数理解析研究所講究録
1004 巻 1997 年 168-189

A Polyhedral Approach for Nonconvex Quadratic Programming Problems with Box Constraints

東京工業大学 矢島 安敏 (Yasutoshi Yajima)
東京工業大学 藤江 哲也 (Tetsuya Fujie)

1 Introduction

We consider the following nonconvex quadratic programming problem with box constraints:

$$(1.1)\qquad (P)\left|\begin{array}{ll}\text{Minimize} & f(x)=x^TQx+c^Tx\\ \text{Subject to} & 0\leq x_i\leq 1,\ i=1,2,\ldots,n,\end{array}\right.$$

where $x^T=(x_1,x_2,\ldots,x_n)$ is a variable vector of size n, Q is a symmetric $n\times n$ matrix, and c is a vector of size n. If f is a convex function, problem (P) is an easy convex minimization problem and a lot of standard convex nonlinear algorithms can be applied for solving (P). Also, if f is a concave function, i.e., matrix Q is negative semidefinite, it is well known that problem (P) has a globally optimal solution at an extreme point of box constraints. Problem (P) is, therefore, equivalent to the following quadratic zero-one programming:

$$(1.2)\qquad (IQ)\left|\begin{array}{ll}\text{Minimize} & x^TQx+c^Tx\\ \text{Subject to} & x_i\in\{0,1\},\ i=1,2,\ldots,n.\end{array}\right.$$

Many methods have been proposed for solving (IQ). Among them are branch and bound algorithms [12, 15], linear relaxation methods and/or cutting plane methods for solving equivalent linear zero-one integer programs or max-cut problems [2, 14, 4], eigenvalue methods [7, 16], and semidefinite relaxation methods [11]. In this article, we consider the problem (P) when Q is indefinite. It seems that the problem is one of the simplest but the toughest global optimization problems.

Only a few methods have been proposed. Coleman and Hulbert [6] propose an efficient algorithm for obtaining a local optimal solution of the problems. Hansen et al.[9] propose necessary conditions for optimality for (P). They also propose some kind of active set strategy and solve the problem optimally by branch and bound methods.

We will propose a polyhedral approach which is closely related to the *linearization technique* proposed by Padberg [14] for solving (IQ). He linearizes the quadratic terms x_ix_j by introducing new variables

$$(1.3)\qquad y_{ij}=x_ix_j,\quad \text{for all}\ 1\leq i<i\leq n.$$

It is easy to verify that problem (IQ) is equivalently reduced into the following linear zero-one integer programming problem:

$$
(1.4)\quad \left|\begin{array}{ll} \text{Minimize} & \sum_{i<j} Q_{ij}y_{ij} + c^T x \\ \text{Subject to} & y_{ij} \le x_i,\ \ y_{ij} \le x_j,\ \ x_i + x_j - 1 \le y_{ij}, \\ & x_i \in \{0,1\},\ i = 1,2,\ldots,n, \\ & y_{ij} \in \{0,1\},\ \ \text{for all}\ 1 \le i < j \le n, \end{array}\right.
$$

where Q_{ij} is a (i,j)–element of matrix Q. We note that $x_i^2 = x_i$ if $x_i \in \{0,1\}$. Therefore, without loss of generalities, we can replace the quadratic terms x_i^2 to x_i for all $i = 1,2,\ldots,n$, and assume Q be a zero diagonal matrix.

He considers the convex hull of zero-one vectors satisfying the constraints of (1.4). He calls it Boolean quadric polytope (BQP) and proposes three families of facets, named, the clique–inequality, the cut–inequality and the generalized cut inequality. Also, Simone [22] shows that the BQP is the image of the cut polytope (CP) defined by [3], and that the polyhedral structure of CP can be easily reduced to those of BQP. See also [4, 5] for further details.

In this article, we will apply the same linearizing technique to the case when x_i's are continuous between 0 and 1. To linearize the problem, we will also introduce new variables

$$
(1.5)\quad y_{ij} = x_i x_j, \quad \text{for all}\ \ 1 \le i \le j \le n,
$$

and consider set QP and its convex hull QP^C defined below:

$$
(1.6)\quad QP = \{(x,y) \in R^n \times R^{\frac{n(n+1)}{2}}\ |\ 0 \le x_i \le 1,\ y_{ij} = x_i x_j \ \text{for all}\ \ 1 \le i \le j \le n\},
$$

$$
(1.7)\quad QP^C = conv\{\ QP\ \}.
$$

Here, the difference between QP^C and BQP must be noted. Firstly, QP has additional variables y_{ii} $(i = 1,2,\ldots,n)$ which correspond to x_i^2. Since x_i takes an arbitrary value between 0 and 1, x_i^2 can not be replaced by x_i. Secondly, QP^C is not a polyhedral set any longer. Vertices of QP^C consists of not only 0-1 vertices but also non integer vertices. However, ignoring these additional variables y_{ii}, any 0-1 vertices of QP^C are identical to those of BQP. QP^C can be viewed as a continuous generalization of BQP.

In a series of articles [18, 19, 21], Sherali et al. developed the same linearization method for solving general nonconvex quadratic programming problem. Their idea can be viewed as a technique for approximating QP. They take all possible pairwise product of the original inequalities

$$
(1.8)\quad \begin{array}{l} x_i \ge 0,\ \ i = 1,2,\ldots,n \\ -x_i \ge -1,\ \ i = 1,2,\ldots,n, \end{array}
$$

and generate the following linear inequalities

$$
\begin{array}{lrcl} (1.9) & x_i + x_j - 1 & \le & y_{ij}, \\ (1.10) & 0 & \le & y_{ij}, \\ (1.11) & y_{ij} & \le & x_i, \\ (1.12) & y_{ij} & \le & x_j, \end{array}
$$

by replacing quadratic term x_ix_j with y_{ij} for all $1 \leq i \leq j \leq n$. Let us define

$$QP^0 = \{(x,y) \mid (x,y) \text{ satisfies } (1.8) \sim (1.12)\ \},$$

and consider the following linear programming problem:

$$(1.13)\quad \text{Minimize} \left\{ 2\sum_{i<j} Q_{ij}y_{ij} + \sum_{i=1}^{n} Q_{ii}y_{ii} + c^Tx \mid (x,y) \in QP^0 \right\},$$

where Q_{ij} is a (i,j) element of matrix Q. Since $QP^0 \supseteq QP$ linear programming problem (1.13) gives a lower bound for (P).

