Journal of Global Optimization **24:** 311–332, 2002.

Generating Convex Polynomial Inequalities for Mixed 0–1 Programs ⋆

ROBERT A. STUBBS⋆⋆ and SANJAY MEHROTRA‡
Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208-3119, USA (e-mail: rstubbs@dashopt.com; mehrotra@iems.nwu.edu)

Abstract. We develop a method for generating valid convex quadratic inequalities for mixed 0–1 convex programs. We also show how these inequalities can be generated in the linear case by defining cut generation problems using a projection cone. The basic results for quadratic inequalities are extended to generate convex polynomial inequalities.

AMS Subject Classification: 90C10, 90C11, 90C25, 65K05, 90C25

Key words: Mixed integer programming, Convex programming, Semidefinite programming, Mixed integer nonlinear programming, Mixed integer convex programming

1. Introduction

We consider the following mixed 0–1 convex program (MICP):

$$\left.\begin{array}{ll} \text{minimize} & c \bullet x \\ \text{subject to} & g_i(x) \leqslant 0, \quad i = 1, \ldots, m, \\ & x_j \in \{0, 1\} \quad j = 1, \ldots, p, \\ & 0 \leqslant x_j \leqslant L \quad j = p+1, \ldots, n, \end{array}\right\} \tag{1}$$

where $c, x \in \mathbb{R}^n$, $n \geqslant p$, and $g_i(x)$ are convex functions. The notation $c \bullet x$ denotes the inner product of vector c with x, hence the objective function in (1) is linear. We assume that $g_i(x) \geqslant -\hat{L}$, for $i = 1, \ldots m$, for all x in the continuous relaxation of the feasible set of MICP. Here L and $\hat{L}$ are some positive constants.

We do not loose generality by considering a linear objective because a problem with a nonlinear convex objective, $g_0(x)$, can be written in the form of (1) by adding a variable x_0 and the constraint $g_0(x) - x_0 \leqslant 0$, and minimizing x_0. The problem in the above form is quite general, since by adding a non-negative multiple of $\sum_{i=1}^{n}(x_i^2 - x_i)$ to the objective and constraints a pure 0–1 nonlinear integer program can be reformulated as a MICP. The problem is considered in this

⋆ Supported in part by NSF-DMI-9500396, NSF-DMI-9908038 and ONR N00014-01-1-0048. An earlier draft under title, 'Generating Convex Quadratic Inequalities for Mixed 0–1 Programs.'

⋆⋆ (Present address: Dash Optimization, Inc., 501-F Johnson Ferry Road, Marietta, GA 30068, USA.)

‡ The author thanks his beloved Supreme Lord for all the revelations that have lead to this paper. All its fruits are surrendered to Him.

form by Stubbs and Mehrotra [15] so that cutting planes can be generated. The boundedness assumption ensures that a convex reformulation of the cut generation problem (discussed in subsequent sections) is possible.

For over a decade, branch-and-cut methods have proven to be an effective solution technique for solving many classes of mixed 0–1 linear programs. Recently, the first such method for solving mixed 0–1 convex programs was developed by Stubbs and Mehrotra [15]. This method can be viewed as an extension of the lift-and-project cutting plane method of Balas et al.[2, 3]. This method generates cuts by solving a projection problem over a disjunctive program that is written as a convex program in a higher dimensional space. The linear cuts are defined from supporting hyperplanes of the projection of this higher dimensional region onto the original space.

The purpose of this paper is to develop a method for generating convex polynomial cuts in the context of branch-and-cut algorithms. It is natural to study the possibility of adding nonlinear cuts in the context of (1) since nonlinear constraints are already present in the original problem, and one may expect that the projected region in Stubbs and Mehrotra [15] is 'more accurately' represented by nonlinear constraints in comparison with the region obtained by adding linear cuts. The nonlinear cuts might also prove valuable for linear problems since tighter relaxations are possible by adding a semidefinite constraint on variables in a higher dimensional relaxation (for theoretical and practical usefulness of semidefinite constraints see Lovász and Schrijver [10], Balas et al. [4] for the maximum clique problem, Goemans and Williamson [7] for the max-cut and max-2sat problems, and Karger et al. [9] for graph coloring problems). To the best of our knowledge, the method presented in this paper is the first known method for generating valid convex nonlinear inequalities for mixed 0–1 programs.

This paper is organized as follows. In Section 2, we review the relaxations of (1) studied in Stubbs and Mehrotra [15]. In Section 3, we develop our procedure for generating valid convex quadratic cuts. The projection problem considered for this purpose uses a differentiable convex function. In Section 4, we study the implication of our procedure for the linear case. In this section we give methods for finding quadratic and convex polynomial cuts using appropriately defined projection cones. In Section 5, we give sufficient conditions, under which projection problem considered for generating convex quadratic cuts can be defined by using non-differentiable convex functions. In Section 6 we extend our results of Section 3 to generate convex polynomial cuts for MICP.

1.1. NOTATION

A vector $v \in \mathbb{R}^n$ is taken to be a row vector, and its ith component is represented by $(v)_i$. This is not the same as v_i. For example, $(v_i)_j$ represents jth element of v_i. The only exception is the use of vector x, where x_i represents its ith component. The explicit use of '(.)' while representing the ith component is needed to keep notation simple while considering hierarchy of relaxations. The ijth element of a matrix Y is represented by Y_{ij}.

The notation $Y \succeq (\preceq) 0$ represents a constraint requiring matrix Y to be symmetric positive (negative) semidefinite. The notation $u \bullet v$ is used to denote the inner-product of vectors u and v. Also the notation $E \bullet Y$ is used to represent sum of component-wise product of matrices E and Y, i.e., $E \bullet Y \equiv \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij} Y_{ij}$ for matrices $E, Y \in \mathbb{R}^{n\times n}$. $E \bullet Y$ can also be thought of as inner product of two vectors generated by writing elements of E and Y in a vector form.

For vectors u and v of equal length, $u \otimes v$ represents a vector whose ith component is $(u)_i(v)_i$. Also $v \oslash u$ represents a vector whose ith component is $(v)_i/(u)_i$. Given square matrices R and E, by $R \otimes E$ we represent a matrix whose ijth element is $R_{ij}E_{ij}$. By $E \oslash R$ we represent a matrix whose ijth element is E_{ij}/R_{ij}, where $R_{ij} \neq 0$.

The set of feasible choices satisfying $\{d_1, \dots, d_n | \sum_{k=1}^{n} d_k = l, d_k \in \mathbb{Z}_+, k = 1, \dots, n.\}$ is represented by Δ^l.

2. Relaxations of MICP

Let

$$C = \left\{ x \;\middle|\; \begin{array}{ll} g_i(x) \leqslant 0, & i = 1, \dots, m, \\ 0 \leqslant x_j \leqslant 1, & j = 1, \dots, p, \\ 0 \leqslant x_j \leqslant L, & j = p+1, \dots, n \end{array} \right\},$$

be the feasible region of the continuous relaxation of (1). Let $G(x) \leqslant 0$ represent the set of constraints giving C, and

$$\mathcal{C} \equiv \operatorname{conv}\{x \in C \mid x_i \in \{0,1\}, \text{ for } i = 1, \dots, p\}$$

be the convex hull of the feasible set of MICP.

2.1. BACKGROUND

The following discussion on developing relaxations of the integer hull $\mathcal{C}$ for MICP is based on Stubbs and Mehrotra [15]. The work in [15] was motivated from earlier developments on disjunctive programming and linearlization techniques developed by Balas [1], Balas et al. [2], and Sherali and Adams [12, 13]. The idea of using a semidefinite inquality to describe a tighter relaxation is originally due to Lovász and Schrijver [10].

For a binary variable x_j, let

$$C_j^0 \equiv \{x \in C | x_j = 0\}, \text{ and } C_j^1 \equiv \{x \in C | x_j = 1\}.$$

The convex hull of sets C_j^0 and C_j^1 is given by

$$M_j(C) \equiv \left\{ \left(x, u_j^0, u_j^1, \lambda_j^0, \lambda_j^1\right) \;\middle|\; \begin{array}{l} x = \lambda_j^0 u_j^0 + \lambda_j^1 u_j^1, \\ \lambda_j^0 + \lambda_j^1 = 1,\ \lambda_j^0 \geqslant 0,\ \lambda_j^1 \geqslant 0, \\ u_j^0 \in C_j^0,\ u_j^1 \in C_j^1 \end{array} \right\},$$

where we have introduced new variables $u_j^0, u_j^1 \in \mathbb{R}^n$, and $\lambda_j^0, \lambda_j^1 \in \mathbb{R}$. The projection of $M_j(C)$ onto the $x-$space is given by

$$N_j(C) \equiv \left\{x \;\middle|\; \left(x, u_j^0, u_j^1, \lambda_j^0, \lambda_j^1\right) \in M_j(C)\right\}.$$

Note that the equality constraints in $M_j(C)$ are not linear, and therefore, the description of $M_j(C)$ involves non-convex constraints. Stubbs and Mehrotra [15] gave a convex reformulation of $M_j(C)$. This is accomplised as follows. Let $z = \lambda x$, and define a new function $q_i(z, \lambda) : \mathbb{R}^{n+1} \to \mathbb{R}$ as follows. Let $q_i(z, \lambda) \equiv \lambda g_i(z/\lambda)$, if $\lambda > 0$ and $q_i(0, 0) \equiv 0$, for $\lambda = 0$. Now define the set

$$\tilde{C} \equiv \left\{ (z, \lambda) \;\middle|\; \begin{array}{ll} q_i(z, \lambda) \leqslant 0, & i = 1, \dots, m \\ 0 \leqslant z_i \leqslant \lambda, & i = 1, \dots, p \\ 0 \leqslant z_j \leqslant \lambda L, & j = p+1, \dots, n \\ 0 \leqslant \lambda \leqslant 1 & \end{array} \right\} \equiv \{(z, \lambda) \,|\, Q(G(z), \lambda) \leqslant 0\}. \tag{2}$$

