathbf{A}\mathbf{z}^I - \mathbf{b} z_0^I \succeq_{\mathcal{K}} \mathbf{0},\ I \subseteq B, \quad \mathbf{M}(\mathbf{y}) \succeq \mathbf{0} \right\}$$

is also a relaxation for $\mathcal{C}^\circ$. We obtain a hierarchy of relaxations as a function of the size of the subset B of the binary indices. Below we illustrate this technique on **Example 1**.

Example 1 (contd): Set $B = \{1, 2\} = P$. Then the corresponding vector $\mathbf{y}$ has the following components

$$\mathbf{y} = \left[y_\emptyset,\ y_1,\ y_2,\ y_3,\ y_{12},\ y_{13},\ y_{23},\ y_{123} \right]$$

The second-order cone constraint $\|\mathbf{x}\| \leq 1$ implies the following constraints on $\mathbf{y}$:

$$x_1 x_2 (1 - \|\mathbf{x}\|) \geq 0 \quad \Rightarrow \quad y_{12} \geq \|[y_{12},\ y_{12},\ y_{123}]\| \quad \Rightarrow \quad y_{12} = y_{123} = 0,$$

$$\begin{aligned}
x_1(1 - x_2)(1 - \|\mathbf{x}\|) \geq 0 \quad &\Rightarrow \quad y_2 - y_{12} \geq \|[y_1 - y_{12},\ 0,\ y_{23} - y_{123}]\| \quad \Rightarrow \quad y_{23} = 0 \\
(1 - x_1)x_2(1 - \|\mathbf{x}\|) \geq 0 \quad &\Rightarrow \quad y_1 - y_{12} \geq \|[0,\ y_2 - y_{12},\ y_{13} - y_{123}]\| \quad \Rightarrow \quad y_{13} = 0 \\
(1 - x_1)(1 - x_2)(1 - \|\mathbf{x}\|) \geq 0 \quad &\Rightarrow \quad 1 - y_1 - y_2 + y_{12} \geq \|[0,\ 0,\ y_3 - y_{13} - y_{23} - y_{123}]\|, \\
&\Rightarrow \quad y_1 + y_2 + |y_3| \leq 1.
\end{aligned}$$

On multiplying the inequalities $0 \leq x_j \leq 1$, $j = 1, 2$, by the pairs $x_1 x_2$, $(1 - x_1)x_2$, $x_1(1 - x_2)$, and $(1 - x_1)(1 - x_2)$; and linearizing, we get

$$y_{12} \geq 0, \quad y_1 \geq y_{12}, \quad y_2 \geq y_{12}, \quad 1 + y_{12} \geq y_1 + y_2.$$

The above constraints and the fact that $x_j = y_j$, $j = 1, 2, 3$ imply that

$$\mathcal{R}_B(\mathcal{C}) = \{\mathbf{x} : x_j \geq 0,\ j = 1, 2,\ x_1 + x_2 + |x_3| \leq 1\} = \mathbf{conv}\left(\mathcal{C}^\circ\right).$$

Theorem 3 establishes it is always the case that $\mathcal{R}_P(\mathcal{C}) = \mathbf{conv}\left(\mathcal{C}^\circ\right)$. ■

A second hierarchy is obtained by setting $B = P$ but limiting the number of terms in the product in (23). Concretely, fix $\mathbf{x} \in \mathcal{C}^\circ$. Let $\mathbf{y}$ denote a vector indexed by the empty set $\emptyset$, subsets $H \subseteq B$ with $|H| \leq l$, and subsets of the form $H \cup \{j\}$, $H \subseteq B$, $j \notin B$, with $|H| \leq l$. For subsets $I, J \subseteq P$, such that $I \cap J = \emptyset$ and $|I \cup J| \leq l$, define

$$z_0^{I,J} = \prod_{j \in I} x_j \prod_{j \in J} (1 - x_j) = \sum_{I \subseteq H \subseteq I \cup J} (-1)^{|(I \cup J) \setminus H|} y_H \geq 0,$$

and $\mathbf{z}^I \in \mathbf{R}^n$ as follows,

$$z_k^{I,J} = x_k \prod_{j\in I} x_j \prod_{j\in J}(1-x_j) = \sum_{I\subseteq H\subseteq I\cup J} (-1)^{|(I\cup J)\setminus H|} y_{H\cup k}, \quad k=1,\ldots,n.$$

Then it follows that $\mathcal{C}^\circ$ is contained in the relaxation

$$\mathcal{R}_l(\mathcal{C}) = \left\{ \mathbf{x} : x_j = y_{\{j\}},\ j=1,\ldots,n, \mathbf{A}\mathbf{z}^{I,J} - \mathbf{b}z_0^{I,J} \succeq_{\mathcal{K}} \mathbf{0},\ I,J\subseteq P, I\cap J=\emptyset, |I\cup J|\leq l. \right\}.$$

Let $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ denote the k-th binomial coefficient. Then $\mathcal{R}_B(\mathcal{C})$ has $\mathcal{O}(n2^l\binom{p}{l})$ variables and $\mathcal{O}(2^l\binom{p}{l})$ conic constraints, i.e. the number of variables and constraints are *exponential* in l. However, as we will see in Theorem 3 the relaxations $\mathcal{R}_B(\mathcal{C})$ and $\mathcal{R}_l(\mathcal{C})$ are significantly tighter than $\mathcal{N}_B(\mathcal{C})$.

Let $\nu = \lfloor \frac{l}{2} \rfloor$. Let $\mathbf{M}_l(\mathbf{y}) \in \mathbf{R}^{2^\nu\times 2^\nu}$ denote a matrix with entries indexed by subsets $I\subseteq P$ with $|I|\leq\nu$. Define

$$\mathbf{M}_t(\mathbf{y})_{I,J} = y_{I\cup J}, \quad I,J\subseteq P, |I|\leq \nu, |J|\leq\nu.$$

Then (22) implies that $\mathbf{M}_l(\mathbf{y}) = \mathbf{y}_\nu\mathbf{y}_\nu^T \succeq \mathbf{0}$, where $\mathbf{y}_\nu$ denotes the subvector corresponding to subsets $I\subseteq P$, $|I|\leq\nu$. Thus, one can add the constraint $\mathbf{M}_l(\mathbf{y})\succeq\mathbf{0}$ to obtain the following relaxation for $\mathcal{C}^\circ$

$$\mathcal{R}_l^+(\mathcal{C}) = \left\{ \mathbf{x} : x_j = y_{\{j\}},\ j=1,\ldots,n, \quad \mathbf{A}\mathbf{z}^I - \mathbf{b}z_0^I \succeq_{\mathcal{K}} \mathbf{0},\ I\subseteq P, |I|\leq l, \quad \mathbf{M}_t(\mathbf{y})\succeq\mathbf{0} \right\}.$$

The following Theorem is a simple extension of the results in [41]. It also follows from the results in [25] (see also [26]).

