J. KSIAM Vol.5, No.2, 137-147, 2001

SOLUTIONS OF NONCONVEX QUADRATIC OPTIMIZATION PROBLEMS VIA DIAGONALIZATION

MOONSOOK YU AND SUNYOUNG KIM

ABSTRACT. Nonconvex Quadratic Optimization Problems (QOP) are solved approximately by SDP (semidefinite programming) relaxation and SOCP (second order cone programmming) relaxation. Nonconvex QOPs with special structures can be solved exactly by SDP and SOCP. We propose a method to formulate general nonconvex QOPs into the special form of the QOP, which can provide a way to find more accurate solutions. Numerical results are shown to illustrate advantages of the proposed method.

1. INTRODUCTION

We consider the following general quadratic optimization problem:

$$(1) \qquad (QOP) \quad z^* := \begin{array}{ll} \text{minimize} & x^TQ_0x + 2q_0^Tx \\ \text{subject to} & x^TQ_ix + 2q_i^Tx + \gamma_i \leq 0,\ i = 1, ..., m, \end{array}$$

where Q_i is an $n \times n$ symmetric matrix, $q_i \in R^n$ and $\gamma_i \in R$ for $i = 0, 1, 2, \ldots, m$.

The feasible region of the QOP by F is denoted by:

$$F \;=\; \left\{x \in C_0 : x^TQ_px + q_p^Tx + \gamma_p \leq 0 \ (p = 1, 2, \ldots, m)\right\}.$$

We assume that C_0 is a bounded polyhedral set represented by a finite number of linear inequalities in practice, although C_0 can be any compact convex subset of R^n in theory.

If all Q_i $(i = 0, 1, \ldots, m)$ are positive semidefinite, (1) becomes a convex problem. Many available software [3, 16, 15] can be used to find a minimizer because in this case every local minimizer is a global minimizer.

The above problem (1) represents indefinite quadratic optimization problem. Many difficult nonconvex optimization problems arise in various combinatorial optimization problems such as linearly constrained nonconvex quadratic programs, maximum clique problems and $0-1$ integer programs. Finding solutions of nonconvex QOP (1) is known to be NP-hard problem. As a consequence, obtaining approximate solutions of (1) has provided as a way to solve (1) currently.

Approximate solutions can be obtained by relaxing a feasible region to a convex region. This gives a larger feasible region and resulting solutions have smaller objective

Key words: second-order-cone program, lift-and-project convex relaxation method, nonconvex quadratic program, global optimization, diagonalization

Research of the author supported in part by Kosef R04-2001-00200

values. One of other popular techniques in finding approximate solutions is to derive a tighter lower bound of the minimal objective value.

Relaxing a feasible region by lift and project convex relaxation methods ([1, 2, 4, 5, 7, 9, 11, 12, 13, 14, 17]) has been one of the most popular methods. The semidefinite programming relaxation (SDP) of (QOP) and linear programming relaxation (LP) have been used extensively to obtain approximate solutions.

The semidefinite programming relaxation of (QOP) is

$$
(SDP) \quad z^{SDP} := \begin{array}{ll} \text{minimize} & M_0 \bullet X \\ \text{subject to} & M_i \bullet X \leq 0,\ i = 1, ..., m \\ & X_{00} = M_{m+1} \bullet X = 1, \\ & X \succeq 0, \end{array} \tag{2}
$$

where

$$
X = \begin{pmatrix} X_{00}, & x^T \\ x, & \bar{X} \end{pmatrix}
$$

and

$$
M_{m+1} = \begin{pmatrix} 1, & 0 \\ 0, & 0 \end{pmatrix}.
$$

If we remove the constraints $X \succeq 0$ in (2), the resulting relaxation is the LP relaxation method. The SDP relaxation of the nonconvex feasible region F of the QOP is in the worst case as effective as the LP relaxation. Also, it is known that the SDP relaxation is more effective the the LP relaxation in theory and practice. However, solving an SDP with large dimension involves expensive computational work while an LP provides a solution in much less time. The SOCP relaxation was proposed in [6] to improve the effectiveness of the LP relaxation and efficiency of SDP relaxation. The SOCP can be viewed as a compromise between the SDP and LP relaxation.

