Journal of Global Optimization **13:** 151–170, 1998.

A Polyhedral Approach for Nonconvex Quadratic Programming Problems with Box Constraints

YASUTOSHI YAJIMA[1] and TETSUYA FUJIE[2]
[1]*Department of Industrial Engineering and Management, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan (e-mail: yasutosi@me.titech.ac.jp)*
[2]*Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan*

(Received 17 April 1997; accepted 28 December 1997)

Abstract. We apply a linearization technique for nonconvex quadratic problems with box constraints. We show that cutting plane algorithms can be designed to solve the equivalent problems which minimize a linear function over a convex region. We propose several classes of valid inequalities of the convex region which are closely related to the Boolean quadric polytope. 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 tight approximation of the convex region.

Key words: Cutting plane method, Linearization technique, Nonconvex quadratic programs, Valid inequalities

1. Introduction

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

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

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, in the concave case, equivalent to the following quadratic zero-one program:

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

Many methods have been proposed for solving (IQ). Among them are branch and bound algorithms [14, 17], linear relaxation methods and/or cutting plane methods for solving equivalent linear zero-one integer programs or max-cut problems [2, 5, 16], eigenvalue methods [9, 19], and semidefinite relaxation methods [13].

In this article, we consider the problem (P) when Q is indefinite. From the complexity point of view, the problem is NP-hard [18]. Thus, it seems that the problem is one of the simplest but the toughest global optimization problems.

Several methods, in the indefinite case, have been proposed. Coleman and Hulbert [7] proposed an efficient algorithm for obtaining a local optimal solution of the problems. Also, several polynomial time algorithms have been proposed by Vavasis [26] and Ye [27] for obtaining approximate solutions. Hansen et al. [11] proposed necessary conditions for optimality for (P). They also proposed some kind of active set strategy and solved the problem optimally by branch and bound methods. Other algorithms can be found in the recent survey of De Angelis et al. [8] and references therein.

We will propose a polyhedral approach which is closely related to the *linearization technique* proposed by Padberg [16] for solving (IQ). He linearizes the quadratic terms $x_i x_j$ by introducing new variables

$$y_{ij} = x_i x_j, \quad \text{for all } 1 \leqslant i < j \leqslant n. \tag{3}$$

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

$$\begin{array}{ll} \text{Minimize} & 2\sum_{i<j} Q_{ij} y_{ij} + c^T x \\ \text{Subject to} & y_{ij} \leqslant x_i, \quad y_{ij} \leqslant x_j, \quad x_i + x_j - 1 \leqslant y_{ij}, \\ & x_i \in \{0,1\}, \quad i = 1, 2, \ldots, n, \\ & y_{ij} \in \{0,1\}, \quad \text{for all } 1 \leqslant i < j \leqslant n, \end{array} \tag{4}$$

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 generality, we can replace the quadratic terms x_i^2 with 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 (4), i.e.,

$$conv\{(x, y) \in R^n \times R^{\frac{n(n-1)}{2}} \mid x_i \in \{0, 1\},\ y_{ij} = x_i x_j \text{ for all } 1 \leqslant i < j \leqslant n\}. \tag{5}$$

Padberg calls this the Boolean quadric polytope (BQP) and proposes three families of facets, named the clique-inequality, the cut-inequality and the generalized cut inequality. Also, Simone [25] 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 [5, 6] 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

$$y_{ij} = x_i x_j, \quad \text{for all } 1 \leqslant i \leqslant j \leqslant n, \tag{6}$$

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

$$QP = \{(x, y) \in R^n \times R^{\frac{n(n+1)}{2}} \mid 0 \leqslant x_i \leqslant 1, \;\; y_{ij} = x_i x_j \text{ for all } \; 1 \leqslant i \leqslant j \leqslant n\}, \tag{7}$$

$$QP^C = conv\{\ QP\ \}. \tag{8}$$

Here, the difference between QP^C and BQP must be noted. Firstly, QP^C 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 consist 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 [21, 22, 24], Sherali et al. developed the same linearization method for solving general nonconvex quadratic programming problems. Their idea can be viewed as a technique for approximating QP. They take all possible pairwise products of the original inequalities

$$\begin{array}{ll} x_i \geqslant 0, & i = 1, 2, \ldots, n \\ -x_i \geqslant -1, & i = 1, 2, \ldots, n, \end{array} \tag{9}$$

and generate the following linear inequalities

$$x_i + x_j - 1 \leqslant y_{ij}, \tag{10}$$

$$0 \leqslant y_{ij}, \tag{11}$$

$$y_{ij} \leqslant x_i, \tag{12}$$

$$y_{ij} \leqslant x_j, \tag{13}$$

by replacing the quadratic term $x_i x_j$ with y_{ij} for all $1 \leqslant i \leqslant j \leqslant n$. Let us define

$$QP^0 = \{(x, y) \mid (x, y) \text{ satisfies (9) through (13)}\},$$

and consider the following linear programming problem:

$$\text{Minimize} \left\{ 2 \sum_{i<j} Q_{ij} y_{ij} + \sum_{i=1}^{n} Q_{ii} y_{ii} + c^T x \mid (x, y) \in QP^0 \right\}, \tag{14}$$

where Q_{ij} is a (i, j) element of matrix Q. Since $QP^0 \supseteq QP$, linear programming problem (14) gives a lower bound for (P).

