SIAM J. Optim.
Vol. 21, No. 1, pp. 391–414

SECOND-ORDER CONE RELAXATIONS FOR BINARY QUADRATIC POLYNOMIAL PROGRAMS*

BISSAN GHADDAR†, JUAN C. VERA‡, AND MIGUEL F. ANJOS§

Abstract. Several types of relaxations for binary quadratic polynomial programs can be obtained using linear, second-order cone, or semidefinite techniques. In this paper, we propose a general framework to construct conic relaxations for binary quadratic polynomial programs based on polynomial programming. Using our framework, we re-derive previous relaxation schemes and provide new ones. In particular, we present three relaxations for binary quadratic polynomial programs. The first two relaxations, based on second-order cone and semidefinite programming, represent a significant improvement over previous practical relaxations for several classes of nonconvex binary quadratic polynomial problems. From a practical point of view, due to the computational cost, semidefinite-based relaxations for binary quadratic polynomial problems can be used only to solve small to midsize instances. To improve the computational efficiency for solving such problems, we propose a third relaxation based purely on second-order cone programming. Computational tests on different classes of nonconvex binary quadratic polynomial problems, including quadratic knapsack problems, show that the second-order-cone-based relaxation outperforms the semidefinite-based relaxations that are proposed in the literature in terms of computational efficiency, and it is comparable in terms of bounds.

Key words. binary quadratic polynomial program, polynomial programming, sum-of-squares, second-order cone

AMS subject classifications. 90C20, 90C26

DOI. 10.1137/100802190

1. Introduction.

Binary quadratic polynomial problems (BQPP) 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 (SOC) [14, 16], or semidefinite techniques [3, 7, 15, 26]. In this paper we study relaxations for general BQPPs based on polynomial programming.

Polynomial programming includes a broad class of problems and is known to be NP-hard. Polynomial programming problems can be relaxed to tractable problems by using sum-of-squares (SOS) decompositions which lead to semidefinite programming (SDP) relaxations. This technique was first proposed by Shor [36] to obtain bounds on the optimal value of the unconstrained case. This idea was then generalized by Parrilo [27, 29] and Lasserre [19] for the constrained case.

In this paper, we use a characterization of nonnegative linear polynomials over the ball to propose SOC relaxations of BQPPs. We use the polynomial programming framework to rederive, compare, and strengthen existing relaxation schemes. We present a new second-order and semidefinite-based construction where we are able to

*Received by the editors July 15, 2010; accepted for publication (in revised form) January 24, 2011; published electronically March 31, 2011.
http://www.siam.org/journals/siopt/21-1/80219.html

†Department of Management Sciences, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (bghaddar@uwaterloo.ca). This author was supported by a Canada Graduate Scholarship from the Natural Sciences and Engineering Research Council of Canada.

‡Tilburg School of Economics and Management, Tilburg University, Tilburg, The Netherlands (j.c.veralizcano@uvt.nl).

§Département de Mathématiques et génie industriel & GERAD, École Polytechnique de Montréal, Montréal, QC, H3C 3A7, Canada (anjos@stanfordalumni.org). This author was partially supported by the Natural Sciences and Engineering Research Council of Canada and by a Humboldt Research Fellowship.

theoretically show that the resulting relaxations provide bounds stronger than other computationally practical semidefinite-based relaxations proposed in the literature. Additionally, our proposed framework enables us to isolate expensive components of existing relaxations, namely, the semidefinite terms. By removing the semidefinite terms, we obtain relaxations based purely on SOCs. We present computational tests exploring the performance of these relaxations, comparing them to existing ones in terms of bounds and computational time on general quadratic constrained problems, quadratic linear constrained problems (QLCPs), and quadratic knapsack problems (QKPs). The computational experiments confirm our theoretical results where we obtain that the SOC-SDP-based relaxations give the best bounds. Our experiments also show that the purely SOC-based relaxations produce bounds that are competitive with the existing SDP bounds but are computationally much more efficient. Furthermore, our approach can be in principle extended to mixed-binary polynomial programs where some of the variables are continuous.

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 BQPP including our three new proposed relaxations. In section 4, we apply our proposed relaxations to the three classes of problems mentioned above, and we theoretically compare them to other existing relaxations from the literature. In section 5, we report computational results for these problems. Finally, conclusions and future research directions are discussed in section 6.