Using the boundedness assumption it can be shown [15, Lemma 1] that $q_i(z, \lambda)$ in (2) are convex over $\tilde{C}$. The constraints defining $\tilde{C}$ are represented by $Q(G(z), \lambda) \leqslant 0$. This notation emphasizes that the constraints in $Q(.) \leqslant 0$ are obtained from the constraints in $G(.) \leqslant 0$ after a variable transformation. By letting $v_j^0 = \lambda_j^0 u_j^0$ and $v_j^1 = \lambda_j^1 u_j^1$, $M_j(C)$ is written as

$$\tilde{M}_j(C) \equiv \left\{ \begin{pmatrix} x, \\ v_j^0, v_j^1, \lambda_j^0, \lambda_j^1 \end{pmatrix} \;\middle|\; \begin{array}{l} x = v_j^0 + v_j^1,\ \ (v_j^0, \lambda_j^0) \in \tilde{C}_j^0,\ \ (v_j^1, \lambda_j^1) \in \tilde{C}_j^1, \\ \lambda_j^0 + \lambda_j^1 = 1,\ \ \lambda_j^0 \geqslant 0, \lambda_j^1 \geqslant 0 \end{array} \right\}, \tag{3}$$

where

$$\tilde{C}_j^0 \equiv \{(z, \lambda) \in \tilde{C} | z_j = 0\},\ \ \tilde{C}_j^1 \equiv \{(z, \lambda) \in \tilde{C} | z_j = \lambda\}.$$

We can eliminate variables $v_j^0, \lambda_j^0, \lambda_j^1$ in (3) by using $x_j = (v_j^1)_j = \lambda_j^1$, $v_j^0 = x - v_j^1$, and $\lambda_j^0 = 1 - x_j$. As a consequence we have

$$\bar{M}_j(C) \equiv \left\{(x, v_j^1) \;\middle|\; Q(G(x - v_j^1), 1 - x_j) \leqslant 0,\ \ Q(G(v_j^1), x_j) \leqslant 0, (v_j^1)_j = x_j\right\}. \tag{4}$$

Note that the constriants in $Q(G(x - v_j^1), 1 - x_j) \leqslant 0$ are from $\mathbb{R}^{2n} \to \mathbb{R}$. It can be shown that these constraints are convex inqualities. Note that the projection of

$\bar{M}_j(C)$ onto the $x-$space is also equal to $N_j(C)$. More generally, the projection of set

$$\bar{M}(C) \equiv \left\{ (x, v_k^1, k=1,\dots,p) \;\middle|\; \begin{array}{ll} Q(G(x-v_j^1), 1-x_j) \leqslant 0, & j=1,\dots,p \\ Q(G(v_j^1), x_j) \leqslant 0, & j=1,\dots,p \\ (v_j^1)_j = x_j, & j=1,\dots,p \\ (v_j^1)_i = (v_i^1)_j, & i=1,\dots p, i<j=2,\dots,p \end{array} \right\}$$

onto the x-space, represented by $\bar{N}(C)$, is equal to $\cap_{j=1}^{p} N_j(C)$ strengthened further by the symmetry constraints.

We can strengthen this set further [15] by introducing a semidefinite inequality $V_{\mathbb{B}} - X_{\mathbb{B}} \succeq 0$, where $X_{\mathbb{B}}$ represents a $p \times p$ matrix whose st element is $x_s x_t$, $s, t, = 1, \dots, p$. $V_{\mathbb{B}}$ is a $p \times p$ matrix whose st element is $(v_s^1)_t$, $s, t = 1, \dots, p$. In particular, let

$$\hat{M}(C) \equiv \left\{ (x, v_k^1, k=1,\dots,p) \;\middle|\; \begin{array}{ll} Q(G(x-v_j^1), 1-x_j) \leqslant 0, & j=1,\dots,p \\ Q(G(v_j^1), x_j) \leqslant 0, & j=1,\dots,p \\ (v_j^1)_j = x_j, & j=1,\dots,p \\ V_{\mathbb{B}} - X_{\mathbb{B}} \succeq 0 & \end{array} \right\}.$$

and

$$\hat{N}(C) \equiv \left\{ x \mid (x, v_k^1, k=1,\dots,p) \in \hat{M}(C) \right\}.$$

Then, $\mathcal{C} \subseteq \hat{N}(C)$ [15, Theorem 7].

In our context of generating convex quadratic inequalities we find the use of additional variables v_j^1, $j = p+1, \dots, n$ and constraints $Q(G(v_j^1), x_j) \leqslant 0$, $j = p+1, \dots, n$, convenient. Let X be a $n \times n$ symmetric matrix whose st element is $x_s x_t$, $s, t = 1, \dots, n$, and V represent the $n \times n$ symmetric matrix whose jth row is v_j^1, and

$$M(C) \equiv \left\{ (x, v_k^1, k=1,\dots,n) \;\middle|\; \begin{array}{ll} Q(G(x-v_j^1), 1-x_j) \leqslant 0, & j=1,\dots,p \\ Q(G(v_j^1), x_j) \leqslant 0, & j=1,\dots,p \\ (v_j^1)_j = x_j, & j=1,\dots,p \\ Q(v_j^1, x_j) \leqslant 0, & j=p+1,\dots,n \\ V - X \succeq 0 & \end{array} \right\}. \quad (5)$$

The following Lemma show that the projection of $M(C)$ onto the x-space is contained in $\hat{N}(C)$, and that it provides a valid relaxation of $\mathcal{C}$.

LEMMA 2.1. *Let*

$$N(C) \equiv \left\{ x \;\middle|\; (x, v_k^1, k=1,\dots,p) \in M(C) \right\},$$

then $\mathcal{C} \subseteq N(C) \subseteq \hat{N}(C)$.

Proof. For any feasible solution $\hat{x}$ of MICP by taking $\hat{v}_j^1 = \hat{x}_j \hat{x},\ \ j = 1, \dots, n$ and $(\hat{v}_j^1)_j = \hat{x}_j,\ \ j = 1, \dots, p$ we can verify that $(\hat{x}, \hat{v}_k^1, k = 1, \dots, p) \in M(C)$. Hence $\hat{x} \in N(C)$. Since $\mathcal{C}$ is the convex hull of all feasible solutions of MICP, and $N(C)$ is a convex set, we must have $\mathcal{C} \subseteq N(C)$. Now we show that $N(C) \subseteq \hat{N}(C)$. Let $\hat{X}$ and $\hat{V}$ be X and V evaluated at $(\hat{x}, \hat{v}_k^1, k = 1, \dots, n)$, and $(\hat{x}, \hat{v}_k^1, k = 1, \dots, n) \in M(C)$. Then $(\hat{x}, \hat{v}_k^1, k = 1, \dots, p) \in \hat{M}(C)$, since $V_{\mathbb{B}} - X_{\mathbb{B}} \succeq 0$ is satisfied because $V - X \succeq 0$, and all other constraints defining $\hat{M}(C)$ are a subset of constraints defining $M(C)$. Hence $N(C) \subseteq \hat{N}(C)$ □

Let $N^0(C) = C$, $N^1(C) \equiv N(C)$, and $N^t(C) \equiv N(N^{t-1}(C))$, for $t \geqslant 2$. Note that the set $N^t(C)$ is obtained by applying the $N(\cdot)$ operator to $N^{t-1}(C)$. The following theorem and its proof is similar to [15, Theorem 6,7]. It shows that if the operator $N(.)$ is applied p times we get the integer hull $\mathcal{C}$.

THEOREM 2.2. *The following properties of $N^t(C)$ hold for $t = 1, \dots, p$:*

1. *$N^t(C)$ is a convex set,*
2. *$N^t(C) \subseteq N^{t-1}(C)$, $\mathcal{C} \subseteq N^t(C)$,*
3. *$N^p(C) = \mathcal{C}$.* □

In fact, a similar theorem [15, Theorem 1] is true for the weaker operator $N_{j_p}(N_{j_{p-1}}(\dots N_{j_1}(C)))$, where $j_1, \dots j_p$ is any permutation of the index set $1, \dots, p$.

Stubbs and Mehrotra [15] used the above results to develop a branch-and-cut method for MICP. In this method at some node of the branch and bound tree, linear cuts are generated. We explain the linear cut generation procedure. Let $\bar{x}$ be an optimal extreme point solution of the current continuous relaxation of MICP. By current continuous relaxation we mean the continuous relaxation of the feasible region of (1) to which the cuts generated thus far are added. Without loss of generality represent this set by C. To understand how a linear cut is generated, first consider the projection problem

$$\min_{x \in T(x)} \|x - \bar{x}\|, \tag{6}$$

where $T(x)$ can be $\bar{N}(C)$, $\hat{N}(C)$, $N(C)$, or more generally, $N^t(C)$, and $\|.\|$ is any vector norm function. Let x^* be an optimal solution of (6). An appropriate subgradient of the objective function of (6) at x^* can be used to get a valid inequality for the set $T(x)$.

Obviously $T(x)$ is not known to us. However, we know that it is a projection of some set $S(x, y)$ in a higher dimensional space whose representation is explicitly known. Therefore, we consider the problem

$$\min_{(x,y) \in S(x,y)} \|x - \bar{x}\|, \tag{7}$$

instead of (6). Here the set $S(x, y)$ can be $\bar{M}(C)$, $\hat{M}(C)$, $M(C)$, or more generally, $M^t(C)$. An appropriate subgradient of the objective function in (7) can be used to generate a valid inequality for $S(x, y)$, which because of the form of the subgradient (only x variables appear in the objective function) remains valid for $T(x)$.

