New Relaxations for Binary Quadratic Problems Using Second-Order Cone Programming

Bissan Ghaddar[*], Juan C. Vera, Miguel F. Anjos[†]
{bghaddar@uwaterloo.ca, jvera@uwaterloo.ca, anjos@stanfordalumni.org}

Department of Management Sciences, University of Waterloo,
200 University Avenue West, Waterloo, ON, Canada, N2L 3G1

October 31, 2009

Abstract

We present a general framework for conic relaxations based on polynomial programming. Using this framework, we can re-derive previous relaxation schemes and provide new ones for general binary quadratic optimization. In particular, we propose a second-order cone programming relaxation that can be strengthened by adding triangle inequalities. Computational tests on the max-cut problem, the unconstrained version of binary quadratic optimization, show that the strengthened second-order cone-based relaxation outperforms the semidefinite-based relaxation in terms of computational efficiency and is comparable in terms of bounds. In order to gain insight into the performance of our approach on constrained quadratic binary problems, we also compare the second-order cone relaxation for the quadratic knapsack problem to the strongest semidefinite relaxation in the literature. The computational results confirm that using the second-order cone relaxation with triangle inequalities provides a bound that is competitive with the semidefinite bound strengthened with triangle inequalities but the former relaxation is computationally more efficient.

1 Introduction

Binary quadratic problems (BQP) can be expressed as optimizing a quadratic polynomial objective subject to quadratic polynomial equalities and inequalities. Several types of relaxations can be obtained using linear, second-order cone, or semidefinite techniques [19, 30, 5, 9]. In this paper we study relaxations for the general BQP based on polynomial programming. We apply these relaxations to the max-cut problem (the unconstrained version of the BQP) and to the quadratic knapsack problem (QKP). We explore the performance of these relaxations, comparing them to previous relaxations in the literature in terms of bounds and computational time.

Polynomial programming includes a broad class of problems and is known to be $\mathcal{NP}$-hard. Polynomial problems can be relaxed to tractable problems by using sum-of-squares (SOS) decompositions which lead to semidefinite relaxations. This technique was first proposed by Shor [41] to

[*]Research supported by a Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada.

[†]Currently on leave at Universität zu Köln, Institut für Informatik, Pohligstrasse 1, 50969 Köln, Germany. Research supported by the Natural Sciences and Engineering Research Council of Canada, and by a Fellowship from the Humboldt Foundation.

obtain bounds for the optimal value for the unconstrained case. This idea was then generalized by Parrilo [31] and Lasserre [23] for the constrained case. In this paper, we use the characterization of linear polynomials that are non-negative over the ball as second-order cone (SOC) vectors. Using this characterization, we propose SOC relaxations to the binary quadratic problem.

Using the polynomial programming framework allows us to re-derive and compare previous relaxation schemes. Further, we can strengthen existing relaxations by using the SOC characterization of degree one polynomials non-negative on the ball. The polynomial programming scheme also enables us to identify and eliminate the expensive components of the relaxations, in our case the semidefinite terms. In this way we obtain second-order cone-based relaxations that produce bounds that are competitive with the existing bounds but computationally more efficient.

The paper is organized as follows. In Section 2 we present an overview of polynomial programming, and its SOS and SOC relaxations. In Section 3, we describe our solution methodology and present several relaxations for the binary quadratic problem. In Section 4, we apply our proposed approach to Max-cut, QKP, and quadratic multiple knapsack problem (QMKP) and report computational results. Finally, conclusions and future research directions are discussed in Section 5.

2 Preliminaries

Given an n-tuple $\alpha = (\alpha_1, \cdots, \alpha_n)$ where $\alpha_i \in \mathbb{Z}_+$, the total degree of the monomial $x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ is the non-negative integer $d = |\alpha| := \sum_i^n \alpha_i$. There are at most $\binom{n+d}{d}$ monomials of degree at most d. A polynomial is a finite linear combination of monomials

$$f(x) = \sum_\alpha c_\alpha x^\alpha = \sum_\alpha c_\alpha x_1^{\alpha_1} \cdots x_n^{\alpha_n},$$

where the coefficients $c_\alpha \in \mathbb{R}$. We denote the cone of real polynomials (of degree at most d) that are SOS by $\Sigma^2 \subset \mathbf{R}[x]$ (resp. Σ_d^2) where $\mathbf{R}[x] := \mathbf{R}[x_1, \cdots, x_n]$ (resp. $\mathbf{R}_d[x]$) denotes the set of polynomials in n variables with real coefficients (resp. of degree at most d). Given $S \subseteq \mathbf{R}$, we denote $\mathcal{P}_d(S) := \{p(x) \in \mathbf{R}_d[x] : p(s) \geq 0 \text{ for all } s \in S\}$ to be the cone of polynomials of degree at most d that are non-negative over S.

Consider the multivariate polynomials $f(x)$ and $g_i(x)$ for $1 \leq i \leq m$ with $x \in \mathbb{R}^n$. A polynomial programming problem has the form:

$$\begin{aligned} \textbf{(PP-P)}\ & \sup\ f(x) \\ & \text{s.t. }\ g_i(x) \geq 0 \quad 1 \leq i \leq m. \end{aligned}$$

We can also include equalities $h_i(x) = 0$, since these can be expressed as inequality constraints ($h_i(x) \geq 0$ and $h_i(x) \leq 0$).

Solving polynomial programming problems is an area being actively studied. In the unconstrained case Shor introduced the idea of computing the minimum value λ such that $\lambda - f(x)$ is a SOS to obtain an upper bound for the supremum of f [41]. Such a minimum λ can be computed in polynomial time using semidefinite programming. This idea was further developed by Parrilo [31] and Parrilo and Sturmfels [34] for the constrained case using SOS decompositions. Lasserre [23] proposed a general solution approach for polynomial optimization problems via semidefinite programming using methods based on moment theory. Refinements of such ideas have been used in several instances. de Klerk and Pasechnik [20] approximated the copositive cone via a hierarchy of linear or semidefinite programs of increasing size using decompositions into sum-of-squares and polynomials with non-negative coefficients. Kojima, Kim, and Waki exploited the sparsity of the polynomials to reduce the size of the semidefinite problem [21]. Peña, Vera, and Zuluaga [35] presented solution schemes exploiting the equality constraints. In addition, the idea of approximating a set of non-negative polynomials is also present in the work of several authors such as Nesterov [29], Parrilo [33, 32], Sturmfels, Demmel, and Nie [42], Laurent [25], and Zuluaga, Vera, and Peña [45].

Consider λ to be the optimal value for (PP-P), then λ is the smallest value such that $\lambda - f(x) \geq 0$ for all $x \in S := \{x : g_i(x) \geq 0;\ 1 \leq i \leq m\}$. As a result, we can express problem (PP-P) as:

$$\begin{aligned} \textbf{(PP-D)}\ & \inf \lambda \\ & \text{s.t. } \lambda - f(x) \geq 0 \quad \forall x \in S. \end{aligned} \tag{1}$$

To obtain computable relaxations (via SDP) of (1), one can use a SOS decomposition with restricted degree of the (unknown) polynomials which can be re-phrased in terms of a linear system of equations involving positive semidefinite matrices [45]. Thus, solving a polynomial problem can be relaxed to solving an easier problem involving SOS which can be re-cast as a semidefinite programming problem [41, 44].

For the constrained version of (PP-P),

$$z^*_{PP} = \max_{x \in S}\ f(x) \qquad \equiv \qquad \min\{\lambda : \lambda - f(x) \geq 0\ \forall x \in S\}.$$

The condition $\lambda - f(x) \in \mathcal{P}_d(S)$ is $\mathcal{NP}$-hard in general. Relaxing this condition to $\lambda - f(x) \in \mathcal{K}$ for a suitable $\mathcal{K} \subseteq \mathcal{P}_d(S)$ and defining

$$\begin{aligned} z^*_{\mathcal{K}} \ =\ & \inf \lambda \\ & \text{s.t. } f(x) - \lambda \in \mathcal{K}, \end{aligned}$$

we have $z^*_{\mathcal{K}} \geq z^*_{PP}$. Finding a good approximation $\mathcal{K}$ of $\mathcal{P}_d(S)$ is a key factor in obtaining a good bound on the problem. At the same time, we need a tractable approximation, i.e., one that uses linear, second-order, and semidefinite cones, and thus can be solved efficiently using interior-point methods.

2.1 Sum-of-Squares and Second-Order Cone Relaxations

Consider a polynomial $p(x)$ of degree d. We are interested in studying when

$$p(x) \geq 0 \qquad \forall x \in \mathbb{R}^n.$$

A necessary condition is that the degree of p be even. A sufficient condition is the existence of a sum-of-squares decomposition, i.e., the existence of polynomials $q_1(x), \cdots, q_k(x)$ such that $p(x) = \sum_{i=1}^k q_i(x)^2$, or equivalently, $p \in \Sigma^2$. If $p(x)$ is a sum-of-squares polynomial then it is a non-negative polynomial for all values of x; however the inverse does not hold. A simple counter-example is the Motzkin form [28].

SOS conditions can be written as SDP problems by applying the following theorem:

Theorem 2.1. *[41] A polynomial $p(x)$ of degree d is SOS if and only if $p(x) = \sigma(x)^T Q \sigma^T(x)$, where σ is a vector of monomials in the x_i variables, $\sigma(x) = [x^\alpha]$ with $|\alpha| \leq \frac{d}{2}$ and $Q \in \mathcal{S}^N_+$, $N = \binom{n+d/2}{d/2} = |\sigma|$.*

As a result, $p(x)$ is SOS for every x can be easily seen to be equivalent to an SDP feasibility problem. The size of the matrix Q in the corresponding SDP is $\binom{n+d/2}{d/2} \times \binom{n+d/2}{d/2}$. In addition, we have $\binom{n+d}{d}$ equality constraints. This problem is polynomial time solvable if d is fixed.

The following result will allow us to use second-order cone relaxations where $f(x)$ is a polynomial of degree one and $\mathcal{B} := \{x : \|x\|^2 = n\}$

Lemma 2.2. *$f(x) \in \mathcal{P}_1(\mathcal{B})$ if and only if $f(x) = f^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix}$ with $f \in \mathcal{L}^n$, where $\mathcal{L}^n$ is the second-order cone.*

Further, by the $\mathcal{S}-$Lemma of Yakubovich (see [38]), the non-negativity of a polynomial $f(x)$ of degree two for $x \in \mathcal{B}$ can be represented using SOS:

Lemma 2.3. *$f(x) \in \mathcal{P}_2(\mathcal{B})$ if and only if $f(x) = s(x) + t(n - \|x\|^2)$, where $s(x)$ is SOS and $t \in \mathbb{R}_+$.*

The key feature that distinguishes the use of semidefinite and second-order cones to solve (PP-P) is their tractability. As a result, we can use these techniques to compute global upper bounds for polynomial functions.

3 Binary Quadratic Programming

The binary quadratic programming problem is a classical combinatorial problem. It is the problem of minimizing or maximizing a quadratic function of several binary variables. We start by considering the constrained version where we minimize a generalized objective involving both linear and quadratic terms with binary decision variables, subject to quadratic and linear constraints on these variables. The problem is known as the binary quadratic problem and is formally expressed as:

$$
\begin{aligned}
\textbf{(BQP-P)}\ \max\ & x^T Q x + p^T x && \\
\text{s.t.}\ & a_i x = b_i && \forall i = 1, \cdots, t \qquad (2) \\
& c_i x \geq d_i && \forall i = 1, \cdots, u \qquad (3) \\
& x^T F_i x + e_i^T x = k_i && \forall i = 1, \cdots, v \qquad (4) \\
& x^T G_i x + h_i^T x \geq l_i && \forall i = 1, \cdots, w \qquad (5) \\
& x \in \{-1, 1\}^n. &&
\end{aligned}
$$

There are many well-known problems that can be naturally written as binary quadratic problems. For example, folding of proteins in three-dimension by Phillips and Rosen [36], machine scheduling and unconstrained task allocation by Alidaee, Kochenberger, and Ahmadian [1], capital budgeting and financial analysis such as in Laughhunn [24], and examples arising in physics and engineering applications such as the spin glass problem and circuit board layout design by Grötschel, Jünger, and Reinelt [13]. Boros and Hammer [6] and Boros and Prekopa [7] formulated many satisfiability problems as BQPs. In addition, there are several applications related to combinatorial problems such as the single-row facility layout problem [3] and the max-cut problem.