Recently, some authors[8, 17] propose semidefinite relaxations for general nonconvex quadratic problems. Let us denote the positive semidefiniteness of a matrix A by $A \succeq 0$. They approximate (1.5) by the positive semidefinite condition $Y - xx^T \succeq 0$, or equivalently,

$$(1.14)\quad \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succeq 0,$$

where Y is a symmetric matrix with element y_{ij}. Therefore, a lower bound for (P) is obtained by solving the semidefinite programming problem:

$$(1.15)\quad \text{Minimize} \left\{ 2\sum_{i<j} Q_{ij}y_{ij} + \sum_{i=1}^{n} Q_{ii}y_{ii} + c^Tx \mid (x,y) \in QP^{SDP} \right\},$$

where

$$(1.16)\quad QP^{SDP} = QP^0 \cap \{(x,y) \mid (x,y) \text{ satisfies } (1.14)\ \}.$$

Many algorithms [1, 10, 13, etc.] have been proposed for solving (1.15).

In this article, we will propose several classes of valid linear inequalities of QP. It will be shown that a polytope defined by our inequalities is tighter than that defined by (1.9) $\sim$ (1.12). We also propose cutting plane algorithms employing these inequalities as cutting planes. The article is organized as follows. In Section 2, we introduce notation and some basic results. Sections 3 is devoted to propose several classes of valid inequalities of QP. We also show that these inequalities are closely related to the facets of BQP. In Section 4, we describe cutting plane algorithms for solving (P). We also describe heuristic procedures for generating cutting planes. Results of preliminary computational experiments show that our inequalities generate a polytope which is a fairly nice approximation of QP.

2 Basic Results and Notation

Let us consider the following indefinite quadratic programming problem:

$$(2.17)\quad (P)\left|\begin{array}{ll}\text{Minimize} & f(x)=x^TQx+c^Tx\\ \text{Subject to} & 0\le x_i\le 1,\ i=1,2,\ldots,n,\end{array}\right.$$

and its associated convex programming problem with linear objective function:

$$(2.18)\quad (P_L)\left|\begin{array}{ll}\text{Minimize} & f_L(x,y)=2\sum_{i<j}Q_{ij}y_{ij}+\sum_{i=1}^{n}Q_{ii}y_{ii}+c^Tx\\ \text{Subject to} & (x,y)\in QP^C.\end{array}\right.$$

Theorem 2.1 *Problem (P_L) has an optimal solution (x^*,y^*) such that x^* is an optimal solution of (P).*

Proof It is obvious that any vertex of QP^C satisfies (1.5), and that problem (P_L) has an optimal solution among the vertices of QP^C. Then, finding an optimal vertex of (P_L) amounts to solve the problem (P). □

In order to propose valid inequalities for QP^C, we will use the following notation. Let N be a set of indices $N=\{1,2,\ldots,n\}$. For any $S\subseteq N$ we define polynomials

$$V_S(x)=\sum_{i\in S}x_i,$$

$$V_S(x^2)=\sum_{i\in S}x_i^2,$$

and

$$E_S(y)=\sum_{i,j\in S,\ i<j}y_{ij}.$$

Moreover, for any $S,T\subseteq N$ such that $S\cap T=\emptyset$ let us denote

$$(S,T)=\{(i,j)\mid i<j\ \text{ and either } i\in S,\ j\in T,\ \text{ or } i\in T,\ j\in S\}.$$

We define

$$E_{S,T}(y)=\sum_{(i,j)\in(S,T)}y_{ij}.$$

We note that if $(x,y)\in QP$ then $E_{S,T}(y)=V_S(x)V_T(x)$.

The following lemma plays an important role in this article.

Lemma 2.2 *Let S be a subset of N and t be a real number between 0 and $|S|$. Then*

$$(2.19)\quad -(\alpha+\beta^2)\le\min\{-V_S(x^2)\mid V_S(x)=t,\ \ 0\le x_i\le 1,\ i\in S\},$$

where α and β are arbitrary nonnegative integer and real number, respectively, such that

$$\alpha+\beta=t.$$

Moreover, equality in (2.19) is established when $\alpha=\lfloor t\rfloor$.

Proof We note that (2.19) is a concave minimization problem and has an optimal solution among the vertices, whose objective values are equal to $-(I+r^2)$, where

$$I = \lfloor t \rfloor, \quad \text{and} \quad r = t - I.$$

Also, it is obvious to see

$$-(I+r^2) \geq -(\alpha+\beta^2)$$

for any nonnegative integer α and real β such that $\alpha+\beta = t$. □

3 Cutting Planes

Now, we are ready to propose several classes of valid inequalities for QP.

Theorem 3.3 (Clique Type Inequality) *For any $S \subseteq N$ and any integer α, $0 \leq \alpha \leq |S|$, the following inequality*

$$\alpha V_S(x) - E_S(y) \leq \frac{\alpha(\alpha+1)}{2} \tag{3.20}$$

is valid for QP.

Proof For any $(x, y) \in QP$, let $t = V_S(x)$. Then we have

$$\begin{aligned} \{V_S(x)\}^2 &= t^2, \\ V_S(x^2) + 2E_S(y) &= t^2, \\ 2E_S(y) &= t^2 - V_S(x^2). \end{aligned}$$

By Lemma 2.2,for any nonnegative integer α and real β such that $\alpha + \beta = t$, $2E_S(y)$ is bounded below by

$$2E_S(y) = t^2 - V_S(x^2) \geq t^2 - (\alpha+\beta^2), \tag{3.21}$$

or equivalently, for any integer α such that $0 \leq \alpha \leq |S|$, we obtain

$$\begin{aligned} 2E_S(y) &\geq (\alpha+\beta)^2 - (\alpha+\beta^2), \\ &= 2\alpha(\alpha+\beta) - \alpha(\alpha+1), \\ &= 2\alpha V_S(x) - \alpha(\alpha+1), \end{aligned}$$

which completes the proof. □

In [14], Padberg shows that for any $S \subseteq N$ with $|S| \leq 3$ and any integer α, $1 \leq \alpha \leq |S| - 2$, inequalities (3.20) define facets of BQP. The idea of this proof can be applied for BQP in the following way. Let

$$\text{(3.22)}\quad \overline{QP} = \{(x,\overline{y}) \in R^n \times R^{n(n-1)/2} \,|\, 0 \le x_i \le 1,\ y_{ij} = x_i x_j \text{ for all } 1 \le i < j \le n\}.$$

We note that vector $\overline{y}$ does not have elements such that y_{ii}, $(i = 1, \ldots, n)$ and that BQP is contained in $\overline{QP}$. It is straight forward to see that the proof of Theorem 3.3 holds true for $\overline{QP}$ as well as for QP. It should be emphasized that inequalities (3.20) are not only valid for the convex hull of $\overline{QP}$ but also facets for BQP.

More generally, we have the following theorem:

Theorem 3.4 (Cut Type Inequality) *For any $S, T \subseteq N$ and integer α, the following inequality*

$$\text{(3.23)}\quad E_S(y) + E_T(y) - E_{S,T}(y) - \alpha V_S(x) + (\alpha+1)V_T(x) + \frac{\alpha(\alpha+1)}{2} \ge 0$$

is valid for QP.