2.2. MOTIVATION FOR GENERATING NONLINEAR CUTS: AN EXAMPLE

The following example demonstrates the value of using nonlinear cuts, and it explains some of the notation introduced earlier. Consider the problem:

$$\left.\begin{array}{ll} \text{maximize} & x_1 + 1.1x_2 + 1.2x_3 \\ \text{subject to} & (x_1 + x_2 - 0.5)^2 + (x_2 + x_3 - 0.5)^2 + (x_3 + x_1 - 0.5)^2 \leqslant 0.75 \\ & x_j \in \{0, 1\}, \quad j = 1, \ldots, 3, \end{array}\right\} \tag{8}$$

It is easy to verify that the optimal solution of (8) is $(0, 0, 1)$.

A solution to the continuous relaxation of (8) is $\approx (0.407, 0.497, 0.586)$ with the optimal objective value ≈ 1.659. The sets C_1^0 and C_1^1 are given by

$$\{x | x_1 = 0, 0 \leqslant x_2 \leqslant 1, 0 \leqslant x_3 \leqslant 1,\ (x_2 - 0.5)^2 + (x_2 + x_3 - 0.5)^2 + (x_3 - 0.5)^2 \leqslant 0.75\}$$

and

$$\{x | x_1 = 1, 0 \leqslant x_2 \leqslant 1, 0 \leqslant x_3 \leqslant 1,\ (x_2 + 0.5)^2 + (x_2 + x_3 - 0.5)^2 + (x_3 + 0.5)^2 \leqslant 0.75\},$$

respectively. These are same as

$$\begin{aligned} C_1^0 &= \left\{x | x_1 = 0, (x_2 + x_3 - 0.5)^2 \leqslant 0.25\right\}, \\ C_1^1 &= \{x | x_1 = 1, x_2 = 0, x_3 = 0\}. \end{aligned}$$

The convex hull of sets C_1^0 and C_1^1 is given by:

$$\left\{(x_1, x_2, x_3) \,\middle|\, \begin{array}{l} x_1 = 1 - \lambda, x_2 = \lambda(u_1^0)_2, x_3 = \lambda(u_1^0)_3, \\ 0 \leqslant \lambda \leqslant 1, ((u_1^0)_2 + (u_1^0)_3 - 0.5)^2 \leqslant 0.25 \end{array}\right\},$$

which after variable transformation can be rewritten as:

$$\left\{(x_1, x_2, x_3) \,\middle|\, \begin{array}{l} (x_2 + x_3 - (1 - x_1)/2)^2 \leqslant (1 - x_1)/4, \\ 0 \leqslant x_2 \leqslant 1 - x_1,\ 0 \leqslant x_3 \leqslant 1 - x_1 \end{array}\right\}.$$

Note that in our example we are able to write the convex hull explicitly in the space of (x_1, x_2, x_3). This is not possible in general. Similarly, we can also write the convex hull $\text{conv}(C_2^0, C_2^1)$ and $\text{conv}(C_3^0, C_3^1)$ as

$$\left\{(x_1, x_2, x_3) \,\middle|\, \begin{array}{l} (x_1 + x_3 - (1 - x_2)/2)^2 \leqslant (1 - x_2)/4, \\ 0 \leqslant x_1 \leqslant 1 - x_2,\ 0 \leqslant x_3 \leqslant 1 - x_2 \end{array}\right\}$$

and

$$\left\{ (x_1, x_2, x_3) \;\middle|\; \begin{array}{l} (x_1 + x_2 - (1-x_3)/2)^2 \leqslant (1-x_3)/4, \\ 0 \leqslant x_1 \leqslant 1 - x_3,\ 0 \leqslant x_2 \leqslant 1 - x_3 \end{array} \right\},$$

respectively. The set $\bar{N}(C)$ is the intersection of sets $\text{conv}(C_1^0, C_1^1)$, $\text{conv}(C_2^0, C_2^1)$, and $\text{conv}(C_3^0, C_3^1)$.

When we minimize our objective function over $\bar{N}(C)$, we obtain the optimal solution (0, 0, 1). The most important inequalities in describing $\text{conv}(C_j^0, C_j^1)$, $j = 1, 2, 3$ are the nonlinear inequalities. Clearly, in the above example such a compact description of $\text{conv}(C_j^0, C_j^1)$, $j = 1, 2, 3$ is not possible using linear inequalities only.

In the above example nonlinear cuts are generated directly by considering the convex hull, and not from solving the projection problem. As mentioned earlier, in general it will not be possible to generate this convex hull directly. In fact, in practice while solving a mixed integer program we do not generate the convex hull. We only generate inequalities that describe this region selectively. In the subsequent sections we give a technique for generating such inequalities.

3. Convex quadratic cuts for MICP

In this section we develop a method for generating valid convex quadratic inequalities. This is accomplished by coming up with an appropriate projection problem. For this purpose by V_{ij} represent the variable $(v_i)_j$, which is also the ijth element of matrix V. We introduce a new set of variables $\Upsilon_{ij},\ i, j = 1, \ldots, n$, corresponding to variables V_{ij} in (5), and consider the problem:

$$\left.\begin{array}{ll} \text{minimize} & h(x, V, \Upsilon) \equiv \frac{1}{2}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(V_{ij} - \bar{x}_i\bar{x}_j - \Upsilon_{ij})^2 \\ \text{subject to} & (x, V_{ij}, i, j = 1, \ldots, n) \in M(C) \\ & \Upsilon \equiv [\Upsilon_{ij}] \succeq 0. \end{array}\right\} \tag{9}$$

In problem (9) the constraint $\Upsilon \succeq 0$ ensures that the solution of (9) can be used to generate a convex constraint (see below). Let $\bar{X}$ represent X evaluated at $\bar{x}$. The second part of the objective function in (9) is $1/2$ the square of the Frobenius norm of the matrix $V - \bar{X} - \Upsilon$, i.e., $\|V - \bar{X} - \Upsilon\|_F \equiv \sum_{i=1}^{n}\sum_{j=1}^{n}(V_{ij} - \bar{X}_{ij} - \Upsilon_{ij})^2$. The gradient of the objective function in (9) is given by

$$\begin{array}{rcll} (\nabla h(x, V, \Upsilon))_{x_j} & = & (x - \bar{x})_j & j = 1, \ldots, n, \\ (\nabla h(x, V, \Upsilon))_{V_{ij}} & = & (V_{ij} - \bar{X}_{ij} - \Upsilon_{ij}), & i, j = 1, \ldots, n, \\ (\nabla h(x, V, \Upsilon))_{\Upsilon_{ij}} & = & -(V_{ij} - \bar{X}_{ij} - \Upsilon_{ij}), & i, j = 1, \ldots, n. \end{array}$$

Let (x^*, V^*, Υ^*) be an optimal solution of (9). The gradient of the objective function at this solution is written as $(\xi^*, E^*, -E^*)$, where $\xi^* = (x^* - \bar{x})$, and $E^* \equiv V^* - \bar{X} - \Upsilon^*$.

LEMMA 3.1. *The matrix E^* is a symmetric negative semidefinite matrix. Furthermore, if $\bar{x} \notin N(C)$, then the optimal objective value of (9) is positive.*

Proof. Since V, $\bar{X}$ and Υ are symmetric matrices, E^* is also a symmetric matrix. Hence, there exists an orthogonal matrix Q such that $Q^T E^* Q = \Lambda \equiv diag(\Lambda_1, \ldots \Lambda_n)$, where $\Lambda_1, \ldots \Lambda_n$ are the eigenvalues of E^* [8, p. 268, Theorem 8.1-1]. The orthogonality of Q implies that $E^* = Q\Lambda Q^T$. Assume, without loss of generality, that the first $k \leqslant n$ eigen values of E^* are positive, i.e., $\Lambda_1, \ldots, \Lambda_k \geqslant 0$. Now take $\hat{\Upsilon} = \Upsilon^* + Q\hat{\Lambda}Q^T$, where $\hat{\Lambda} = diag(\Lambda_1, \ldots \Lambda_k, 0, \ldots 0)$. Clearly, $(x^*, V^*, \hat{\Upsilon})$ is a feasible solution for (9). Let $\hat{E} = V^* - \bar{X} - \hat{\Upsilon}$. Since $\|\hat{E}\|_F^2 = \|Q[\Lambda - \hat{\Lambda}]Q^T\|_F^2 = \sum_{i=k+1}^n \Lambda_i^2 < \sum_{i=1}^n \Lambda_i^2 = \|Q^T \Lambda^* Q\|_F^2 = \|E^*\|_F^2$, we have a contradiction with the assumption that (x^*, V^*, Υ^*) is optimal.

Since $\bar{x} \notin N(C)$ we do not have a V so that $(\bar{x}, V) \in M(C)$. Therefore, for any value of Υ, $(\bar{x}, \bar{X} + \Upsilon) \notin M(C)$. Hence, the objective value of (9) is positive. □

We note that if $\bar{x}$ is an extreme point optimal solution of the current continuous relaxation of MICP, then $\bar{x} \notin N(C)$. The inequality

$$\xi^* \bullet x + E^* \bullet (V - \Upsilon) \geqslant \xi^* \bullet x^* + E^* \bullet \left(V^* - \Upsilon^*\right)$$

is a valid inequality because (9) is convex program with differentiable objective function [5, Theorem 3.4.3]. Furthermore,

$$\xi^* \bullet x + E^* \bullet V \geqslant \xi^* \bullet x^* + E^* \bullet \left(V^* - \Upsilon^*\right) \tag{10}$$

is also valid for (9) because for every feasible (x, V, Υ), $(x, V, 0)$ is also feasible. The above inequality is now used to generate a valid convex quadratic inequality in the space of $x-$variables only.