3.1 Polynomial-based Relaxations

In this section, we propose and describe several relaxations for the binary quadratic problem based on polynomial programming. Using (1), we obtain that (BQP-P) is equivalent to

$$\begin{aligned}
&\min \lambda \\
&\text{s.t. } \lambda - \sum_{i,j} Q_{ij} x_i x_j - \sum_i p_i x_i \in \mathcal{P}_2(\{-1,1\}^n \cap S),
\end{aligned}$$

where $S = \{x : a_i x = b_i, c_i x \geq d_i, x^T F_i x + e_i^T x = k_i, x^T G_i x + h_i^T x \geq l_i\}$. Note that even checking if a polynomial is in $\mathcal{P}_2(\{-1,1\}^n)$ is $\mathcal{NP}$-hard. Hence, we need tractable approximations of $\mathcal{P}_2(\{-1,1\}^n \cap S)$. One approximation of $\mathcal{P}_2(\{-1,1\}^n \cap S)$ is obtained using the cone

$$\begin{aligned}
\mathcal{K}_r := &[\Sigma^2_{r+2} + \sum_i (1 - x_i^2)\mathbf{R}_r[x] + \sum_i (b_i - a_i x)\mathbf{R}_{r+1}[x] + \sum_i (d_i - c_i x)\mathcal{P}_{r+1}(S) \\
&+ \sum_i (k_i - x^T F_i x - e_i^T x)\mathbf{R}_r[x] + \sum_i (l_i - x^T G_i x - h_i^T x)\Sigma^2_r] \cap \mathbf{R}_2[x] \\
&\subseteq (\mathcal{P}_r(\{-1,1\}^n) + \mathcal{P}_r(S)) \cap \mathbf{R}_2[x] \\
&\subseteq \mathcal{P}_2(\{-1,1\}^n \cap S),
\end{aligned}$$

where r is an non-negative even number. Since there are no good representations of odd degree polynomials that are non-negative in S, we will approximate $\mathcal{P}_{r+1}(S)$ by using other tractable cones.

The relaxed problem can be expressed as

$$\begin{aligned}
(\mathbf{BQP}_{\mathcal{K}_r})\ &\min \lambda \\
&\text{s.t. } \lambda - \sum_{i,j} Q_{ij} x_i x_j - \sum_i p_i x_i \in \mathcal{K}_r.
\end{aligned} \tag{6}$$

The first and simplest relaxation is to consider $r = 0$, hence, we use

$$\begin{aligned}
\mathcal{K}_0 = &\Sigma^2_2 + \sum_i (1 - x_i^2)\mathbf{R}_0 + \sum_i (b_i - a_i x)\mathbf{R}_1[x] + \sum_i (d_i - c_i x)\mathcal{P}_1(S) \\
&+ \sum_i (k_i - x^T F_i x - e_i^T x)\mathbf{R}_0 + \sum_i (l_i - x^T G_i x - h_i^T x)\mathbf{R}_0^+.
\end{aligned}$$

3.2 Second-Order Cone Relaxation of BQP

Recall the previous polynomial formulation of the binary quadratic problem. Notice first that $\|x\|_2 = \sqrt{n}$ is a redundant constraint as

$$x_i^2 = 1 \Rightarrow \sum_i x_i^2 = n \Rightarrow \|x\|_2 = \sqrt{n}.$$

Therefore, we can write the BQP problem as:

$$
\begin{aligned}
\max\ & x^T Q x + p^T x \\
\text{s.t.}\ & x \in S \\
& \|x\|^2 = n \\
& x \in \{-1,1\}^n.
\end{aligned}
$$

Which can be reformulated using (1):

$$
\begin{aligned}
\min\ & \lambda \\
\text{s.t.}\ & \lambda - q(x) \geq 0 \\
& \forall x \in S \cap \mathcal{B} \cap \{-1,1\}^n,
\end{aligned}
$$

where $q(x) = \sum_{i,j} Q_{ij} x_i x_j + \sum_i p_i x_i$.

Defining

$$
\begin{aligned}
\bar{\mathcal{K}}_{\prime} = \mathcal{P}_2(\mathcal{B}) + \sum_i (1+x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1-x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1-x_i^2)\mathbf{R}_0 + \sum_i (b_i - a_i x)\mathbf{R}_1[x] \\
+ \sum_i (d_i - c_i x)\mathcal{P}_1(\mathcal{B}) + \sum_i (k_i - x^T F_i x - e_i^T x)\mathbf{R}_0 + \sum_i (l_i - x^T G_i x - h_i^T x)\mathbf{R}_0^+,
\end{aligned}
$$

we have $\bar{\mathcal{K}}_{\prime} \subseteq \mathcal{P}_2(S \cap \mathcal{B} \cap \{-1,1\}^n)$.

Using Lemmas 2.2 and 2.3, we can write the condition $\lambda - q(x) \in \bar{\mathcal{K}}_0$ as

$$
\begin{aligned}
\lambda - q(x) =& s(x) + \sum_i (1+x_i)\alpha_i(x) + \sum_i (1-x_i)\beta_i(x) + \sum_i \gamma_i(1-x_i^2) + \sum_i \delta_i(x)(b_i - a_i x) \\
& + \sum_i \eta_i(x)(d_i - c_i x) + \sum_i \theta_i(k_i - x^T F_i x - e_i^T x) + \sum_i \xi_i(l_i - x^T G_i x - h_i^T x),
\end{aligned}
$$

with $s(x) = \begin{pmatrix} 1 & x^T \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix}$ where $S \in \mathcal{S}_+^{n+1}$, $\alpha_i(x) = \alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix}$, $\beta_i(x) = \beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix}$, and $\eta_i(x) = \eta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix}$ where $\alpha_i, \beta_i, \eta_i \in \mathcal{L}^n$, $\delta_i(x) \in R_1[x]$, $\gamma_i, \theta_i \in \mathbb{R}$, and $\xi_i \in \mathbb{R}_+$. The formulation becomes

$$
\begin{aligned}
(\mathbf{BQP}_{\bar{\mathcal{K}}_0})\ \min\ & \lambda \\
\text{s.t.}\ & \lambda - q(x) = \begin{pmatrix} 1 & x^T \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + \sum_i (1+x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
& \quad + \sum_i \gamma_i(1-x_i^2) + \sum_i \delta_i(x)(b_i - a_i x) + \sum_i (d_i - c_i x)\eta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
& \quad + \sum_i \theta_i(k_i - x^T F_i x - e_i^T x) + \sum_i \xi_i(l_i - x^T G_i x - h_i^T x), \\
& S \in \mathcal{S}_+^{n+1}, \qquad \alpha_i, \beta_i, \eta_i \in \mathcal{L}^n, \qquad \gamma_i, \theta_i \in \mathbb{R}, \qquad \xi_i \in \mathbb{R}_+.
\end{aligned}
$$

The $(\mathrm{BQP}_{\bar{\mathcal{K}}_{\prime}})$ can further be relaxed by removing the positive semidefinite variable. This gives the following:

$$
\begin{aligned}
&\min\ \lambda \\
&\text{s.t. } \lambda - q(x) = \sum_i (1+x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \gamma_i (1-x_i^2) \\
&\qquad + \sum_i \delta_i(x)(b_i - a_i x) + \sum_i (d_i - c_i x)\eta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \theta_i (k_i - x^T F_i x - e_i^T x) \\
&\qquad + \sum_i \xi_i (l_i - x^T G_i x - h_i^T x), \\
&\quad \alpha_i, \beta_i, \eta_i \in \mathcal{L}^n, \qquad \gamma_i, \theta_i \in \mathbb{R}, \qquad \xi_i \in \mathbb{R}_+.
\end{aligned}
$$

To strengthen the SOCP relaxation for the BQP we can add cuts to the original problem, which is equivalent to adding more variables to the above problem due to the next lemma.

Lemma 3.1. *For any S, d, and $f \in \mathbf{R}_d[x]$*

$$
\mathcal{P}_d\left(S \cap \{x : f(x) \geq 0\}\right) \supseteq \mathcal{P}_d(S) + f(x)\mathcal{P}_{d-\deg(f)}(S).
$$

The first type of valid inequalities that we consider for the binary quadratic problem is adding the following constraints:

$$
-1 \leq x_i x_j \leq 1. \tag{7}
$$

Thus we obtain the following relaxation:

$$
\begin{aligned}
&(\mathbf{BQP_{SOCP}})\ \min\ \lambda \\
&\text{s.t. } \lambda - q(x) = \sum_i (1+x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \gamma_i (1-x_i^2) \\
&\qquad + \sum_i \delta_i(x)(b_i - a_i x) + \sum_i (d_i - c_i x)\eta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \theta_i (k_i - x^T F_i x - e_i^T x) \\
&\qquad + \sum_i \xi_i (l_i - x^T G_i x - h_i^T x) + \sum_{i,j} \nu_{ij}(1 - x_i x_j) + \sum_{i,j} \rho_{ij}(1 + x_i x_j) \\
&\quad \alpha_i, \beta_i, \eta_i \in \mathcal{L}^n, \qquad \gamma_i, \theta_i \in \mathbb{R}, \qquad \xi_i, \nu_{ij}, \rho_{ij} \in \mathbb{R}_+.
\end{aligned}
$$

The second type of valid inequalities correspond to the triangle inequalities, i.e., the inequalities of the form

$$
\begin{aligned}
x_i x_j + x_j x_k + x_k x_i &\geq -1 \qquad (8) \\
x_i x_j - x_j x_k - x_k x_i &\geq -1 \qquad (9) \\
-x_i x_j + x_j x_k - x_k x_i &\geq -1 \qquad (10) \\
-x_i x_j - x_j x_k + x_k x_i &\geq -1. \qquad (11)
\end{aligned}
$$

Using Lemma 3.1, we have that the following is a relaxation for the BQP:

$$
\begin{aligned}
(\mathbf{BQP}_{\mathbf{SOCP-\triangle}})\ \min\ & \lambda \\
\text{s.t. } \lambda - q(x) = & \sum_i (1+x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \gamma_i(1-x_i^2) \\
& + \sum_i \delta_i(x)(b_i - a_i x) + \sum_i (d_i - c_i x)\eta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \theta_i(k_i - x^T F_i x - e_i^T x) \\
& + \sum_i \xi_i(l_i - x^T G_i x - h_i^T x) + \sum_{i,j} \nu_{ij}(1 - x_i x_j) + \sum_{i,j} \rho_{ij}(1 + x_i x_j) \\
& + \sum_{i<j<k} \mu_{ijk}^1 (1 + x_i x_j + x_i x_k + x_j x_k) \\
& + \sum_{i<j<k} \mu_{ijk}^2 (1 + x_i x_j - x_i x_k - x_j x_k) \\
& + \sum_{i<j<k} \mu_{ijk}^3 (1 - x_i x_j + x_i x_k - x_j x_k) \\
& + \sum_{i<j<k} \mu_{ijk}^4 (1 - x_i x_j - x_i x_k + x_j x_k) \\
& \alpha_i, \beta_i, \eta_i \in \mathcal{L}^n, \qquad \gamma_i, \theta_i \in \mathbb{R}, \qquad \xi_i,\ \nu_{ij}, \rho_{ij},\ \mu_{ijk}^1,\ \mu_{ijk}^2,\ \mu_{ijk}^3,\ \mu_{ijk}^4 \in \mathbb{R}_+.
\end{aligned}
$$

There are $4\binom{n}{3}$ of the variables corresponding to the triangle inequalities. As n increases then the problem becomes too large to consider all these variables explicitly. ($\text{BQP}_{\text{SOCP}-\triangle}$) has $2n+u$ second-order cone variables, $n+t(n+1)+v+w+2\binom{n}{2}+4\binom{n}{3}$ linear variables, and $\binom{n+2}{2}$ constraints. Most of the μ_{ijk} variables will be non-basic and would have a value of zero at optimality, therefore only a subset of variables needs to be considered when solving the problem. The μ_{ijk} variables can be viewed as the dual variables of constraints (8)-(11). Therefore, we use an LP-column-generation type approach to iteratively generate some of these variables that have the potential to improve the objective function. This is done according to Algorithm 1.

Algorithm 1 (BQP_{SOCP})-Column Generation

1. Solve (BQP_{SOCP}) and obtain the dual solution X^*.
2. Sort the triangle inequalities (8)-(11) by magnitude of violation.
3. If no violated inequalities are found, then stop.
4. Generate a variable for each of the ϱ most violated triangle inequalities.
5. Add these variables to (BQP_{SOCP}).
6. Go back to Step 1.

4 Applications

In this section we apply the solution methodology proposed in Section 3.2 to two well-known problems, the max-cut problem and the quadratic knapsack problem.

4.1 Maximum-Cut Problem

The max-cut problem is one of the basic $\mathcal{NP}$-hard combinatorial optimization problems and has attracted scientific interest from both the discrete [10, 26] and the continuous optimization communities [14]. The max-cut problem has applications arising in statistical physics namely the spin glass problem [26], and in telecommunications for signal decoding [27]. An LP-based [17] and an SDP-based server [39] are available in the public domain to solve max-cut problems.

The max-cut problem is: Given an edge-weighted undirected graph $G = (V, E, W)$ where V is the vertex set, E is the edge set, and $W = \{w_e\}_{e \in E}$ are the edge weights, partition the vertex set V into two disjoint sets, V_1 and V_2 (where $V_1 \cap V_2 = \phi$ and $V_1 \cup V_2 = V$), such that the sum of the weights of the edges cut (having one endpoint in V_1 and the other in V_2) is maximized.

To formulate the max-cut problem as a BQP, define the binary variable x_i by

$$x_i = \begin{cases} 1 & \text{if } i \in V_1 \\ -1 & \text{if } i \in V_2. \end{cases}$$

The product of $x_i x_j$ is 1 if vertices i and j are not cut and -1 otherwise. A standard formulation is the following:

$$\begin{aligned} \textbf{(MC)} \quad & \max \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) \\ & \text{s.t. } x_i \in \{-1, 1\} \qquad\qquad \forall 1 \le i \le n. \end{aligned}$$

By setting $\hat{L} = \frac{1}{4} L$ where $L = \operatorname{Diag}(We) - W$ is known as the Laplacian matrix of G, the objective function can be written as $x^T \hat{L} x$. Hence finding the maximal cut in a graph is equivalent to solving the unconstrained binary quadratic problem. We note that in general, the Laplacian matrix L (and hence $\hat{L}$) is not necessary negative semidefinite and hence the problem may be non-convex.