We note that inequality (3.23) includes (3.20) as a special case when $T = \emptyset$.

Proof For any $(x, y) \in QP$, let

$$I_S = \lfloor V_S(x) \rfloor, \quad r_S = V_S(x) - I_S$$

and

$$I_T = \lfloor V_T(x) \rfloor, \quad r_T = V_T(x) - I_T.$$

From (3.21), we have

$$2E_S(y) = \{V_S(x)\}^2 - V_S^2(x) \ge (I_S + r_S)^2 - (I_S + r_S^2),$$

and

$$2E_T(y) = \{V_T(x)\}^2 - V_T^2(x) \ge (I_T + r_T)^2 - (I_T + r_T^2).$$

Then, we have the following inequality.

$$\begin{aligned} & E_S(y) + E_T(y) - E_{S,T}(y) - \alpha V_S(x) + (\alpha+1)V_T(x) + \frac{\alpha(\alpha+1)}{2} \\ &= E_S(y) + E_T(y) - V_S(x)V_T(x) - \alpha V_S(x) + (\alpha+1)V_T(x) + \frac{\alpha(\alpha+1)}{2} \\ &\ge \frac{1}{2}\{(I_S - I_T - \alpha)(I_S - I_T - \alpha - 1 + 2r_S - 2r_T) + 2r_T(1 - r_S)\}. \end{aligned}$$

Let $I = I_S - I_T - \alpha$ and $\theta = -1 + 2r_S - 2r_T$, we define

$$F(I) = I(I + \theta) + 2r_T(1 - r_S).$$

Since $0 \le r_T, r_S < 1$, we have $-3 < \theta < 1$ and $r_T(1 - r_S) \ge 0$. Then, it is easy to see that for any integer I such that $I \le 0$, or $I \ge 3$

$F(I) \geq 0.$

When $I = 1$ we have

$$F(1) = 2r_S(1 - r_T) \geq 0$$

and also when $I = 2$

$$F(2) = 2(1 - r_T)(1 + r_S) + 2r_S \geq 0.$$

$F(I)$ is, therefore, nonnegative for any integer I. We have

$$E_S(y) + E_T(y) - V_S(x)V_T(x) - \alpha V_S(x) + (\alpha + 1)V_T(x) + \frac{\alpha(\alpha+1)}{2} \geq 0,$$

and the proof is complete. □

In [14], Padberg shows that for any $S, T \subseteq N$ such that $S \cap T = \emptyset$, $|S| \geq 1$, and $|T| \geq 2$,inequalities (3.23) define facets of BQP when $\alpha = |T| - |S|$. Also in [4, 20], inequalities (3.23) have been introduced by considering the product of two linear functions below:

(3.24) $l(x) = (V_S(x) - V_T(x) - \alpha)(V_S(x) - V_T(x) - \alpha - 1),$

where α is an arbitrary integer. The nonnegativity of $l(x)$ is obvious if x is integer. Expanding (3.24) and replacing $x_i x_j$ to y_{ij} and x_i^2 to x_i, we can obtain (3.23), which are considered as valid inequalities for BQP. In our proof, however, the same inequalities can be obtained without using 0-1 properties.

Finally, we will introduce some classes of inequalities which are obtained easily.

Theorem 3.5 *For any $i \in N$ and real r, the following inequality*

(3.25) $y_{ii} - 2rx_i + r^2 \geq 0$

is valid for QP. Moreover, for any $i, j \in N$ such that $i < j$, and any $r_1, r_2 \in R$, the following inequality

(3.26) $r_1^2 y_{ii} + r_2^2 y_{jj} - 2r_1 r_2 y_{ij} \geq 0$

is valid for QP.

Proof For any x_i and real $r \in R$, the following inequality

$$(x_i - r)^2 \geq 0$$

holds. Expanding the left–hand–side and replacing x_i^2 to y_{ii}, we obtain (3.25), which holds true for any $(x, y) \in QP$. It is easy to show inequality (3.26) in the same way.

□

Inequalities (3.25) and (3.26) are closely related to positive semidefinite cone (1.14). Let

$$X = \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix},$$

and let us consider the determinant of 2×2 principal minors which consist of the first and the ith row of X. We have the following convex sets

$$\text{(3.27)} \quad \{(x, y) \,|\, y_{ii} - x_i^2 \geq 0\}, \quad i = 1, 2, \ldots, n,$$

which include QP^{SDP}. We see that for any $r \in R$,

$$y_{ii} - 2rx_i + r^2 = 0$$

defines a supporting hyperplane of (3.27) at $x_i = r$, $y_{ii} = r^2$, and that this hyperplane generates inequality (3.25). Also, the determinant of 2×2 principal minors not containing the first row of X define the following convex sets

$$\text{(3.28)} \quad \{(x, y) \,|\, y_{ii}y_{jj} - y_{ij}^2 \geq 0\}, \quad \text{for all } i < j.$$

For any $r_1, r_2 \in R$,

$$r_1^2 y_{ii} + r_2^2 y_{jj} - 2r_1 r_2 y_{ij} = 0$$

defines a supporting hyperplane of (3.28) at $y_{ii} = r_2^2$, $y_{jj} = r_1^2$, $y_{ij} = r_1 r_2$, and generates inequality (3.26).

Moreover, let $(\overline{x}, \overline{y})$ be a given vector which does not satisfy the positive semi–definite condition (1.14), and let $\overline{X}$ be an $n + 1$ dimensional square matrix defined below:

$$\text{(3.29)} \quad \overline{X} = \begin{bmatrix} 1 & \overline{x}^T \\ \overline{x} & \overline{Y} \end{bmatrix},$$

where $\overline{Y}$ is a symmetric matrix with element $\overline{y_{ij}}$. The following lemma has been shown.

Lemma 3.6 *If $(\overline{x}, \overline{y}) \notin QP^{SDP}$, then the following inequality separates $(\overline{x}, \overline{y})$ from QP^{SDP}.*

$$\text{(3.30)} \quad v^T \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} v \geq 0,$$

where v is an eigenvector associated with a negative eigenvalue of $\overline{X}$.

Proof See [17]. □

4 Algorithms

In this section, we describe our algorithms and the results of our numerical experiments. Firstly, we show the details of the cutting plane algorithm. Section 4.1 is devoted to describe procedures for generating the violated inequalities. We will also describe strategies for selecting, adding and dropping these violated inequalities.