Notice that $X \succeq 0$ indicates that for any nonzero vector v, $v^T X v \geq 0$. Instead of requiring $X \succeq 0$ for any nonzero vector, we only choose convenient vectors to satisfy the condition. Therefore, a solution obtained by the SOCP relaxation may not be as good as the one from SDP relaxation, but can be found in much less time. In the same token, a solution from the SOCP relaxation is more effective than the one from an LP and takes more time than the LP relaxation in general. If the matrix Q_i's have a special structure, *e.g.*, all off-diagonal elements of Q_i are nonpositive, then the SOCP relaxation is shown to yield the exact solution of (QOP) [8].

The aim of this paper is to solve (QOP) by diagonalization of Q_i $(i = 0, 1, \ldots, m)$. Transforming Q_i into a diagonal matrix results in the SOCP relaxation with more variables. The changes are proposed based on the results that (QOP) with diagonal Q_i can be solved exactly, not approximately in [8]. This formulation is also easy to handle in terms of the matrix Q_i and gives effective soltuions. Although it is difficult to obtain the exact solution of (QOP), diagonalization is proved to give better solutions of (QOP).

This paper is organized as follows: Section 2 includes preliminary results for (QOP) with special structures. We describe the diagonal transformation in Section 3 and as a result, obtain SDP and SOCP relaxations. Section 4 contains numerical experiments for the transformed problems. Section 5 is devoted to concluding discussions.

2. Solving quadratic optimization problems with SDP and SOCP

We describe the conditions of (QOP) to be solved exactly by SDP and SOCP in [8] in this section. As mentioned in Section 1, solving a SDP or SOCP provides an approximate solution for (QOP) in general. A class of (QOP) with special structures was shown to be solved with no gap. We start by showing the conditions on (QOP) to produce exact solutions.

We rewrite (QOP) of (1) as follows:

$$\left.\begin{array}{ll} \text{minimize} & (x_0; x)^T M_0 (x_0; x) \\ \text{subject to} & (x_0; x)^T M_i (x_0; x) \leq 0 \ (1 \leq i \leq m), \ x_0 = 1, \end{array}\right\} \tag{3}$$

where

$$M_i = \begin{pmatrix} \gamma_i, & q_i^T \\ q_i, & Q_i \end{pmatrix}.$$

In addition to (3), the following condition is imposed.

(i) All off-diagonal elements of M_i $(0 \leq i \leq m)$ are nonpositive.

The additional condition (i) enables us to reach the following theorem.

Theorem 2.1. [8] *Let X^* be an optimal solution of the SDP relaxation (2). Then*

$$(\hat{x}_0; \hat{x}) == \begin{pmatrix} \sqrt{X^*_{00}} \\ \sqrt{X^*_{11}} \\ \vdots \\ \sqrt{X^*_{nn}} \end{pmatrix}$$

is an optimal solution of the QOP problem (3).

The condition (i) seems very restrictive, but there exists a way to expand the class of Q_i that satisfies the condition (i). The condition (i) can be interpreted as

(i') There exists an $(e_0, e_1, \ldots, e_n)^T \in R^{1+n}$ such that

$$\begin{array}{l} e_0 = 1, \ e_j \in \{-1, 1\} \ (j = 1, 2, \ldots, n), \\ [M_i]_{kj} e_k e_j \leq 0 \ (0 \leq k < j \leq n, \ 0 \leq i \leq m). \end{array}$$

Then, the variable x_j by $-x_j$ can be replace accordingly to make condition (i) hold.

Next, we describe conditions imposed on (QOP) to be solved by SOCP exactly. We may assume the following conditions.