Theorem 3

(i) For all $B\subseteq P$ the relaxation $\mathcal{R}_B^+(\mathcal{C}) = \mathcal{R}_B(\mathcal{C}) = \mathbf{conv}\,(\mathcal{C}\cap\{\mathbf{x} : x_j\in\{0,1\}, j\in B\})$. Thus, $\mathcal{R}_P^+(\mathcal{C}) = \mathcal{R}_P(\mathcal{C}) = \mathbf{conv}\,(\mathcal{C}^\circ)$

(ii) Define $\mathcal{R}_0^+(\mathcal{C}) = \mathcal{R}_0(\mathcal{C}) = \mathcal{C}$. Then $\mathcal{R}_l(\mathcal{C})\subseteq\mathcal{R}_{l-1}(\mathcal{C})$ and $\mathcal{R}_l^+(\mathcal{C})\subseteq\mathcal{R}_{l-1}^+(\mathcal{C})$, for all $1\leq l\leq p$. Moreover, $\mathcal{R}_p(\mathcal{C}) = \mathcal{R}_p^+(\mathcal{C}) = \mathbf{conv}\,\left(\mathcal{C}^0\right)$.

Part (i) shows that the additional semidefinite constraint $\mathbf{M}_B(\mathbf{Y})\succeq\mathbf{0}$ does not improve the relaxation. Next, we characterize the set valid inequalities for the relaxation $\mathcal{R}_B(\mathcal{C})$. We first establish the following intermediate result.

Theorem 4 *Fix $B\subseteq P=\{1,\ldots,p\}$ and suppose $\mathbf{int}(\mathbf{conv}\,(\mathcal{C}^\circ))\neq\emptyset$. Then $\boldsymbol{\alpha}^T\mathbf{y}\geq 0$ is a valid inequality for the $\{\mathbf{y} : \mathbf{A}\mathbf{z}^I\succeq\mathbf{b}z_0^I,\ I\subseteq B\}$, $(z_0^I,\mathbf{z}^I)$, $I\subseteq B$, given by (23)-(24), if, and only if, there exist vectors $\mathbf{u}^I\in\mathbf{R}^m$, $\mathbf{u}^I\succeq_{\mathcal{K}^*} 0$, $I\subseteq B$, such that*

$$\begin{array}{rcll} \alpha_\emptyset &=& (-1)^{|B|+1}\mathbf{b}^T\mathbf{u}^\emptyset, & \\ \alpha_H &=& (-1)^{|B\setminus H|+1}\sum_{I\subseteq H}\mathbf{b}^T\mathbf{u}^I & \\ && +\,(-1)^{|B\setminus H|}\sum_{k\in H}\sum_{I\subseteq H}\mathbf{a}_k^T\mathbf{u}^I, & H\subseteq B, \\ \alpha_{H\cup\{j\}} &=& (-1)^{|B\setminus H|}\sum_{I\subseteq H}\mathbf{a}_j^T\mathbf{u}^I, & H\subseteq B, j\notin B. \end{array} \tag{26}$$

This result follows from a straightforward application of conic duality. The dual set is characterized by $\mathcal{O}(m2^{|B|})$ variables and $\mathcal{O}(2^{|B|})$ conic constraints. Constructing the set of valid inequalities for the relaxation $\mathcal{R}_l(\mathcal{C})$ is left to the reader.

By suitably restricting the components of $\boldsymbol{\alpha}$, one can generate linear and convex quadratic cuts for the relaxation $\mathcal{R}_B(\mathcal{C})$.

1. $\mathcal{C}^{(1)} \leftarrow \mathcal{C} = \{\mathbf{x} : \mathbf{A}\mathbf{x} \succeq_{\mathcal{K}} \mathbf{b}\}$ and $t \leftarrow 1$
2. $\mathbf{x}^{(t)} \leftarrow \text{argmin}\left\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in \mathcal{C}^{(t)}\right\}$ and $(p^{(t)}, d^{(t)}) \leftarrow$ primal/dual value
   (a) "Round" $\mathbf{x}^{(t)}$ to obtain a feasible solution $\widehat{\mathbf{x}}^{(t)}$
   (b) If $\mathbf{c}^T\widehat{\mathbf{x}}^{(t)} \leq (1+\epsilon)d^{(t)}$, stop.
3. Generate a concave inequality $f(\mathbf{x}) \geq 0$ valid for $\mathcal{C}^{\circ}$ such that $f(\mathbf{x}^{(t)}) < 0$.
4. $\mathcal{C}^{(t+1)} \leftarrow \left\{\mathbf{x} : f(\mathbf{x}) \geq 0\right\} \cap \mathcal{C}^{(t)}$, and $t \leftarrow t+1$. Go to Step 2.

Figure 1: Cut Algorithm

Lemma 5 *Let $B = \{1, \ldots, l\} \subseteq P = \{1, \ldots, p\}$ and suppose $\mathbf{int}(\mathbf{conv}(\mathcal{C}^{\circ})) \neq \emptyset$.*

(i) The inequality $\boldsymbol{\alpha}^T\mathbf{x} \geq \beta$ is valid for $\mathcal{R}_B(\mathcal{C})$ if, and only if, there exists $\bar{\boldsymbol{\alpha}} \in \mathbf{R}^{2^l(n-l+1)}$ such that $\bar{\alpha}_{\emptyset} = -\beta$, $\bar{\alpha}_{\{j\}} = \alpha_j$, $j = 1, \ldots, n$, and $\bar{\boldsymbol{\alpha}}$ satisfies (26).

(ii) The convex quadratic inequality $\mathbf{x}_B^T\mathbf{Q}\mathbf{x}_B + \boldsymbol{\alpha}^T\mathbf{x} \geq \beta$, $-\mathbf{Q} \succeq \mathbf{0}$, is valid for $\mathcal{R}_B(\mathcal{C})$ if there exists $\bar{\boldsymbol{\alpha}} \in \mathbf{R}^{2^l(n-l+1)}$ such that $\bar{\alpha}_{\emptyset} = -\beta$, $\bar{\alpha}_{\{j\}} = \alpha_j$, $j = 1, \ldots, n$, $\bar{\alpha}_{\{i,j\}} = Q_{ij}$, $i, j \in B$, all other components $\bar{\alpha}_H = 0$, and $\bar{\boldsymbol{\alpha}}$ satisfies (26).

Part (i) follows from Theorem 4 and part (ii) is established by an argument identical to the one used to establish Lemma 4 part (iii).