4.1 Generating Cutting Planes

For simplicity, in the rest of this section, let us denote the clique and the cut type inequality by

$$l_S^{cl}(x,y\,;\,\alpha) \equiv \alpha V_S(x) - E_S(y) - \frac{\alpha(\alpha+1)}{2} \leq 0$$

and

$$\begin{aligned} & l_{S,T}^{ct}(x,y\,;\,\alpha) \\ \equiv & \; -E_S(y) - E_T(y) + E_{S,T}(y) + \alpha V_S(x) - (\alpha+1)V_T(x) - \frac{\alpha(\alpha+1)}{2} \leq 0, \end{aligned}$$

respectively. Associated with these cuts, let us define the following quadratic functions:

(4.31) $$q_S^{cl}(x,y) = \frac{1}{2}\{V_S(x)\}\{V_S(x)-1\} - E_S(y)$$

and

(4.32) $$q_{S,T}^{ct}(x,y) = q_S^{cl}(x,y) + q_T^{cl}(x,y) + E_{S,T}(y) - V_S(x)V_T(x),$$

which can be considered as the lower bounds for $l_S^{cl}(x,y\,;\,\alpha)$ and $l_{S,T}^{ct}(x,y\,;\,\alpha)$, respectively, in the following sense.

Lemma 4.7 *For any x, y and $S \subseteq N$, if $\alpha = \lfloor V_S(x) \rfloor$, then*

(4.33) $$q_S^{cl}(x,y) \leq l_S^{cl}(x,y\,;\,\alpha).$$

Also, for any x, y and $S, T \subseteq N$ $S \cap T = \emptyset$, if $\alpha = \lfloor V_S(x) - V_T(x) \rfloor$, then

(4.34) $$q_{S,T}^{ct}(x,y) \leq l_{S,T}^{ct}(x,y\,;\,\alpha).$$

Proof It is obvious to see that

$$q_S^{cl}(x,y) - l_S^{cl}(x,y\,;\,\alpha) = \frac{1}{2}\{V_S(x) - \alpha\}\{V_S(x) - \alpha - 1\} \leq 0$$

if $\alpha = \lfloor V_S(x) \rfloor$.

Also

$$q_{S,T}^{ct}(x,y) - l_{S,T}^{ct}(x,y\,;\,\alpha) = \frac{1}{2}\{V_S(x) - V_T(x) - \alpha\}\{V_S(x) - V_T(x) - \alpha - 1\} \leq 0$$

if $\alpha = \lfloor V_S(x) - V_T(x) \rfloor$.

□

Lemma 4.7 gives sufficient conditions for generating the cutting planes. Therefore, given a vector $(\overline{x}, \overline{y})$, if we find $S \subseteq N$ such that

$$q_S^{cl}(\overline{x}, \overline{y}) > 0,$$

then we can generate the clique inequality

$$l_S^{cl}(x,y\,;\,\lfloor V_S(\overline{x}) \rfloor) \leq 0$$

which cuts off $(\overline{x}, \overline{y})$.

Moreover, suppose that we find $S, T \subseteq N$ $S \cap T = \emptyset$ such that

$$q_S^{cl}(\overline{x}, \overline{y}) + q_T^{cl}(\overline{x}, \overline{y}) > 0.$$

Now, let us consider the clique inequality generated by the union of S and T. We see that $q_{S\cup T}^{cl}(x,y)$ can be calculated in the following way:

$$\begin{aligned} q_{S\cup T}^{cl}(x,y) &= \frac{1}{2}\{V_{S\cup T}(x)\}\{V_{S\cup T}(x) - 1\} - E_{S\cup T}(y) \\ &= q_S^{cl}(x,y) + q_T^{cl}(x,y) + V_S(x)V_T(x) - E_{S,T}(y). \end{aligned}$$

Then, we can obtain at least either the clique inequality

$$l_{S\cup T}^{cl}(x,y\,;\,\lfloor V_{S\cup T}(\overline{x}) \rfloor) \leq 0$$

or the cut inequality

$$l_{S,T}^{ct}(x,y\,;\,\lfloor V_S(\overline{x}) - V_T(\overline{x}) \rfloor) \leq 0,$$

which cut off the vector $(\overline{x}, \overline{y})$.

In our cutting plane algorithms, we solve (1.13) as the initial relaxation problem and repeatedly solve LPs by adding violated linear inequalities until a termination criterion holds or no cutting planes are found. We adopt the following relative error criterion

(4.35) $\quad f(x^*) - \epsilon|f(x^*)| \leq f_L(\overline{x}, \overline{y}),$

where $(\overline{x}, \overline{y})$ is an optimal solution of the current LP, x^* is a feasible solution of QP, and $0 < \epsilon < 1$. We note that since $\overline{x}$ is a feasible solution of (P), $f_L(\overline{x}, \overline{y})$ and $f(x^*)$ give a lower and an upper bounds of (P), respectively.

In our algorithm, we use the following procedures to generate violating inequalities for a given point $(\overline{x}, \overline{y})$.

Procedure **TRI**
First enumerate all triples $i, j, k \in N$, and generate violating clique type inequalities (3.20) with $|S| = 3$ and $\alpha = 1$, then enumerate all triples again and generate violating cut type inequalities (3.23) with $|S| = 1$, $|T| = 2$ and $\alpha = 1$.

We note that it requires $O(n^3)$ computational time to perform this procedure and that for each triple i, j, k, we can generate one clique type and three cut type inequalities.

Procedure **DIAG**

1. *For all $i \in N$, if $\overline{y_{ii}} < \overline{x_i}^2$ then generate inequalities (3.25) by setting $r = \overline{x_i}$.*
2. *For all pairs $i, j \in N$, if $\overline{y_{ii}}\,\overline{y_{jj}} < \overline{y_{ij}}^2$ then generate inequalities (3.26) by setting $r_1^2 = \overline{y_{jj}}$, $r_2^2 = \overline{y_{ii}}$.*
3. *If some inequalities have been generated, then terminate.*
4. *Let $\overline{X}$ be a matrix defined in Lemma 3.6. For all eigenvectors v which are associated with negative eigenvalues of $\overline{X}$, generate inequalities (3.30).*

We call the procedure **DIAG** since (3.25), (3.26) and (3.30) are the only inequalities that contain "diagonal" variables y_{ii}.

Procedure **HYP**

1. *Enumerate all subsets S of N such that $|S| \leq 3$ and $q_S^{cl}(\overline{x}, \overline{y}) > 0$, and let $\mathcal{S}$ be a family of these subsets.*
2. *For all disjoint subset $S, T \in \mathcal{S}$, generate inequalities*

$$l^{cl}_{S\cup T}(x, y\,;\ \lfloor V_{S\cup T}(\overline{x})\rfloor) \leq 0$$

and/or

$$l^{ct}_{S,T}(x, y\,;\ \lfloor V_S(\overline{x}) - V_T(\overline{x})\rfloor) \leq 0.$$

Procedure **HEU**
The procedure consists of four subprocedures **HEU1**,..., **HEU4**.