(ii) Each Q_i is a diagonal matrix $(i = 0, 1, 2, \ldots, m)$..

(iii) All the elements of vectors

$$q_0, q_1, q_2, \ldots, q_m \tag{4}$$

are nonpositive.

Under the condition (ii), we can reformulate the relaxation (2) as an SOCP.

$$\left.\begin{array}{ll} \text{minimize} & Q_0 \bullet \bar{X} + 2q_0^T x \\ \text{subject to} & Q_i \bullet \bar{X} + 2q_i^T x + \gamma_i \leq 0 \ (i = 1, 2, \ldots, m), \\ & x_j^2 \leq \bar{X}_{jj} \ (j = 1, 2, \ldots, n). \end{array}\right\} \tag{5}$$

Suppose that (iii) holds in addition to (ii). Let $(x^*, \bar{X}^*)$ be an optimal solution of the SOCP (5). Then, similar to the proof of the last theorem, $\hat{x} = (\sqrt{\bar{X}_{11}^*}, \sqrt{\bar{X}_{22}^*}, \ldots, \sqrt{\bar{X}_{nn}^*})^T$ is an optimal solution of (1).

Note that condition (iii) can be relaxed by

(iii)' For every $j = 1, 2, \ldots, n$, all the jth elements of (4) are either nonnegative or nonpositive,

since we can replace the variable x_j by $-x_j$ and $[q_i]_j$ by $-[q_i]_j$ $(i = 0, 1, 2, \ldots, m)$, respectively, in (1) when all the jth elements in (4) are nonnegative.

3. Diagonalization of quadratic optimization problems

As shown in Section 2, the quadratic optimization problem with diagonal matrix M_i $(0 \leq i \leq m)$ satisfying (ii) can be solved by SDP and SOCP. In other words, diagonalization of (QOP) may provide a tool to find near exact solutions. We first show that the quadratic optimization problem (1) is diagonalized by introducing new variables, and then solved by SDP or SOCP. The resulting formulation is expected to induce better solutions in terms of objective values.

Let Q_i be diagonalized as $Q_i = P_i D_i P_i^T$, where P_i is the orthogonal matrix and D_i is the diagonal matrix. Then, (1) can be written as

$$\begin{array}{lll} z^* := & \text{minimize} & x^T P_0 D_0 P_0^T x + 2q_0^T x \\ & \text{subject to} & x^T P_i D_i P_i^T x + 2q_i^T x + \gamma_i \leq 0, \ (i = 1, \ldots, m). \end{array} \tag{6}$$

Let $y^i = P_i^T x$ $(0 \leq i \leq m)$. Then (6) is transformed into

$$\begin{array}{lll} (DQOP) & \text{minimize} & {y^0}^T D_0 y^0 + 2q_0^T P_0 y^0 \\ & \text{subject to} & {y^i}^T D_i y^i + 2q_i^T P_i y^i + \gamma_i \leq 0, \ (i = 1, \ldots, m). \\ & & P_i y^i = P_j y^j \ \ (0 \leq i < j \leq m). \end{array} \tag{7}$$

If (7) is solved by SDP or SOCP to obtain a solution, the condition (ii) needs to be satisfied. For each i $(0 \leq i \leq m)$, consider an $(n \times n)$ diagonal matrix

$$E_i = \begin{bmatrix} e_1 & & \cdots & & \\ \vdots & e_2 & \cdots & & \\ \vdots & & \ddots & & \\ & & & & e_n \end{bmatrix},$$

where $e_j = 1$ or -1, $j = 1, \ldots, n$. If $(g_i^T P_i)_j$ is positive, then e_j in E_i is determined as -1. Otherwise, e_j in E_i becomes 1. With the E_i for $i = 0, 1 \ldots, m$, we have $g_i^T P_i E_i$ satisfying (ii). Note that $E_i E_i = I$. Let $E_i y^i = \boldsymbol{y}^i$. It follows that $g_i^T P_i y^i = g_i^T P_i E_i \boldsymbol{y}^i$, and $g_i^T P_i E_i$ is a vector with nonpositive elements.