We conclude this section by showing that an MCP is equivalent to a mixed 0-1 polynomial program. A vector $\mathbf{x} = (x_0, \bar{\mathbf{x}}) \in \mathcal{K}_q$ if, and only if, $x_0 \geq 0$ and $x_0^2 - \|\mathbf{x}\|^2 \geq 0$. Similarly a matrix $\mathbf{X} \succeq \mathbf{0}$ if, and only if, $\mathbf{det}(\mathbf{X}_l) \geq 0$, $l = 1, \ldots, n$, where $\mathbf{X}_l = [x_{ij}]_{i,j \in \{1,\ldots,l\}}$, $l = 1, \ldots, n$. Thus, the MCP (1) can be reformulated as follows.

$$\begin{array}{rll} \text{minimize} & \mathbf{c}^T\mathbf{x}, & \\ \text{subject to} & g_i(\mathbf{x}) \geq 0, & i = 1, \ldots, r, \\ & x_j \in \{0, 1\}, & j = 1, \ldots, p, \end{array} \qquad (27)$$

where $g_i(\mathbf{x})$, $i = 1, \ldots, r$, are polynomials in $\mathbf{x}$ with the property that the degree of the binary variables x_j, $j = 1, \ldots, p$, in each $g_i(\mathbf{x})$, $i = 1, \ldots, r$, is at most 1. This reformulation allows one to use the techniques developed in [25] (see also [26]) to construct a hierarchy of relaxations and the corresponding set of valid inequalities. The size of these relaxations are also exponential in the "level" of the hierarchy.

4 Cut algorithm

The general outline of the cut algorithm is shown in Figure 1. Although the cut algorithm appears to be very similar to the cut algorithm for mixed 0-1 LPs, there are several unresolved computational issues. To begin with, the conic program in Step 2 of the cut algorithm is solved using an interior point method, and thus, $\mathbf{x}^{(t)}$ is unlikely to be feasible for the MCP – making the rounding (step 2(a)) necessary. Rounding for general MCPs is hard; therefore, the cut algorithm is likely to yield lower bounds and not feasible solutions.

The crucial step in the algorithm is cut generation (step 3). The valid concave inequality $f(\mathbf{x}) \geq 0$ is generated using either the C-G procedure (see Section 2) or from a tighter relaxation (see Section 3). Since the iterate $\mathbf{x}^{(t)}$ is likely to be an interior point, it is not clear how to generate a Gomory cut. Mitchell [32] discusses techniques for generating Gomory cuts from an interior point in the context of mixed 0-1 LPs. Extensions of these techniques may lead to a method for generating Gomory cuts for MCPs. The recently proposed simplex algorithm for conic programming [20] may also lead to a method for generating Gomory cuts. There exists a convex cut separating a fractional iterate $\mathbf{x}^{(t)}$ from $\mathcal{C}^\circ$ only if $\mathbf{x}^{(t)}$ is an extreme point; consequently, it would appear that the conic program has to be solved to optimality. However, Mitchell [33] and Mitchell and Borchers [34] note that interior-point methods for solving mixed integer LPs perform better when cuts are generated from a sub-optimal (therefore, not extreme) point. The techniques in [33, 34] can also be employed in the context of MCPs.

There is no analog of the dual simplex method for restarting the solution of a general MCP over the restricted set obtained by adding a cut. For the specific case of the maxcut and related quadratic optimization problems, Helmberg and Rendl [23] and Mitchell [35] have proposed restart strategies that employ variants of the dual-scaling interior point algorithms. The simplex method proposed in [20] may provide an efficient restart strategy for general MCPs. The conic programs in all our computational experiments were solved using SeDuMi [44]. SeDuMi does not allow one to specify an initial dual feasible point, and therefore, we had to solve each new conic program from scratch. Thus, the run times reported in Section 5 are not representative – our results can only be seen as evidence (or lack thereof) of the "quality" of conic cuts.

Since conic programming relaxations typically have uncountably many extreme points, the cut algorithm is not likely to terminate finitely, therefore branching is unavoidable. Although Mitchell [35] have proposed restart strategies for the specific case of maxcut and related problems, good warm start strategies for general MCPs are not known. As noted above, in our computational experiments we do not employ any restart strategies.

4.1 Cut generation

In this section we detail strategies for generating a valid inequality $f(\mathbf{x}) \geq 0$ that is violated by the current iterate $\mathbf{x}^{(t)}$ (see step 3 in Figure 1).

Unlike in mixed 0-1 LPs, there is no known systematic methods for generating Gomory cuts from a fractional iterate $\mathbf{x}^{(t)}$. For specific combinatorial optimization problems, such as the TSP and maxcut, a subset of Gomory cuts are well known families of cuts and can be generated easily. See Section 2 for details.

Let $F = \{i : 0 \leq i \leq p, 0 < x_i^{(t)} < 1\}$ denote the set of fractional elements of the current iterate $\mathbf{x}^{(t)}$ and suppose $\mathbf{x}^{(t)}$ is an extreme point of the current relaxation $\mathcal{C}^{(t)}$. Then for all $B \cap F \neq \emptyset$ the iterate $\mathbf{x}^{(t)} \notin \mathcal{N}_B(\mathcal{C}^{(t)}) \supseteq \mathcal{N}_B^+(\mathcal{C}^{(t)})$ and $\mathbf{x}^{(t)} \notin \mathcal{R}_B(\mathcal{C}^{(t)})$. For all $\nu \geq 1$, the iterate $\mathbf{x}^{(t)} \notin \mathcal{R}_\nu(\mathcal{C}^{(t)}) \supseteq \mathcal{R}_\nu^+(\mathcal{C}^{(t)})$. Let $\mathcal{N}$ denote any of these relaxations. Then, minimum norm duality implies that

$$\min_{\{\mathbf{x} \in \mathcal{N}\}} \|\mathbf{x} - \mathbf{x}^{(t)}\| = \begin{array}{ll} \text{maximize} & \beta - \boldsymbol{\alpha}^T \mathbf{x}^{(t)}, \\ \text{subject to} & (\boldsymbol{\alpha}, \beta) \in \mathcal{N}^*, \\ & \|\boldsymbol{\alpha}\|_* \leq 1, \end{array} \tag{28}$$

where $\|\cdot\|_*$ is the dual norm of the norm $\|\cdot\|$ and $\mathcal{N}^*$ denotes the polar set of $\mathcal{N}$, i.e $(\boldsymbol{\alpha}, \beta) \in \mathcal{N}^*$ iff $\boldsymbol{\alpha}^T \mathbf{x} \geq \beta$, for all $\mathbf{x} \in \mathcal{N}$. Thus, the "deepest" linear cut in the $\|\cdot\|$-norm is given by the solution of (28). The optimization problem (28) can be solved efficiently provided $\mathcal{N}^*$ can be described by conic constraints and $\|\boldsymbol{\alpha}\|_* \leq 1$ is itself a conic constraint. Theorem 4 and Theorem 2 respectively imply that $\mathcal{R}_B(\mathcal{C}^{(t)})^*$ and $\mathcal{M}_B^+(\mathcal{C}^{(t)})^*$ (consequently, $\mathcal{N}_B(\mathcal{C}^{(t)})^*$) are given by conic constraints. The dual-norm constraint $\|\boldsymbol{\alpha}\|_* \leq 1$ corresponding to $\mathcal{L}_1$ and $\mathcal{L}_\infty$ norms is linear; and is second-order cone representable when $\|\cdot\|$ is the $\mathcal{L}_p$ norm, $p \geq 1$, rational (see Section 6.2.3 of [37] for details). Thus, for all $\mathcal{L}_p$-norms, (28) is a conic program.