THEOREM 3.2. *Let $\bar{x} \notin N(C)$, and E^* be defined as in Lemma 3.1 at an optimal solution (x^*, V^*, Υ^*) of (9). Then,*

$$\xi^* \bullet x + E^* \bullet X \geqslant \xi^* \bullet x^* + E^* \bullet (V^* - \Upsilon^*) \tag{11}$$

is a valid convex quadratic inequality that cuts away $\bar{x}$.

Proof. If (11) is not valid then for some feasible solution $\hat{x}$ of MICP,

$$\xi^* \bullet \hat{x} + E^* \bullet \hat{X} < \xi^* \bullet x^* + E^* \bullet (V^* - \Upsilon^*),$$

where $\hat{X}$ is X evaluated at $\hat{x}$. Since, $(\hat{x}, \hat{v}_j^1 = \hat{x}_j \hat{x}, j = 1, \ldots, n) \in M(C)$, it means that the inequality (10) is not valid for $M(C)$, which is a contradiction. Now we show that (11) cuts away $\bar{x}$. Note that

$$\begin{aligned}\xi^* \bullet (\bar{x} - x^*) + E^* \bullet \bar{X} = -\xi^* \bullet \xi^* + E^* \bullet \bar{X} = -\xi^* \bullet \xi^* \\ + E^* \bullet (V^* - \Upsilon^* - E^*) < E^* \bullet (V^* - \Upsilon^*),\end{aligned}$$

where the last inequality follows by observing that the optimal objective value of (9) is given by $\xi^* \bullet \xi^* + E^* \bullet E^*$, and from Lemma 3.1 it is positive. The convexity of (11) follows because the Hessian matrix of the function defining this constraint is $2E^*$, which is negative semidefinite from Lemma 3.1. □

4. Nonlinear cuts for linear problems

In this section, we study a method for generating convex cuts for mixed binary linear problems. We do this for two reasons. First, it would help the reader to motivate our developments in Sections 3, 5 and 6 from the known linear cut generation techniques in the lift-and-project method of Balas et al. [2] and the reformulation-linearization technique of Sherali and Adams [12, 13]. The second reason is algorithmic. Lovász and Schrijver [10] state that including a semidefinite constraint may provide better approximations to the integer hull of a polytope. This is shown to be true for several structured combinatorial problems [7, 9, 11]. The projection $N(C)$ of the convex regions in higher dimensional space defined by including semidefinite constraint $M(C)$ is not necessarily a polytope. Therefore, it seems natural to consider representing $N(C)$ by using convex nonlinear constraints as well as linear constraints. Furthermore, considering the linear case has resulted in additional insights for constructing alternative objective functions in the projection problem (9), as well as a methodology for generating convex polynomial cuts. We develop methods for generating convex quadratic cuts in the next subsection. This is generalized to convex polynomial cuts in Section 4.2.

4.1. CONVEX QUADRATIC CUTS FOR LINEAR PROBLEMS

Consider the problem

$$\left.\begin{array}{lll} \text{minimize} & c \bullet x & \\ \text{subject to} & a^i \bullet x \geqslant b_i, & i = 1, \ldots m, \\ & x_j \in \{0, 1\}, & j = 1, \ldots, p, \\ & x_j \geqslant 0, & j = p+1, \ldots, n, \end{array}\right\} \tag{12}$$

and consider the set

$$\begin{aligned} K &= \left\{x \in \mathbb{R}^n | a^i \bullet x \geqslant b_i,\ i = 1, \ldots m,\ x \geqslant 0,\ x_j \leqslant 1,\ j = 1, \ldots, p\right\}, \\ &\equiv \left\{x \in \mathbb{R}^n | \tilde{a}^i \bullet x \geqslant \tilde{b}_i,\ i = 1, \ldots \bar{m}\right\}, \end{aligned} \tag{13}$$

where $\bar{m} = m + n + p$. Sherali and Adams [12, 13] developed the following lifting procedure that provides a strengthened relaxation of (12) in a higher dimensional space. They multiply each of the constraint $\tilde{a}^i \bullet x \geqslant \tilde{b}_i$ by x_j and $(1 - x_j)$ for

$j = 1, \ldots, p$ to obtain the nonlinear system

$$\left.\begin{array}{rcl} (1-x_1)(\tilde{a}^i \bullet x - \tilde{b}_i) & \geqslant & 0, \;\; i = 1, \ldots \bar{m}, \\ x_1(\tilde{a}^i \bullet x - \tilde{b}_i) & \geqslant & 0, \;\; i = 1, \ldots \bar{m}, \\ & \vdots & \\ (1-x_p)(\tilde{a}^i \bullet x - \tilde{b}_i) & \geqslant & 0, \;\; i = 1, \ldots \bar{m}, \\ x_p(\tilde{a}^i \bullet x - \tilde{b}_i) & \geqslant & 0, \;\; i = 1, \ldots \bar{m}. \end{array}\right\} \tag{14}$$

They substitute x_j for x_j^2, $j = 1, \ldots, p$ in (14). Next they linearize (14) by substituting a new variable y_{ij} for $x_i x_j = x_j x_i$, $i = 1, \ldots, n$, $i < j = 2, \ldots, n$. The projection of the polyhedron obtained in this way onto the x-space is equal to $\bar{N}(K)$.

Let us represent the constraints resulting from (14) and subsequent linearization by

$$e^i + f^i \bullet x + g^i \bullet y \geqslant 0, \;\; i = 1, \ldots, \tilde{m},$$

where $\tilde{m} = 2p\bar{m}$. A valid inequality, $\sum_{i=1}^{\tilde{m}} w_i(e^i + f^i \bullet x) \geqslant 0$, is obtained by using the projection cone

$$\left\{ w \,\middle|\, \sum_{i=1}^{\tilde{m}} w_i g^i = 0, w \geqslant 0 \right\}. \tag{15}$$

In the context of generating cuts in a branch-and-cut method, Balas, Ceria, and Cornuéjols [2] suggest several different optimization problems to determine w, while using the projection cone for constraints obtained from multiplying with only one of the variables (i.e., x_j and $(1 - x_j)$ for some j) in (14). In the subsequent discussion we develop these concepts further to generate convex quadratic cuts. Consider the procedure described above until we substitute for $x_j^2 = x_j$, however do not linearize by replacing $x_i x_j$ with y_{ij}. Add constraints $x_j(a^i \bullet x - \tilde{b}_i) \geqslant 0$, for $j = p+1, \ldots n$, $i = 1, \ldots, m$, and constraints $-x_j^2 + x_j \geqslant 0$ for $j = 1, \ldots, p$. The resulting system is written as a set of quadratic inequalities of the form

$$e^i + f^i \bullet x + \frac{1}{2} F^i \bullet X \geqslant 0, \;\; i = 1, \ldots \hat{m}, \tag{16}$$

where F^i are $n \times n$ symmetric matrices, and $\hat{m}$ is the number of constraints obtained in this way. The projection cone

$$\left\{ w \,\middle|\, \sum_{i=1}^{\hat{m}} w_i F^i \preceq 0, w \geqslant 0 \right\}$$

is used to generate a convex quadratic inequality

$$\sum_{i=1}^{\hat{m}} w_i \left(e^i + f^i \bullet x + \frac{1}{2} F^i \bullet X \right) \geqslant 0. \tag{17}$$

Note that valid linear inequalities are generated by requiring $\sum_{i=1}^{\hat{m}} w_i F^i = 0$ in the above cone.

We can now write several different cut generation problems by using this projection cone to determine a proper choice of w. The following cut generation problem is similar to the first normalization problem considered in Balas, Ceria, and Cornuéjols [2]:

$$\left.\begin{array}{rl} \text{maximize} & -\sum_{i=1}^{\hat{m}} w_i(e^i + f^i \bullet \bar{x} + \frac{1}{2}F^i \bullet \bar{X}) \\ \text{subject to} & \sum_{i=1}^{\hat{m}} w_i F^i \preceq 0, \\ & \sum_{i=1}^{\hat{m}} w_i = 1, \\ & w \geqslant 0. \end{array}\right\} \tag{18}$$

The objective in (18) tries to find a cut that is violated by $\bar{x}$ by maximum amount.

We can also construct several alternative objectives measuring the quality of a cut differently. The analogue of the cut generation problem in Balas et al. [2] for which the 1-norm of the coefficients of the generated cut is bounded by one is given below. This normalization (whose dual is the $\infty-$norm projection problem) was found most effective by Balas et al. [2, 3] while generating linear cuts. This cut generation problem is written as

$$\left.\begin{array}{rl} \text{maximize} & -\left(\sum_{i=1}^{\hat{m}} e^i w_i + (\alpha^+ - \alpha^-) \bullet \bar{x} + (E^+ - E^-) \bullet \bar{X}\right) \\ \text{subject to} & \sum_{i=1}^{\hat{m}} w_i f^i - \alpha^+ + \alpha^- = 0, \\ & \sum_{i=1}^{\hat{m}} w_i F^i - E^+ + E^- \preceq 0, \\ & \sum_{i=1}^{n}(\alpha^+)_i + \sum_{i=1}^{n}(\alpha^-)_i + \sum_{i=1}^{n}\sum_{j=1}^{n}([E^+]_{ij} + [E^-]_{ij}) \leqslant 1 \\ & E^+ - E^- \preceq 0, \\ & w \geqslant 0,\ \alpha^+ \geqslant 0,\ \alpha^- \geqslant 0,\ [E^+]_{ij} \geqslant 0,\ [E^-]_{ij} \geqslant 0, i, j = 1, \ldots n. \end{array}\right\} \tag{19}$$

Here $\alpha^+, \alpha^- \in \mathbb{R}^n$, and E^+, E^- are $n \times n$ symmetric matrices of variables. Note that an optimal solution of (18) or (19) gives a convex quadratic inequality from (17).