Recently, some authors [10, 20] propose semidefinite relaxations for general nonconvex quadratic problems. Let us denote the positive semidefiniteness of a matrix A by $A \succeq 0$. They approximate (6) by the positive semidefinite condition $Y - xx^T \succeq 0$, or equivalently,

$$\begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succeq 0, \tag{15}$$

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

$$\text{Minimize} \left\{ 2\sum_{i<j} Q_{ij} y_{ij} + \sum_{i=1}^{n} Q_{ii} y_{ii} + c^T x \mid (x, y) \in QP^{SDP} \right\}, \tag{16}$$

where

$$QP^{SDP} = QP^0 \cap \{(x, y) \mid (x, y) \text{ satisfies (15)}\}. \tag{17}$$

Many algorithms [1, 12, 15, etc.] have been proposed for solving (16).

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 (9) through (13). 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. Section 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 as well as the positive semidefinite inequality (15). 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:

$$(P) \left| \begin{array}{ll} \text{Minimize} & f(x) = x^T Q x + c^T x \\ \text{Subject to} & 0 \leqslant x_i \leqslant 1,\ i = 1, 2, \dots, n \end{array} \right. \tag{18}$$

and its associated convex programming problem with linear objective function:

$$(P_L) \left| \begin{array}{ll} \text{Minimize} & f_L(x, y) = 2\sum_{i<j} Q_{ij} y_{ij} + \sum_{i=1}^{n} Q_{ii} y_{ii} + c^T x \\ \text{Subject to} & (x, y) \in QP^C. \end{array} \right. \tag{19}$$

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 (6), 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 $N = \{1, 2, \ldots, n\}$. 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

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

Then, 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*

$$-(\alpha + \beta^2) \leqslant \min\{-V_S(x^2) \mid V_S(x) = t,\ \ 0 \leqslant x_i \leqslant 1,\ i \in S\}, \tag{20}$$

where α is an arbitrary integer and β is a number such that

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

Moreover, equality in (20) is established when $\alpha = \lfloor t \rfloor$.

Proof. We note that the right-hand side of (20) 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

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

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

3. Valid inequalities

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

THEOREM 3.1 (clique type inequality). *For any $S \subseteq N$ and any integer α ($0 \leqslant \alpha \leqslant |S|$), the following inequality*

$$\alpha V_S(x) - E_S(y) \leqslant \frac{\alpha(\alpha+1)}{2} \tag{21}$$

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

$$2E_S(y) = t^2 - V_S(x^2) \geqslant t^2 - (\alpha + \beta^2), \tag{22}$$

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

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

which completes the proof. □

In [16], Padberg shows that for any $S \subseteq N$ with $|S| \geqslant 3$ and any integer α, $1 \leqslant \alpha \leqslant |S| - 2$, inequalities (21) define facets of BQP. The idea of our proof can be applied for BQP in the following way. Let

$$\begin{aligned} \overline{QP} = \{&(x, \overline{y}) \in R^n \times R^{n(n-1)/2} \mid 0 \leqslant x_i \leqslant 1, \\ &\overline{y}_{ij} = x_i x_j \text{ for all } 1 \leqslant i < j \leqslant n\}. \end{aligned} \tag{23}$$

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 straightforward to see that the proof of Theorem 3.1 holds true for $\overline{QP}$ as well as for QP. It should be emphasized that inequalities (21) are not only valid for the convex hull of $\overline{QP}$ but also facets for BQP.

More generally, we have the following theorem:

THEOREM 3.2 (cut type inequality). *For any* $S, T \subseteq N$ *such that* $S \cap T = \emptyset$ *and integer* α, *the following inequality*

$$E_S(y) + E_T(y) - E_{S,T}(y) - \alpha V_S(x) + (\alpha+1)V_T(x) + \frac{\alpha(\alpha+1)}{2} \geqslant 0 \quad (24)$$

is valid for QP.

We note that inequality (24) includes (21) as a special case when $T = \emptyset$.