The "deepest" quadratic cuts [14] are generated by solving the conic program

$$\begin{array}{rl} \text{maximize} & \beta - \boldsymbol{\alpha}^T \mathbf{x}^{(t)} - \mathbf{x}_B^T \mathbf{Q} \mathbf{x}_B, \\ \text{subject to} & (\mathbf{Q}, \boldsymbol{\alpha}, \beta) \text{ satisfy } (19) - (20), \\ & \mathbf{Q} \preceq \mathbf{0}, \quad -\mathbf{Tr}\,\mathbf{Q} + \|\boldsymbol{\alpha}\|_2 \leq 1, \end{array}$$

where $\|\boldsymbol{\alpha}\|_2 = \sqrt{\boldsymbol{\alpha}^T \boldsymbol{\alpha}}$ is the usual Euclidean norm.

4.2 Cut lifting

For large problems the complexity of the cut generation can be quite high. A computationally efficient alternative is to first project the feasible set onto the space of fractional variables, generate the cut in this lower dimensional space and then "lift" the cut. Although projection does compromise the quality of the cut, the reduction in complexity more often than not offsets this loss. Cut lifting is also important in a branch-and-cut setting since it allows one to "lift" a cut generated at a specific node of the branch-and-cut tree to one that is valid for all the nodes.

We illustrate the cut lifting using the relaxation $\mathcal{N}_k(\mathcal{C}^{(t)})$, $k \in F$, i.e. the analog of the Balas, Ceria, Cornuéjols lift-and-project set. Let $F = \{j \in \{1, \ldots, p\} : 0 < x_j^{(t)} < 1\} \cup \{j \geq p+1\}$. Since one can complement the variables $x_j^{(t)} = 1$ by redefining

$$\mathbf{b} \leftarrow \mathbf{b} - \mathbf{a}_j, \quad \mathbf{a}_j \leftarrow -\mathbf{a}_j,$$

we will, without loss of generality, assume that $x_j^{(t)} = 0$ for all $j \notin F$. Let $\mathbf{A}_F$ denote the sub-matrix of $\mathbf{A}$ obtained by removing the columns corresponding to $j \notin F$ and the rows corresponding to the constraints $0 \leq x_j \leq 1$, $j \notin F$, and let $\mathbf{b}_F$ denote the sub-vector obtained by removing the components corresponding to the constraints $0 \leq x_j \leq 1$, $j \notin F$ – note that we are only removing linear constraints. Let $R = \{i : i\text{-th row of } \mathbf{A} \text{ included in } \mathbf{A}_F\}$. Let F_j denote the index of the element $j \in F$, s_i (resp. t_i) denote the row index of the constraint $x_j \geq 0$ (resp. $x_j \leq 1$), $j \notin F$, and r_i, $i \in R$, denote the index of the i-th row of $\mathbf{A}$ in $\mathbf{A}_F$. Part (ii) of Lemma 4 implies that the restricted cut generation program is given by

$$\begin{array}{rlllll} \text{maximize} & \beta - \boldsymbol{\alpha}_F^T \mathbf{x}^{(t)}, & & & & \\ \text{subject to} & \boldsymbol{\alpha}_F & -\gamma^0 \mathbf{e}_k & -\mathbf{A}_F^T \mathbf{v}_F^0 & & = 0, \\ & \boldsymbol{\alpha}_F & -\gamma^1 \mathbf{e}_k & -\mathbf{A}_F^T \mathbf{v}_F^1 & & = 0, \\ & & & \mathbf{b}_F^T \mathbf{v}_F^0 & -\beta & = 0, \\ & & \gamma^1 & +\mathbf{b}_F^T \mathbf{v}_F^1 & -\beta & = 0, \\ & \|\boldsymbol{\alpha}_F\|_* \leq 1, & & \mathbf{v}_F^0,\ \mathbf{v}_F^1 \succeq_{\mathcal{K}^*} 0, & & \end{array} \tag{29}$$

where $\mathbf{e}_k$ denotes the k-th standard basis vector in $\mathbf{R}^n$. Let $(\bar{\boldsymbol{\alpha}}_F, \bar{\beta}, \bar{\gamma}^0, \bar{\gamma}^1, \bar{\mathbf{v}}_F^0, \bar{\mathbf{v}}_F^1)$ denote the optimal solution of (29). Define $\bar{\boldsymbol{\alpha}} \in \mathbf{R}^n$, $\bar{\mathbf{v}}^0, \bar{\mathbf{v}}^1 \in \mathbf{R}^m$ as follows:

$$\bar{\alpha}_j = \begin{cases} \bar{\alpha}_F(F_j), & j \in F, \\ \max\{\mathbf{A}_F^T \mathbf{v}_F^0, \mathbf{A}_F^T \mathbf{v}_F^1\}, & j \notin F, \end{cases} \tag{30}$$

where $\bar{\alpha}_F(F_j)$ denotes the F_j-th element of $\bar{\boldsymbol{\alpha}}_F$,

$$\bar{v}_i^0 = \begin{cases} \bar{v}_F^0(r_i), & i \in R, \\ (\alpha_j - \mathbf{A}_F^T \mathbf{v}_F^0)^+, & i = s_j, j \notin F, \\ 0, & i = t_j, j \notin F, \end{cases} \tag{31}$$

$$\bar{v}_i^1 = \begin{cases} \bar{v}_F^1(r_i), & i \in R, \\ (\alpha_j - \mathbf{A}_F^T \mathbf{v}_F^1)^+, & i = s_j, j \notin F, \\ 0, & i = t_j, j \notin F, \end{cases} \tag{32}$$

Since only the rows corresponding to the linear constraints, $0 \leq x_i \leq 1$, $i \notin F$, are missing in $(\mathbf{A}_F, \mathbf{b}_F)$, it follows that $\bar{\mathbf{v}}^0$, $\bar{\mathbf{v}}^1 \succeq_{\mathcal{K}^*} \mathbf{0}$. $(\bar{\boldsymbol{\alpha}}, \bar{\beta}, \bar{\gamma}^0, \bar{\gamma}^1, \bar{\mathbf{v}}^0, \bar{\mathbf{v}}^1)$ is an extreme point of the dual set defined by (21), i.e. $\bar{\boldsymbol{\alpha}}^T\mathbf{x} \geq \beta$ is a valid inequality for the set $\mathcal{N}_k(\mathcal{C}^{(t)})$. However, $\|\bar{\boldsymbol{\alpha}}\|_* \neq 1$, i.e. the cut $\bar{\boldsymbol{\alpha}}^T\mathbf{x} \geq \beta$ is not the "deepest" cut, and therefore, its performance is likely to be worse than the cut generated in the full space.