(DQOP) now can be transformed to the following form which satisfies the conditions (i) and (ii).

$$\begin{array}{lll} \text{(DQOP')} & \text{minimize} & \boldsymbol{y}^{0^T} D_0 \boldsymbol{y}^0 + 2q_0^T P_0 E_0 \boldsymbol{y}^0 \\ & \text{subject to} & \boldsymbol{y}^{i^T} D_i \boldsymbol{y}^i + 2q_i^T P_i E_i \boldsymbol{y}^i + \gamma_i \leq 0,\ i = 1, ..., m. \\ & & P_i E_i y^i = P_j E_j y^j \quad (0 \leq i < j \leq m). \end{array}$$

Notice that (DQOP') contains equality constraints as $P_i E_i y^i = P_j E_j y^j$. It should be noted that (3) satisfying the condition (ii) does not involve equality constraints. In order to obtain exact solutions using the formulation of (3), equality constraints are to be absent. As in (DQOP'), if the equality constraints are obtained during the process of transformation, the equality constraints are needed, furthermore can be relaxed to inequality constraints. Then the resulting problem is a relaxed problem of (3) and the solutions from (DQOP') are approximate solutions than accurate solutions. We discuss this issue in Section 4 when numerical experiments are explained.

Let $\boldsymbol{y} = [P_0 E_0 \boldsymbol{y}^0, P_1 E_1 \boldsymbol{y}^1, \ldots, P_m E_m \boldsymbol{y}^m]$. We consider an $(m+1)n \times (m+1)n$ diagonal matrix $\boldsymbol{M}_i$ that includes only D_i in its i-th block diagonal element $(0 \leq i \leq m)$.

$$\boldsymbol{M}_i = \begin{bmatrix} \boldsymbol{O} & \cdots & \cdots & \cdots & \cdots \\ \vdots & \boldsymbol{O} & \cdots & \cdots & \cdots \\ \vdots & \cdots & D_i & \cdots & \cdots \\ \vdots & \cdots & \cdots & \boldsymbol{O} & \cdots \\ \vdots & \cdots & \cdots & \cdots & \boldsymbol{O} \end{bmatrix}$$

Let $\boldsymbol{q}_i = [\boldsymbol{0}, \boldsymbol{0}, q_i, \boldsymbol{0}, \boldsymbol{0}]$, where $\boldsymbol{0}$ is an $1 \times n$ zero vector. Then,

$$\begin{array}{llll} & (DQOPV) & \text{minimize} & \boldsymbol{y}^T \boldsymbol{M}_0 \boldsymbol{y} + 2\boldsymbol{q}_0^T \boldsymbol{y} \\ (8) & & \text{subject to} & \boldsymbol{y}^T \boldsymbol{M}_i \boldsymbol{y} + 2\boldsymbol{q}_i^T \boldsymbol{y} + \gamma_i \leq 0,\ i = 1, ..., m. \\ & & & P_i E_i y^i = P_j E_j y^j \quad (0 \leq i < j \leq m). \end{array}$$

The matrices $\boldsymbol{M}_i$ of the problem (DQOPV) are diagonal, and the (DQOPV) can be solved by SDP and SOCP to obtain an approximate solution as described in Section 2.1. A drawback of this approach is the increased number of variables. If (DQOPV) is solved by SDP relaxation, the size of the variable matrix is $(1+n+mn)\times(1+n+mn)$. However, we can see that the matrix is very sparse.