HEU1 *Execute the following n times.*

- **step 1** *Generate a subset* $S \subset N$ *such that* $|S| = 3$ *randomly.*
- **step 2** *For all* $i \in N \setminus S$ *calculate* $s_i := \overline{x}_i - \sum_{j\in S} \overline{y}_{ij}$. *Let* $s^* := \max_{i\in N\setminus S} s_i$ *and* i^* *be the corresponding index.*
- **step 3** *If* $g^* \leq 0$, *quit the subprocedure. Otherwise let* $S := S \cup \{i^*\}$.
- **step 4** *Generate (3.20) with* S *and* $\alpha = \lfloor V_S(\overline{x}) \rfloor$. *If* $S \neq N$, *go to step 2.*

HEU2 *The same as* **HEU1** *except that we let* $s_i := \overline{x}(\overline{x}-1) - 2\sum_{j\in S}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j)$ *in step 2.*

HEU3 *Execute the following n times.*

- **step 1** *Generate subsets* $S, T \subset N$ *such that* $|S| = 1, |T| = 2$ *and* $S \cap T = \emptyset$ *randomly.*
- **step 2** *For all* $i \in N \setminus (S \cup T)$ *calculate* $s_i := -\overline{x}_i - \sum_{j\in S} \overline{y}_{ij} + \sum_{j\in T} \overline{y}_{ij}$ *and* $t_i := \sum_{j\in S} \overline{y}_{ij} - \sum_{j\in T} \overline{y}_{ij}$. *Let* $s^* := \max_{i\in N\setminus(S\cup T)} s_i$ *and* i^*_S *be the corresponding index. Similarly, let* $t^* := \max_{i\in N\setminus(S\cup T)} t_i$ *and* i^*_T *be the corresponding index. If* $s^* > t^*$, *go to step 3. Otherwise go to step 4.*
- **step 3** *If* $s^* \leq 0$, *quit the subprocedure. Otherwise let* $S := S \cup \{i^*_S\}$.
- **step 4** *If* $t^* \leq 0$, *quit the subprocedure. Otherwise let* $T := T \cup \{i^*_T\}$.
- **step 5** *Generate (3.23) with* S, T *and* $\alpha = \lfloor V_S(\overline{x}) - V_T(\overline{x}) \rfloor$. *If* $S \cup T \neq N$, *go to step 2.*

HEU4 *The same as* **HEU3** *except that we let* $s_i := \overline{x}(\overline{x}-1) - 2\sum_{j\in S}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j) + 2\sum_{j\in T}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j)$ *and* $t_i := \overline{x}(\overline{x}-1) + 2\sum_{j\in S}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j) - 2\sum_{j\in T}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j)$ *in step 2.*

In **HEU1** (resp. **HEU2**), we maximize $l^{cl}_S(\overline{x}, \overline{y}; 1)$ (resp. $q^{cl}_S(\overline{x}, \overline{y})$) increasing S one by one. In **HEU3** (resp. **HEU4**), we maximize $l^{cl}_{S,T}(\overline{x}, \overline{y}; -1)$ (resp. $q^{cl}_{S,T}(\overline{x}, \overline{y})$) increasing $S \cup T$ one by one.

4.2 Computational Experience

In this subsection, we show our computational experiences of the cutting plane methods and the branch and bound methods.

Test problems are generated as follows. The coefficients $Q_{ij} (1 \leq i \leq j \leq n)$ and $c_i (1 \leq i \leq n)$ of the objective function in (1.1) are integers assigned randomly between -100 and 100 with density d. We generate ten problems for each n and d. Table 4.1

Table 4.1: Sign of the eigenvalues of Q

	$n=20$			$n=30$			$n=40$			$n=50$		
d	pos.	neg.	zero	pos.	neg.	zero	pos.	neg.	zero	pos.	neg.	zero
0.1	7.9	8.0	4.1	13.5	13.6	2.9	19.4	18.5	2.1	25.0	25.0	0.0
0.2	9.5	10.1	0.4	15.0	15.0	0.0	19.9	20.1	0.0			
0.3	9.7	10.3	0.0	15.0	15.0	0.0	19.6	20.4	0.0			
0.4	10.0	10.0	0.0	14.9	15.1	0.0	19.9	20.1	0.0			
0.5	10.0	10.0	0.0	14.5	15.5	0.0	19.8	20.2	0.0			
0.6	9.8	10.2	0.0	14.6	15.4	0.0						
0.7	10.1	9.9	0.0	14.5	15.5	0.0						
0.8	9.9	10.1	0.0	15.0	15.0	0.0						
0.9	10.0	10.0	0.0	14.8	15.2	0.0						
1.0	9.9	10.1	0.0	14.9	15.1	0.0						

displays the average number of positive (pos.), negative (neg.) and zero eigenvalues of Q. We can see that most of the randomly generated matrices Q are full rank and have almost the same number of positive and negative eigenvalues. All problems are solved on a SUN SparcStation 1 and we use CPLEX 2.0 callable library as an LP solver. Throughout the rest of the article we set $\epsilon = 0.01$.

We consider the following four strategies (`cut0`, `cut1`, `cut2`, `cut3`) for the cutting plane generating phase, which is denoted by CUT phase for short.

- `cut0` : generate no cutting planes, that is, just solve initial LP(1.13).
- `cut1` : execute **TRI**
- `cut2` : first execute **TRI**. If some cutting planes are found then quit the phase, otherwise execute **DIAG**.
- `cut3` : first execute **TRI**. If some cutting planes are found then quit the phase, otherwise execute **DIAG**. If some cutting planes are found then quit the phase, otherwise execute **HEU**.

We detect a violating inequality as a cutting plane if the distance between $(\overline{x}, \overline{y})$ and the inequality is no less than δ. We first set $\delta = 0.1$ and dynamically change δ from one phase to the next. In our experiments, we terminate the cutting plane algorithm if no cutting plane is found with $\delta = 10^{-5}$ or an ϵ optimal solution is found. The total number of inequalities added to the initial LP is limited to 1500. We provide a routine for deleting inequalities whose slacks are greater than 0.01.

Results of the cutting plane method are given in Tables 4.2 $\sim$ 4.4. In Tables 4.2 and 4.3, m denotes the number of problems which are solved to ϵ optimality, and Ave. and Max. denote the average and the maximum cpu time in seconds, respectively. In Table 4.4, the average and the maximum number of the total generated cutting planes are shown.