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

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

and

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

From (22), we have

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

and

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

Then, we have the following inequality:

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

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

$$F(I) \geqslant 0.$$

When $I = 1$ we have

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

and also when $I = 2$

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

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

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

and the proof is complete. □

In [16], Padberg shows that for any $S, T \subseteq N$ such that $S \cap T = \emptyset$, $|S| \geqslant 1$, and $|T| \geqslant 2$, inequalities (24) define facets of BQP when $\alpha = |T| - |S|$. Also in [5, 23], inequalities (24) have been introduced by considering the product of two linear functions below:

$$l(x) = (V_S(x) - V_T(x) - \alpha)(V_S(x) - V_T(x) - \alpha - 1), \tag{25}$$

where α is an arbitrary integer. The nonnegativity of $l(x)$ is obvious if x is integer. Expanding (25) and replacing $x_i x_j$ to y_{ij} and x_i^2 to x_i, we can obtain (24), which are considered as valid inequalities for BQP. In our proof, however, the same inequalities can be obtained without using 0-1 properties.

Finally, in Theorem 3.3 and Lemma 3.4 given below, we will introduce two classes of useful inequalities.

THEOREM 3.3. *For any $i \in N$ and real r, the following inequality*

$$y_{ii} - 2rx_i + r^2 \geqslant 0 \tag{26}$$

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

$$r_1^2 y_{ii} + r_2^2 y_{jj} - 2r_1 r_2 y_{ij} \geqslant 0 \tag{27}$$

is valid for QP.

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

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

holds. Expanding the left–hand–side and replacing x_i^2 to y_{ii}, we obtain (26), which holds true for any $(x, y) \in QP$. It is easy to show inequality (27) in the same way. □

Inequalities (26) and (27) are closely related to the positive semidefinite cone (15). 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

$$\{(x, y) \mid y_{ii} - x_i^2 \geqslant 0\}, \quad i = 1, 2, \ldots, n, \tag{28}$$

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

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

defines a supporting hyperplane of (28) at $x_i = r,\ y_{ii} = r^2$, and that this hyperplane generates inequality (26). Also, the determinant of 2×2 principal minors not containing the first row of X define the following convex sets:

$$\{(x, y) \mid y_{ii}y_{jj} - y_{ij}^2 \geqslant 0,\ y_{ii},\ y_{jj} \geqslant 0\}, \quad \text{for all } i < j. \tag{29}$$

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 (29) at $y_{ii} = r_2^2,\ y_{jj} = r_1^2,\ y_{ij} = r_1 r_2$, and generates inequality (27).

Moreover, let $(\overline{x}, \overline{y})$ be a given vector which does not satisfy the positive semidefinite condition (15), and let $\overline{X}$ be an $n + 1$ dimensional square matrix defined below:

$$\overline{X} = \begin{bmatrix} 1 & \overline{x}^T \\ \overline{x} & \overline{Y} \end{bmatrix}, \tag{30}$$

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

LEMMA 3.4. *If* $(\overline{x}, \overline{y}) \notin QP^{SDP}$, *then the following inequality separates* $(\overline{x}, \overline{y})$ *from* QP^{SDP}.

$$v^T \begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} v \geqslant 0, \tag{31}$$

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

Proof. See [20]. □

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

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

and

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

respectively.

In our cutting plane algorithm, we solve (14) 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.

Firstly, we use the following procedure to generate violated inequalities for a given point $(\overline{x}, \overline{y})$.

Procedure **TRI**

First enumerate all triples $i, j, k \in N$, and generate violated clique type inequalities (21) with $|S| = 3$ and $\alpha = 1$, then enumerate all triples again and generate violated cut type inequalities (24) 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. We will refer to these inequalities as ‘triangle inequalities’ in the remainder of this paper.

If no violated inequalities have been found, we apply the following procedure:

Procedure **DIAG**

1. *For all $i \in N$, if $\overline{y}_{ii} < \overline{x}_i^2$ then generate inequalities (26) 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 (27) by setting $r_1^2 = \overline{y}_{jj}, \quad r_2^2 = \overline{y}_{ii}$.*
3. *If some inequalities have been generated, then terminate.*
4. *Let $\overline{X}$ be a matrix defined in Lemma 3.4. For all eigenvectors v which are associated with negative eigenvalues of $\overline{X}$, generate inequalities (31).*

We call this procedure ‘diag’ since (26), (27) and (31) are the only inequalities that contain ‘diagonal’ variables, y_{ii}.

Finally, we perform several heuristic procedures for detecting violated inequalities. In these heuristics, starting from randomly generated triangle inequalities with three indices, we try to find violated inequalities by increasing the cardinality of S or T one by one.

Since inequalities (21) and (24) have a lot of freedom, i.e., S, T and α, we restrict clique and cut type inequalities within $\alpha = 1$ and $\alpha = -1$, respectively.