4.1.1 Polynomial-based Relaxations of Max-Cut

In this section, we propose and describe several relaxations for max-cut based on polynomial programming. Using (1), we obtain that (MC) is equivalent to

$$\begin{aligned} & \min \lambda \\ & \text{s.t. } \lambda - \frac{1}{4} \sum_{i,j} L_{ij} x_i x_j \in \mathcal{P}_2(\{-1, 1\}^n). \end{aligned}$$

Let $\bar{q}(x) = \lambda - \frac{1}{4} \sum_{i,j} L_{ij} x_i x_j$. The simplest tractable approximations of $\mathcal{P}_2(\{-1, 1\}^n)$ is using the cone $\mathcal{K}_0'$:

$$\mathcal{P}_2(\{-1, 1\}^n) \supseteq \mathcal{K}_0' = \Sigma_2^2 + \sum_i (1 - x_i^2) \mathbf{R}_0.$$

$$\begin{aligned} \textbf{(MC}_{\mathcal{K}_0'}\textbf{)} \quad & \min \lambda \\ & \text{s.t. } \lambda - \frac{1}{4} \sum_{i,j} L_{ij} x_i x_j = \begin{pmatrix} 1 & x^T \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + \sum_{i=1}^{n} c_i (1 - x_i^2) \\ & c \in \mathbb{R}^n \qquad S \in \mathcal{S}_+^{n+1}. \end{aligned}$$

Hence, we obtain a relaxation of (MC). For every monomial we have a constraint, so there are $1+n+\binom{n+1}{2}$ equality constraints and the problem can be then written as follows:

$$\begin{aligned}
&\min\ \lambda && (12)\\
&\text{s.t. } \lambda-\sum_i c_i - S_{00} = 0 && (13)\\
&\qquad S_{ij} = -\frac{1}{4}L_{ij} \qquad 1\le i,j\le n,\ i\neq j && (14)\\
&\qquad S_{0i} = 0 \qquad 1\le i\le n && (15)\\
&\qquad S_{ii} - c_i = -L_{ii} \qquad 1\le i\le n && (16)\\
&\qquad c\in\mathbb{R}^n \qquad S\in\mathcal{S}_+^{n+1}. && (17)
\end{aligned}$$

The Goemans and Williamson [12] SDP-based relaxation of (MC) is:

$$\begin{aligned}
\textbf{(MC}_{\textbf{GW}}\textbf{)}\ \ &\max\ \frac{1}{4}L\bullet X\\
&\text{s.t. } X_{ii}=1 \qquad (18)\\
&\qquad X\succeq 0,
\end{aligned}$$

where $\bullet$ denotes the inner product.

Lemma 4.1. *The optimal values of ($MC_{\mathcal{K}_0'}$) and (MC_{GW}) are equal.*

Proof. From constraint (16) $S_{ii}+L_{ii}=c_i$. So, we can re-write ($\mathrm{MC}_{\mathcal{K}_0'}$) as follows:

$$\begin{aligned}
&\min\ \sum_{i=1}^n S_{ii} + \sum_{i=1}^n L_{ii} + S_{00}\\
&\text{s.t. } S_{ij} = -\frac{1}{4}L_{ij} \qquad 1\le i,j\le n,\ i\neq j\\
&\qquad S\in\mathcal{S}_+^n, \quad S_{00}\ge 0.
\end{aligned}$$

We have $S_{00}=0$ at optimality, then the above problem can be viewed as the dual of ($\mathrm{MC}_{\mathrm{GW}}$). For the primal problem ($\mathrm{MC}_{\mathrm{GW}}$), Slater constraint qualification holds since taking $\tilde{X}=I$, we get $\tilde{X}$ is strictly feasible to ($\mathrm{MC}_{\mathrm{GW}}$). In addition, the optimal value of ($\mathrm{MC}_{\mathrm{GW}}$) is finite. Thus strong duality holds for ($\mathrm{MC}_{\mathrm{GW}}$) and hence the optimal values of ($\mathrm{MC}_{\mathcal{K}_0'}$) and ($\mathrm{MC}_{\mathrm{GW}}$) are equal. $\square$

4.1.2 Second-Order Cone Relaxation of Max-Cut

Using the results of Section 3.2, we can formulate the max-cut problem as follows:

$$\begin{aligned}
\textbf{(MC}_{\textbf{SS}}\textbf{)}\ &\min\ \lambda\\
&\text{s.t. } \lambda - q(x) = \begin{pmatrix}1 & x^T\end{pmatrix} S \begin{pmatrix}1\\ x\end{pmatrix} + \sum_i (1+x_i) f_i^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_i (1-x_i) g_i^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix}\\
&\qquad + \sum_i c_i(1-x_i^2)\\
&\qquad f_i, g_i\in\mathcal{L}^n, \quad S\in\mathcal{S}_+^{n+1}, \quad c_i\in\mathbb{R}.
\end{aligned}$$

(MC$_{\text{SS}}$) can be further relaxed by removing the positive semidefinite variable and adding the valid inequalities $-1 \leq x_i x_j \leq 1$. This gives the following SOCP relaxation:

$$
\begin{aligned}
\textbf{(MC}_{\textbf{SOCP}}\textbf{)}\ \min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = \sum_i (1+x_i) f_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i) g_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i c_i (1-x_i^2) \\
& \qquad + \sum_{i<j} d_{ij}(1-x_i x_j) + \sum_{i<j} e_{ij}(1+x_i x_j) \\
& f_i, g_i \in \mathcal{L}^n, \qquad c_i \in \mathbb{R}, \qquad d_{ij}, e_{ij} \in \mathbb{R}_+.
\end{aligned}
$$

The SOCP relaxation can be improved by adding triangle inequalities as discussed in Section 3.2:

$$
\begin{aligned}
\textbf{(MC}_{\textbf{SOCP}-\triangle}\textbf{)}\ \min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = \sum_i (1+x_i) f_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i) g_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
& \qquad + \sum_i c_i (1-x_i^2) + \sum_{i<j} d_{ij}(1-x_i x_j) + \sum_{i<j} e_{ij}(1+x_i x_j) \\
& \qquad + \sum_{i<j<k} h^1_{ijk}(1 + x_i x_j + x_i x_k + x_j x_k) \\
& \qquad + \sum_{i<j<k} h^2_{ijk}(1 + x_i x_j - x_i x_k - x_j x_k) \\
& \qquad + \sum_{i<j<k} h^3_{ijk}(1 - x_i x_j + x_i x_k - x_j x_k) \\
& \qquad + \sum_{i<j<k} h^4_{ijk}(1 - x_i x_j - x_i x_k + x_j x_k) \\
& f_i, g_i \in \mathcal{L}^n, \quad c_i \in \mathbb{R}, \quad d_{ij}, e_{ij},\ h^1_{ijk},\ h^2_{ijk},\ h^3_{ijk},\ h^4_{ijk} \in \mathbb{R}_+.
\end{aligned}
$$

Remark 4.2. *For the max-cut problem, an unconstrained binary quadratic problem, the objective function involves only quadratic terms. In Proposition 4.3 we show that removing the second-order cone variables from (MC$_{SOCP}$) will not affect its optimal objective function value. However, without loss of generality if we fix one of the vertices of the graph to be in one partition (e.g. fix x_1=1) then the objective function will have linear terms and computational results show that having the second-order cone variables in (MC$_{SOCP}$) will then improve the objective bound.*

Proposition 4.3. *Let (MC'$_{SOCP}$) be the problem obtained when removing the second-order cone variables from (MC$_{SOCP}$). Then the optimal value of both problems is the same.*

Proof. Let (λ, f, g, c, d, e) be a feasible solution for (MC$_{\text{SOCP}}$) then

$$
\begin{aligned}
\lambda - q(x) &= \sum_i (1+x_i) f_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i) g_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i c_i (1-x_i^2) + \sum_{i<j} d_{ij}(1-x_i x_j) + \sum_{i<j} e_{ij}(1+x_i x_j) \\
&= \sum_i (c_i - f_{ii} + g_{ii})(1-x_i^2) + \sum_{i<j} (d_{ij} + d'_{ij})(1 - x_i x_j) + \sum_{i<j} (e_{ij} + e'_{ij})(1 + x_i x_j) \\
&\quad + \sqrt{n} \sum_i (f_{i0} + g_{i0}) + \sum_i (f_{ii} - g_{ii}) - \sum_{i<j} (d'_{ij} + e'_{ij}),
\end{aligned}
$$

where $d'_{ij} = \min(0, -f_{ij} - f_{ji} + g_{ij} + g_{ji})$ and $e'_{ij} = \min(0, f_{ij} + f_{ji} - g_{ij} - g_{ji})$.

and thus, we have a feasible solution for

$$\begin{aligned} &\min \lambda \\ &\text{s.t. } \lambda - q(x) = \sum_i c_i(1 - x_i^2) + \sum_{i<j} d_{ij}(1 - x_i x_j) + \sum_{i<j} e_{ij}(1 + x_i x_j) \\ &\quad f_i, g_i \in \mathcal{L}^n, \qquad c_i \in \mathbb{R}, \qquad d_{ij}, e_{ij} \in \mathbb{R}_+. \end{aligned}$$

with objective value

$$\bar{\lambda} := \lambda - \sqrt{n} \sum_i (f_{i0} + g_{i0}) - \sum_i (f_{ii} - g_{ii}) + \sum_{i<j} (d'_{ij} + e'_{ij}) \geq \lambda.$$

□

To generate a feasible solution from ($\mathrm{MC}_{\mathrm{SOCP}-\triangle}$), we can use the dual solution to form the matrix X^*. We cannot apply Goemans and Williamson rounding on X^*, as this matrix is most likely not positive semidefinite. In Algorithm 2, we suggest a slightly different approach from [12] to be able to generate feasible solutions for the max-cut problem.

Algorithm 2 ($\mathrm{MC}_{\mathrm{SOCP}}$)-Rounding

1. Solve ($\mathrm{MC}_{\mathrm{SOCP}-\triangle}$) to obtain the dual solution matrix X^*.
2. Find $(\lambda_1, \cdots, \lambda_n)$, all (with repetitions) the eigenvalues of X^* with $L = (l_1, \cdots, l_n)$ the corresponding eigenvectors.
3. Remove eigenvectors that have a zero eigenvalue to get $\tilde{L} = (l_1, \cdots, l_k)$.
4. Choose a unit random vector $r \in \mathbb{R}^k$ and compute $\tilde{l} = \tilde{L}r$.
5. Partition the vertex set V into V_1 and V_2 according to $V_1 = \{i : \tilde{l}_i \geq 0\}$ and $V_2 = \{i : \tilde{l}_i < 0\}$.

4.2 Higher Order Relaxations of Max-Cut

Lasserre [22] introduced SDP relaxations corresponding to liftings of polynomial 0-1 problems into higher dimensions. He applied the SDP liftings to quadratic 0-1 programs and max-cut problems. A higher level relaxation of the max-cut using $\mathcal{K}'_2$ is as follows [22]:

$$\begin{aligned} (\mathbf{MC}_{\mathcal{K}'_2})\ &\min \lambda \\ &\text{s.t. } \lambda - \frac{1}{4}\sum_{i,j} L_{ij} x_i x_j = s(x) + \sum_{i=1}^{n} c_i(x)(1 - x_i^2) \\ &\quad c_i(x) \in \mathbf{R}_2[x] \qquad s(x) \in \Sigma_4^2. \end{aligned}$$

($\mathrm{MC}_{\mathcal{K}'_2}$) has one SDP variable of dimension $\binom{n+2}{2} \times \binom{n+2}{2}$, $n\binom{n+2}{2}$ linear variables, and $\binom{n+4}{4}$ constraints. By eliminating the $c_i(x)$'s variables, we can reduce the problem to a problem with one SDP variable of dimension $(\binom{n+1}{2} + 1) \times (\binom{n+1}{2} + 1)$ and $1 + n + \binom{n}{2} + \binom{n}{3} + \binom{n}{4}$ constraints. Even after the reduction, the problem is still impractical to solve as n, the size of the graph, becomes large.

A lifting of the SDP relaxation of max-cut was independently developed by Anjos and Wolkowicz [4]. The ($\mathrm{MC}_{\mathrm{AW}_p}$) relaxation presented in [4] is:

$$
\begin{aligned}
(\mathbf{MC}_{\mathbf{AW}_p})\ \max\ & \hat{Q} \bullet Y \\
\text{s.t. } & Y_{00,00} = 1 && && (19)\\
& Y_{ij,ij} = 1 && 1 \le i < j \le n, && (20)\\
& Y_{ij,jk} = Y_{0,ik} && \forall\, 1 \le i < k \le n,\ 1 \le j \le n\ j \ne k,\ j \ne i, && (21)\\
& Y \in \mathcal{S}_+^{\binom{n}{2}+1}, && && (22)
\end{aligned}
$$

where

$$
\hat{Q} = \left(\begin{array}{c|c} \sum_{i=1}^{n} \hat{L}_{ii} & \hat{L}_{ij} \\ \hline \hat{L}_{ij} & 0 \end{array} \right),
$$

for $i < j$.