Table 4.2: The number of solved problems (m) and cpu time in seconds (1)

	$n = 20$											
	cut0			cut1			cut2			cut3		
		TIME			TIME			TIME			TIME	
d	m	ave.	max.	m	ave.	max.	m	ave.	max.	m	ave.	max.
0.1	8	2.9	4.0	8	3.1	4.6	10	3.4	6.2	10	3.3	6.2
0.2	5	4.9	5.9	6	7.0	20.1	10	9.3	30.4	10	9.4	30.3
0.3	4	5.8	7.3	8	7.1	11.1	10	7.0	11.1	10	7.0	11.1
0.4	3	6.2	7.6	7	11.3	21.8	10	11.6	22.3	10	11.6	22.4
0.5	2	6.3	8.1	4	17.7	27.1	10	20.3	34.3	10	20.3	34.2
0.6	2	6.6	8.2	6	15.4	26.6	10	21.3	49.1	10	21.4	50.1
0.7	0	6.6	8.3	3	22.8	37.6	10	28.9	53.1	10	28.9	53.4
0.8	0	6.7	7.7	7	22.9	30.6	10	28.7	67.4	10	28.8	68.0
0.9	0	6.8	7.7	3	20.7	27.5	10	23.8	50.9	10	23.9	51.2
1.0	0	6.5	7.1	6	26.6	49.4	10	30.3	75.2	10	30.3	75.8

Table 4.3: The number of solved problems (m) and cpu time in seconds (2)

	$n = 30$											
	cut0			cut1			cut2			cut3		
		TIME			TIME			TIME			TIME	
d	m	ave.	max.	m	ave.	max.	m	ave.	max.	m	ave.	max.
0.1	8	15.4	20.5	8	15.9	21.4	10	16.2	24.4	10	16.4	26.6
0.2	5	24.3	29.3	9	26.3	33.6	10	25.7	33.0	10	25.7	33.0
0.3	1	27.2	29.1	6	139.3	279.1	10	142.4	313.8	10	142.4	312.9
0.4	0	28.1	30.6	7	181.2	247.8	10	185.1	253.5	10	185.5	255.8
0.5	0	28.4	31.6	4	199.9	247.1	10	200.8	247.5	10	200.9	247.4
0.6	0	28.4	30.5	5	225.6	398.8	10	339.2	857.8	10	330.2	861.4
0.7	0	29.5	32.7	2	259.8	500.1	10	507.3	2188.6	10	515.3	2262.3
0.8	0	30.9	33.5	5	217.5	287.3	10	254.9	488.6	10	255.6	491.1
0.9	0	31.1	32.9	7	217.9	389.7	10	223.9	447.4	10	224.0	447.6
1.0	0	31.5	34.0	4	320.8	528.5	10	380.0	752.3	10	380.9	755.4

Table 4.4: The number of generated cutting planes

	$n = 20$						$n = 30$					
	cut1		cut2		cut3		cut1		cut2		cut3	
	CUT		CUT		CUT		CUT		CUT		CUT	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	1.0	9	8.7	51	8.7	51	0.0	0	9.2	71	9.2	71
0.2	89.5	686	109.1	749	109.1	749	77.0	386	81.1	386	81.1	386
0.3	99.6	440	104.5	440	104.5	440	1964.9	3307	2000.6	3478	2000.6	3478
0.4	220.2	630	234.2	656	234.2	656	2486.8	2914	2508.1	2914	2508.1	2914
0.5	490.9	760	535.6	884	535.6	884	2519.5	2807	2574.8	2888	2574.8	2888
0.6	414.1	630	444.0	685	444.0	685	2383.5	3168	2492.4	3651	2482.1	3548
0.7	579.2	625	615.6	711	615.6	711	2086.7	3067	2395.3	5947	2405.1	6045
0.8	566.4	614	584.1	685	584.1	685	1663.1	2347	1646.3	1835	1646.3	1835
0.9	566.6	598	598.0	642	598.0	642	1559.1	1888	1553.7	1672	1553.7	1672
1.0	581.4	616	596.2	648	596.2	648	1707.9	2022	1627.6	1932	1627.6	1932

From these tables, we can see that the performance of the cutting plane methods depend on the density of the matrix Q as well as n. Note that algorithms given by Hansen et al.[9] behave similarly, although their test problems are generated in a somewhat different way from ours. When matrix Q is sparse solving (1.13) is sufficient for most of the problems to obtain an ϵ optimal solution. Whereas, it is necessary to add the elaborate cutting planes when the density increases. In our computational experiments, all test problems are solved to ϵ optimality by strategy cut2.

See Table 4.5. Here, diag shows the results of the cutting plane method which uses procedure **DIAG** alone in CUT phase. cut1 and cut2 are taken from Table 4.2. This table indicates that combination of the triangle inequalities and the eigenvalue inequalities is important to generate an ϵ optimal solution.

Next, we show results of a branch and bound method for (P) combined with the cutting plane method. The outline of the method is described as follows:

step 0 Let $\mathcal{N} := \{(0,1)\}$ and $z^* := +\infty$.

step 1 If $\mathcal{N} = \emptyset$, stop. Otherwise pick $(\ell, u) \in \mathcal{N}$: If $|\mathcal{N}| \leq n$, the way to pick is followed by a breadth first fashion. Otherwise it is followed by a depth first fashion. Let $\mathcal{N} := \mathcal{N} \setminus \{(\ell, u)\}$.

step 2 Solve a (sub)problem

$$(P_{\ell,u}) \left| \begin{array}{ll} \text{Minimize} & f(x) = x^T Q x + c^T x \\ \text{Subject to} & \ell_i \leq x_i \leq u_i, \ i = 1, 2, \ldots, n. \end{array} \right.$$

For this purpose, use a linear transformation to yield an equivalent problem

Table 4.5: The effectiveness of the triangle inequalities

	$n = 20$								
	diag			cut1			cut2		
		TIME			TIME			TIME	
d	m	ave.	max.	m	ave.	max.	m	ave.	max.
0.1	10	3.5	6.5	8	3.1	4.6	10	3.4	6.2
0.2	10	155.4	1432.6	6	7.0	20.1	10	9.3	30.4
0.3	9	70.2	259.6	8	7.1	11.1	10	7.0	11.1
0.4	9	485.5	3723.9	7	11.3	21.8	10	11.6	22.3
0.5	6	1338.3	5732.1	4	17.7	27.1	10	20.3	34.3
0.6	7	917.6	7039.7	6	15.4	26.6	10	21.3	49.1
0.7	7	1386.5	5208.3	3	22.8	37.6	10	28.9	53.1
0.8	8	748.8	2276.9	7	22.9	30.6	10	28.7	67.4
0.9	8	655.8	2865.0	3	20.7	27.5	10	23.8	50.9
1.0	8	549.9	2222.7	6	26.6	49.4	10	30.3	75.2

$$(P'_{\ell,u})\left|\begin{array}{ll}\text{Minimize} & f(x) = x'^T Q' x' + c'^T x' + d \\ \text{Subject to} & 0 \le x'_i \le 1,\ i = 1,2,\ldots,n,\end{array}\right.$$

where d is a scalar. Solve $(P'_{\ell,u})$ by the cutting plane method. Let $(\overline{x'}, \overline{y'})$ be its solution. Then use the transformation again to obtain a solution $(\overline{x}, \overline{y})$ corresponding to $(P_{\ell,u})$. If $z^* - \epsilon|z^*| \le f_L(\overline{x}, \overline{y})$, go to step 1. If $f(\overline{x}) < z^*$, let $x^* := \overline{x}$ and $z^* = f(\overline{x})$.