For each $S \subset N$ and $i \in N \setminus S$, we define

$$\begin{aligned} g_i &= l^{cl}_{S\cup\{i\}}(\overline{x}, \overline{y}, 1) - l^{cl}_{S}(\overline{x}, \overline{y}, 1) \\ &= \overline{x}_i - \sum_{j\in S} \overline{y}_{ij}. \end{aligned}$$

Also, for any $S, T \subset N$, $S \cap T = \emptyset$ and $i \in N \setminus (S \cup T)$, define

$$\begin{aligned} s_i &= l^{ct}_{S\cup\{i\},T}(\overline{x}, \overline{y}, -1) - l^{ct}_{S,T}(\overline{x}, \overline{y}, -1) \\ &= -\overline{x}_i - \sum_{j\in S} \overline{y}_{ij} + \sum_{j\in T} \overline{y}_{ij} \end{aligned}$$

and

$$\begin{aligned} t_i &= l^{ct}_{S,T\cup\{i\}}(\overline{x}, \overline{y}, -1) - l^{ct}_{S,T}(\overline{x}, \overline{y}, -1) \\ &= \sum_{j\in S} \overline{y}_{ij} - \sum_{j\in T} \overline{y}_{ij}. \end{aligned}$$

Then, we have the following heuristics:

Procedure **HEU1**

Execute the following n times.

1. *Generate a subset $S \subset N$ such that $|S| = 3$ randomly.*
2. *Let $g_{i^*} := \max_{i\in N\setminus S} g_i$.*
3. *If $g_{i^*} \le 0$, then quit. Otherwise let $S := S \cup \{i^*\}$.*
4. *Generate (21) with S and $\alpha = \lfloor V_S(\overline{x})\rfloor$. If $S \neq N$, go to 2. Otherwise quit.*

Procedure **HEU2**

Execute the following n times.

1. *Generate subsets $S, T \subset N$ such that $|S| = 1, |T| = 2$ and $S \cap T = \emptyset$ randomly.*
2. *Let $s_{i^*} := \max_{i\in N\setminus(S\cup T)} s_i$ and $t_{k^*} := \max_{k\in N\setminus(S\cup T)} t_k$. If $s_{i^*} > t_{k^*}$, go to 3. Otherwise go to 4.*
3. *If $s_{i^*} \le 0$, then quit. Otherwise let $S := S \cup \{i^*\}$. Go to 5.*
4. *If $t_{k^*} \le 0$, then quit. Otherwise let $T := T \cup \{k^*\}$. Go to 5.*
5. *Generate (24) with S, T and $\alpha = \lfloor V_S(\overline{x}) - V_T(\overline{x})\rfloor$. If $S \cup T \neq N$, go to 2. Otherwise quit.*

At the last steps in these heuristics, we set $\alpha = \lfloor V_S(\overline{x})\rfloor$ or $\alpha = \lfloor V_S(\overline{x}) - V_T(\overline{x})\rfloor$, which seems to work better.

Moreover, we introduce quadratic functions to take α into consideration. Let

$$q^{cl}_S(x, y) = \frac{1}{2}\{V_S(x)\}\{V_S(x) - 1\} - E_S(y) \tag{32}$$

and

$$q^{ct}_{S,T}(x, y) = q^{cl}_S(x, y) + q^{cl}_T(x, y) + E_{S,T}(y) - V_S(x)V_T(x), \tag{33}$$

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

LEMMA 4.1. *For any* x, y *and* $S \subseteq N$, *if* $\alpha = \lfloor V_S(x) \rfloor$, *then*

$$q^{cl}_S(x, y) \leqslant l^{cl}_S(x, y\,;\,\alpha). \tag{34}$$

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*

$$q^{ct}_{S,T}(x, y) \leqslant l^{ct}_{S,T}(x, y\,;\,\alpha). \tag{35}$$

Proof. It is obvious to see that

$$q^{cl}_S(x, y) - l^{cl}_S(x, y\,;\,\alpha) = \frac{1}{2}\{V_S(x) - \alpha\}\{V_S(x) - \alpha - 1\} \leqslant 0$$

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

Also

$$\begin{aligned} & q^{ct}_{S,T}(x, y) - l^{ct}_{S,T}(x, y\,;\,\alpha) \\ & \quad = \frac{1}{2}\{V_S(x) - V_T(x) - \alpha\}\{V_S(x) - V_T(x) - \alpha - 1\} \leqslant 0 \end{aligned}$$

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

Lemma 4.1 gives sufficient conditions for generating the cutting planes. For instance, given a vector $(\overline{x}, \overline{y})$, if we find $S \subseteq N$ such that