Lemma 4.4. *Let $\lambda^*_{MC_{AW_p}}$ and $\lambda^*_{\mathcal{K}'_2}$ be the optimal values of (MC_{AW_p}) and ($MC_{\mathcal{K}'_2}$) respectively, then*

$$
\lambda^*_{MC_{AW_p}} \ge \lambda^*_{MC_{\mathcal{K}'_2}} \ge z^*_{MC}.
$$

Proof. Consider the dual of ($\mathrm{MC}_{\mathrm{AW}_p}$):

$$
\begin{aligned}
(\mathbf{MC}_{\mathbf{AW}_p\text{-}\mathbf{D}})\ \ \min\ & \mu_0 + \sum_{i<j} \mu_{ij} \\
\text{s.t. } & -\hat{Q} + \mathrm{Diag}(\mu_0, \mu_{ij}) - \sum_{i \ne j \ne k,\ i<k} \nu_{ijk} A_{ijk} = S && (23)\\
& S \in \mathcal{S}_+^{\binom{n}{2}+1}, \quad \mu,\ \nu \in \mathbb{R}.
\end{aligned}
$$

Let $\langle i, j\rangle := \frac{j(j-1)}{2} + i$ for $i \le j$ and $\langle i, j\rangle := \frac{i(i-1)}{2} + j$ otherwise. We have μ_0, μ_{ij} and ν_{ijk} are the dual variables of constraints (19), (20) and (21) respectively and $A_{ijk} \in \mathcal{S}^{\binom{n}{2}+1}$ is equal to 1 in the $(\langle i, j\rangle, \langle j, k\rangle)$ position, -1 in the $(0, \langle i, k\rangle)$ position, and zero elsewhere. From (23),

$$
\begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}^T \left(-\hat{Q} + \mathrm{Diag}(\mu_0, \mu_{ij}) - \sum_{i \ne j \ne k,\ i<k} \nu_{ijk} A_{ijk} \right) \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix} = \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}^T S \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}.
$$

Using

$$
\begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}^T \hat{Q} \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix} = q(x) + \sum_i \hat{L}_{ii}(1 - x_i^2)
$$

and

$$
\begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}^T A_{ijk} \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix} = x_i x_j x_j x_k - x_i x_k
$$

and defining

$$
s(x) = \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}^T S \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix},
$$

we obtain

$$\mu_0 - q(x) - \sum_i \hat{L}_{ii}(1 - x_i^2) = s(x) - \sum_{i<j} \mu_{ij} x_i^2 x_j^2 + \sum_{i \neq j \neq k,\, i<k} \nu_{ijk}(x_i x_j x_j x_k - x_i x_k).$$

Then, we have that ($\text{MC}_{\text{AW}_p\text{-D}}$) is equivalent to

$$\begin{aligned}
\textbf{(MC}_{\textbf{AW}_p\textbf{-DP}}\textbf{)} \quad & \min\ \mu_0 + \sum_{i<j} \mu_{ij} \\
& \text{s.t.}\ \ \mu_0 + \sum_{i<j} \mu_{ij} - q(x) = \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix}^T S \begin{pmatrix} 1 \\ x_i x_j \end{pmatrix} + \sum_i \hat{L}_{ii}(1 - x_i^2) \\
& \qquad\qquad - \sum_{i<j} \mu_{ij} \left(x_i^2 (x_j^2 - 1) + x_i^2 - 1 \right) + \sum_{i \neq j \neq k,\, i<k} \nu_{ijk} x_i x_k (x_j^2 - 1) \\
& \qquad S \in \mathcal{S}_+^{\binom{n}{2}+1}, \quad \mu,\ \nu \in \mathbb{R}.
\end{aligned}$$

Now given (S^*, μ^*, ν^*) to be an optimal solution to ($\text{MC}_{\text{AW}_p\text{-DP}}$) we define $\lambda = \mu_0^* + \sum_{i<j} \mu_{ij}^*$, $S = S^*$, and $c_i(x) = \hat{L}_{ii} + \sum_j \mu_{ij}^* x_j^2 + \sum_j \mu_{ij}^* + \sum_{i \neq j \neq k,\, j<k} \nu_{ijk} x_j x_k$. Hence, $(\lambda, S, c_i(x))$ is a feasible solution for ($\text{MC}_{\mathcal{K}_2'}$). Therefore,

$$\lambda^*_{\text{MC}_{\mathcal{K}_2'}} \leq \lambda = \mu_0 + \sum_{i<j} \mu_{ij} = \lambda^*_{\text{MC-AW}_p}\,.$$

We note that $\mu_0 + \sum_{i<j} \mu_{ij} = \lambda^*_{\text{MC}_{\text{AW}_p}}$ holds since both the primal (MC_{AW_p}) and the dual ($\text{MC}_{\text{AW}_p\text{-D}}$) problem satisfy Slater's constraint qualification [2]. □

Using the same idea as presented in Section 3.2 for relaxing BQP using second-order cones, we can apply this approach to the lifted max-cut relaxation. Taking the vector $\bar{X}$ to be the vector consisting of the upper diagonal entries of matrix $X = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$, i.e., $\bar{X} = \begin{pmatrix} x_i \\ X_{ij} \end{pmatrix}$ where $i < j$, we have

$$\begin{aligned}
\textbf{(MC}_{\textbf{SS-Lift}}\textbf{)} \quad & \min\ \lambda \\
& \text{s.t.}\ \ \lambda - \frac{1}{4} \sum_{i<j} L_{ij} \bar{X}_{\langle i,j \rangle} = \begin{pmatrix} 1 \\ \bar{X} \end{pmatrix}^T S \begin{pmatrix} 1 \\ \bar{X} \end{pmatrix} + \sum_{t=1}^{\binom{n+1}{2}} (1 + \bar{X}_t) f_t^T \begin{pmatrix} \sqrt{\binom{n+1}{2}} \\ \bar{X} \end{pmatrix} \\
& \qquad + \sum_{t=1}^{\binom{n+1}{2}} (1 - \bar{X}_t) g_t^T \begin{pmatrix} \sqrt{\binom{n+1}{2}} \\ \bar{X} \end{pmatrix} + \sum_{t=1}^{\binom{n+1}{2}} c_t (1 - \bar{X}_t \bar{X}_t) \\
& \qquad + \sum_{i \neq j \neq k, i<k} h_{ijk} (\bar{X}_{\langle i,j \rangle} \bar{X}_{\langle j,k \rangle} - \bar{X}_{\langle i,k \rangle}) \\
& \qquad S \in \mathcal{S}_+^{\binom{n+1}{2}+1}, \quad f_t, g_t \in \mathcal{L}^{\binom{n+1}{2}}, \quad c_t,\ h_{ijk} \in \mathbb{R}.
\end{aligned}$$

We further relax ($\text{MC}_{\text{SS-Lift}}$) by removing the positive semidefinite variable S to obtain a second-order cone relaxation as follows:

$$
\begin{aligned}
(\mathbf{MC_{SOCP\text{-}Lift}})\ \min\ & \lambda \\
\text{s.t.}\ & \lambda - \frac{1}{4}\sum_{i<j} L_{ij}\bar{X}_{\langle i,j\rangle} = \sum_{t=1}^{\binom{n+1}{2}} (1+\bar{X}_t) f_t^T \begin{pmatrix} \sqrt{\binom{n+1}{2}} \\ \bar{X} \end{pmatrix} + \sum_{t=1}^{\binom{n+1}{2}} (1-\bar{X}_t) g_t^T \begin{pmatrix} \sqrt{\binom{n+1}{2}} \\ \bar{X} \end{pmatrix} \\
& + \sum_{t=1}^{\binom{n+1}{2}} c_t(1-\bar{X}_t\bar{X}_t) + \sum_{i\neq j\neq k, i<k} h_{ijk}(\bar{X}_{\langle i,j\rangle}\bar{X}_{\langle j,k\rangle} - \bar{X}_{\langle i,k\rangle}) \\
& + \sum_{i<j} e_{ij}(\bar{X}_{\langle i,j\rangle} - \bar{X}_i\bar{X}_j) \\
& f_t, g_t \in \mathcal{L}^{\binom{n+1}{2}}, \quad c_t,\ h_{ijk},\ e_{ij} \in \mathbb{R}.
\end{aligned}
$$

Hence, we obtain an SOCP relaxation for the lifted max-cut problem. One can also add the triangle inequalities to ($\mathrm{MC_{SOCP\text{-}Lift}}$) to strengthen the relaxation further.

4.3 Quadratic Knapsack Problem

In order to gain insight into the quality of our polynomial programming-based relaxations when applied to constrained quadratic binary problems, we study in this section the quadratic knapsack problem (QKP) and the quadratic multiple knapsack problem (QMKP). For the QKP, we are given n items with a non-negative weight w_i assigned to item i, and an $n \times n$ symmetric profit matrix P with real entries, where P_{ij} is the profit achieved if both items i and j are selected and P_{ii} is the profit achieved once item i is selected. The QKP is the problem of selecting a subset of items so as to maximize the overall profit such that the total weight of the selected items does not exceed a given capacity c. The QKP was introduced by Gallo, Hammer, and Simeone [11] and is $\mathcal{NP}$-hard. Applications of the QKP include portfolio selection [24], where one can invest in a pool of assets subject to budget constraint and the objective is to maximize the overall profit (or minimize the risk) with the coefficients in the objective being the pairwise interrelation between the assets.

The QKP is a particular case of the QMKP where we have a single knapsack constraint as opposed to having multiple knapsack constraints. Hence, the QMKP maximizes a quadratic objective function subject to m linear capacity constraints and the variables being binary. By introducing binary variables x_i such that

$$
x_i = \begin{cases} 1 & \text{if item } i \text{ is selected} \\ 0 & \text{otherwise,} \end{cases}
$$

the problem may be formulated as:

$$
\begin{aligned}
(\mathbf{QMKP})\ \max\ & \sum_{i,j} P_{ij}x_ix_j \\
\text{s.t.}\ & \sum_i w_{ij}x_i \leq c_j & 1 \leq j \leq m \\
& x \in \{0,1\}^n.
\end{aligned}
$$

The QMKP is a generalization of the linear knapsack problem (where the objective function is linear). As in the case of the linear knapsack problem, the QMKP can be used as a sub-problem to other complex problems such as the graph partitioning problem described in Johnson, Mehrotra, and Nemhauser [16]. Since the QMKP is a constrained version of the binary quadratic problem, all valid inequalities for the unconstrained BQP problem are also valid for the QKP and QMKP and hence they can be used to tighten bounds for these problems.

4.3.1 Polynomial-based Relaxations of QMKP

For convenience, we replace the discrete set $\{0,1\}^n$ by $\{-1,1\}^n$ by making the change of variables $x_i = \frac{1+\bar{x}_i}{2}$. The objective function can be expressed as

$$p(x) = \sum_{i,j} P_{ij}x_ix_j = \frac{1}{4}\sum_{i,j} P_{ij}(1+\bar{x}_i)(1+\bar{x}_j) = \frac{1}{4}\sum_{i,j} P_{ij} + \frac{1}{4}\sum_{i} P_{ij}\bar{x}_i + \frac{1}{4}\sum_{j} P_{ij}\bar{x}_j + \frac{1}{4}\sum_{i,j} P_{ij}\bar{x}_i\bar{x}_j.$$

Each knapsack constraint j can be written as

$$\frac{1}{2}\sum_i w_{ij}(1+\bar{x}_i) \le c_j \Rightarrow \sum_i w_{ij}\bar{x}_i \le 2c_j - \sum_i w_{ij}.$$

Taking $\bar{c}_j = 2c_j - \sum_i w_{ij}$ and the matrix $Q \in \mathcal{S}^{n+1}$ as follows

$$Q = \begin{pmatrix} \frac{1}{4}\sum_{i,j} P_{ij} & \frac{1}{4}e^T P \\ \frac{1}{4}Pe & \frac{1}{4}P \end{pmatrix},$$

the QMKP can be written as the polynomial programming problem

$$\begin{aligned} \max\ & q(\bar{x}) = \begin{pmatrix} 1 & \bar{x} \end{pmatrix} Q \begin{pmatrix} 1 \\ \bar{x} \end{pmatrix} \\ \text{s.t. }\ & w_j^T\bar{x} \le \bar{c}_j \qquad\qquad 1 \le j \le m \\ & \bar{x} \in \{-1,1\}^n. \end{aligned}$$

Using (1), we obtain the equivalent problem,

$$\begin{aligned} \min\ & \lambda \\ \text{s.t. }\ & \lambda - q(\bar{x}) \in \mathcal{P}_2(\{-1,1\}^n \cap \{\bar{x} : w_j^T\bar{x} \le \bar{c}_j\}). \end{aligned}$$