5 Computational results

In this section, we discuss the results of our preliminary computational experiments with the cut algorithm displayed in Figure 1. We restricted ourselves to the special case of mixed 0-1 SDPs. Since the relaxations in Section 3.2 involve an exponential number of conic constraints and variables, we restricted ourselves to the relaxations $\mathcal{N}_B(\mathcal{C}^{(t)})$ introduced in Section 3.1. In particular, we tested linear cuts generated from $\mathcal{N}_k(\mathcal{C}^{(t)})$, $k \in F$. Since $\mathcal{N}_k(\mathcal{C}^{(t)})$ is the analog of the Balas, Ceria and Cornuéjols lift-and-project set, we will refer to the variable k as the *disjunction* variable and a cut generated from $\mathcal{N}_k(\mathcal{C}^{(t)})$ as the *disjunctive* cut [5].

Our primary objective in these computational experiments was to investigate the "quality" of the cuts where "quality" was interpreted as the decrease in integrality gap per cut. We considered mixed 0-1 SDPs resulting from the pure 0-1 SDP formulation of maxcut [19], the pure 0-1 SDP formulation of the traveling salesman problem [15, 13] and Lovász-Schrijver lifting of mixed 0-1 LPs. Since SDPs are inherently harder to solve and there are no general purpose warm-start strategies for SDPs, it is unlikely that the 0-1 SDP formulation will at present be the method of choice for solving these problems. In light of this, our work should be interpreted as an attempt to identify instances where such a formulation might be of interest when (and if) SDP solvers become competitive.

All the conic programs were solved using SeDuMi [44] within MATLAB R11 on a Dell Precision 340. All the computations were carried out using compiled MATLAB codes. Since SeDuMi does not allow the user to specify a starting solution, all the conic programs resulting from adding cuts or branching were solved from scratch; therefore, the run times reported here are not representative of run times achievable by a conic cut algorithm (for more representative run times see [22]).

5.1 Computational results for the maxcut problem

The $\{+1, -1\}$-SDP formulation of the maxcut is given in (6). Using the change of variables $\mathbf{Y} = \frac{1}{2}(\mathbf{X} + \mathbf{1}\mathbf{1}^T)$, (6) can be reformulated as the equivalent 0-1 SDP

$$\begin{array}{ll} \text{maximize} & \frac{1}{2}\mathbf{L} \bullet \mathbf{Y}, \\ \text{subject to} & \mathbf{diag}(\mathbf{Y}) = \mathbf{1}, \\ & 2\mathbf{Y} - \mathbf{1}\mathbf{1}^T \succeq \mathbf{0}, \\ & Y_{ij} \in \{0, 1\}. \end{array} \tag{33}$$

We used the 0-1 formulation (33) in our computational experiments. The maxcut instances were randomly generated with the weights chosen independently and identically from a uniform distribution on $[1, 100]$. In every iteration we chose the "most" fractional variable, i.e. one closest to 0.5, as the disjunction variable and generated a single disjunctive cut.

The results are summarized in Table 1. The column labeled "size" lists the number of nodes in the underlying graph, the column labeled "gap" lists the integrality gap of the initial SDP relaxation, the column labeled "cuts" lists the number of disjunctive cuts needed to solve the instance to optimality, and column labeled "time" lists the time elapsed. All instances were solved to optimality using a modest number of cuts. This might be explained by the experimental observation that a single disjunctive cut was often enough to drive a fractional variable to one of the boundaries. It does help the cut algorithm that the initial integrality gap of the maxcut SDP is small. The number of cuts required to solve problems of same size vary considerably and appear to increase with the initial integrality gap. The rate of improvement in the objective function decreases as more cuts are generated and on termination only 40% of the cuts are active.

Comparing our results with interior point approaches that exploit structure [23, 22] suggests the following conclusions. For small maxcut problems, the number of disjunctive cuts required to solve a problem to optimality appear to be smaller than the corresponding number of triangle, clique and hypermetric inequalities. However, the cut generation conic program (28) is computationally expensive and this is reflected in an overall larger run time. It is likely that odd-cycle inequalities [7] are more efficient than the disjunctive cuts. Since the spectral bundle method [22] appears to be the most efficient for solving maxcut problems, combining disjunctive cuts with such an approach would be an interesting exercise.

5.2 Computational results for the traveling salesman problem

We solved the $\{0, 1\}$-SDP formulation for the TSP given in (3) [15] (see also [13]). In every iteration of the cut algorithm, we chose the "most" fractional variable, i.e. one closest to 0.5, as our disjunction variable and generated one disjunctive cut. The cut algorithm was run for a maximum of 50 iterations. Table 2 lists the CPU time for generating 50 cuts for small TSPLIB instances. From these run time results it is clear that this SDP-based method is not yet likely to competitive with the LP-based methods. In the rest of this section we comment on the "quality" of the disjunctive cuts.

Table 3 details the performance of the cuts. The first column lists the name of the TSP instance – the first eight are instances from the TSPLIB [39] and the rest are random. The random TSP instances were solved using CONCORDE [4]. The second column lists the integrality gap of the initial SDP relaxation. Columns 3 and 4 list the fraction of the gap closed after the addition of 25 and 50 cuts respectively. There was, on average, a 48.47% reduction in the integrality gap with the

size	gap (%)	cuts	time (sec)
5	2.81	36	5.40
	2.13	21	3.57
10	1.87	217	47.74
	0.81	164	34.44
20	0.63	261	75.69
	1.28	375	108.75
30	1.14	297	118.90
	2.14	334	136.94
40	2.39	450	616.50
	2.41	532	718.20
50	1.43	663	2705.34
	2.36	428	1746.52

Table 1: Computational results for the maxcut problem

		cpu time (sec)	
instance	**size**	25 cuts	50 cuts
burma14	14	48.6	101.3
ulysses16	16	92.8	192.9
gr17	17	85.1	185.2
ulysses22	22	291.2	622.7
gr24	24	511.6	1028.6
fri26	26	722.8	1517.2
bayg29	29	1571.0	3224.0
bays29	29	1438.0	2982.0

Table 2: CPU time for solving small instances of the TSP

first 25 cuts, whereas an additional 25 cuts decreased the gap by only, on average, an additional 9.02%. On random instances disjuntive cuts perform remarkably well – 50 cuts are adequate for solving most of the instances.