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

then we can generate the clique inequality

$$l^{cl}_S(x, y\,;\,\lfloor V_S(\overline{x}) \rfloor) \leqslant 0,$$

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

Then, g_i, s_i and t_i in *Procedure* **HEU1** and *Procedure* **HEU2** can be replaced by g'_i, s'_i and t'_i, respectively, in the following way. Let

$$\begin{aligned} g'_i &= q^{cl}_{S\cup\{i\}}(\overline{x}, \overline{y}, 1) - q^{cl}_S(\overline{x}, \overline{y}) \\ &= \frac{1}{2}\overline{x}_i(\overline{x}_i - 1) - \sum_{j\in S}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j), \end{aligned}$$

where $S \subset N$ and $i \in N \setminus S$. Also, let

$$\begin{aligned} s_i' &= q^{ct}_{S\cup\{i\},T}(\overline{x},\overline{y}) - q^{ct}_{S,T}(\overline{x},\overline{y}) \\ &= \frac{1}{2}\overline{x}_i(\overline{x}_i - 1) - \sum_{j\in S}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j) + \sum_{j\in T}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j) \end{aligned}$$

and

$$\begin{aligned} t_i' &= q^{ct}_{S,T\cup\{i\}}(\overline{x},\overline{y}) - q^{ct}_{S,T}(\overline{x},\overline{y}) \\ &= \frac{1}{2}\overline{x}_i(\overline{x}_i - 1) + \sum_{j\in S}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j) - \sum_{j\in T}(\overline{y}_{ij} - \overline{x}_i\overline{x}_j), \end{aligned}$$

where $S, T \subset N,\ \ S \cap T = \emptyset$ and $i \in N \setminus (S \cup T)$.

4.2. COMPUTATIONAL EXPERIMENTS

Test problems are generated as follows. Each of the coefficients $Q_{ij}(1 \leqslant i \leqslant j \leqslant n)$ and $c_i(1 \leqslant i \leqslant n)$ of the objective function in (P) are set as integers randomly distributed between -100 and 100 with density d. We generate ten problems for each n and d. Table 4.1 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 have full rank and have almost the same number of positive and negative eigenvalues. Later, we also consider the behavior of our algorithm when the number of positive and negative eigenvalues are different. All problems are solved on DEC Alpha (CPU 21164-300MHz), and we use the CPLEX 4.0 library as an LP solver.

We add the violated inequalities to the LP if the Euclidean distance between the point $(\overline{x},\overline{y})$ and the cut defining hyperplane exceeds δ, which is first set $\delta = 10^{-1}$ and dynamically changed from one iteration to the next. We terminate the cutting plane algorithm if no cutting plane is found with $\delta = 10^{-6}$ or the following relative error criterion holds with $\epsilon = 10^{-4}$:

$$f(\hat{x}) - \epsilon|f(\hat{x})| \leqslant f_L(\overline{x},\overline{y}), \tag{36}$$

where $(\overline{x},\overline{y})$ is an optimal solution of the current LP and $\hat{x}$ is the best feasible solution obtained so far. We call $\hat{x}$ an ϵ-optimal solution. We note that since $\overline{x}$ is a feasible solution of the original problem (P), $f(\overline{x})$ gives an upper bound of (P). After each LP has been solved, we try to update the best feasible solution $\hat{x}$ by using $\overline{x}$. We also provide a procedure for deleting inequalities whose slacks are greater than 10^{-6} when the total number of inequalities added to the initial LP exceeds 5000.

First, we show a result with $n = 20, 30$ in Table 4.2. In the table, m denotes the number of problems which are solved to ϵ-optimality. For those which terminate

Table 4.1. The number of positive, negative, and zero eigenvalues of Q at different densities

	$n = 20$			$n = 30$			$n = 40$			$n = 50$			$n = 60$			$n = 70$			$n = 80$		
d	pos.	neg.	zero	pos.	neg.	zero	pos.	neg.	zero	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	29.8	29.8	0.4	34.6	35.4	0.0	39.5	40.5	0.0
0.2	9.5	10.1	0.4	15.0	15.0	0.0	19.9	20.1	0.0	25.2	24.6	0.2	29.9	30.1	0.0						
0.3	9.7	10.3	0.0	15.0	15.0	0.0	19.6	20.4	0.0	25.2	24.8	0.0	29.7	30.3	0.0						
0.4	10.0	10.0	0.0	14.9	15.1	0.0	19.9	20.1	0.0	24.8	25.2	0.0									
0.5	10.0	10.0	0.0	14.5	15.5	0.0	19.8	20.2	0.0	25.1	24.9	0.0									
0.6	9.8	10.2	0.0	14.6	15.4	0.0	19.6	20.4	0.0												
0.7	10.1	9.9	0.0	14.5	15.5	0.0	19.6	20.4	0.0												
0.8	9.9	10.1	0.0	15.0	15.0	0.0	20.3	19.7	0.0												
0.9	10.0	10.0	0.0	14.8	15.2	0.0	20.4	19.6	0.0												
1.0	9.9	10.1	0.0	14.9	15.1	0.0	20.0	20.0	0.0												