It is interesting to see the relationship of (19) with the cut generation problem discussed in Section 3. For this purpose consider the dual of (19):

$$\left.\begin{array}{rll} \text{minimize} & \pi & \\ \text{subject to} & e^i + f^i \bullet x + F^i \bullet V \geqslant 0, & i = 1, \ldots, \hat{m}, \\ & -\bar{x}_i + x_i - \pi \leqslant 0, & i = 1, \ldots, n \\ & \bar{x}_i - x_i - \pi \leqslant 0, & i = 1, \ldots, n \\ & \left[-\bar{x}\bar{x}^T + V - \Upsilon\right]_{ij} - \pi \leqslant 0, & i, j = 1, \ldots, n \\ & \left[\bar{x}\bar{x}^T - V + \Upsilon\right]_{ij} - \pi \leqslant 0, & i, j = 1, \ldots, n \\ & \pi \geqslant 0,\ V \succeq 0,\ \Upsilon \succeq 0. & \end{array}\right\} \tag{20}$$

Problem (20) is written as the following $\infty-$norm projection problem:

$$\left.\begin{array}{ll} \text{minimize} & \|x-\bar{x}\|_\infty + \left\|V-\bar{X}-\Upsilon\right\|_\infty \\ \text{subject to} & e^i + f^i \bullet x + F^i \bullet V \geqslant 0 \qquad i=1,\ldots,m, \\ & V \succeq 0,\, \Upsilon \succeq 0, \end{array}\right\} \tag{21}$$

where the norm in the second term of the objective function of (21) is written by treating the matrix as a vector. The objective function in (21) suggested us the form of objective function in (9). Note that we do not have the projection cone explicitly available to us when considering MICP.

4.2. CONVEX POLYNOMIAL CUTS FOR LINEAR PROBLEMS

The reformulation-linearization described in the previous section is called level-1 relaxation by Sherali and Adams [12, 13]. Sherali and Adams [12, 13] consider a hierarchy of relaxations indexed by $d = 1, \ldots, p$ leading up to the convex hull representation at level-p. The relaxation at level d is constructed by multiplying each of the equations in $\tilde{a}^i \bullet x \geqslant b_i$ with all d-degree polynomial factors of the form:

$$\prod_{j\in J1} x_j \prod_{j\in J2}(1-x_j) \text{ for each } J1, J2 \subseteq \{1,\ldots,p\},\ \ J1\cap J2=\emptyset,$$
$$\text{and } |J1\cup J2| = d.$$

Note that there are $C_d^p\ 2^d$ such factors. The reformulation-linearization technique proceeds with substituting $x_j^2 = x_j$, for $j = 1, \ldots, p$, and linearizing the nonlinear terms by a new variable for each of the nonlinear terms.

We consider the same procedure until we substitute for $x_j^2 = x_j$, however we do not linearize. Now add all convex polynomial constraints of degree $d+1$ satisfied by an $x \in \mathbb{B}^n$. For example, constraints $-(x_j)^k + x_j \geqslant 0$ for $k = 2, \ldots, d$ are added. Also multiply each of the equations in $\tilde{a}^i \bullet x \geqslant b_i$ with all $d-$degree polynomial factors of the form:

$$\prod_{j\in J3} x_j \text{ for each } J3 \subseteq \{1,\ldots,n\},\ \text{and } |J3| = d.$$

The resulting system is written as a set of $d+1$-degree polynomial inequalities of the form

$$P^i(x) \equiv \sum_{l=0}^{d+1} \sum_{\{d_1,\ldots,d_n\}\in\Delta^l} \left(f^i_{d_1 d_2 \ldots d_n} \prod_{j=1}^n x_j^{d_j} \right) \geqslant 0, \quad i = 1, \ldots, m_d. \tag{22}$$

The second summation sign in (22) means that the sum is taken over all nonnegative integer choices of $d_1, \ldots, d_n$ that satisfy $\sum_{k=1}^n d_k = l$. $f^i_{d_1 d_2 \ldots d_n}$ represents coefficients in the polynomial. The Hessian matrix of $P^i(x)$ is written as

$$H^i(x) \equiv \sum_{l=0}^{d-1} \sum_{\{d_1,\dots,d_n\}\in\Delta^l} \left(\prod_{j=1}^{n} x_j^{d_j}\right) F^i_{d_1 d_2 \dots d_n} \quad i = 1,\dots,m_d, \tag{23}$$

where $F^i_{d_1\dots d_n}$ are $n \times n$ symmetric matrices, whose coefficients are explicitly known from the coefficients in (22). In particular, the matrix multiplied with $\prod_{j=1}^{n} x_j^{d_j}$ is given by

$$\left[F^i_{d_1 d_2 \dots d_n}\right]_{st} = \begin{cases} (d_s+1)(d_t+1) f^i_{d_1\dots d_s+1\dots d_t+1\dots d_n}, & s \neq t, \\ (d_s+2)(d_s+1) f^i_{d_1\dots d_s+2\dots d_n}, & s = t. \end{cases}$$

As in the quadratic case, we use a projection cone to generate a convex polynomial inequality from the inequalities in (22). The following proposition gives such a projection cone.

PROPOSITION 4.1. *Let* w *satisfy*

$$\left\{ w \,\middle|\, \sum_{i=1}^{m_d} w_i F^i_{d_1\dots d_n} \preceq 0,\ w \geqslant 0,\ \forall \{d_1,\dots,d_n\} \in \Delta^l, l = 1,\dots,d-1 \right\}.$$

Then, the inequality $\sum_{i=1}^{m_d} w_i P^i(x) \geqslant 0$ *is a convex polynomial inequality over* $\mathbb{R}^n_+$.

Proof. The Hessian matrix of the polynomial $\sum_{i=1}^{m_d} w_i P^i(x)$ is given by

$$H(x) \equiv \sum_{i=1}^{m_d} w_i H^i(x) = \sum_{l=0}^{d-1} \sum_{\{d_1,\dots,d_n\}\in\Delta^l} \left(\prod_{j=1}^{n} x_j^{d_j}\right) \sum_{i=1}^{m_d} w_i F^i_{d_1 d_2 \dots d_n},$$
$$i = 1,\dots,m_d.$$

The product $\prod_{j=1}^{n} x_j^{d_j}$ is non-negative since $x \in \mathbb{R}^n_+$, and $\sum_{i=1}^{m_d} w_i F^i_{d_1 d_2\dots d_n} \preceq 0$ from the definition of the projection cone. The result follows because the positive scalar product and the sum of negative semidefinite matrices gives a negative semidefinite matrix. □

We can now write the generalizations of problems (18) and (19) for polynomial cut generation. The analogue of problem (18) is:

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^{m_d} w_i P^i(\bar{x}) \\ \text{subject to} & \sum_{i=1}^{m_d} w_i F^i_{d_1\dots d_n} \preceq 0, \quad \forall \{d_1,\dots,d_n\} \in \Delta^l, l = 0,\dots d-1 \\ & \sum_{i=1}^{m_d} w_i = 1,\ w \geqslant 0. \end{array}$$

The analogue of problem (19) is:

$$
\left.\begin{array}{ll}
\max & -\sum_{i=1}^{m_d} e^i w_i - (\alpha^+ - \alpha^-) \bullet \bar{x} - \sum_{l=2}^{d+1} \sum_{\{d_1,\dots,d_n\}\in\Delta^l} \left(e^+_{d_1\dots d_n} - e^-_{d_1\dots d_n}\right) \prod_{j=1}^{n} \bar{x}_j^{d_j} \\
\text{s.t.} & \sum_{i=1}^{m_d} w_i f^i - \alpha^+ + \alpha^- = 0, \\
& \sum_{i=1}^{m_d} w_i F^i_{d_1\dots d_n} - G_{d_1\dots d_n} \preceq 0,\ \forall\{d_1,\dots,d_n\} \in \Delta^l, l = 1,\dots d-1, \\
& [G_{d_1\dots d_n}]_{st} = (d_s+1)(d_t+1)(e^+_{d_1\dots d_s+1\dots d_t+1\dots d_n} - e^-_{d_1\dots d_s+1\dots d_t+1\dots d_n}),\ s = 1,\dots,n, \\
& \qquad t = s+1,\dots,n,\ \text{and}\ \forall\{d_1,\dots,d_n\} \in \Delta^l, l = 0,\dots d-1, \\
& [G_{d_1\dots d_n}]_{ss} = (d_s+2)(d_s+1)(e^+_{d_1\dots d_s+2\dots d_n} - e^-_{d_1\dots d_s+2\dots d_n}), \\
& \qquad s = 1,\dots,n, \text{and}\ \forall\{d_1,\dots,d_n\} \in \Delta^l, l = 0,\dots,d-1 \\
& \sum_{i=1}^{n} (\alpha^+)_i + \sum_{i=1}^{n} (\alpha^-)_i + \sum_{l=2}^{d+1} \sum_{\{d_1,\dots,d_n\}\in\Delta^l} \left(e^+_{d_1\dots d_n} + e^-_{d_1\dots d_n}\right) \leqslant 1, \\
& G_{d_1\dots d_n} \preceq 0, \quad \forall\{d_1,\dots,d_n\} \in \Delta^l, l = 0,\dots d-1, \\
& w \geqslant 0,\ \alpha^+ \geqslant 0,\ \alpha^- \geqslant 0, \\
& e^+_{d_1\dots d_n}, e^-_{d_1\dots d_n} \geqslant 0,\ \forall\{d_1,\dots,d_n\} \in \Delta^l, l = 2,\dots,d+1.
\end{array}\right\} \tag{24}
$$

Note that the second, third and fourth set of constraints in (24) are specified to ensure that the Hessian of the function giving the cut is negative semidefinite on $\mathbb{R}^n_+$.