Using the same approach as in Section 4.1.1, we can approximate the cone $\mathcal{P}_2(\{-1,1\}^n \cap \{\bar{x} : w_j^T\bar{x} \le \bar{c}_j\})$ using

$$\begin{aligned} \mathcal{P}_2(\{-1,1\}^n \cap \{\bar{x} : w_j^T\bar{x} \le \bar{c}_j\} \cap \mathcal{B}) \supseteq \tilde{\mathcal{K}} =& \Sigma_2^2 + \sum_{j=1}^m (\bar{c}_j - w_j^T\bar{x})\mathcal{P}_1(\mathcal{B}) + \sum_{i=1}^n (1+\bar{x}_i)\mathcal{P}_1(\mathcal{B}) \\ &+ \sum_{i=1}^n (1-\bar{x}_i)\mathcal{P}_1(\mathcal{B}) + \sum_{i=1}^n (1-\bar{x}_i^2)\mathbf{R}_0. \end{aligned}$$

Therefore, we have $\lambda - q(\bar{x}) \in \tilde{\mathcal{K}}$ if and only if

$$\lambda - q(\bar{x}) = s(\bar{x}) + \sum_{j=1}^m (\bar{c}_j - w_j^T\bar{x})d_j(\bar{x}) + \sum_{i=1}^n (1+\bar{x}_i)f_i(\bar{x}) + \sum_{i=1}^n (1-\bar{x}_i)g_i(\bar{x}) + \sum_{i=1}^n c_i(1-\bar{x}_i^2),$$

with $f_i(\bar{x}), g_i(\bar{x})$, and $d_j(\bar{x}) \in \mathcal{P}_1(\mathcal{B})$, $s(\bar{x})$ is SOS, and $c_i \in \mathbb{R}$. Hence, we obtain a relaxation of (QMKP):

$$
\begin{aligned}
\textbf{(QMKP}_{\textbf{ss}}\textbf{)}\ \min\ & \lambda \\
\text{s.t. }\ & \lambda - q(\bar{x}) = \begin{pmatrix}1 & \bar{x}\end{pmatrix} S \begin{pmatrix}1\\ \bar{x}\end{pmatrix} + \sum_{j=1}^{m}(\bar{c}_j - w_j^T\bar{x})d_j^T\begin{pmatrix}\sqrt{n}\\ \bar{x}\end{pmatrix} + \sum_{i=1}^{n}(1+\bar{x}_i)f_i{}^T\begin{pmatrix}\sqrt{n}\\ \bar{x}\end{pmatrix} \\
& \qquad + \sum_{i=1}^{n}(1-\bar{x}_i)g_i{}^T\begin{pmatrix}\sqrt{n}\\ \bar{x}\end{pmatrix} + \sum_{i=1}^{n} c_i(1-\bar{x}_i^2) \\
& c_i \in \mathbb{R}, \qquad f_i, g_i, d_j \in \mathcal{L}^n, \qquad S \in \mathcal{S}_+^{n+1}.
\end{aligned}
$$

Notice that $(1+\bar{x}_i)(\bar{c}_k - w_k^T\bar{x}) \geq 0$, $(1-\bar{x}_i)(\bar{c}_k - w_k^T\bar{x}) \geq 0$, $(1+\bar{x}_j)(1+\bar{x}_i) \geq 0$, $(1-\bar{x}_j)(1-\bar{x}_i) \geq 0$, $(1-\bar{x}_j)(1+\bar{x}_i) \geq 0$, and $(\bar{c}_k - w_k^T\bar{x})(\bar{c}_l - w_l^T\bar{x}) \geq 0$ for $1 \leq i \leq j \leq n$ and $1 \leq k \leq l \leq m$ are all valid inequalities for (QMKP). Using Lemma 3.1, we obtain the following stronger relaxation:

$$
\begin{aligned}
\textbf{(QMKP}_{\widehat{\mathcal{K}}}\textbf{)}\ \min\ & \lambda \\
\text{s.t. }\ & \lambda - q(\bar{x}) = \begin{pmatrix}1 & \bar{x}\end{pmatrix} S \begin{pmatrix}1\\ \bar{x}\end{pmatrix} + \sum_{j=1}^{m}(\bar{c}_j - w_j^T\bar{x})d_j^T\begin{pmatrix}\sqrt{n}\\ \bar{x}\end{pmatrix} + \sum_{i=1}^{n}(1+\bar{x}_i)f_i{}^T\begin{pmatrix}\sqrt{n}\\ \bar{x}\end{pmatrix} \\
& \qquad + \sum_{i=1}^{n}(1-\bar{x}_i)g_i{}^T\begin{pmatrix}\sqrt{n}\\ \bar{x}\end{pmatrix} + \sum_{i=1}^{n}\sum_{k=1}^{m}\alpha_{ik}(1+\bar{x}_i)(\bar{c}_k - w_k^T\bar{x}) \\
& \qquad + + \sum_{i=1}^{n}\sum_{k=1}^{m}\beta_{ik}(1-\bar{x}_i)(\bar{c}_k - w_k^T\bar{x}) + \sum_{i=1}^{n}\sum_{j=i}^{n}\gamma_{ij}(1+\bar{x}_i)(1+\bar{x}_j) \\
& \qquad + \sum_{i=1}^{n}\sum_{j=i}^{n}\delta_{ij}(1-\bar{x}_i)(1-\bar{x}_j) + \sum_{i=1}^{n}\sum_{j=1}^{n}\zeta_{ij}(1+\bar{x}_i)(1-\bar{x}_j) \\
& \qquad + \sum_{k=1}^{m}\sum_{l=k}^{m}\eta_{kl}(\bar{c}_k - w_k^T\bar{x})(\bar{c}_l - w_l^T\bar{x}) + \sum_{i=1}^{n} c_i(1-\bar{x}_i^2) \\
& c_i \in \mathbb{R}, \qquad \alpha_{ik}, \beta_{ik}, \gamma_{ij}, \delta_{ij}, \zeta_{ij}, \eta_{kl} \in \mathbb{R}_+^n, \qquad f_i, g_i, d_j \in \mathcal{L}^n, \qquad S \in \mathcal{S}_+^{n+1}.
\end{aligned}
$$

Another approach to strengthen the relaxation (QKP_{SS}) is to add the triangle inequalities:

$$
\begin{aligned}
\textbf{(QMKP}_{\textbf{SS}-\triangle}\textbf{)}\ \min\ & \lambda \\
\text{s.t. } & \lambda - q(\bar{x}) = \begin{pmatrix} 1 & \bar{x} \end{pmatrix} S \begin{pmatrix} 1 \\ \bar{x} \end{pmatrix} + \sum_{j=1}^{m} (\bar{c}_j - w_j^T \bar{x}) d_j^T \begin{pmatrix} \sqrt{n} \\ \bar{x} \end{pmatrix} + \sum_{i=1}^{n} (1 + \bar{x}_i) f_i{}^T \begin{pmatrix} \sqrt{n} \\ \bar{x} \end{pmatrix} \\
& \quad + \sum_{i=1}^{n} (1 - \bar{x}_i) g_i{}^T \begin{pmatrix} \sqrt{n} \\ \bar{x} \end{pmatrix} + \sum_{i=1}^{n} c_i (1 - \bar{x}_i^2) \\
& \quad + \sum_{i<j<k} h^1_{ijk} (1 + \bar{x}_i \bar{x}_j + \bar{x}_i \bar{x}_k + \bar{x}_j \bar{x}_k) \\
& \quad + \sum_{i<j<k} h^2_{ijk} (1 + \bar{x}_i \bar{x}_j - \bar{x}_i \bar{x}_k - \bar{x}_j \bar{x}_k) \\
& \quad + \sum_{i<j<k} h^3_{ijk} (1 - \bar{x}_i \bar{x}_j + \bar{x}_i \bar{x}_k - \bar{x}_j \bar{x}_k) \\
& \quad + \sum_{i<j<k} h^4_{ijk} (1 - \bar{x}_i \bar{x}_j - \bar{x}_i \bar{x}_k + \bar{x}_j \bar{x}_k) \\
& c_i \in \mathbb{R}, \qquad f_i, g_i, d_j \in \mathcal{L}^n, \qquad S \in \mathcal{S}_+^{n+1}, \qquad h^1_{ijk},\ h^2_{ijk},\ h^3_{ijk},\ h^4_{ijk} \in \mathbb{R}_+.
\end{aligned}
$$

As in Section 3.2, one can drop the SDP term and add valid inequalities of the form $-1 \leq x_i x_j \leq 1$ to obtain a second-order cone relaxation for the QMKP problem. We refer to these relaxations as ($\text{QMKP}_{\text{SOCP}}$) or ($\text{QMKP}_{\text{SOCP}-\triangle}$), depending on whether or not we add the triangle inequalities. We note that all the relaxations mentioned in this section can be applied to the QKP problem by considering $m = 1$.

4.3.2 Helmberg, Rendl, and Weismantel QKP Relaxation

Helmberg et al. [15] presented four SDP-based relaxations for the QKP. All of them are obtained by considering the semidefinite matrix $X = xx^T$. In particular they studied the relaxation

$$
\begin{aligned}
\textbf{(QKP}_{\textbf{HRW4}}\textbf{)}\ \max\ & \langle P, X \rangle \\
\text{s.t. } & \sum_j w_j X_{ij} - c X_{ii} \leq 0 \quad 1 \leq i \leq n \\
& X - \operatorname{diag}(X) \operatorname{diag}(X)^T \succeq 0,
\end{aligned}
$$

and showed that $z^*_{\text{QKP}_{\text{HRW4}}}$ provides the best bound among the SDP relaxations they provided. Actually, (QKP_{HRW4}) provides the tightest known SDP-relaxation for the QKP in the literature. We will be using this relaxation for comparison purposes in our computational results. In addition, Helmberg et al. [15] use cutting planes from linear inequalities that are valid for the binary quadratic problem. They present computational results instances with up to 61 items to illustrate the quality of these SDP relaxations and the cutting planes.

4.3.3 Burer QMKP Relaxation

Burer [8] presented an SDP-based relaxation for the QMKP. Consider the following relaxation:

$$
\begin{aligned}
(\mathbf{QKP_{Burer'_m}})\ \min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = \begin{pmatrix} 1 & x & s & t \end{pmatrix} (M+N) \begin{pmatrix} 1 \\ x \\ s \\ t \end{pmatrix} + \sum_{i=1}^{n} c_i x_i (1-x_i) + \sum_{i=1}^{n} (1 - x_i - s_i) l_i(x) \\
& \qquad + \sum_{j=1}^{m} (c_j - w_j^T x - t_j) k_j(x) \\
& c_i \in \mathbb{R}, \qquad l_i, k_i \in \mathbf{R}_1[x,s,t], \qquad M \in \mathcal{S}_+^{2n+m+1}, \qquad N \in \mathbb{R}_+^{2n+m+1},
\end{aligned}
$$

where m is the number of knapsack constraints. From [43], it follows that the above relaxation is at least as strong as the relaxation presented by Burer. Further, $(\mathrm{QKP_{Burer'_m}})$ is equivalent to:

$$
\begin{aligned}
\min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = \begin{pmatrix} 1 & x & 1-x & c - w^T x \end{pmatrix} (M+N) \begin{pmatrix} 1 \\ x \\ 1-x \\ c - w^T x \end{pmatrix} + \sum_{i=1}^{n} c_i x_i (1-x_i) \\
& c_i \in \mathbb{R}, \qquad M \in \mathcal{S}_+^{2n+m+1}, \qquad N \in \mathbb{R}_+^{2n+m+1},
\end{aligned}
$$

which can be written as

$$
\begin{aligned}
\min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = \begin{pmatrix} 1 & x \end{pmatrix} M' \begin{pmatrix} 1 \\ x \end{pmatrix} + \sum_{i=1}^{n} c_i x_i (1-x_i) \\
& \qquad + \sum_{i=1}^{n} \sum_{k=1}^{m} \alpha_{ik} x_i (c_k - w_k^T x) + \sum_{i=1}^{n} \sum_{k=1}^{m} \beta_{ik} (1-x_i)(c_k - w_k^T x) \\
& \qquad + \sum_{i=1}^{n} \sum_{j=i}^{n} \gamma_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=i}^{n} \delta_{ij} (1-x_i)(1-x_j) \\
& \qquad + \sum_{i=1}^{n} \sum_{j=1}^{n} \zeta_{ij} x_i (1-x_j) + \sum_{k=1}^{m} \sum_{l=k}^{m} \eta_{kl} (c_k - w_k^T x)(c_l - w_l^T x) \\
& c_i \in \mathbb{R}, \qquad \alpha_{ik}, \beta_{ik}, \gamma_{ij}, \delta_{ij}, \zeta_{ij}, \eta_{kl} \in \mathbb{R}_+, \qquad M' \in \mathcal{S}_+^{n+1}.
\end{aligned}
$$

4.3.4 Comparing the relaxations for QKP and QMKP

We compare $(\mathrm{QKP_{HRW4}})$ and our proposed relaxation. To do this, first we re-derive $(\mathrm{QKP_{HRW4}})$ in a different way. Consider the problem

$$
\begin{aligned}
(\mathbf{QKP'})\ \min\ & \lambda \\
\text{s.t. } & \lambda - p(x) \in \mathcal{P}_2(\{0,1\}^n \cap \{x : (c - w^T x) \geq 0\}).
\end{aligned}
$$

(QKP') is equivalent to (QMKP) where $m = 1$. This problem can be relaxed using

$$\mathcal{P}_2(\{0,1\}^n \cap \{x : (c - w^T x) \geq 0\}) \supseteq \Sigma_2^2 + \sum_i \Sigma_0^2 x_i(c - w^T x) + \sum_i x_i(1 - x_i)\mathbf{R},$$

obtaining

$$\begin{aligned} \textbf{(QKP}_{\widehat{\mathcal{K}}'}\textbf{)} \ & \min \ \lambda \\ & \text{s.t.} \ \lambda - p(x) = \begin{pmatrix} 1 & x \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + \sum_i d_i x_i (c - w^T x) + \sum_i c_i x_i (1 - x_i), \end{aligned}$$

where $S \in \mathcal{S}_+^{n+1}$, $d_i \in \mathbb{R}_+$, and $c_i \in \mathbb{R}$. By equating the coefficients of the monomials of the above problem, we rewrite it as

$$\begin{aligned} \textbf{(QKP}_{\textbf{HRW4-D}}\textbf{)} \ & \min \ \lambda \\ & \text{s.t.} \ \lambda - S_{00} = 0 \\ & \quad c_i + c d_i + S_{i0} + S_{0i} = 0 \\ & \quad \frac{d_i w_j + d_j w_i}{2} - S_{ij} + c_i \delta_{i=j} = P_{ij} \qquad 1 \leq i \leq j \leq n \\ & \quad S \succeq 0, \quad d_i \geq 0. \end{aligned}$$

where $\delta_{i=j}$ equals 1 if $i = j$ and zero otherwise.