Table 4.2. Result of the cutting plane algorithm for $\epsilon = 10^{-4}$

		$n = 20$									
		ERROR $\times 10^{-4}$		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	9	9.80	9.80	26.4	224	14.0	112	0.0	0	2.0	4.1
0.2	9	4.62	4.62	52.0	202	20.0	99	0.0	0	2.1	2.4
0.3	10	–	–	44.8	144	6.0	31	0.0	0	2.1	2.9
0.4	10	–	–	147.0	389	14.0	52	0.0	0	2.2	2.8
0.5	9	2.23	2.23	365.8	609	15.1	58	0.0	0	2.3	3.0
0.6	10	–	–	390.6	648	24.4	81	0.0	0	2.4	3.2
0.7	10	–	–	641.9	731	29.7	81	0.0	0	2.6	3.8
0.8	10	–	–	692.8	839	33.0	190	0.0	0	2.7	4.4
0.9	9	2.98	2.98	738.4	863	59.8	132	1.5	15	2.8	5.6
1.0	10	–	–	616.7	694	25.1	70	0.0	0	2.6	4.1
		$n = 30$									
		ERROR $\times 10^{-4}$		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	–	–	20.6	74	20.0	68	0.0	0	17.5	24.6
0.2	10	–	–	25.0	130	11.9	76	0.0	0	18.5	22.3
0.3	10	–	–	585.5	1061	38.5	118	0.0	0	20.3	24.0
0.4	10	–	–	1230.9	1411	25.8	156	0.0	0	20.0	23.5
0.5	10	–	–	1582.5	1878	45.9	107	0.0	0	21.8	25.6
0.6	10	–	–	2125.7	2546	52.0	166	2.7	27	28.2	51.6
0.7	10	–	–	2496.0	3025	74.2	176	0.2	2	31.3	71.2
0.8	9	1.40	1.40	2572.4	2693	63.6	183	0.0	0	24.9	31.1
0.9	10	–	–	2555.0	4209	53.0	264	7.9	79	45.6	257.4
1.0	9	3.79	3.79	2147.4	2288	71.7	152	0.0	0	31.1	53.7

with failure to generate cutting planes, ERROR gives the relative error;

$$\frac{f(\hat{x}) - f_L(\overline{x}, \overline{y})}{|f(\hat{x})|}. \tag{37}$$

Also, CUT denotes the number of the total generated cutting planes, CUT(**DIAG**) and CUT(**HEU**) denote the total number of generated cutting planes by *procedure* **DIAG** and *procedure* **HEU**, respectively, and TIME denotes the CPU time in seconds. Moreover, ave. and max. denote the average and the maximum, respectively.

The table shows that all problems are solved to $\epsilon = 10^{-3}$ optimality, and that most of the problems are solved by the triangle and diagonal inequalities. It seems that these inequalities play an important role in the polyhedral relaxation of QP.

Table 4.3. Result of the cutting plane algorithm with heuristics for $\epsilon = 10^{-4}$

		$n = 20$									
		ERROR $\times 10^{-4}$		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	9	1.18	1.18	24.2	224	12.4	112	0.0	0	2.5	5.1
0.2	10	–	–	50.9	202	18.9	99	0.0	0	2.7	3.6
0.3	10	–	–	44.8	144	6.0	31	0.0	0	2.8	3.4
0.4	10	–	–	147.0	389	14.0	52	0.0	0	2.9	3.5
0.5	10	–	–	363.2	598	12.5	43	0.0	0	2.8	3.5
0.6	10	–	–	389.3	648	23.9	80	0.0	0	3.1	4.1
0.7	10	–	–	640.7	731	29.4	81	0.0	0	3.3	4.6
0.8	10	–	–	692.8	839	33.0	190	0.0	0	3.5	5.6
0.9	10	–	–	726.4	822	50.0	104	0.0	0	3.4	4.7
1.0	10	–	–	616.2	690	24.6	69	0.0	0	3.4	4.7

		$n = 30$									
		ERROR $\times 10^{-4}$		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	–	–	17.4	74	16.8	68	0.0	0	21.2	26.7
0.2	10	–	–	23.3	130	11.9	76	0.0	0	21.6	24.6
0.3	10	–	–	573.1	1014	38.5	118	0.0	0	24.5	29.1
0.4	10	–	–	1225.4	1411	20.3	101	0.0	0	23.9	27.6
0.5	10	–	–	1580.3	1878	43.7	106	0.0	0	26.2	29.4
0.6	10	–	–	2117.1	2504	49.5	159	5.8	58	31.7	51.7
0.7	10	–	–	2473.5	2995	72.0	161	0.0	0	34.3	70.4
0.8	10	–	–	2568.4	2689	59.6	147	0.0	0	28.7	35.1
0.9	10	–	–	2557.9	4238	53.0	264	10.4	104	48.4	249.1
1.0	9	3.79	3.79	2143.4	2288	68.8	152	0.0	0	33.3	49.3

We also see that the performance of the cutting plane algorithm depends on the density of Q and c as well as n. For dense problems, many cutting planes are required to achieve an ϵ-optimal solution.