Next, we discuss the results of a branch-and-cut algorithm. The experiments were stopped after 50 nodes unless the optimal solution was encountered earlier. We employed a depth-first-search branching rule and branched on the fractional variable closest to 1. Once the variable was chosen, we always branched on $x_j = 1$ first. In the branch-and-cut approach we branched whenever the improvement in the objective on the addition of the disjunctive cut was less than 0.1%. At every node we also maintained a feasible tour by a simple rounding procedure (for details see [13]). Columns 5 and 6 in Table 3 list the relative gaps of the best tour encountered in the course of the pure branch-and-bound (B-B) and the branch-and-cut (B-C) approach respectively. Even with a limit of 50 nodes, the B-C procedure produces good results; thereby, affirming the quality of the disjunctive cuts. The average gap of the B-C solution is 4.3% for the TSPLIB instances and 1.14% for the random TSP instances. The performance of the B-C procedure is significantly superior to the B-B procedure – especially in case of random TSP instances. Since the number of branching nodes was restricted, the performance of both the B-B and B-C is poor when the gap of the initial SDP relaxation is large. This partially explains the poor performance on the TSPLIB instances.

instance	gap (%)	% closed		% closed	
		25 cuts	50 cuts	B-and-B	B-and-C
burma14	4.45	20.3	25.0	100.00	100.00
ulysses16	7.30	5.7	6.3	97.08	98.58
gr17	13.2	12.0	13.8	90.12	90.94
ulysses22	9.78	7.70	7.85	92.73	100.00
gr24	3.30	19.05	21.43	76.58	92.80
fri26	5.12	5.26	7.02	82.29	97.55
bayg29	3.54	12.50	13.89	95.47	95.47
bays29	3.56	12.50	13.89	82.98	90.40
Random $n = 16$	11.27	31.25	31.25	95.07	100.00
$n = 16$	3.11	50.00	100.00	92.23	100.00
$n = 16$	1.33	100.00	100.00	100.00	100.00
Random $n = 20$	6.84	38.46	46.15	93.16	94.22
$n = 20$	6.70	42.86	42.86	81.34	93.30
$n = 20$	6.77	44.45	55.56	93.23	94.74
Random $n = 25$	3.39	50.00	66.67	83.05	100.00
$n = 25$	4.24	60.00	80.00	79.05	100.00
$n = 25$	3.39	80.00	90.00	95.25	97.97
Random $n = 30$	0.69	100.00	100.00	97.24	100.00
$n = 30$	0.86	100.00	100.00	97.41	100.00
$n = 30$	0.77	50.00	100.00	96.93	100.00
Random $n = 35$	1.04	100.00	100.00	92.19	100.00
$n = 35$	0.51	100.00	100.00	96.46	100.00
$n = 35$	1.39	75.00	100.00	78.47	100.00

Table 3: Computational results for the TSP

The initial gap typically increased with problem size; and hence, required a search over a large number of branching nodes. As mentioned above, in the absence of general warm start techniques, we solved all the SDPs from scratch. This severely limited the number of B-C nodes and, in turn, the problem sizes. Warm start methods for SDP-based B-C approaches for maxcut and related quadratic problems have been proposed [23, 35]. In order for the SDP-based methods to be competitive, these approaches will have to be extended to a wider class of problems.

5.3 Semidefinite lift-and-project cuts for mixed 0-1 linear programs

In this section we compare the performance of disjunctive cuts generated from the semidefinite lifting [31] of the mixed 0-1 LPs with that of the lift-and-project cuts introduced by Balas, Ceria and Cornuéjols [6]. We will refer to disjunctive cuts generated from the semidefinite lifting as *semidefinite cuts* and those generated by the lift-and-project approach in [6] as *linear cuts*. Since semidefinite lifting provides a tighter relaxation for the feasible set of the mixed 0-1 LP, we hypothesized that semidefinite cuts would be "deeper" and, therefore, fewer cuts would be needed to achieve a certain gap. The results of our preliminary computational experiments appear to support this hypothesis.

Consider a generic mixed 0-1 linear program,

$$\begin{array}{rl} \text{minimize} & \mathbf{c}^T\mathbf{x}, \\ \text{subject to} & \mathbf{A}_0\mathbf{x} = \mathbf{b}_0, \\ & \mathbf{A}_1\mathbf{x} \leq \mathbf{b}_1, \\ & \mathbf{x} \geq 0, \\ & x_i \in \{0,1\}, \quad i = 1,\ldots,p, \end{array} \tag{34}$$

where $\mathbf{x} \in \mathbf{R}^n$. Let $\bar{\mathbf{x}}$ be the optimal solution LP obtained by relaxing the binary constraints.

Fix a subset $B \subseteq \{1,\ldots,p\}$. Then the semidefinite lifting $\mathcal{C}_B$ of $\mathcal{C}^\circ$ with respect to the variables in B is given by $(\mathbf{x} \in \mathbf{R}^n, \mathbf{Y} \in \mathbf{R}^{n\times|B|})$ that satisfy,

$$\begin{array}{rcll} \mathbf{A}_0\mathbf{x} & = & \mathbf{b}_0, & \\ \mathbf{A}_1\mathbf{y}_k - \mathbf{b}_1 x_k & \leq & \mathbf{0}, & k \in B, \\ -\mathbf{A}_1\mathbf{y}_k + \mathbf{A}_1\mathbf{x} + \mathbf{b}_1 x_k & \leq & \mathbf{b}_1, & k \in B, \\ \mathbf{Y} & \geq & \mathbf{0}, & \\ \mathbf{y}_k - \mathbf{x} & \leq & \mathbf{0}, & k \in B, \\ y_{ik} - x_k & \leq & 0, & i \leq p, \quad k \in B, \\ -y_{ik} + x_i + x_k & \leq & 1, & i \leq p, \quad k \in B, \\ \begin{bmatrix} \mathbf{Y}_{BB} & \mathbf{x}_B \\ \mathbf{x}_B^T & 1 \end{bmatrix} & \succeq & 0, & \end{array} \tag{35}$$

where $\mathbf{y}_k$ denotes the k-th column of the matrix $\mathbf{Y}$, $\mathbf{x}_B = [x_i]_{(i\in B)}$ and $\mathbf{Y}_{BB} = [y_{ij}]_{(i,j\in B)}$. (See [31] for the details of this semidefinite lifting.) Thus, we have an SDP relaxation for the mixed 0-1 LP (34). We use this SDP formulation to generate disjunctive cuts separating the current LP optimal $\bar{\mathbf{x}}$ from $\mathcal{C}^\circ$. This cut is added to the constraint set of (34) and the process is repeated.