5. Convex quadratic cuts using nonsmooth objectives

In Section 3, we presented a method for generating valid convex quadratic inequalities for MICP. For this purpose we used the Frobenius matrix norm in the objective function. In the previous section the dual (21) of the cut generation problem (19) is an $\infty-$norm projection problem. It is natural to ask whether it is possible to use other matrix norms in the objective function of (9). In this section we show that other matrix norms can be used instead of the Frobenius norm in the projection problem, provided that certain assumptions are satisfied. In particular, we need that Lagrange multipliers exist at the optimal solution found for the projection problem, and that the subgradient of the objective function satisfies certain additional properties.

Let us consider the projection problem:

$$
\left.\begin{array}{ll}
\text{minimize} & \|x - \bar{x}\| + \left\|V - \bar{X} - \Upsilon\right\| \\
\text{subject to} & (x, V_{ij}, i, j = 1,\dots,n) \in M(C) \\
& \Upsilon \succeq 0,
\end{array}\right\} \tag{25}
$$

where we follow the notation used for describing (9). The second term in the objective function of (25) can now be any matrix norm. First we give sufficient conditions for a feasible solution to be optimal for (25). We rewrite (25) using generic notation.

PROPOSITION 5.1. *Let us consider the problem*

$$\left.\begin{array}{ll} \textit{minimize} & h(u, Y, \Upsilon) \\ \textit{subject to} & h_i(u, Y) \leqslant 0, \qquad\qquad\qquad i = 1, \ldots, \hat{m}, \\ & (A_u)_j \bullet u + \left[(A_Y)_j\right] \bullet Y = b_j, \quad j = 1, \ldots \hat{l}, \\ & Y, \Upsilon \succeq 0, \end{array}\right\} \tag{26}$$

The set of inequality and linear equality constraints in (26) represent the constraints defining $M(C)$. The function $h(u, Y, \Upsilon)$ is any convex function, $\hat{m}$ is the number of constraints, and A_u, A_Y represent columns of equality constraint matrix A corresponding to variables in u and Y, and $(A_u)_j$ represent the jth row of A_u and $\left[(A_Y)_j\right]$ represent the jth row of A_Y written in matrix form. Assume that at a feasible solution (u^*, Y^*, Υ^*) we have Lagrange multipliers (μ, π) satisfying

$$\left.\begin{array}{l} \xi_h(u^*) + \sum_{i=1}^{\hat{m}} \mu_i \xi_{h_i}(u^*) + A_u^T \pi = 0, \\ F^* \equiv [\xi_h(Y^*)] + \sum_{i=1}^{\hat{m}} \mu_i \left[\xi_{h_i}(Y^*)\right] + \sum_{j=1}^{\hat{l}} \pi_j \left[(A_Y)_j\right] \succeq 0, \\ E^* \equiv -[\xi_h(\Upsilon^*)] \preceq 0, \ \ \mu_i \geqslant 0, i = 1, \ldots, \hat{m}, \\ \mu_i h_i(u^*, Y^*, \Upsilon^*) = 0, \ F^* \bullet Y^* = 0, \ E^* \bullet \Upsilon^* = 0, \end{array}\right\} \tag{27}$$

where $\xi_h(.)$, and $\xi_{h_i}(.)$ represent a subgradient of $h(.)$ and $h_i(.)$ at (u^*, Y^*, Υ^*), and the notation $\left[\xi_{h_i}()\right]$ represent the $()$ components of the subgradient vector written as a matrix. Then, (u^*, Y^*, Υ^*) is an optimal solution.

Proof. Assume that we have a direction (d_u, D_Y, D_Υ) such that $h(u^* + d_u, Y^* + D_Y, \Upsilon^* + D_\Upsilon) < h(u^*, Y^*, \Upsilon^*)$, and $(u^* + d_u, Y^* + D_Y, \Upsilon^* + D_\Upsilon)$ is feasible. Then, because of the convexity of h, we have $h(u^*, Y^*, \Upsilon^*) > h(u^* + d_u, Y^* + D_Y, \Upsilon^* + D_\Upsilon) \geqslant h(u^*, Y^*, \Upsilon^*) + \xi_h(u^*)^T d_u + [\xi_h(Y^*)] \bullet D_Y + [\xi_h(\Upsilon^*)] \bullet D_\Upsilon$. Hence,

$$0 > \xi_h(u^*)^T d_u + \left[\xi_h(Y^*)\right] \bullet D_Y + \left[\xi_h(\Upsilon^*)\right] \bullet D_\Upsilon. \tag{28}$$

Similarly, for the constraints satisfying $h_i(u^*, Y^*, \Upsilon^*) = 0$, we have

$$0 \geqslant \xi_{h_i}(u^*)^T d_v + \left[\xi_{h_i}(Y^*)\right] \bullet D_Y. \tag{29}$$

Now by taking the inner product of first three sets of equations in (27) with $(d_u, Y^* + D_Y, \Upsilon^* + D_\Upsilon)$, we have

$$0 \leqslant (Y^* + D_Y) \bullet F^* - (\Upsilon^* + D_\Upsilon) \bullet E^* = D_Y \bullet F^* - D_\Upsilon \bullet E^*,$$

where the inequality follows from the fact that for any symmetric positive semi-definite matrices E and Y, $E \bullet Y \geqslant 0$ and the equality uses complementarity conditions. The above inequality contradicts with (28) and (29). $\square$

The next theorem shows that a valid convex quadratic inequality can be generated if the subgradient of the objective function satisfies some additional properties. This theorem can be proved using arguments similar to those used in the proof of Theorem 3.2.

THEOREM 5.2. *Let* $(\xi^*, E^* \equiv [\xi(V^*)], \xi(\Upsilon^*))$ *represent the subgradient of the objective function of (25) used in verifying the optimality condition. Assume that* $E^* \preceq 0$. *In addition, assume that* $(\xi^*, \xi(V^*), \xi(\Upsilon^*))$ *satisfies conditions*

$$[\xi(V^*)]_{ij} = -[\xi(\Upsilon^*)]_{ij}, \text{ and } E^* \bullet \left(V^* - \bar{X} - \Upsilon^*\right) > 0. \tag{30}$$

Then,

$$\xi^* \bullet x + E^* \bullet X \geqslant \xi^* \bullet x^* + E^* \bullet (V^* - \Upsilon^*) \tag{31}$$

is a valid convex quadratic inequality that cuts away $\bar{x}$. □

In Theorem 5.2 the assumption that $E^* \preceq 0$ is satisfied from Proposition 5.1, which assumes the existence of Lagrange multipliers at the optimal solution x^*. The assumptions in (30) are satisfied in the two important cases where the objective function is defined using 1-norm or ∞-norm by treating the matrix as a vector.

6. Convex polynomial cuts for MICP

The technique for generating convex polynomial cuts in the linear case can not be used to generate such cuts for MICP. The reason is that we can no longer give the projection cone explicitly, which was possible in the linear case. Our approach in this section extends the approach for generating convex quadratic cuts in Section 3. We define an appropriate projection problem, whose solution is used to generate a polynomial cut. For this purpose, we consider sets generated by multiple application of operator $M(.)$. We need to introduce some additional notation to describe our approach.

Let n_l represent the number of variables used to represent $M^l(C)$, $l = 1, \dots, d$, $d \leqslant p$. Note the difference between n_l and n^j, where n^j is n multiplied j times. Let $n_0 = n$. We use n new variable vectors of length n_{l-1} (total $n_{l-1} + nn_{l-1}$ variables) to represent $M^l(C)$. We represent these vectors by v_j^l, $j = 1, \dots, n$. The vector v^l represents $(v_1^l, \dots, v_n^l)$. By $V^l_{j_1 \cdots j_{l+1}}$ we represent the $n_{l-1}(j_{l+1} - 1) + n_{l-2} + \sum_{k=0}^{l-2} n^{l-k-1}(j_{l-k} - 1) + j_1$ element of v^l. Loosely speaking, this element corresponds to variable product $x_{j_1} x_{j_2} \dots x_{j_{l+1}}$ in the x-space. By V we represent the vector of variables $(V^d_{j_1 \cdots j_{l+1}}, l = 1, \dots, d)$. By ψ^l we represent $(x, v^1, \dots, v^l)$.

For example, v_j^1, $j = 1, \dots, n$, represent n vectors of length n introduced while writing $M^1(C) \equiv M(C)$ in (5). Also, $v^1 = (v_j^1, j = 1, \dots, n)$, $\psi^1 = (x, v^1)$, $n_1 = n + n^2$, and $V_{j_1 j_2}$ is the $(j_2 - 1)n + j_1$th element of v^1. Similarly, to write

$M^2(C)$ we introduce n new variable vectors v_j^2 of length $n+n^2$ each, $v^2 = (v_j^2, j = 1, \dots, n)$, $\psi^2 = (x, v^1, v^2)$, $n_2 = n_1 + nn_1 = n + 2n^2 + n^3$, and $V_{j_1 j_2 j_3}$ is the $(j_3 - 1)n_1 + n_0 + (j_2 - 1)n + j_1$th element of v^2.