In addition to the cutting plane procedure, we also implement local search procedures to update the best feasible solution, $\hat{x}$. After each LP has been solved, we try to improve the solution by a variant of active set strategy [4] using $\overline{x}$ as an initial point. Furthermore, we implement a multiple start local search procedure to get an initial feasible solution before starting the cutting plane procedure.

Table 4.3 gives results for the same problems as in Table 4.2. From these tables, we see that several unsolved instances in Table 4.2 can be terminated successfully with a good feasible solution generated by the local search procedures. Thus, our cutting plane algorithm gives a sufficiently tight lower bound which guarantees an

Table 4.4. Result of the cutting plane algorithm with heuristics for $\epsilon = 10^{-4}$

		$n = 40$							
		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	77.3	577	17.6	97	0.0	0	22.9	40.6
0.2	8*	957.0	2363	56.9	216	0.2	2	29.5	44.0
0.3	10	1927.2	2253	35.5	121	0.0	0	31.1	55.8
0.4	10	3482.8	8467	88.1	506	17.8	178	156.8	1254.7
0.5	10	3856.4	4440	43.7	118	0.0	0	38.8	48.6
0.6	10	5757.3	11144	112.2	576	36.1	235	330.8	2215.4
0.7	10	5196.4	6729	54.0	188	6.3	63	82.9	294.7
0.8	10	5844.3	8145	127.5	301	39.0	221	187.6	532.8
0.9	10	5646.5	9960	142.8	475	27.5	188	282.1	1633.8
1.0	10	5591.8	8318	155.3	301	23.7	146	236.4	717.1
		$n = 50$							
		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	80.6	309	47.0	208	0.0	0	80.0	124.6
0.2	10	1856.4	2346	50.2	127	0.0	0	90.2	128.9
0.3	10	3676.3	4224	50.4	134	0.0	0	90.0	124.3
0.4	10	5639.8	10375	90.2	299	14.4	144	371.4	2367.1
0.5	10	6587.6	9371	108.8	233	48.5	272	573.8	1553.8
		$n = 60$							
		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	287.5	1646	18.0	84	0.0	0	151.6	226.1
0.2	10	3589.5	5899	59.4	195	1.5	15	246.5	650.5
0.3	10	5880.1	8212	103.6	282	7.2	72	464.5	1744.1
		$n = 70$							
		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	1518.5	3956	41.6	135	0.0	0	328.0	509.3
		$n = 80$							
		CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
d	m	ave.	max.	ave.	max.	ave.	max.	ave.	max.
0.1	10	2123.1	4145	69.8	234	0.0	0	651.9	1104.0

*One instance terminates $\epsilon = 1.75 \times 10^{-4}$ and the other terminates $\epsilon = 10.79 \times 10^{-4}$.

Table 4.5. Results by Hansen et al. [11]

d	n = 30	40	50	60	70	80
10	0.13	0.36	1.14	92.32	81.13	488.06
20	0.70	8.90	132.30			
30	3.39	143.08				
40	9.12					
50	50.85					
60	116.26					
70	160.94					

Table 4.6. Result of the cutting plane algorithm for $\epsilon = 10^{-4}$

$n = 40$								
	CUT		CUT(**DIAG**)		CUT(**HEU**)		TIME	
n^+/n^-	ave.	max.	ave.	max.	ave.	max.	ave.	max.
35/5	4348.2	5357	585.7	1160	0.0	0	186.4	840.4
30/10	4693.1	5471	379.8	1003	0.0	0	98.9	252.8
25/15	4871.5	5190	187.4	286	0.0	0	61.0	108.3
20/20	4528.3	4982	33.0	168	0.0	0	35.2	50.9
15/25	4362.9	5069	15.1	69	0.0	0	27.9	46.1
10/30	3917.4	5000	0.0	0	0.0	0	23.4	25.3
5/35	1595.6	4546	0.0	0	0.0	0	19.5	22.9

ϵ-optimality. Finally, we show results with up to $n = 80$ in Table 4.4. We see that our algorithm can generate an ϵ-optimal solution in a reasonable computational time when Q is sparse.

To our knowledge, only a few papers report numerical experiments for the indefinite problem (P). Table 4.5 shows the results of the branch and bound algorithm by Hansen et al. presented in [11]. They use similar randomly generated test problems, and show the average CPU time of ten problems in each case. Since the computation environment used in their experiments is different, we can not compare them against ours directly. However, the computational effort increases dramatically as the density of the matrix Q increases. It is obvious to see that moderate density problems with 40 or 50 variables would be out of reach by their algorithm.

Finally, we show the behavior of our algorithm under different eigenvalue structures of the matrix Q. Let $A\Lambda A^T$ be a diagonalized decomposition of the randomly generated fully dense matrix Q, where A is an orthogonal matrix and Λ is a diagonal matrix. Changing the matrix Λ, we redefine the matrix Q as $Q = A\Lambda' A^T$, where Λ' is also a randomly generated diagonal matrix which has n^+ positive elements and n^- negative ones. Thus, Q has also n^+ positive eigenvalues and n^- negative ones. We note that the matrix Q also becomes dense.