The goal in this set of experiments was to compare the "quality" of semidefinite and linear cuts. Since the semidefinite cuts are likely to be more expensive we generated these cuts by first projecting (34) to the space of the fractional components of the current iterate and then constructing the SDP lifting of the projected LP. The set B was chosen to be the 10 most fractional variables. Thus, each cut generation SDP is very small – only two 11×11 matrix variables. We generated the "deepest" disjunctive cut for every $k \in B$. The cuts were lifted to the entire space by using a lifting procedure similar to one described in Section 4. The algorithm was iterated for a maximum of 50 cut generation steps unless an optimal solution was found earlier. In order to further reduce the complexity of semidefinite cut generation, we removed inactive cuts after every 10 iterations of the algorithm.

The linear cuts were generated using the normalizations $\|\alpha\|_1 \leq 1$ and $\|\alpha\|_2 \leq 1$. The first normalization results in an LP whereas the second normalization results in a second-order cone program. One cut was generated for every fractional element and we did not prune any inactive cuts. Also, the linear cuts were generated in the full space.

We attempted to make a fair comparison of the two classes of cuts by controlling the complexity of semidefinite cut generation and erring in the direction of "quality" instead efficiency in linear cut generation by keeping all the linear cuts. Since we did not optimize the run times in the base case (i.e. linear cuts), we have not reported times in our experimental results.

The first set of test problems consisted of pure 0-1 LPs. See Table 4 for a description of these problems. The LSB and LSC problems are capital budgeting problems taken from Lemke and

problem name	**# constraints**	**# variables**	**LP optimal**	**IP optimal**	**gap %**
LSB	28	35	521.05	550	5.26
LSC	12	44	56.61	73	22.45
PE4	10	20	-6155.33	-6120	0.57
PE5	10	28	-12462.10	-12400	0.50
PE6	5	39	-10672.34	-10618	0.51
PE7	5	50	-16612.82	-16537	0.45
Stein27	118	27	13	18	27.78
Stein45	331	45	22	30	26.67

Table 4: Description of pure 0-1 problems

problem name	**semidefinite cuts**			**linear cuts:** $\|\alpha\|_2 \leq 1$			**linear cuts:** $\|\alpha\|_1 \leq 1$		
	cuts	**iter**	**% closed**	**cuts**	**iter**	**% closed**	**cuts**	**iter**	**% closed**
LSB	120	32	100	182	35	100	477	45	100
LSC	520	50	94	627	50	90	767	50	89
PE4	104	25	100	175	31	100	265	42	100
PE5	250	40	100	305	45	100	348	50	98
PE6	342	42	100	442	50	97	540	50	89
PE7	470	50	100	576	50	91	688	50	78
Stein27	430	50	92	546	50	82	646	50	72
Stein45	467	50	84	620	50	73	890	50	57

Table 5: Performance of cutting plane algorithms

problem name	# constraints	# variables	LP optimal	IP optimal	gap %
CTN1	102	83	219.21	245.64	10.76
CTN2	153	124	371.84	410.35	9.39
CTN3	206	167	450.26	472.14	4.6
Danoint	664	56	62.64	65.67[a]	–

Table 6: Description of mixed 0-1 problems ([a]Best known solution)

Spielberg [28], the PE problems are capital budgeting problems taken from [38] and the Stein problems are from MIPLIB3.0 [11]. Table 5 contrasts the performance of the semidefinite cuts with linear cuts generated using the two different normalizations. In Table 5 the column labeled "cuts" lists the number of cuts generated by the cut algorithm, the column labeled "iter" lists the number of iterations and the column labeled "% closed" lists the fraction of the initial integrality gap closed by the cutting plane algorithm. From the computational results, it is clear that the performance of the semidefinite cuts is superior to that of linear cuts with either of the two normalizations. Also, the performance of linear cuts with $\mathcal{L}_2$-normalization is considerably better than linear cuts with $\mathcal{L}_1$-normalization. In fact, when the number of binary variables is small, the performance of linear cuts with $\mathcal{L}_2$-normalization is very close to that of semidefinite cuts. The performance of the semidefinite cuts does not appear to be affected by the number of binary variables but begins to degrade sharply as the number of constraints in the problem increases.

The second set of test problems were mixed 0-1 LPs. The first three problems were fixed charge capacitated transportation problems of the form,

$$\begin{array}{rll} \text{minimize} & \sum_{ij} c_{ij}x_{ij} + \sum_{ij} v_{ij}y_{ij}, & \\ \text{subject to} & \sum_j y_{ij} - \sum_j y_{ji} = b_i, & \text{for all } i, \\ & y_{ij} \leq u_{ij}x_{ij}, & \text{for all arcs } (i,j), \\ & \mathbf{y} \geq 0, & \\ & x_{ij} \in \{0,1\}, & \text{for all arcs } (i,j). \end{array}$$

The problem instances were randomly generated as follows. The size of each of two node sets in the problems CTN1, CTN2 and CTN3 were 10, 15 and 20, and densities 45%, 30% and 30% respectively. The fixed cost c_{ij} for each arc was a uniformly distributed integer in $[0, 20]$, the variable cost v_{ij} was a real number uniformly distributed on $[0, 2]$, the capacity u_{ij} was a uniformly distributed integer in $[1, 20]$, and the supplies b_i were uniformly distributed integer in $[-20, 20]$ with the constraint that $\sum_i b_i = 0$. The last problem Danoint is a fixed charge network design problem taken from MIPLIB3.0 [11]. See Table 6 for a description of the problems. All problems, except Danoint, were solved using CPLEX6.0. The value listed in the IP Optimal column for Danoint is the value of the best known integer solution. The CTN problems were solved using the cutting plane algorithm described above. The semidefinite cut generation was modified for solving the Danoint problem – instead of lifting all of the initial linear constraints, we only lifted those that were active. This modification reduced the complexity of semidefinite cut generation without compromising the efficiency of the cuts significantly. The linear cut generation was not modified, i.e. we used all of the initial linear constraints in the cut generation program.

Table 7 summarizes the performance of the semidefinite cuts on these mixed integer programs. The pattern of performance is similar to that obtained for pure 0-1 programs. In general, employing

problem name	semidefinite cuts			linear cuts: $\|\alpha\|_2 \leq 1$			linear cuts: $\|\alpha\|_1 \leq 1$		
	cuts	**iter**	**% closed**	**cuts**	**iter**	**% closed**	**cuts**	**iter**	**% closed**
CTN1	212	24	100	254	28	100	540	32	100
CTN2	289	32	100	378	42	100	613	50	95
CTN3	264	36	100	476	45	100	765	50	96
Danoint	445	50	91	647	50	66	876	50	54

Table 7: Performance of cutting plane algorithms

semidefinite cuts reduces both the number of iterations and the number of cuts required. Linear cuts with $\mathcal{L}_2$-normalization perform better than linear cuts with $\mathcal{L}_1$-normalization. Although the performance of semidefinite cuts degrades as the number of linear constraints increases, they still perform significantly better than linear cuts.