2. Background.

2.1. 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 nonnegative integer $d = |\alpha| := \sum_i^n \alpha_i$. We denote by $\mathcal{M}_d(x)$ the set of monomials of x up to degree d. There are $N = \binom{n+d}{d}$ monomials of degree at most d. A polynomial is a finite linear combination of monomials

$$f(x) = \sum_\alpha c_\alpha x^\alpha = \sum_\alpha c_\alpha x_1^{\alpha_1} \cdots x_n^{\alpha_n} = \langle c, \mathcal{M}_d(x) \rangle ,$$

where the vector of coefficients $c \in \mathbb{R}^N$. We denote the cone of real polynomials (of degree at most d) that are SOS by $\Psi \subset \mathbf{R}[x]$ (resp. Ψ_d), 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). Notice that, in particular, $\Psi_d = \{\sum_{i=1}^N p_i(x)^2 : p(x) \in \mathbf{R}_{\lfloor \frac{d}{2} \rfloor}[x]\}$ and $\Psi_d = \Psi_{d-1}$ for every odd d. Given $S \subseteq \mathbb{R}^n$, 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 nonnegative over S.

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

$$\textbf{(PP-P)} \quad \sup f(x) \quad \text{s.t. } g_j(x) \geq 0, \quad 1 \leq j \leq m.$$

Equality constraints of the form $h_j(x) = 0$ can be included, as they can be expressed as the inequality constraints $h_j(x) \geq 0$ and $h_j(x) \leq 0$.

Solving polynomial programming problems is an area being actively studied. For the unconstrained case, Shor introduced the idea of computing the minimum value λ such that $\lambda - f(x)$ is an SOS to obtain an upper bound for the supremum of f [36]. Such an minimum λ can be computed in polynomial-time using SDP. This idea was further developed by Parrilo [27] and Parrilo and Sturmfels [30] for the constrained case using SOS decompositions. Lasserre [19] proposed a general solution approach for polynomial optimization problems via SDP using methods based on moment theory. Refinements of such ideas have been used in several instances. de Klerk and Pasechnik [8] approximated the copositive cone via a hierarchy of linear or semidefinite programs of increasing size using decompositions into sum-of-squares and polynomials with nonnegative coefficients. Kojima, Kim, and Waki exploited the sparsity of the polynomials to reduce the size of the semidefinite problem [17]. Peña, Vera, and Zuluaga [31] presented solution schemes exploiting the equality constraints. In addition, the idea of approximating a set of nonnegative polynomials is also present in the work of several authors such as Nesterov [25], Parrilo [29, 28], Sturmfels, Demmel, and Nie [38], Laurent [21], and Zuluaga, Vera, and Peña [41].

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 $x \in S := \{x : g_j(x) \geq 0;\ 1 \leq j \leq m\}$. As a result, we can express problem (PP-P) as

$$\textbf{(PP-D)}\ \inf \lambda \qquad \text{s.t. } \lambda - f(x) \geq 0 \quad \forall x \in S. \tag{2.1}$$

To obtain computable relaxations (via SDP) of (2.1), one can use an SOS decomposition with restricted degree of the (unknown) polynomials. This can be rephrased in terms of a linear system of equations involving positive semidefinite matrices [41]. Thus, solving a polynomial problem can be relaxed to solving an easier problem involving SOS which can be recast as an SDP problem [36, 40].

The condition $\lambda - f(x) \geq 0$ for all $x \in S$ is 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 of the original problem. At the same time, having a tractable approximation, i.e., one that uses linear, second-order, and semidefinite cones, is essential for solving the resulting relaxation efficiently using interior-point methods.

2.3. SOS and SOC relaxations. Consider a polynomial $p(x)$ of degree d. A necessary condition for the polynomial $p(x)$ to be nonnegative for all $x \in \mathbb{R}^n$ is that the degree of p is even. A sufficient condition is the existence of an SOS 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, $p \in \Psi$. If $p(x)$ is an SOS polynomial, then it is a nonnegative polynomial for all values of x; however, the inverse does not hold. A simple counterexample is the Motzkin polynomial [24].

SOS conditions can be written as SDP constraints by applying the following theorem.

THEOREM 2.1 (see [36]). *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 $Q \in \mathcal{S}^N_+$, $N = \binom{n+d/2}{d/2} = |\sigma|$.*

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. If d is fixed, then this problem is solvable in polynomial-time.

The following results will allow us to use SOC relaxations when working with nonnegative polynomials over the ball $\mathcal{B} := \{x : \|x\|^2 = n\}$.