Table 4.6 shows the results when $n = 40$ and $\epsilon = 10^{-4}$. We also solve ten problems in each case, and all these problems are solved to ϵ-optimality. Obviously, when the number of positive eigenvalues increases, we need to add many inequalities associated with the positive semidefinite condition (15). Therefore, incorporating the semidefinite condition explicitly, i.e., solving the positive semidefinite problem (16) instead of the LP, we could improve the performance of our algorithm. A similar attempt has been made by Helmberg and Rendl [13] for the 0–1 quadratic problem.

5. Conclusions

We have formulated the indefinite quadratic problem with box constraints as a convex minimization problem with a linear objective function. It has been shown that several classes of facet defining inequalities for BQP can also be valid for QP. Our numerical results indicate that the positive definite constraints (15) and the only triangle inequalities provide a fairly tight polyhedral relaxation of QP. It is worth noting that our cutting plane algorithm will also work well as a bounding procedure in a branch and bound framework, which is now underway.

When the density of Q increases, it seems that we should take more complicated clique or cut type inequalities into consideration efficiently. We will need to devise more sophisticated procedures for generating and selecting good inequalities. We believe that our polyhedral approach could be an efficient method for nonconvex quadratic problems.

Acknowledgments

The authors thank Masakazu Kojima for his suggestions and warm encouragement. We would also like to acknowledge helpful comments from two anonymous referees.

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., Jünger, 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.
4. Bazaraa, M., Sherali, H.D. and Shetty, C.M. (1993), *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, New York.
5. 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.

6. Boros, E. and Hammer, P.L. (1993), Cut polytopes, Boolean quadric polytopes and nonnegative pseudo–Boolean functions, *Mathematics of Operations Research* 18: 245–253.
7. 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.
8. De Angelis, P.L., Pardalos, P.M. and Toraldo, G. (1997), Quadratic Programming with box constraints, in I.M. Bomze et al. (eds.), *Developments in Global Optimization* (pp. 73–93), Kluwer Academic Publishers, Dordrecht, Boston, London.
9. Delorme, C. and Poljak, S. (1993), Laplacian eigenvalues and the maximum cut problem, *Mathematical Programming* 62: 557–574.
10. Fujie, T. and Kojima, M. (1997), Semidefinite programming relaxation for nonconvex quadratic programs, *Journal of Global Optimization* 10: 367–380.
11. Hansen, P., Jaumard, B., Ruiz, M. and Xiong, J. (1993), Global minimization of indefinite quadratic functions subject to box constraints, *Naval Research Logistics Quarterly* 40: 373–392.
12. 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.
13. Helmberg, C. and Rendl, F. (1995), Solving quadratic (0,1)-problems by semidefinite programs and cutting planes, ZIB Preprint SC-95-35.
14. Kalantari, B. and Bagchi, A. (1990), An algorithm for quadratic zero–one programs, *Naval Research Logistics Quarterly* 37: 527–538.
15. Kojima, M., Shindoh, S. and Hara, S. (1997), Interior-point methods for the monotone linear complementarity problem in symmetric matrices, *SIAM Journal on Optimization* 7: 86–125.
16. Padberg, M. (1989), The Boolean quadric polytope: Some characteristics, facets and relatives, *Mathematical Programming* 45: 139–172.
17. 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.
18. Pardalos, P.M. and Vavasis, S.A. (1991), Quadratic programming with one negative eigenvalue is NP–hard, *Journal of Global Optimization* 1: 15–22.
19. Poljak, S. and Rendl, F. (1995), Solving the max–cut problem using eigenvalues, *Discrete Applied Mathematics* 62: 249–278.
20. Ramana, M. (1993), *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems*, PhD thesis, Johns Hopkins University, Baltimore, MD.
21. Sherali, H. and Alameddine, A. (1990), An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes, *Annals of Operations Research* 25: 197–214.
22. Sherali, H. and Alameddine, A. (1992), A new reformulation–linearization for solving bilinear programming problems, *Journal of Global Optimization* 2: 397–410.
23. 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.
24. Sherali, H. and Tuncbilek, C.H. (1995), A reformulation–convexification approach for solving nonconvex quadratic programming problems, *Journal of Global Optimization* 7: 1–31.
25. Simone, C.D. (1989), The cut polytope and the Boolean quadric polytope, *Discrete Mathematics* 79: 71–75.
26. Vavasis, S.A. (1992), Approximate algorithms for indefinite quadratic programming, *Mathematical Programming* 57: 279–311.
27. Ye, Y. (1992), On the affine scaling algorithm for nonconvex quadratic programming, *Mathematical Programming* 56: 285–300.