step 3 Select a branching variable by the following rule [21]. Calculate

$$d_1 = \min \left\{ \begin{array}{l} \min\limits_{1\le k<\ell\le n} \{\min\{0, 2q_{k\ell}(\overline{y}_{k\ell} - \overline{x}_k)\overline{x}_\ell\}\}, \\ \min\limits_{1\le k\le n} \{\min\{0, q_{k\ell}(\overline{y}_{kk} - \overline{x}_k^2)\}\} \end{array} \right\}.$$

If $d_1 = 0$, go to step 1. Otherwise let k^*, ℓ^* be indices which give d_1. If $k^* = \ell^*$ then $i^* = k^*$. Otherwise calculate

$$\begin{aligned} d_2(t) = & \sum_{j=1}^{t-1} \min\{0, 2q_{jt}(\overline{y}_{jt} - \overline{x}_j\overline{x}_t)\} + \sum_{j=t+1}^{n} \min\{0, 2q_{tj}(\overline{y}_{tj} - \overline{x}_t\overline{x}_j)\} \\ & + \min\{0, q_{tt}(\overline{y}_{tt} - \overline{x}_t^2)\} \end{aligned}$$

for $t = k^*, \ell^*$ and let $i^* := \text{argmin}\{d_2(t) \ : \ t = k^*, \ell^*\}$. Let $\overline{\ell}$ be the same vector as ℓ except that $\overline{\ell}_{i^*} = \overline{x}_{i^*}$, and $\overline{u}$ be the same vector as u except that $\overline{u}_{i^*} = \overline{x}_{i^*}$. Let $\mathcal{N} := \mathcal{N} \cup \{(\overline{\ell}, u), (\ell, \overline{u})\}$. Go to step 1.

Note that there is a choice how to solve subproblems by the cutting plane method. Since, as seen in the previous section, all test problems are solved to ϵ optimality by cut2, cut1 is used to solve subproblems to compare with the cutting plane method with cut2.

Tables 4.6 and 4.7 show the results, where BRANCH denotes the number of branching nodes and UPDATE denotes the number of updates of an incumbent solution x^* of (P).

We also implemented a local search heuristics for (P). The branch and bound method is modified as follows:

- In step 0.
  Solve (P) by a multiple start local search. Thus, let $z^* := f(x^*)$ instead of $z^* := \infty$, where x^* is a solution of the heuristics.

- In step 2.
  Apply a local search starting from $\overline{x}$ to obtain a better feasible point $\tilde{x}$. Then the last statement in step 2 becomes "If $f(\tilde{x}) < z^*$ then let $x^* := \tilde{x}$ and $z^* = f(\tilde{x})$ ".

Tables 4.8 and 4.9 show that the local search procedures reduce a considerable amount of cpu time.

To conclude, we show results of the branch and bound method with adopting heuristics and cut2 in Tables 4.10 $\sim$ 4.13. Compared with [9], our method needs more cpu time. This is because we must solve many linear programming problems. Our method, however, can be applied for more general quadratic programs which have linear and/or quadratic constraints. Sherali et al. [18, 19, 21] proposed the linear programming relaxation for these problems, and then our proposed valid inequalities can be applied as cutting planes since the valid inequalities for QP are also valid for general quadratic programs. On the other hand, since QP strictly contains the new linearized set of general quadratic programs in general, our proposed valid inequalities might not contribute as the cutting plane. Therefore new valid inequalities including the structure of linear and/or quadratic constraints should be proposed.

References

[1] Alizadeh, W. F. (1995), Interior point methods in semidefinite programming with application to combinatorial optimization, *SIAM Journal on Optimization*, **5**, 13–51.

[2] Barahona, F., Jönger, M. and Reinelt, G. (1989), "Experiments in Quadratic 0–1 Programming," *Mathematical Programming*, **44**, 127–137.

[3] Barahona, F. and Mahjoub, A. R. (1986), "On the Cut Polytope," *Mathematical Programming*, **36**, 157–173.

Table 4.6: Result of the branch and bound method (1)

	$n = 20$, cut1							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.8	6	2.9	28	1.7	6	4.2	13.2
0.2	1.4	6	218.1	1485	2.0	4	11.0	39.3
0.3	0.4	2	99.6	440	1.6	2	7.2	11.0
0.4	0.6	4	331.6	1670	2.1	6	14.8	55.4
0.5	1.4	4	1133.5	3245	2.9	5	37.8	102.9
0.6	3.2	20	890.6	4070	2.8	7	66.3	376.4
0.7	5.2	16	3213.1	8577	4.8	13	108.5	329.2
0.8	2.0	12	1616.5	7536	2.9	7	63.4	287.2
0.9	2.8	8	2042.4	5070	3.1	5	66.5	201.6
1.0	2.0	8	1695.9	5481	3.0	5	77.1	305.5

Table 4.7: Result of the branch and bound method (2)

	$n = 20$, cut1							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.4	2	0.0	0	1.3	3	16.0	21.6
0.2	0.2	2	77.9	386	1.6	3	27.3	47.6
0.3	2.0	12	2124.0	5540	3.4	7	317.6	1536.3
0.4	0.8	4	3263.0	7272	3.4	7	257.2	575.4
0.5	2.2	6	3951.9	8485	4.6	7	401.7	885.6
0.6	2.4	8	3026.3	8710	4.9	11	585.3	1930.5
0.7	5.8	18	2596.0	6069	6.5	13	1334.3	4450.6
0.8	3.8	16	1884.3	8050	4.4	10	733.2	2348.9
0.9	1.0	4	2716.3	5435	3.2	6	368.9	957.2
1.0	5.6	20	1549.8	6498	5.7	15	1819.5	5967.3

Table 4.8: Result of the branch and bound method with heuristics (1)

	$n = 20$, cut1							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.2	2	1.0	9	0.0	0	3.9	9.0
0.2	0.8	4	196.4	1425	0.2	1	11.7	38.5
0.3	0.0	0	99.5	440	0.0	0	7.4	11.1
0.4	0.2	2	317.3	1534	0.2	1	14.5	48.0
0.5	0.8	4	962.2	3262	0.4	2	29.8	108.2
0.6	0.8	4	843.2	2415	0.3	1	29.3	72.6
0.7	2.0	6	1711.6	4074	0.4	3	61.3	166.1
0.8	1.0	4	1150.2	2991	0.4	3	47.1	139.3
0.9	1.0	4	1131.6	2802	0.3	1	42.0	124.0
1.0	1.0	4	1167.0	3048	0.7	2	54.5	179.6