Let $G^1(\psi^1) \leqslant 0$ represent the set of constraints in $M^1(C)$, and more generally, let $G^l(\psi^l) \leqslant 0$, represent the set of constraints giving $M^l(C)$, $l = 1, \dots, d$. Here we let $G^0(.) \equiv G(.)$. The set $M^2(C)$ is now written as follows:

$$M^2(C) \equiv \left\{ (x, v^1, v^2) \;\middle|\; \begin{array}{ll} Q(G^1(\psi^1 - v_j^2), 1 - x_j) \leqslant 0, & j = 1, \dots, p, \\ Q(G(v_j^2), x_j) \leqslant 0, & j = 1, \dots, p, \\ (v_j^2)_j = x_j, & j = 1, \dots p, \\ Q(v_j^2, x_j) \leqslant 0, & j = p+1, \dots, n, \\ V_{j_1 j_2 j_3} = V_{\mathcal{P}(j_1 j_2 j_3)}, & \forall \mathcal{P}, j_1 \geqslant j_2 \geqslant j_3 = 1, \dots, n \end{array} \right\}.$$

Note that we have not written the semidefinite constraint while writing $M^2(C)$, however, semidefinite constraints are implicitly present in $G^1(.)$. The symmetry constraints $V_{j_1 j_2 j_3} = V_{\mathcal{P}(j_1 j_2 j_3)}$ require variables to take the same value if the indices of one can be obtained from other by a permutation. Here $\mathcal{P}$ represents a permutation of the argument index sets. More generally, the set $M^l(C)$, $l = 1, \dots d$, is written as

$$M^l(C) \equiv$$
$$\equiv \left\{ (x, v^1, \dots, v^l) \;\middle|\; \begin{array}{ll} Q(G^{l-1}(\psi^{l-1} - v_j^l), 1 - x_j) \leqslant 0, & j = 1, \dots, p, \\ Q(G^{l-1}(v_j^l), x_j) \leqslant 0, & j = 1, \dots, p, \\ (v_j^l)_j = x_j, & j = 1, \dots p, \\ Q(v_j^l, x_j) \leqslant 0, & j = p+1, \dots n, \\ V_{j_1 \dots j_{l+1}} = V_{\mathcal{P}(j_1 \dots j_{l+1})}, & \forall \mathcal{P}, j_1 \geqslant \dots \geqslant j_{l+1} = 1, \dots, n \end{array} \right\}. \tag{32}$$

The constraints written as equalities in (32) can be explicitly substituted to reduce the size of the problem, however, we retained them in (32) for notational convenience. It can be shown that the projection of $M^l(C)$ onto the x-space is $N^l(C)$.

We now introduce new variables which allow us to write the projection problem in a way so that its relation is clear with the convex cut generation method described in Section 4.2. For $l = 1, \dots, d$, let $y_{d_1 \dots d_n} = V_{j_1 \dots j_{l+1}}$, where d_j is the number of times subscript j repeats in $V_{j_1 \dots j_{l+1}}$. For example, $y_{0 \dots 1 \dots 0 \dots 1 \dots 0} = V_{j_1 j_2}$, where 1s in the subscript of y are at j_1 and j_2 locations only. Note that $y_{0 \dots 0}$ and $y_{0 \dots 1 \dots 0}$ are not defined. Also note that we can explicitly substitute variables $y_{\dots}$ in (32), however, we will keep them in (33) for notational convenience. Corresponding to each $y_{d_1 \dots d_n}$ we introduce weights $\rho_{d_1 \dots d_n} > 0$ and variables $\gamma_{d_1 \dots d_n}$. These are used in ensuring the convexity of generated polynomial cuts. The weights $\rho_{d_1 \dots d_n}$ are specified in Theorem 6.2 below.

We now introduce additional notation to express the separation problem for generating convex polynomial cuts as clearly as possible. By $X_{d_1 \dots d_n}$ we represent a $n \times n$ symmetric matrix whose st element is $x^{d_1} \dots x^{d_s+1} \dots x^{d_t+1} \dots x^{d_n}$, $s = 1, \dots, n$, $s > t$, $t = 1, \dots, n$. Similarly, by $Y_{d_1, \dots, d_n}$, $R_{d_1, \dots, d_n}$, and $\Upsilon_{d_1, \dots, d_n}$

we represent a matrix whose st elements are $y_{d_1\ldots d_s+1\ldots d_t+1\ldots d_n}$, $\rho_{d_1\ldots d_s+1\ldots d_t+1\ldots d_n}$, and $\gamma_{d_1\ldots d_s+1\ldots d_t+1\ldots d_n}$, respectively. The diagonal elements (ss, $s = 1, \ldots, n$) of $X_{d_1\ldots d_n}$, $Y_{d_1\ldots d_n}$, $R_{d_1\ldots d_n}$, and $\Upsilon_{d_1\ldots d_n}$ are given by $x^{d_1}\ldots x^{d_s+2}\ldots x^{d_n}$, $y_{d_1\ldots d_s+2\ldots d_n}$, $\rho_{d_1\ldots d_s+2\ldots d_n}$, and $\gamma_{d_1\ldots d_s+2\ldots d_n}$, respectively. We let $\bar{X}_{d_1\ldots d_n}$ be $X_{d_1\ldots d_n}$ evaluated at $\bar{x}$. Also we let $E_{d_1\ldots d_n} \equiv R_{d_1\ldots d_n} \otimes \left(Y_{d_1\ldots d_n} - \bar{X}_{d_1\ldots d_n} - \Upsilon_{d_1\ldots d_n}\right)$. The matrices $X_{d_1\ldots d_n}$, $\bar{X}_{d_1\ldots d_n}$, $Y_{d_1\ldots d_n}$, $R_{d_1\ldots d_n}$, $\Upsilon_{d_1\ldots d_n}$, and $E_{d_1\ldots d_n}$ are defined for $\{d_1, \ldots, d_n\} \in \Delta^l, l = 0, \ldots d-1$. We use Y, Υ and E to represent all the variables appearing in $Y_{d_1\ldots d_n}$, $\Upsilon_{d_1\ldots d_n}$ and $E_{d_1\ldots d_n}$ in a vector form.

Now we consider the following projection problem to generate $d+1$ degree polynomial cuts.

$$\left.\begin{array}{ll}\text{minimize} & h(x, Y, V, \Upsilon) \equiv \frac{1}{2}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2 + \dfrac{1}{2}\sum_{l=0}^{d-1}\sum_{\{d_1,\ldots,d_n\}\in\Delta^l}\left\|E_{d_1\ldots d_n}\right\|_F^2 \\ \text{subject to} & y_{d_1\ldots d_n} = V_{j_1\ldots j_{l+1}}, \quad \{d_1, \ldots, d_n\} \in \Delta^l, l = 1, \ldots d \\ & (x, v^1, \ldots, v^d) \in M^d(C) \\ & R_{d_1\ldots d_n} \otimes \Upsilon_{d_1\ldots d_n} \succeq 0, \quad \{d_1, \ldots, d_n\} \in \Delta^l, l = 0, \ldots, d-1. \end{array}\right\} \tag{33}$$

By $(x^*, Y^*, V^*, \Upsilon^*)$ we represent an optimal solution of (33). Note that V^* can be constructed from Y^* and vice-versa, and Y^* and Υ^* contain $Y^*_{d_1\ldots d_n}$ and $\Upsilon^*_{d_1\ldots d_n}$, $\{d_1\ldots d_n\} \in \Delta^l, l = 0, \ldots, d-1$, in a vector form. Let $\xi^* \equiv (\nabla h(x, Y, V, \Upsilon))_{x^*}$ and $Z^* \equiv (\nabla h(x, Y, V, \Upsilon))_{Y^*}$. Note that $(\nabla h(x, Y, V, \Upsilon))_{\Upsilon^*} = -Z^*$, and $Z^* = R \otimes E^*$, where E^* is E evaluated at $(x^*, Y^*, V^*, \Upsilon^*)$. The following lemma is a generalization of Lemma 3.1.

LEMMA 6.1. *Let* $E^*_{d_1\ldots d_n} = R_{d_1\ldots d_n} \otimes \left(Y^*_{d_1\ldots d_n} - \bar{X}_{d_1\ldots d_n} - \Upsilon^*_{d_1\ldots d_n}\right)$, $\{d_1, \ldots, d_n\} \in \Delta^l$, $l = 0, \ldots d-1$. *The matrices* $E^*_{d_1\ldots d_n}$ *are symmetric negative semidefinite. Furthermore, if* $\bar{x} \notin N^d(C)$*, then the optimal objective value of (33) is positive.*