Taking the dual of ($\mathrm{QKP}_{\mathrm{HRW4\text{-}D}}$), we obtain

$$\begin{aligned} \max \ & \langle \bar{P}, \bar{X} \rangle \\ \text{s.t.} \ & \bar{X}_{00} = 1 && (24) \\ & \bar{X}_{ii} - \bar{X}_{i0} = 0 \quad 1 \leq i \leq n && (25) \\ & \sum_{j=1}^{n} w_j \bar{X}_{ij} - c\bar{X}_{ii} \leq 0 \quad 1 \leq i \leq n && (26) \\ & \bar{X} \succeq 0, && (27) \end{aligned}$$

where $\bar{P} = \begin{pmatrix} 0 & 0 \\ 0 & P \end{pmatrix}$.

In addition, $X - \mathrm{diag}(X)\mathrm{diag}(X)^T \succeq 0$ is equivalent to $\bar{X} = \begin{pmatrix} 1 & \mathrm{diag}(X)^T \\ \mathrm{diag}(X) & X \end{pmatrix} \succeq 0$, so the above problem is a reformulation of ($\mathrm{QKP}_{\mathrm{HRW4}}$). Taking $X = I$, X is strictly feasible for ($\mathrm{QKP}_{\mathrm{HRW4}}$), therefore Slater's constraint qualification is satisfied for ($\mathrm{QKP}_{\mathrm{HRW4}}$). In addition, $X - \mathrm{diag}(X)\mathrm{diag}(X)^T \succeq 0$ implies $-\frac{1}{8} \leq X_{ij} \leq 1$ [15]. As a result, the objective $\langle P, X \rangle$ is bounded by $\sum_{i,j} |P_{ij}|$ and we have strong duality.

Theorem 4.5. *1. When $m = 1$, let $\lambda^*_{QKP_{HRW4\text{-}D}}$, $\lambda^*_{QKP_{\widehat{\mathcal{K}}}}$, and $\lambda^*_{QKP_{Burer'_1}}$ be the optimal solution value of ($QKP_{HRW4\text{-}D}$), ($QKP_{\widehat{\mathcal{K}}}$), and ($QKP_{Burer'_1}$) respectively, then*

$$\lambda^*_{QKP_{HRW4\text{-}D}} = \lambda^*_{QKP_{HRW4}} \geq \lambda^*_{QKP_{Burer'_1}} \geq \lambda^*_{QKP_{\widehat{\mathcal{K}}}} \geq z^*_{QKP}.$$

2. Let $\lambda^*_{QMKP_{Burer'_m}}$ and $\lambda^*_{QMKP_{\widehat{\mathcal{K}}}}$ be the optimal solution value of ($QMKP_{Burer'_m}$) and ($QMKP_{\widehat{\mathcal{K}}}$)

respectively, then

$$\lambda^*_{QMKP_{Burer'_m}} \geq \lambda^*_{QMKP_{\widehat{\mathcal{K}}}} \geq z^*_{QMKP}.$$

Proof. For part 1, define

$$\begin{aligned}
\mathcal{H}_1 =& \Sigma_2^2 + \sum_i \Sigma_0^2(1+\bar{x}_i)(\bar{c}-w^T\bar{x}) + \sum_i (1-\bar{x}_i^2)\mathbf{R} \\
\mathcal{H}_2 =& \mathcal{H}_1 + \sum_i \Sigma_0^2(1-\bar{x}_i)(\bar{c}-w^T\bar{x}) + \sum_{i\leq j} \Sigma_0^2(1+\bar{x}_i)(1+\bar{x}_j) \\
&+ \sum_{i\leq j} \Sigma_0^2(1-\bar{x}_i)(1-\bar{x}_j) + \sum_{i,j} \Sigma_0^2(1+\bar{x}_i)(1-\bar{x}_j) + \Sigma_0^2(\bar{c}-w^T\bar{x})(\bar{c}-w^T\bar{x}) \\
\mathcal{H}_3 =& \mathcal{H}_2 + (\bar{c}-w^T\bar{x})\mathcal{P}_1(\mathcal{B}) + \sum_i (1+\bar{x}_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1-\bar{x}_i)\mathcal{P}_1(\mathcal{B}).
\end{aligned}$$

Hence,

$$\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \mathcal{H}_3.$$

After changing variables, $\mathcal{H}_1$ corresponds to the approximation of $\mathcal{P}_2(\{0,1\}^n \cap \{x : (c - w^T x) \geq 0\})$ used in (QKP$_{\widehat{\mathcal{K}}'}$) which is equivalent to (QKP$_{\text{HRW4-D}}$), $\mathcal{H}_2$ corresponds to the representation (QKP$_{\text{Burer'}_1}$), while $\mathcal{H}_3$ corresponds to the representation (QMKP$_{\widehat{\mathcal{K}}}$) with m=1, we refer to it as (QKP$_{\widehat{\mathcal{K}}}$). Similarly, we can show part 2 of the theorem. □

Remark 4.6. *In some instances, even when using the weaker relaxation (QKP$_{SS}$) (the (QMKP$_{SS}$) relaxation with $m = 1$), we obtain a strictly better bound than (QKP$_{HRW4}$) as shown in Section 5.2. For those instances (QKP$_{\widehat{\mathcal{K}}}$) is also strictly better than (QKP$_{HRW4}$).*

5 Computational Results

In this section, we conduct comparisons based on computational time and quality of the bounds for the three applications discussed earlier. All the relaxations were coded in MATLAB and the computations were done on a 1200 MHz Sun Sparc machine and using the SDPT3 solver [18].

5.1 Max-Cut Computational Results

In order to test the performance of the max-cut relaxations discussed in Section 4.1.1, we generated test instances using the graph generator rudy of Rinaldi [40]. We generated several instances for each of the following types of graphs:

- Spinglass2G: generates a toroidal 2 dimensional-grid for a spin glass model with Gaussian interactions.
- Spinglass2pm: generates a toroidal 2 dimensional-grid for a spin glass model with ±1 interactions. The percentage of negative interactions in this case is taken 50%.
- Random: generates graphs with random edge weights taken from a discrete uniform distribution between 1 and 10 for our test instances and the graph density is varied from 10% to 100%.

The computational results presented are based on three types of relaxations for max-cut:

SDP: the Goemans and Williamson SDP relaxation, ($\mathrm{MC}_{\mathrm{GW}}$).

SOCP: the proposed SOCP relaxation, ($\mathrm{MC}_{\mathrm{SOCP}}$).

SOCP-SDP: the proposed SOCP-SDP relaxation, ($\mathrm{MC}_{\mathrm{SS}}$).

In addition, for each of the above mentioned relaxations we add the valid triangle inequalities, obtaining relaxations denoted by $\triangle$. The relaxations are compared in terms of the upper bound, the lower bound from the feasible solution obtained using Algorithm 2, and the computational time.

From Tables 1-3, we see that the SOCP relaxations without the triangle inequalities have weak upper and lower bounds as compared to the SDP and the SOCP-SDP relaxation. However, when the triangle inequalities are added, the bounds provided by the SOCP relaxation are comparable to those of the SDP and SOCP-SDP relaxations. In terms of computational time, the SOCP relaxation is the most efficient one. The SOCP relaxation is around 4 times faster than the SDP relaxation for the instances tested and for larger instances the difference in computational time becomes more significant. The SDP and SOCP-SDP relaxations with triangle inequalities are the best in terms of bounds but require high computational time. On the other hand, the SOCP relaxation with triangle inequalities provides good upper and lower bounds and has a reasonable computational time. From the results, the percentage gap between the strengthened SOCP relaxation and the strengthened SOCP-SDP relaxation ranges between 0%-17%, with an average of 3.5%.

In Table 4, we present results for max-cut on graphs of larger sizes using the SOCP-$\triangle$ relaxation. The instances are generated using the rudy graph generator and they vary in size from 60 to 80 vertices. We report the upper bounds, computational time, optimal solution, and the percentage gap between the upper bound and the optimal solution value. The optimal solution was obtained from [39]. The results show that the proposed relaxation is efficient in terms of computational time and provides good bounds as the % gap between the upper bound and the optimal ranges between 0% and 16% with an average of 1%.

Next we compare the higher order relaxations presented in Section 4.1.1. The relaxations compared are:

SOCP: the second-order cone relaxation presented in Section 4.1.1, ($\mathrm{MC}_{\mathrm{SOCP}}$)

SDP: the Goemans and Williamson SDP relaxation presented in Section 4.1, ($\mathrm{MC}_{\mathrm{GW}}$).

SDP-2: the Lasserre relaxation presented in Section 4.1.1, ($\mathrm{MC}_{\mathcal{K}_2'}$).

SOCP-Lift: the lifted second-order cone relaxation presented in Section 4.2, ($\mathrm{MC}_{\mathrm{SOCP\text{-}Lift}}$).

SS-Lift: the lifted relaxation presented in Section 4.2, ($\mathrm{MC}_{\mathrm{SS\text{-}Lift}}$).

SDP-AW: the Anjos-Wolkowicz relaxation presented in Section 4.2, ($\mathrm{MC}_{\mathrm{AW}_p}$).

The instances presented in Table 5 are taken from Anjos [2]. The SDP-based higher order relaxations, SS-Lift, SDP-2, and SDP-AW are always better than the basic SDP relaxation. The SOCP-Lift relaxation is much better than the SOCP relaxation but still not as good as SS-Lift, SDP-2, and SDP-AW in terms of bounds. The last two rows of Table 5 show the number and the dimension (number,dimension) of the SDP and SOCP variables for each of the relaxation (only the leading terms are shown). The last row gives the number of the linear variables.

Table 1: Max-Cut upper bound.

Instance	SOCP	SDP	SOCP-SDP	SOCP-△	SDP-△	SOCP-SDP-△
3×3_2g_33	338829.0	161890.0	161890.0	140130.0	140130.0	140130.0
4×4_2g_44	823634.0	632730.0	632730.0	562720.0	562720.0	562720.0
5×5_2g_55	1832274.0	1481900.0	1481900.0	1422600.0	1422600.0	1422600.0
6×6_2g_66	2826173.0	2347900.0	2347900.0	2290300.0	2290300.0	2290300.0
7×7_2g_77	4764025.0	4192200.0	4192200.0	4078200.0	4078200.0	4078200.0
3×3_2pm_33	9.0	5.1	5.1	4.0	4.0	4.0
4×4_2pm_44	16.0	12.1	12.1	10.0	10.0	10.0
5×5_2pm_55	25.0	19.2	19.2	18.0	18.0	18.0
6×6_2pm_66	36.0	28.8	28.8	26.0	26.0	26.0
7×7_2pm_77	49.0	37.5	37.5	34.0	34.0	34.0
20_rnd_10	100.0	94.9	94.9	94.0	94.0	94.0
20_rnd_30	295.0	243.1	243.1	239.0	239.0	239.0
20_rnd_50	496.0	375.9	375.9	368.0	368.0	368.0
20_rnd_70	695.0	472.3	472.3	463.3	457.0	457.0
20_rnd_90	870.0	546.5	546.5	580.0	534.0	534.0
20_rnd_100	971.0	588.9	588.9	647.3	583.0	583.0
30_rnd_10	220.0	201.5	201.5	194.0	194.0	194.0
30_rnd_30	663.0	502.4	502.4	480.0	480.0	480.0
30_rnd_50	1091.0	761.8	761.8	736.0	736.0	736.0
30_rnd_70	1498.0	979.6	979.6	998.7	963.0	963.0
30_rnd_90	1918.0	1152.5	1152.5	1278.7	1133.0	1133.0
30_rnd_100	2139.0	1271.9	1271.9	1426.0	1262.0	1262.0
40_rnd_10	429.0	393.1	393.1	380.0	380.0	380.0
40_rnd_30	1165.0	860.7	860.7	831.1	831.0	831.0
40_rnd_50	1929.0	1292.2	1292.2	1287.3	1250.8	1250.8
40_rnd_70	2693.0	1698.7	1698.7	1795.3	1655.9	1655.9
40_rnd_90	3513.0	2100.7	2100.7	2342.0	2062.0	2062.0
40_rnd_100	3905.0	2263.8	2263.8	2603.3	2229.5	2229.5
50_rnd_10	617.3	530.6	530.6	518.0	518.0	518.0
50_rnd_30	1882.0	1338.2	1338.2	1310.5	1289.7	1289.6
50_rnd_50	3069.0	1985.0	1985.0	2046.09	1931.40	1931.40
50_rnd_70	4287.06	2629.9	2629.9	2858.1	2588.8	2588.8
50_rnd_90	5495.0	3219.2	3219.2	3663.4	3170.64	3170.64
50_rnd_100	6089.1	3458.9	3458.9	4059.3	3424.8	3424.8

Table 2: Computational time in seconds.