Table 4.9: Result of the branch and bound method with heuristics (2)

	$n = 30$, cut1							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.2	2	0.0	0	0.2	2	19.9	33.9
0.2	0.0	0	39.5	386	0.1	1	27.4	38.0
0.3	0.2	2	2419.8	7936	0.2	1	172.7	626.5
0.4	0.4	2	3104.7	6017	0.4	1	230.4	449.7
0.5	0.4	2	3191.0	5580	0.3	2	252.7	463.8
0.6	1.4	8	3224.7	9977	1.1	5	476.7	1503.1
0.7	1.6	8	2099.6	5570	0.9	3	665.3	3028.4
0.8	1.2	6	2397.4	5437	0.5	3	414.5	1193.5
0.9	0.0	0	1561.9	1888	0.1	1	214.4	365.8
1.0	1.6	8	2962.2	9047	0.9	6	972.7	3744.8

Table 4.10: Final result of the branch and bound method with heuristics (1)

	$n = 20$, cut2							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.0	0	4.8	48	0.0	0	3.3	4.8
0.2	0.0	0	102.5	748	0.1	1	7.6	21.5
0.3	0.0	0	99.5	440	0.0	0	7.4	11.1
0.4	0.0	0	230.8	656	0.1	1	12.0	23.1
0.5	0.0	0	504.8	821	0.2	1	18.0	27.6
0.6	0.0	0	431.3	683	0.1	1	16.6	28.9
0.7	0.0	0	605.6	689	0.2	2	24.6	36.1
0.8	0.0	0	577.7	653	0.2	2	24.3	37.3
0.9	0.0	0	586.3	618	0.0	0	22.3	36.2
1.0	0.0	0	593.0	645	0.5	1	27.4	51.3

Table 4.11: Final result of the branch and bound method with heuristics (2)

	$n = 30$, cut2							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.0	0	3.4	34	0.0	0	18.6	25.6
0.2	0.0	0	39.5	386	0.1	1	27.5	38.2
0.3	0.0	0	1973.7	3475	0.2	1	139.3	288.9
0.4	0.0	0	2329.4	2802	0.4	1	168.5	212.1
0.5	0.0	0	2444.8	2785	0.1	1	195.8	241.4
0.6	0.0	0	2343.0	2908	0.2	1	231.2	395.4
0.7	0.0	0	1988.3	2680	0.3	1	285.9	669.4
0.8	0.0	0	1622.1	1832	0.1	1	213.9	293.6
0.9	0.4	4	2273.0	8902	0.3	2	634.9	4527.0
1.0	0.0	0	1582.3	1743	0.1	1	296.2	499.9

Table 4.12: Final result of the branch and bound method with heuristics (3)

	$n = 40$, cut2							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.0	0	149.0	1490	0.0	0	73.6	192.4
0.2	0.0	0	3199.9	6358	0.2	1	454.2	956.0
0.3	0.0	0	5335.8	7640	0.3	1	741.4	1101.7
0.4	0.6	6	4224.7	5275	1.1	7	1242.2	6269.4
0.5	0.0	0	4531.0	5113	0.1	1	818.7	1315.7

Table 4.13: Final result of the branch and bound method with heuristics (4)

	$n = 50$, cut2							
	BRANCH		CUT		UPDATE		TIME	
d	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	0.0	0	97.3	973	0.1	1	181.8	269.7

[4] Boros, E. and Hammer, P. L. (1991), "The Max–Cut Problem and Quadratic 0–1 Optimization; Polyhedral Aspect, Relaxations and Bounds," *Annals of Operations Research*, **33**, 151–180.

[5] Boros, E. and Hammer, P. L. (1993), "Cut Polytopes, Boolean Quadric Polytopes and Nonnegative Pseudo–boolean Functions," *Mathematics of Operations Research*, **18**, 245–253.

[6] Coleman, T. F. and Hulbert, L. A. (1989), "A Direct Active Set Algorithm for Large Sparse Quadratic Programs with Simple Bounds," *Mathematical Programming*, **45**, 373–406.

[7] Delorme, C. and Poljak, S. (1993), "Laplacian Eigenvalues and the Maximun Cut Problem," *Mathematical Programming*, **62**, 557–574.

[8] Fujie, T., and Kojima, M. (1995), "Semidefinite Programming Relaxation for Nonconvex Quadratic Programs," Research Report #298, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology.

[9] Hansen, P., Jaumard, B., Ruiz, M. and Xiong, J. (1993), "Global Minimization of Indefinite Quadratic Functions Subject to Box Constraints," *Naval Research Logistics*, **40**, 373–392.

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

[11] Helmberg, C. and Rendl, F. (1995), "Solving Quadratic (0,1)-Problems by Semidefinite Programs and Cutting Planes," ZIB Preprint SC-95-35.

[12] Kalantari, B. and Bagchi, A. (1990), "An Algorithm for Quadratic Zero–One Programs," *Naval Research Logistics*, **37**, 527–538.

[13] Kojima, M., Shindoh, S. and Hara, S. (1995), Interior-point methods for the monotone semidefinite linear complementarity problems, Research Report #282, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology.

[14] Padberg, M. (1989), "The Boolean Quadratic Polytope: Some Characteristics, Facets and Relatives," *Mathematical Programming*, **45**, 139–172.

[15] Pardalos, P. M. and Rodgers, G. P. (1990), "Computational Aspects of a Branch and Bound Algorithm for Quadratic Zero–One Programming," *Computing*, **45**, 131–144.

[16] Poljak, S. and Rendl, F. (1995), "Solving the Max–Cut Problem using Eigenvalues, " *Discrete Applied Mathematics*, **62**, 249–278.

[17] Ramana, M. (1993), *An algorithmic analysis of multiquadratic and semidefinite programming problems*, PhD thesis, Johns Hopkins University, Baltimore, MD.

[18] Sherali, H. and Alameddine, A. (1990), "An Explicit Charatcterization of the Convex Envelope of a Bivariate Bilinear Function Over Special Polytopes," *Annals of Operations Research*, **25**, 197–214.

[19] Sherali, H. and Alameddine, A. (1992), "A New Reformulation–Linearization for Solving Bilinear Programming Problems," *Journal of Global Optimization*, **2**, 397–410.

[20] Sherali, H., Lee, Y. and Adams, W. P. (1995), "A Simultaneous Lifting Strategy for Identifying New Class of Facets for Boolean Quadric Polytope," *Operations Research Letters*, **17**, 19–26.

[21] Sherali, H. and Tuncbilek, C. H. (1995), "A Reformulation–Convexification Approach for Solving Nonconvex Quadratic Programming Problems," *J. of Global Optimization*, **7**, 1–31.

[22] Simone, C. D. (1989), "The Cut Polytope and The Boolean Quadric Polytope," *Discrete Mathematics*, **79**, 71–75.