The SDP relaxation of (8) is formulated as follows. Let us introduce the following matrix notation:

$$\bar{\boldsymbol{M}}_i = \begin{bmatrix} \gamma_i & \boldsymbol{q}_i^T \\ \boldsymbol{q}_i & \boldsymbol{M}_i \end{bmatrix}$$

and $\boldsymbol{Y} = (1; \boldsymbol{y})$,

$$\bar{\boldsymbol{M}}_{m+1} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Using $(P_iE_iy^i)_k = (P_{i+1}E_{i+1}y^{i+1})_k \quad (1 \le k \le n, \quad 0 \le i \le m-1)$,

$$\bar{\boldsymbol{M}}_{m+1+ik} = \begin{bmatrix} 0 & \mathbf{0} & (P_iE_i)_{k.}/2 & -(P_{i+1}E_{i+1})_{k.}/2 & \mathbf{0} & \cdots \\ 0 & \mathbf{0} & \cdots & & & \\ (P_iE_i)_{k.}^T/2 & \mathbf{0} & & \cdots & & \\ -(P_{i+1}E_{i+1})_{k.}^T/2 & \mathbf{0} & & \cdots & & \\ \mathbf{0} & & & \boldsymbol{O} & & \\ \vdots & & & & \mathbf{0} & \end{bmatrix}$$

where $1 \le ik \le mn$. The problem now can be rewritten as

$$\text{(9)} \qquad \left.\begin{array}{ll} \text{minimize} & \bar{\boldsymbol{M}}_0 \bullet \boldsymbol{Y} \\ \text{subject to} & \bar{\boldsymbol{M}}_i \bullet \boldsymbol{Y} \le 0 \quad (i = 1, 2, \ldots, m), \\ & \boldsymbol{Y} \succeq 0, \\ & \bar{\boldsymbol{M}}_{m+1} = 1, \\ & \bar{\boldsymbol{M}}_{m+1+ik} = 0. \quad (1 \le ik \le mn) \end{array}\right\}$$

The SOCP relaxation of (8) is formulated as follows.

$$\left.\begin{array}{ll} \text{minimize} & \boldsymbol{y}^T\boldsymbol{M}_0\boldsymbol{y} + 2\boldsymbol{q}_0^T\boldsymbol{y} \\ \text{subject to} & \boldsymbol{y}^T\boldsymbol{M}_i\boldsymbol{y} + 2\boldsymbol{q}_i^T\boldsymbol{y} + \gamma_i \le 0 \ (i = 1, 2, \ldots, m), \\ & \boldsymbol{y}_j^2 \le \boldsymbol{Y}_{jj} \ (j = 1, 2, \ldots, n(m+1)), \\ & P_iE_iy^i = P_jE_jy^j \text{ for } (0 \le i < j \le m). \end{array}\right\}$$

Let $\boldsymbol{z}_j = \boldsymbol{y}_j^2, \ j = 1, \ldots, mn+n$.

$$\text{(10)} \qquad \left.\begin{array}{ll} \text{minimize} & \operatorname{diag}(\boldsymbol{M}_0)^T\boldsymbol{z} + 2\boldsymbol{q}_0^T\boldsymbol{y} \\ \text{subject to} & \operatorname{diag}(\boldsymbol{M}_i)^T\boldsymbol{z} + 2\boldsymbol{q}_i^T\boldsymbol{y} + \gamma_i \le 0 \ (i = 1, 2, \ldots, m), \\ & \boldsymbol{y}_j^2 \le \boldsymbol{z}_j \ (j = 1, 2, \ldots, n(m+1)), \\ & P_iE_iy^i = P_{i+1}E_{i+1}y^{i+1} \quad (0 \le i \le m-1). \end{array}\right\}$$

n	the number of variables
m	the number of quadratic inequality constraints
m_l	the number of linear constraints
$\#\lambda$	the number of negative eigenvalues of Q_p
SDP	the SDP relaxation (2)
DSDP	the diagonalized SDP relaxation (9)
DSOCP	the diagonalized SOCP relaxation (10)
obj.val.	the value of objective function obtained
cpu	the cpu time in seconds
it.	the number of iterations that the corresponding relaxation takes