Instance	SOCP	SDP	SOCP-SDP	SOCP-△	SDP-△	SOCP-SDP-△
3×3_2g_33	0.44	1.40	1.80	1.40	1.20	1.70
4×4_2g_44	0.56	1.00	2.10	2.20	2.30	2.80
5×5_2g_55	0.84	1.80	4.80	5.40	5.50	7.30
6×6_2g_66	1.62	4.60	11.90	22.50	19.90	25.40
7×7_2g_77	4.10	25.10	41.00	112.90	83.70	100.10
3×3_2pm_33	0.34	0.70	1.60	1.30	1.20	1.80
4×4_2pm_44	0.60	0.80	2.10	2.00	2.20	3.90
5×5_2pm_55	0.79	1.60	4.70	6.50	5.60	8.50
6×6_2pm_66	1.35	5.00	10.90	26.40	19.80	33.40
7×7_2pm_77	4.73	20.20	40.30	117.80	78.50	124.90
20_rnd_10	0.51	1.20	3.20	3.40	3.00	4.70
20_rnd_30	0.58	1.30	3.10	2.90	3.30	4.80
20_rnd_50	0.57	1.20	2.70	2.70	3.30	4.50
20_rnd_70	0.74	1.30	2.60	2.40	4.90	6.80
20_rnd_90	0.86	1.30	2.60	2.40	3.90	5.10
20_rnd_100	0.70	1.30	2.70	2.30	3.80	5.10
30_rnd_10	0.92	2.90	7.50	9.20	10.20	14.90
30_rnd_30	0.91	3.00	6.60	12.60	12.90	16.50
30_rnd_50	1.00	3.10	5.50	11.50	12.90	16.10
30_rnd_70	0.96	3.30	6.10	8.20	13.60	17.00
30_rnd_90	1.20	3.10	5.80	7.00	14.70	18.90
30_rnd_100	0.72	2.80	6.40	7.10	14.70	17.90
40_rnd_10	1.47	9.50	17.10	36.30	37.80	53.80
40_rnd_30	1.48	9.10	14.00	45.80	46.70	55.60
40_rnd_50	1.99	9.80	15.70	37.80	62.40	79.50
40_rnd_70	1.78	7.90	15.60	29.00	54.80	69.20
40_rnd_90	2.45	8.40	16.40	25.00	62.70	79.10
40_rnd_100	2.08	7.90	16.40	26.30	57.20	72.10
50_rnd_10	7.16	18.28	37.80	72.25	97.39	106.17
50_rnd_30	7.41	14.81	30.55	74.44	114.70	137.40
50_rnd_50	8.26	16.49	31.07	78.14	96.56	120.15
50_rnd_70	6.09	16.09	30.43	64.49	116.38	139.91
50_rnd_90	8.89	16.59	31.62	64.35	114.73	135.89
50_rnd_100	6.31	15.22	31.60	63.98	109.33	134.28

Table 3: Max-Cut lower bound (feasible solution).

Instance	SOCP	SDP	SOCP-SDP	SOCP-△	SDP-△	SOCP-SDP-△
3×3_2g_33	5843	140128	140128	140128	140128	140128
4×4_2g_44	307702	562722	562722	562722	562722	562722
5×5_2g_55	363179	1422618	1422618	1422618	1422618	1422618
6×6_2g_66	965019	2290275	2290275	2290275	2290275	2290275
7×7_2g_77	2065916	4052837	4052837	4078176	4078176	4078176
3×3_2pm_33	2	4	4	4	4	4
4×4_2pm_44	4	10	10	6	8	6
5×5_2pm_55	6	18	18	18	18	18
6×6_2pm_66	6	26	26	26	26	26
7×7_2pm_77	16	32	32	34	34	34
20_rnd_10	79	94	94	94	94	94
20_rnd_30	185	239	239	239	239	239
20_rnd_50	291	368	368	368	368	368
20_rnd_70	410	457	457	404	457	457
20_rnd_90	472	534	534	498	534	534
20_rnd_100	536	583	583	542	583	583
30_rnd_10	147	193	193	194	194	194
30_rnd_30	382	379	477	413	480	480
30_rnd_50	611	728	728	736	736	736
30_rnd_70	833	962	962	835	963	963
30_rnd_90	1055	1121	1121	1037	1114	1133
30_rnd_100	1142	1188	1262	1162	1262	1262
40_rnd_10	262	380	380	380	380	380
40_rnd_30	662	821	821	642	831	831
40_rnd_50	1055	1226	1226	1067	1142	1142
40_rnd_70	1440	1457	1637	1438	1561	1561
40_rnd_90	1861	1858	2046	1882	2050	1967
40_rnd_100	2038	2058	2226	2069	2108	2108
50_rnd_10	352	356	368	389	369	362
50_rnd_30	1023	1013	1039	1039	1036	1035
50_rnd_50	1642	1628	1658	1652	1636	1647
50_rnd_70	2289	2231	2242	2211	2250	2231
50_rnd_90	2863	2896	2859	2916	2894	2863
50_rnd_100	3174	3158	3159	3168	3189	3207

Table 4: Results for the SOCP-$\triangle$ relaxation for Max-Cut.

Instance	UB	Time	Optimal Value	% Gap
8×8_2g_88	3641835.59	299.03	3556450	2.40
9×9_2g_99	5662335.85	965.59	4851202	16.72
8×8_2pm_88	44.00	328.00	44	0.01
9×9_2pm_99	56.01	1243.71	56	0.01
60_rnd_10	685.02	320.80	685	0.00
60_rnd_30	1749.23	431.50	1749	0.01
60_rnd_50	2702.17	436.80	2702	0.01
60_rnd_70	3578.82	358.68	3575	0.11
60_rnd_90	4416.86	342.75	4406	0.25
60_rnd_100	4826.31	341.13	4816	0.21
70_rnd_10	948.02	661.57	948	0.00
70_rnd_30	2448.18	707.57	2439	0.38
70_rnd_50	3750.07	655.63	3733	0.46
70_rnd_70	5005.70	671.06	4983	0.46
70_rnd_90	6243.00	678.37	6229	0.22
70_rnd_100	6786.38	989.75	6786	0.01
80_rnd_10	1202.07	1931.18	1202	0.01
80_rnd_30	3064.73	1558.12	3057	0.25
80_rnd_50	4766.79	1328.71	4743	0.50
80_rnd_70	6380.85	1209.27	6335	0.72
80_rnd_90	7888.82	1311.46	7858	0.39
80_rnd_100	8590.86	1238.85	8552	0.45

Table 5: Comparison of higher order relaxation in terms of upper bound.

Instance	Size $\|V\|$	Optimal	SOCP	SDP	SDP-2	SOCP-Lift	SS-Lift	SDP-AW
C_5	5	4.00	5.00	4.52	4.00	4.24	4.00	4.00
K_5-e(2,4)	5	6.00	9.00	6.25	6.00	6.62	6.00	6.00
K_5	5	6.00	10.00	6.25	6.25	7.16	6.25	6.25
K_5-$A(G)$	5	9.28	14.08	9.60	9.28	10.27	9.28	9.28
AW_9^2	9	12.00	18.00	13.50	12.41	14.63	12.50	12.50
Petersen	10	12.00	15.00	12.50	12.00	13.08	12.00	12.00
$n = 12$-$A(G)$	12	88.00	134.00	90.39	88.00	103.65	88.00	88.00
SDP Variables			-	1, n^2	$1, \frac{n^4}{4}$	-	$1, \frac{n^4}{4}$	$1, \frac{n^4}{4}$
SOCP Variables			$2n$,n	-	-	$\frac{n^2}{2}, \frac{n^2}{2}$	$\frac{n^2}{2}, \frac{n^2}{2}$	-
Linear Variables			$\frac{n^2}{2}$	-	-	$\frac{n^3}{2}$	$\frac{n^3}{2}$	-

5.2 QKP Computational Results

In this section we compare the performance of our proposed relaxation for the QKP with the relaxation of Helmberg et al. [15] discussed in Section 4.3.2. We generated test instances using the idea proposed in [37]. The P_{ij} and w_j values are discrete taken from a uniform random distribution in $[1, 100]$ and $[1, 50]$ respectively. The capacity c is uniformly distributed in $[50, \sum_{j=1}^{n} w_j]$. The density of the P matrix, d, varies from 10 to 90 %.

The computational results presented are based on five types of relaxations for the quadratic knapsack problem:

HRW: the Helmberg et al. SDP relaxation presented in Section 4.3.2, ($\mathrm{QKP}_{\mathrm{HRW4}}$).

SOCP: the second-order cone relaxation presented in Section 4.3.1, ($\mathrm{QKP}_{\mathrm{SOCP}}$).

SOCP-SDP: the SDP-SOCP relaxation presented in Section 4.3.1, ($\mathrm{QKP}_{\mathrm{SS}}$).

SOCP-△: the strengthened second-order cone relaxation with triangle inequalities presented in Section 4.3.1, ($\mathrm{QKP}_{\mathrm{SOCP}-\triangle}$).

SOCP-SDP-△: the strengthened SDP-SOCP relaxation with triangle inequalities presented in Section 4.3.1, ($\mathrm{QKP}_{\mathrm{SS}-\triangle}$).

In Table 6, we report results for 30 instances. These instances vary in size and density. The size varies from 20 to 70 items and density from 10 to 90% with a step size of 20 %. For each instance we report the upper bound and the solution time in seconds for each of the relaxations above. It is seen that the bounds based on the semidefinite relaxation in [15] are strictly weaker than the ones based on our proposed relaxation for all instances (except for 2 instances where both relaxations are optimal). We note that the bounds for the HRW are already strong [15, 37] but require increasing computational time. On average for the large instances, the SOCP relaxation is almost 50 times faster than the SOCP-SDP relaxation and the HRW relaxation. The percentage gap of the SOCP relaxation with respect to HRW relaxation ranges from -7.7% to around 25% with an average of 3.4%, where a negative sign implies that the SOCP relaxation is better.

Since SOCP-SDP provides better bounds than HRW, adding the triangle inequalities to this relaxation will strengthen the bound further making it more promising but computationally expensive. In addition, although SOCP provides weak bounds, once the triangle inequalities are added, SOCP-△ provides almost the same bound as SOCP-SDP-△ but requires much less computational time. The computational results show that SOCP-△ provides better bounds than HRW and the percentage gap with respect to HRW is then always negative and can be up to -14%. In terms of computational time, SOCP-△ is faster than SOCP-SDP-△ by a factor of 4 for the large instances.

5.2.1 QMKP Computational Results

In this section, we compare our solution approach with the approach adopted by Burer [8] to solve the quadratic multiple knapsack problem. Table 7 compares the average % gap between each relaxation upper bound and the optimal objective value and the average computational time. We compare three relaxations:

Burer: the relaxation presented by Burer [8].

SDP: the strengthened SDP relaxation presented in Section 4.3.1, ($\mathrm{QMKP}_{\widehat{\mathcal{K}}}$).

SOCP-△: the second-order cone relaxation with triangle inequalities presented in 4.3.1, ($\mathrm{QKP}_{\mathrm{SOCP}-\triangle}$).

Table 6: Comparison in terms of upper bound and computational time (seconds) for the QKP.