Proof. We will follow arguments similar to those in the proof of Lemma 3.1. Since $E^*_{d_1\ldots d_n}$ is a symmetric matrix there exist an orthogonal matrix $Q_{d_1\ldots d_n}$ such that $Q^T_{d_1\ldots d_n} E^*_{d_1\ldots d_n} Q_{d_1\ldots d_n} = \Lambda_{d_1\ldots d_n} \equiv diag(\Lambda^1_{d_1\ldots d_n}, \ldots \Lambda^n_{d_1\ldots d_n})$, where $\Lambda^1_{d_1\ldots d_n}, \ldots \Lambda^n_{d_1\ldots d_n}$ are the eigenvalues of $E^*_{d_1\ldots d_n}$. The orthogonality of $Q_{d_1\ldots d_n}$ implies that $E^*_{d_1\ldots d_n} = Q_{d_1\ldots d_n} \Lambda_{d_1\ldots d_n} Q^T_{d_1\ldots d_n}$. Assume, without loss of generality, that the first $k \leqslant n$ eigen values of $E^*_{d_1\ldots d_n}$ are positive, i.e., $\Lambda^1_{d_1\ldots d_n}, \ldots, \Lambda^k_{d_1\ldots d_n} > 0$. Now construct symmetric matrix $\hat{\Upsilon}_{d_1\ldots d_n} \equiv \Upsilon^*_{d_1\ldots d_n} + \left[Q_{d_1\ldots d_n} \hat{\Lambda}_{d_1\ldots d_n} Q^T_{d_1\ldots d_n}\right] \oslash R_{d_1\ldots d_n}$. We can verify that $\hat{\Upsilon}_{d_1\ldots d_n}$ constructed in this way satisfies the constraint $R_{d_1\ldots d_n} \otimes \Upsilon_{d_1\ldots d_n} \succeq 0$. Hence $(x^*, Y^*, V^*, \hat{\Upsilon}^*)$ is feasible for (33). Let $\hat{E}_{d_1\ldots d_n} \equiv R_{d_1\ldots d_n} \otimes (Y^*_{d_1\ldots d_n} - \bar{X}_{d_1\ldots d_n} + -\hat{\Upsilon}^*_{d_1\ldots d_n})$. Now,

$$\begin{aligned}\|\hat{E}^*_{d_1\ldots d_n}\|_F &= \|E^*_{d_1\ldots d_n} - Q_{d_1\ldots d_n}\hat{\Lambda}_{d_1\ldots d_n} Q^T_{d_1\ldots d_n}\|_F \\ &= \|Q_{d_1\ldots d_n}(\Lambda_{d_1\ldots d_n} - \hat{\Lambda}_{d_1\ldots d_n}) Q^T_{d_1\ldots d_n}\|_F < \|E^*_{d_1\ldots d_n}\|_F.\end{aligned}$$

We have constructed a feasible solution with a smaller objective value, which is a contradiction.

Since $\bar{x} \notin N^d(C)$, we do not have a $(v^1, \dots v^d)$ so that $(\bar{x}, v^1, \dots, v^d) \in M^d(C)$. Therefore, for any value of Υ, $(\bar{x}, \bar{X} + \Upsilon, V(\bar{X} + \Upsilon), \Upsilon)$ is not a feasible solution of (33). Here the notation $V(\bar{X}, \Upsilon)$ denotes that the value of variables in V are constructed from those in Y to satisfy the equality constraints. Hence, the objective value of (33) is positive. □

Following arguments similar to those used to get (10), we can show that the inequality

$$\xi^* \bullet x + Z^* \bullet Y \geqslant \xi^* \bullet x^* + Z^* \bullet \left(Y^* - \Upsilon^*\right) \tag{34}$$

is a valid inequality for $M^d(C)$. The following theorem is a generalization of Theorem 3.2 to the convex polynomial case.

THEOREM 6.2. *Let $\bar{x} \notin N^d(C)$, and $Z^*_{d_1 \dots d_n}$ be defined as above. Define $n \times n$ symmetric matrix $R_{d_1 \dots d_n}$ so that its st element is $1/(d_s + 1)(d_t + 1)$, if $s \neq t$. The diagonal element (ss, $s = 1, \dots, n$) of $R_{d_1 \dots d_n}$ is $1/(d_s + 2)(d_s + 1)$. Then, the inequality*

$$\xi^* \bullet x + \sum_{l=0}^{d-1} \sum_{\{d_1 \dots d_n\} \in \Delta^l} Z^*_{d_1 \dots d_n} \bullet X_{d_1 \dots d_n} \geqslant \xi^* \bullet x^* + \sum_{l=0}^{d-1} \sum_{\{d_1 \dots d_n\} \in \Delta^l} Z^*_{d_1 \dots d_n} \bullet (Y^*_{d_1 \dots d_n} - \Upsilon^*_{d_1 \dots d_n}) \tag{35}$$

is a valid convex polynomial inequality that cuts away $\bar{x}$.

Proof. The inequality (35) can be rewritten as

$$\xi^* \bullet x + Z^* \bullet X \geqslant \xi^* \bullet x^* + Z^* \bullet (Y^* - \Upsilon^*). \tag{36}$$

If (36) is not valid then for some feasible solution $\hat{x}$ of MICP,

$$\xi^* \bullet \hat{x} + Z^* \bullet \hat{X} < \xi^* \bullet x^* + Z^* \bullet (Y^* - \Upsilon^*)$$

where $\hat{X}$ is obtained by evaluating X at $\hat{x}$.

By taking $V^l_{j_1 \dots j_{l+1}} = \prod_{k=1}^{l+1} \hat{x}_{j_k}$, $j_1, \dots j_{l+1} = 1, \dots, n$, $l = 1, \dots, d$, $y_{d_1 \dots d_n} = \prod_{k=1}^{n} \hat{x}_k^{d_k}$, and $\Upsilon = 0$, we have a feasible solution of (33), which is violated by (34). This is a contradiction. Now we show that (36) cuts away $\bar{x}$. Let $\bar{X}$ be X evaluated at $\bar{x}$. Now,

$$\begin{aligned} \xi^* \bullet (\bar{x} - x^*) + Z^* \bullet \bar{X} &= -\xi^* \bullet \xi^* + Z^* \bullet \bar{X} \\ &= -\xi^* \bullet \xi^* + E^* \bullet (R \otimes (Y^* - \Upsilon^*) - E^*) \\ &< Z^* \bullet (Y^* - \Upsilon^*), \end{aligned}$$

where we have used $Z^* = R \otimes E^*$, and the last inequality follows by observing that the optimal objective value of (33) is given by $\frac{1}{2}\xi^* \bullet \xi^* + \frac{1}{2}E^* \bullet E^*$, which is positive from Lemma 6.1. By using the definition of Z^* and R, and taking partial derivatives of elements of $X_{d_1 \ldots d_n}$ it can be verified that the Hessian matrix of (35) is given by

$$\sum_{l=1}^{d-1} \sum_{\{d_1 \ldots d_n\} \in \Delta^l} E^*_{d_1 \ldots d_n}.$$

From Lemma 6.1 this is a negative semidefinite matrix, hence (35) is a convex inequality. □

By following the developments in Section 5, the methodology of this section can also be developed for situations where more general norms functions are used while defining the projection problem (33). We leave the details of this to the reader.

7. Conclusions

We have shown that by using appropriately defined projection problems it is possible to generate convex quadratic and convex polynomial cuts for MICP. These are first such results in the context of 0–1 programs. With limited computational effort, we have not found practical examples where these cuts benefit the branch-and-cut procedure, therefore, our results remain theoretical. However, given the nature of results, we think that such cuts could be valuable, particularly in the context of general 0–1 convex programs. The main difficulty in testing our ideas is that the existing state of the art optimization software available (for example, from NEOS server www.ece.nwu.edu/OTC) is unable to solve the cut generation sub-problems. We also tried a commercially available software package, and it was also unable to produce satisfactory results. Typically the solvers either produce infeasible solutions and claim them to be optimum, or the implemented optimization algorithms fail. As a result we are unable to generate the correct subgradient information needed to write the cut even in the quadratic case.

In summary, we conclude by saying that the use of nonlinear cuts provides a new possible strategy for 0–1 convex programs. This paper has presented a basic technique for how these cuts can be generated, however, computational testing is needed to know the practical viability of our ideas.

Acknowledgement

We would like to thank anonymous referees who refereed this paper for *Operations Research Letters* and *Journal of Global Optimization*. The suggestions from these referees helped in improving this paper. In particular, we thank one of the referees who asked us to develop results for generating convex polynomial cuts.

References

1. Balas E. (1985), Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM J. Alg. Disc. Math.*, 6(3): 466–486.
2. Balas E., Ceria S. and Cornuéjols, G. (1993), A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(3): 295–324.
3. Balas E., Ceria S., and Cornuéjols, G. (1996), Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42: 1229–1246.
4. Balas E., Ceria S., Cornuéjols G. and Pataki, G. (1994), Polyhedral methods for the maximum clique problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*.
5. Bazaraa M.S, Sherali H.D. and C.M. Shetty (1993), *Nonlinear Programming: Theory and Algorithms*, 2nd edition, John Wiley & Sons, New York.
6. Fujie T. and Kojima M. (1997), Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 4: 367–380.
7. Goemans M. and Williamson D. (1995), Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J ACM*, 42(6): 1115–1145.
8. Golub, G.H. and Van Loan, C.F. (1983), *Matrix Computations*, (1st Edition). The Johns Hopkins University Press, Baltimore, MD.
9. Karger D., Motwani R. and Sudan M. (1998), Approximate graph coloring by semidefinite programming. *J ACM*, 45(2): 246–265.
10. Lovász L. and Schrijver A. (1991), Cones of matrices and set-functions and 0–1 optimization. *SIAM Journal on Optimization*, 1(2): 166–190.
11. Poljak S. and Rendl, F. (1995), Nonpolyhedral relaxations of graph bisection problems. *SIAM Journal on Optimization*, 5: 467–487.
12. Sherali, H.D. and Adams, W.P. (1990), A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3): 411–430.
13. Sherali, H.D. and Adams, W.P. (1994), A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, 52: 83–106.
14. Stoer, J. and Bulirsch, R. (1983), *Introduction to Numerical Analysis*. Springer, Berlin.
15. Stubbs, R.A. and Mehrotra, S. (1999), A branch-and-cut method for 0–1 mixed convex programming. *Mathematical Programming, Series A*, 86: 515–532.
16. Vandenberghe, L. and Boyd, S. (1996), Semidefinite programming. *Siam Review*, 38(1): 49–95.