TABLE 1. Notation

Note that (10) has the equality constraints produced by transformation to a diagonal form. We can relax the equality constrains to the inequality constraints as follows.

$$(11)\quad \left.\begin{array}{lll} (DSOCP) \text{ minimize} & \operatorname{diag}(\boldsymbol{M}_0)^T\boldsymbol{z} + 2\boldsymbol{q}_0^T\boldsymbol{y} \\ \text{subject to} & \operatorname{diag}(\boldsymbol{M}_i)^T\boldsymbol{z} + 2\boldsymbol{q}_i^T\boldsymbol{y} + \gamma_i \leq 0 \ (i = 1, 2, \dots, m), \\ & \boldsymbol{y}_j^2 \leq \boldsymbol{z}_j \ (j = 1, 2, \dots, n(m+1)), \\ & P_iE_iy^i \leq P_{i+1}E_{i+1}y^{i+1} \quad (0 \leq i \leq m-1). \end{array}\right\}$$

The number of variables in (DSOCP) is $2(mn+n)$. Recall that the number of variables in (SDP) is $(n+1)^2$. If $m = \frac{1}{2}n$, the number of the variables for the problems are the same. As m increase, the number of variables in (DSOCP) increases. Diagonalization of the matrix $Q_i(0 \leq i \leq m)$ has implemented to have similar formulation in Theorem 2.1, where the exact solutions can be obtained. (DSDP) and (DSOCP) may not provide exact solutions because of the equality constraints, however, as we make the formulation of (QOP) closer to that of (3) with the condition (ii), (DSDP) and (DSOCP) can give a better approximate solution than SDP relaxation (2), which we show with numerical experiments in the following section.

4. NUMERICAL EXPERIMENTS

We present computational results on the SDP relaxation (2), the diagonalized SDP relaxation (9) and the diagonalized SOCP relaxation (10). All the computation was implemented using a MATLAB toolbox, SeDuMi Version 1.03 [15] on Sun Enterprise 4500 (400MHz CPU with 6 GB memory). We use the notation described in Table 1 in the discussion of computational results.

The test problems in our numerical experiments consists of Box constraint QOPs in (1). That is, in addition to the constraints in QOPs, we added the box constraint of $-1 \leq x_j \leq 1$ $(j = 1, 2, \dots, n)$ to have a bounded feasible region for the QOP (1). In this case, C_0 is the region represented by the box constraint. Random numbers from a uniform distribution on the interval $(-1.0, 0.0)$ are assigned to the real number

γ_p and each component of the vector q_p ($p = 0, 1, 2, \ldots, m$). The random number generator in MATLAB was also used to create elements in Q_p ($p = 0, 1, \ldots, m$). The convexity of each Q_p was varied to test stability of the proposed method. The number of negative eigenvalues of each Q_p determines the convexity of the objective function and constraints. When we generate QOPs, we give the number of negative eigenvalues for each Q_p as input, so that the resulting problems have the degree of convexity that we want.

To create Q_p, we first generate $D_p = \mathrm{diag}[1, \lambda_2, \ldots, \lambda_n]$ with a predetermined number of negative/positive diagonal entries, where each λ_i denotes a random number uniformly distributed either in the interval $\in (0.0, 1.0)$ if $\lambda_i > 0$ or in the interval $\in (-1.0, 0.0)$ if $\lambda_i < 0$ ($i = 2, \ldots, n$). An orthogonal matrix P_p is generated by taking the orthogonal matrix from schur decomposition of an $n \times n$ matrix, whose element is a random number in the interval $(0.0, 1.0)$. Finally we generate each $Q_p \in \mathbb{S}^n$ such that

$$Q_p = P_p D_p P_p^T.$$

This construction is aimed to have $\boldsymbol{x} = \mathbf{0}$ all the time as an interior feasible solution of the QOP (1) since $\gamma_p < 0$ ($p = 1, 2, \ldots, m$).