The computational experiments, although preliminary, suggest the conclusion that the quality of semidefinite cuts is significantly superior to that of linear cuts. However, the complexity of semidefinite cut generation is a couple of orders of magnitude higher than that of linear cut generation. Since semidefinite cuts are not likely to be facet defining, we suspect that these cuts perform well because the convex hull of the integer feasible solutions is significantly smaller than the initial linear relaxation. We expect the performance of linear cuts to improve and, in fact, be better than semidefinite cuts when the integrality gap is sufficiently small and facet defining cutting planes become important. This suggests that a branch-and-cut algorithm should employ semidefinite cuts in the initial stages and then revert to linear cuts.

6 Conclusion

In this paper we show that many of the techniques developed for generating linear cuts for mixed 0-1 LPs, such as the Gomory cuts, the disjunctive cuts, and cuts from other hierarchies of tighter relaxations, extend in a straightforward manner to MCPs. We also show that simple extensions of these techniques lead to methods for generating convex quadratic cuts. The Gomory cuts for MCPs have interesting implications for comparing SDP and LP relaxations of combinatorial optimization problems, e.g. all the subtour elimination inequalities for the TSP can be obtained by applying one round of the C-G procedure (see Section 2) to the semidefinite constraint in (3).

The computational performance of these cuts needs to be further investigated. Our experiments were limited to the conic analog of disjunctive cuts proposed by Balas, Ceria and Cornuéjols [6]. The preliminary computational results suggest that the "quality" of the cuts obtained from conic formulations (in particular, SDP formulations) are superior to those generated from linear formulations. However, generating cuts from a conic formulation is considerably more expensive. For MCPs that admit an LP relaxation, it appears that the cuts from the conic formulation are attractive only when the LP relaxation of the problem has a large integrality gap and the problem has a small number of binary variables. However, for true MCPs, i.e. MCPs that do not have any LP relaxation, these cuts are the only recourse.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

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

[3] M. F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the max-cut problem. *Disc. Appl. Math.*, 119(1-2):79–106, 2002.

[4] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. On the solution of traveling salesman problems. *Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung*, ICM III:645–656, 1998.

[5] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

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

[7] F. Barahona and A. R. Mahjoub. On the cut polytope. *Math. Prog.*, 36:157–173, 1986.

[8] A. Ben-Tal, T. Margalit, and A. Nemirovski. Robust modeling of multi-stage portfolio problems. In *High performance optimization*, pages 303–328. Kluwer Acad. Publ., Dordrecht, 2000.

[9] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23(4):769–805, 1998.

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

[11] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P. Savelsberg. *A mixed integer programming library:* MIPLIB3.0. Available at http://www.caam.rice.edu/∼bixby/miplib/miplib.html.

[12] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory.* SIAM, Philadelphia, PA, 1994.

[13] M. T. Çezik. *Semidefinite methods for the traveling salesman and other combinatorial problems.* PhD thesis, IEOR Dept., Columbia University, 2000.

[14] M. T. Çezik and G. Iyengar. Quadratic cuts for mixed 0-1 quadratic programs. Technical Report TR-2001-02, CORC, IEOR Dept., Columbia University, 2001. Available at http://www.corc.ieor.columbia.edu/reports/techreports/tr-2001-02.ps.gz

[15] D. Cvetković, M. Čangalović, and V. Kovačević-Vujčić. Semidefinite programming methods for the symmetric traveling salesman problem. In *IPCO VII (Graz, 1999)*, pages 126–136. Springer, Berlin, 1999.

[16] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.*, 9(1):33–52, 1998.

[17] M. Goemans. Semidefinite programming in combinatorial optimization. *Math. Prog.*, 79:143–161, 1997.

[18] M. Goemans and F. Rendl. Combinatorial optimization. In *Handbook of Semidefinite Programming*. Kluwer, 2000.

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

[20] D. Goldfarb. The simplex method for conic programming. Technical Report TR-2002-05, CORC, IEOR Dept., Columbia University, 2002. Available at http://www.corc.ieor.columbia.edu/reports/techreports/tr-2002-05.ps

[21] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. *Math. Oper. Res.*, 28(1):1–38, 2003.

[22] C. Helmberg. A cutting plane algorithm for large scale semidefinite relaxations. In *The sharpest cut*, MPS/SIAM Ser. Optim., pages 233–256. SIAM, Philadelphia, PA, 2004.

[23] C. Helmberg and F. Rendl. Solving quadratic $(0, 1)$-problems by semidefinite programs and cutting planes. *Math. Prog.*, 82(3):291–315, 1998.

[24] H. Karloff. How good is the Goemans-Williamson maxcut algorithm. *SIAM J. Comp.*, 29(1):336–350, 1999.

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

[26] M. Laurent. A comparison of Sherali-Adams, Lovász-Schrijver and lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, 28(3):470–496, 2003.

[27] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The traveling salesman problem*. John Wiley & Sons Ltd., Chichester, 1990.

[28] C. Lemke and K. Spielberg. A capital budgeting heuristic algorithm using exchange operations. *AIEE Transactions*, 6:264–280, 1974.

[29] M. S. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs and bounds on risk. Under review in *Ann. Oper. Res.*. Available at http://www.stanford.edu/∼boyd/reports/portfolio.pdf.

[30] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra Appl.*, 284(1-3):193–228, 1998.

[31] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991.

[32] J. B. Mitchell. Fixing variables and generating classical cutting planes when using an interior point branch-and-cut method to solve integer programming problems. *Eur. J. Oper. Res.*, 97:139–148, 1997.

[33] J. B. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM J. Optim.*, 10(4):1212–1227, 2000.

[34] J. B. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In *High performance optimization*, pages 349–366. Kluwer Acad. Publ., Dordrecht, 2000.

[35] J. E. Mitchell. Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems. *J. Comb. Opt.*, 5(2):151–166, 2001.

[36] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience series in Discrete Mathematics. John Wiley & Sons, Inc., 1999.

[37] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, Philadelphia, 1993.

[38] C. Petersen. Computational experience with variants of the Balas algorithm applied to the selection of R&D projects. *Mgmt. Sc.*, 13:736–750, 1967.

[39] G. Reinelt. TSPLIB— A traveling salesman problem library. *ORSA J. Comp.*, 3(4):376–384, 1991.

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

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

[42] R. A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Math. Prog.*, 86:515–532, 1999.

[43] R. A. Stubbs and S. Mehrotra. Generating convex polynomial inequalities for mixed 0-1 programs. *J. Global Optim.*, 24(3):311–332, 2002.

[44] J. Stürm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[45] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 1996.