Instance	SOCP		SOCP-SDP		SOCP-$\triangle$		SOCP-SDP-$\triangle$		HRW	
n_d_rnd	UB	Time	UB	Time	UB	Time	UB	Time	UB	Time
20_10_20	811.74	2.39	811.22	1.28	809.81	3.34	809.81	6.14	814.84	1.09
20_30_20	2619.48	1.92	2619.30	1.38	2618.60	4.32	2618.60	7.86	2624.00	1.16
20_50_20	1262.98	1.48	1137.30	1.22	1120.70	2.49	1120.60	4.22	1175.10	1.05
20_70_20	2540.40	1.43	2356.30	1.22	2338.70	2.84	2338.70	4.97	2397.20	1.14
20_90_20	6083.80	1.69	6083.70	1.31	6081.00	2.89	6081.00	5.25	6086.10	1.11
30_10_30	1129.01	3.17	1022.20	7.96	1011.60	22.35	1011.60	36.96	1044.40	6.46
30_30_30	3939.00	2.89	3471.00	7.59	3443.16	20.52	3442.40	40.34	3511.30	5.95
30_50_30	8127.76	4.02	8125.20	7.55	8104.70	18.27	8104.70	29.43	8142.10	7.27
30_70_30	8108.38	2.70	8047.00	8.30	8033.40	19.50	8033.40	40.06	8073.10	6.62
30_90_30	5136.34	3.36	5127.60	8.36	5111.40	23.42	5111.30	35.85	5150.80	5.18
40_10_40	3875.12	7.12	3853.50	37.31	3840.80	99.37	3840.60	207.20	3864.50	28.62
40_30_40	11811.54	8.92	11809.44	39.11	11802.77	112.40	11802.73	204.80	11828.42	28.81
40_50_40	5161.31	6.71	4309.76	39.18	4296.53	92.05	4295.37	182.80	4365.56	35.57
40_70_40	17447.01	7.13	17424.10	39.21	17406.60	111.20	17406.60	208.30	17446.14	30.42
40_90_40	25615.00	8.81	25612.48	37.26	25585.86	83.30	25585.86	189.40	25630.04	30.00
50_10_50	2846.05	10.91	2353.89	122.40	2316.14	386.90	2316.11	755.30	2412.48	121.10
50_30_50	12050.94	11.20	11433.16	134.60	11403.16	326.90	11403.16	688.40	11485.59	122.80
50_50_50	23850.99	13.66	23846.12	122.70	23815.76	320.30	23815.06	727.50	23863.04	116.50
50_70_50	32575.12	15.33	32571.10	139.50	32555.54	260.10	32555.54	601.00	32626.49	94.14
50_90_50	17672.78	13.96	17671.03	128.40	17651.55	381.00	17651.49	723.60	17682.63	99.88
60_10_60	7410.08	25.82	7188.68	670.40	7170.75	1336.00	7170.71	3769.00	7215.96	540.80
60_30_60	26502.66	26.14	26496.51	644.90	26488.87	1796.00	26488.78	5420.00	26530.82	572.30
60_50_60	14396.64	25.22	13871.42	703.30	13845.65	1344.00	13844.49	3597.00	13895.51	564.40
60_70_60	56561.38	28.33	56561.20	868.20	56552.56	1502.00	56552.55	3645.00	56583.47	540.70
60_90_60	62009.00	23.34	62009.00	520.00	62008.99	450.20	62008.99	1815.00	62015.61	556.20
70_10_70	5104.22	43.78	4036.66	2295.00	3958.10	2192.70	3957.38	12460.00	4109.61	2074.00
70_30_70	21826.79	48.53	20208.57	2360.00	20190.03	1747.00	20190.03	9209.00	20275.13	2124.00
70_50_70	45752.77	56.19	45507.07	2806.00	45483.17	3212.00	45483.14	10260.00	45573.21	2107.00
70_70_70	1737.92	46.38	1631.57	1631.60	1619.57	4226.00	1619.56	11780.00	1882.75	2453.00
70_90_70	32876.13	58.37	32857.31	2716.00	32837.20	2825.00	32837.20	9840.00	32913.98	2137.00

We consider 732 instances that vary in size, n, from 10 items up to 50 and the density varies randomly from 1% to 100%. In addition the number of the knapsack constraints, m, vary from 1 to 25. The data for the instances and their optimal objective values, as well as the upper bounds and computational time of Burer's relaxation (reported as Time1 in Table 7) were all provided by Burer. We also implemented Burer's relaxation using our code (as described in Section 4.3.3 above) and we report the average computational time we obtained for it as Time2 in Table 7.

From Table 7, we see that the average $\text{Gap}_{\text{Burer}}$ is always greater than or equal to the average Gap_{SDP}. We note that for all the instances solved the bounds of the ($\text{QMKP}_{\widehat{\mathcal{K}}}$) relaxation were at least as good as Burer's relaxation bounds. For 373 instances, the ($\text{QKP}_{\text{SOCP}-\triangle}$) provided a better upper bound than Burer's relaxation while for 346 the latter relaxation provided better upper bounds and for 12 instances both relaxations gave the same bound. In terms of computational time, Burer's relaxation was the most efficient since Burer's algorithm is very efficient is solving such problems. However, in theory, it is an SDP-based relaxation and thus the computational time has a higher order of complexity than the SOCP-based relaxation and this can be seen when comparing Time2 with the computational time of the SOCP-$\triangle$ relaxation.

Table 7: Comparison of the QMKP in terms of the average % gap of the upper bound from the optimal and the average computational time (sec).

n	m	Burer			SDP		SOCP-△	
		Gap	Time1	Time2	Gap	Time	Gap	Time
10	1	7.77	0.66	2.13	7.76	1.78	7.66	1.71
	5	11.30	0.72	3.03	11.29	3.62	14.11	1.91
20	1	3.75	1.24	5.42	3.73	5.24	3.70	6.64
	5	8.48	2.01	7.68	8.46	10.99	9.66	7.16
	10	10.64	2.22	11.67	10.64	15.30	12.46	7.56
30	1	1.80	2.17	18.71	1.75	19.10	1.67	29.27
	5	5.79	3.35	27.82	5.77	43.80	6.07	32.61
	15	10.14	4.76	72.35	10.12	93.04	11.12	36.64
40	1	1.31	3.76	65.13	1.27	66.37	1.22	103.28
	5	2.81	5.30	95.69	2.79	146.39	2.71	117.7
	20	10.01	10.21	382.73	9.99	442.05	11.72	140.09
50	1	1.11	4.96	121.62	1.08	98.27	1.04	227.99
	5	2.71	8.00	192.18	2.66	199.37	2.68	284.68
	25	9.77	18.13	713.48	9.71	744.50	12.41	340.47

6 Conclusion and Future Work

In this research we used polynomial programming approaches to produce tractable relaxations to general binary quadratic optimization problems. These approximations utilize linear, second-order and semidefinite cones over which it is known how to optimize efficiently. We proposed a second-order cone relaxation to the general BQP and applied it to three binary quadratic problems, the max-cut problem, the quadratic knapsack problem, and the quadratic multiple knapsack problem. These SOCP-based relaxations have much lower running times with only a small degradation of bounds when compared to SDP-based relaxations.

For the max-cut problem, we developed and implemented several relaxations based on SDP and SOCP. We compared these relaxations with other relaxations in literature. When adding the triangle inequalities to the SOCP relaxation, our method gives a better performance in terms of computational time when solving max-cut problems as compared to the SDP relaxation while the optimality gap is still comparable between the two approaches. In addition, we developed higher order relaxations based on SOCP and conducted test results on a set of instances to compare them in terms of bounds.

For the quadratic knapsack problem, we developed an SOCP-SDP-based relaxation that is competitive with the best relaxations in the literature. Theoretical results as well as computational experiments show that our approach outperforms that relaxation in terms of bound and both relaxations are comparable in terms of computational time. We also relaxed our proposed relaxation to obtain a weaker SOCP relaxation. Adding the triangle inequalities, we were able to obtain comparable bounds to the SOCP-SDP-based relaxation, but were computationally more efficient. We also conducted computational results for the multiple quadratic knapsack problem and showed the quality of the bounds provided by our SOCP-based relaxation is competitive with those from the very recent specialized relaxation of Burer for this problem [8].

The main focus of our research is to develop an exact algorithm to be able to solve general binary quadratic problems. Our SOCP relaxation show potential in terms of bound and computational time to be used in an exact algorithm scheme to find optimal solutions for large instances of such problems in a reasonable time. Our findings motivate the use of SOCP-based relaxations with additional inequalities since they are efficient and provide competitive bounds. We are looking into developing general cuts for the binary quadratic problems based on polynomial programming. In particular, we could add non-linear cuts to the problem to strengthen the relaxation further.

Acknowledgments

The authors thank Sam Burer for providing us with the QMKP instances and their optimal values.

References

[1] B. Alidaee, G. Kochenberger, and A. Ahmadian. 0-1 quadratic programming approach for the optimal solution of two scheduling problems. *International Journal of Systems Science*, 25:401–408, 1994.

[2] M. F. Anjos. *New Convex Relaxations for the Maximum Cut and VLSI Layout Problems*. PhD thesis, Department of Combinatorics and Optimization, University of Waterloo, Canada., 2001.

[3] M. F. Anjos, A. Kennings, and A. Vannelli. A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization*, 2(2):113–122, 2005.

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

[5] F. Barahona, M. Jünger, and G. Reinelt. Experiments in quadratic 0-1 programming. *Mathematical Programming*, 44:127–137, 1989.

[6] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[7] E. Boros and A. Prekopa. Probabilistic bounds and algorithms for the maximum satisfiability problem. *Annals of Operations Research*, 21:109–126, 1989.

[8] S. Burer. Optimizing a polyhedral-semidefinite relaxation of completely positive programs. *The 20th International Symposium of Mathematical Programming (ISMP)*, 2009.

[9] S. Burer, R. Monteiro, and Y. Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal of Optimization*, 12(2):503–521, 2002.

[10] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer, Berlin, 1997.

[11] G. Gallo, P. Hammer, and B. Simeone. Quadratic knapsack problems. *European Journal of Operational Research*, 12:132–149, 1980.

[12] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[13] M. Grötschel, M. Jünger, and G. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

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

[15] C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Combinatorial Oprimization*, 4(2):197–215, 2000.

[16] E. L. Johnson, A. Mehrotra, and G. L. Nemhauser. Min-cut clustering. *Mathematical Programming*, 62:133–151, 1993.

[17] M. Jünger and F. Liers. Spin-glass server. *http://www.informatik.uni-koeln.de/ls_juenger/research/sgs/index.html.*

[18] R. H. T. K. C. Toh and M. J. Todd. SDPT3 - a MATLAB software for semidefinite-quadratic-linear programming. *http://www.math.nus.edu.sg/∼mattohkc/sdpt3.html*, 2006.

[19] S. Kim and M. Kojima. Exact solutions of some nonconvex quadratic optimization problems via sdp and socp relaxations. *Computational Optimization and Applications*, 26(2):143–154, 2003.

[20] E. d. Klerk and D. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12(4):875–892, 2002.

[21] M. Kojima, S. Kim, and H. Waki. A general framework for convex relaxation of polynomial optimization problems over cones. *Journal of Operations Research Society of Japan*, 46:125–144, 2003.

[22] J. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12(3):756–769, 2001.

[23] J. Lasserre. Global optimization problems with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11:796–817, 2001.

[24] D. Laughhunn. Quadratic binary programming with application to capital-budgeting problems. *Operations Research*, 18:454–461, 1970.

[25] M. Laurent. Semidefinite representations for finite varieties. *Mathematical Programming*, 109(Ser. A):1–26, 2007.

[26] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi. *New Optimization Algorithms in Physics: Computing exact ground states of hard Ising spin glass problems by branch-and-cut*, pages 47–68. Wiley, London, 2004.

[27] A. Mobasher, M. Taherzadeh, R. Sotirov, and A. K. Khandani. A near maximum likelihood decoding algorithm for MIMO systems based on semi-definite programming. *IEEE Transactions Information Theory*, pages 4–9, 2005.

[28] T. S. Motzkin. The arithmetic-geometric inequality. In *Inequalities (Proceedings of Symposium Wright Patterson Air Force Base, Ohio, 1965)*, pages 205–224. Academic Press, New York, 1967.

[29] Y. Nesterov. Semidefinite relaxation and non-convex quadratic optimization. *Optimization Methods and Software*, 12:1–20, 1997.

[30] P. Pardalos and G. Rodgers. Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45:131–144, 1990.

[31] P. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA., 2000.

[32] P. Parrilo. An explicit construction of distinguished representations of polynomials nonnegative over finite sets. Technical report, IfA Technical Report AUT02-02, Zurich - Switzerland, 2002.

[33] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[34] P. Parrilo and B. Sturmfels. Minimizing polynomial functions, algorithmic and quantitative real algebraic geometry. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 60:83–89, 2003.

[35] J. F. Peña, J. C. Vera, and L. F. Zuluaga. Exploiting equalities in polynomial programming. *Operations Research Letters*, 36(2), 2008.

[36] A. Phillips and J. Rosen. A quadratic assignment formulation of the molecular conformation problem. *Journal of Global Optimization*, 4:229–241, 1994.

[37] D. Pisinger. The quadratic knapsack problem-a survey. *Discrete Applied Mathematics*, 155(5):623–648, 2007.

[38] I. Pólik and T. Terlaky. A survey of the $\mathcal{S}$-lemma. *SIAM Review*, 49:371–418, 2007.

[39] F. Rendl, G. Rinaldi, and A. Wiegele. Biq mac solver. (http://biqmac.uni-klu.ac.at/).

[40] G. Rinaldi. Rudy graph generator. *http://www-user.tu-chemnitz.de/∼helmberg/rudy.tar.gz*.

[41] N. Shor. A class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

[42] B. Sturmfels, J. Demmel, and J. Nie. Minimizing polynomials via sum of squares over the gradient ideal. *Mathematical Programming*, 106:587–606, 2006.

[43] J. C. Vera, L. Zuluaga, and J. Peña. Positive polynomials on equality constrained domains. *Working Paper*.

[44] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of Semidefinite programming - Theory, Algorithms, and Applications.* Kluwer, 2000.

[45] L. Zuluaga, J. C. Vera, and J. Peña. LMI approximations for cones of positive semidefinite forms. *SIAM Journal on Optimization*, 16(4), 2006.