The numerical results are obtained by comparing (2), (9) and (10). The motivation for this setting of the numerical experiments is to see whether better approximate solutions can be attained using one of the three relaxations of (QOP), and compare their numerical efficiency in terms of cpu time. Table 2 shows the numerical results from (2), (9) and (10), and Table 3 show the comparison between (2) and (10). Recall that diagonalization of (QOP) into (9) and (10) increases the number of variables. As a result, (9) involves a large number of variables and in numerical experiments, it takes much more cpu time than (2) or (10). Therefore, for the large value of n and m as shown in Table 3, the test results from (2) and (10) are included.

The numerical results for various n, m, and $\#\lambda$ on each relaxation are shown in Table 2 and Table 3. We have chosen $n \leq 30$, $m \leq n$ and $\#\lambda = 2, n/2, n-2$ to observe the effects on different numbers of variables and constraints, and increasing/decreasing convexity for n in Table 2. As we increase the negative number of eigenvalues of Q_i, the concavity of the feasible region grows. In Table 2, the numerical results from three relaxation methods are shown for relatively small n and m. This choice of n is from the increased number of varibles in (9) and (10). Especially, in SDP relaxation (9), the number of varibles is $(mn + n)^2$. Computing time for solving diagonalized SDP relaxation (9) takes much more time compared to (2) or (10), *e.g.*, 1944 seconds for $n = 30, m = 15$ and $\#\lambda = 28$. Our problem is a minimization over a relaxed region, therefore, large object values indicate better approximate solutions of (QOP). Objective values from (DSDP) and (DSOCP) are almost similar in all cases of Table 2. The values are the same theoretically, but diagonalization of a matrix involves errors in practice and the resulting objective values are not exactly the same. We can see that (DSOCP) is better than (SDP) and (DSDP) in time and objective values. Hence, we compare (SDP) and (DSOCP) for larger n and m in Table 3.

Since the increased number of the variables in the diagonalization depends on the number of constraints m, the cases of $m \leq n$ were tested. More precisely, the number of variables in (2) and (10) is equal when $m = \frac{1}{2}n$. If $m \geq \frac{1}{2}n$, then more time is expected to take for solutions of (10) than (2), though we have observed that (10) yields better objective values than (2). The objective values obtained from (SDP) and (DSOCP) show that (DSOCP) provides better approximations for all n and m in the numerical experiments. The cpu time for finding approximate solutions depends heavily on the number of variables. In Table 3, it is shown that (DSOCP) achieve solutions in much less time than (SDP) because the size of m is less than or equal to half of n except $n = 100$ and $m = 50$ $\#\lambda = 98$, in which the problem is almost concave.

n	m	$\#\lambda$	SDP			DSDP			DSOCP		
			obj.val.	cpu	it.	obj.val.	cpu	it.	obj.val	cpu	it.
20	10	2	-42.44	0.6	15	-16.27	157.1	16	-16.28	0.6	13
20	10	10	-39.70	0.6	15	-10.25	130.8	14	-10.20	0.6	14
20	10	18	-50.66	0.5	12	-5.59	119.3	13	-5.54	0.6	14
30	10	2	-99.03	1.3	14	-44.04	719.5	19	-43.56	1.1	15
30	10	15	-79.87	1.6	14	-37.31	653.3	18	-36.83	1.1	15
30	10	28	-45.06	1.3	14	-26.61	593.5	16	-26.59	1.0	14
30	15	2	-65.01	2.0	19	-45.71	2141.1	17	-45.65	1.6	14
30	15	15	-54.29	1.7	16	-35.77	1841.8	16	-35.71	2.0	16
30	15	28	-57.30	2.3	19	-26.26	1944.0	17	-25.93	1.7	15

TABLE 2. Numerical results from SDP, DSDP and DSOCP

n	m	$\#\lambda$	SDP			DSOCP		
			obj.val.	cpu	it.	obj.val.	cpu	it.
100	10	2	-227.02	37.3	23	-61.75	2.3	14
100	10	50	-239.64	32.7	20	-38.86	2.7	16
100	10	98	-309.36	31.7	19	-11.72	2.3	15
100	40	2	-273.09	89.4	22	-62.83	67.8	17
100	40	50	-297.76	88.6	23	-34.10	71.8	17
100	40	98	-262.75	103.2	26	-11.74	10.7	16
100	50	2	-216.17	156.7	33	-11.63	145.8	17
100	50	98	-225.15	93.2	20	-61.80	126.6	15
150	50	2	-391.88	410.2	27	-100.33	223.5	16
150	50	75	-389.35	472.6	31	-56.17	248.8	17
150	50	148	-434.30	499.5	32	-142.81	276.5	17

TABLE 3. Numerical results on SDP and DSOCP

5. Concluding discussions

We have discussed finding approximate solutions of general quadratic problems in the form of (1). The diagonalized second order cone programming relaxation has been proposed for improving the effectiveness and efficiency for solving the problem based on Theorem 2.1, which states the conditions for solving the quadratic optimization problem with no gap. Though the proposed formulation requires the increased number of variables, the proposed method works well in terms of time and objective values for $m \leq \frac{1}{2}n$. In most of the test problems that we listed in the previous section, the diagonalized SOCP relaxation has attained much better lower bounds than the lift-and-project SDP relaxation with smaller amount of cpu time.

References

[1] Balas, E., Ceria, S., Cornuéjols, G. (1993): A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming* **58**, 295–323

[2] Fujie, T., Kojima, M. (1997): Semidefinite relaxation for nonconvex programs. *Journal of Global Optimization* **10**, 367–380

[3] Fujisawa, K., Kojima, M., Nakata, K. (1995): SDPA (SemiDefinite Programming Algorithm) user's manual, Version 5.0. Research Report B-308, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, Japan.

[4] Goemans, M. X., Williamson, D. P. (1995): Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of Assoc. Comput. Mach.* **42**, 1115–1145

[5] Grötschel, M., Lovász, L., Schrijver, A. (1988): *Geometric algorithms and combinatorial optimization.* Springer, New York, NY.

[6] Kim, S. and Kojima M. (2001): ·, Second Order Cone Programming Relaxation Programming of Quadratic Optimization Problems, *Optimization Methods and Software*, **15**, 201-224.

[7] Kojima, M., Tunçel, L. (2000): Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM J. on Optimization* **10**, 750–778.

[8] Kojima M. and Kim S: "On Quadratic Minimization with Quadratic Constraints," a working paper, Department of Mathematical and Computing Science, Tokyo Institute of Technology, (2001).

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

[10] Lovász, L.(1998): "Applications in Combinatorial Optimization," presented at Workshop *Semidefinite Programming: Applications and Algorithms,"* ZIB-Berlin, November 15-17.

[11] Nesterov, Yu. E. (1998): Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software* **9**, 141–160

[12] Poljak, S., Rendl, F., Wolkowicz, H. (1995): A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization* **7**, 51–73

[13] Sherali, H. D., Tuncbilek, C. H. (1992): A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization* **2**, 101–112

[14] Shor, N. Z. (1990): Dual quadratic estimates in polynomial and boolean programming. *Annals of Operations Research* **25**, 163–168

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

[16] Toh, K., Todd, M. J., Tütüntü, R. H. (1998): SDPT3 — a MATLAB software package for semidefinite programming. Dept. of Mathematics, National University of Singapore, Singapore.
[17] Ye, Y. (1999): Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming* **84**, 219-226

Department of Mathematics, Ewha Womans University
11-1 Dahyun-dong, Sudaemoon-gu, Seoul 120-750 Korea
email: sadad@math.ewha.ac.kr skim@ewha.ac.kr