LEMMA 2.2. *$f(x) \in \mathcal{P}_1(\mathcal{B})$ if and only if $f(x) = f^T\binom{\sqrt{n}}{x}$ with $f \in \mathcal{L}^{n+1}$, where $\mathcal{L}^{n+1}$ is the SOC.*

Further, by the $\mathcal{S}-$Lemma of Yakubovich (see [34]), the nonnegativity over the ball of a polynomial $f(x)$ of degree 2 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 $s(x) \in \Psi_2$ and $t \in \mathbb{R}_+$.*

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

3. Binary quadratic polynomial programming.

The binary quadratic polynomial programming problem is a classical combinatorial problem. It is the problem of minimizing or maximizing a quadratic function of several binary variables, subject to quadratic and linear constraints. The problem can be formally expressed as

$$\textbf{(BQPP)} \quad \max\ x^TQx + p^Tx$$

$$\text{s.t. } a_j^T x = b_j \qquad \forall j \in \{1, \ldots, t\}, \tag{3.1}$$

$$c_j^T x \le d_j \qquad \forall j \in \{1, \ldots, u\}, \tag{3.2}$$

$$x^T F_j x + e_j^T x = k_j \qquad \forall j \in \{1, \ldots, v\}, \tag{3.3}$$

$$x^T G_j x + h_j^T x \le l_j \qquad \forall j \in \{1, \ldots, w\}, \tag{3.4}$$

$$x_i \in \{-1, 1\} \qquad \forall i \in \{1, \ldots, n\}. \tag{3.5}$$

Note that constraint (3.5) can be modified to allow some continuous variables. In this paper, we focus on pure binary quadratic polynomial programs although our solution methodology can be applied to mixed-binary quadratic polynomial programs with bounded continuous variables. Furthermore, although one could consider an equivalent form of the problem with only inequality constraints, we treat equality and inequality constraints separately because this is beneficial from a computational perspective (see the discussion in section 4.1).

There are many well-known problems that can be naturally written as BQPPs. For instance, folding of proteins in three-dimension by Phillips and Rosen [32], machine scheduling and unconstrained task allocation by Alidaee, Kochenberger, and Ahmadian [1], capital budgeting and financial analysis such as in Laughhunn [20], as well as other examples arising in physics and engineering applications such as the spin glass problem and circuit board layout design by Grötschel, Jünger, and Reinelt [11]. Furthermore, Boros and Hammer [4] and Boros and Prekopa [5] formulated many satisfiability problems as BQPPs. In addition, there are several applications related to combinatorial problems such as the single-row facility layout problem [2] and the quadratic assignment problem [22].

3.1. Polynomial programming-based relaxations.

Using (2.1), (BQPP) is equivalent to

$$\begin{aligned}&\min \lambda\\ &\text{ s.t. } \lambda - q(x) \in \mathcal{P}_2(H \cap S),\end{aligned}$$

where $q(x) = \sum_{i,j} Q_{ij}x_ix_j + \sum_i p_ix_i$, $S = \{x : a_j^T x = b_j, c_j^T x \geq d_j, x^T F_j x + e_j^T x = k_j, x^T G_j x + h_j^T x \geq l_j\}$, and $H := \{-1, 1\}^n$. Note that even checking if a polynomial is in $\mathcal{P}_2(H)$ is NP-hard, therefore tractable approximations of $\mathcal{P}_2(H \cap S)$ are needed. A hierarchy of approximations to $\mathcal{P}_2(H \cap S)$ is obtained using the cones

$$\begin{aligned}\mathcal{K}_r := &\left(\Psi_{r+2} + \sum_i (1 - x_i^2)\mathbf{R}_r[x] + \sum_j (b_j - a_j^T x)\mathbf{R}_{r+1}[x] + \sum_j (d_j - c_j^T x)\Psi_{r+1}\right.\\ &\left. + \sum_j (k_j - x^T F_j x - e_j^T x)\mathbf{R}_r[x] + \sum_j (l_j - x^T G_j x - h_j^T x)\Psi_r\right) \cap \mathbf{R}_2[x]\\ &\subseteq \mathcal{P}_2(H \cap S)\end{aligned}$$

for an integer $r \geq 0$. The result is a hierarchy of relaxations:

$$\begin{aligned}(\mathbf{BQPP}_{\mathcal{K}_r}) \; &\min \lambda\\ &\text{ s.t. } \lambda - q(x) \in \mathcal{K}_r,\end{aligned} \tag{3.6}$$

whose optimal value converges to the optimal value of (BQPP) due to the fact that at the limit the cone $\mathcal{K}_r$ contains the interior of $\mathcal{P}_2(H \cap S)$. The following theorem follows by applying Corollary 1 of [31] and Putinar's theorem [35].

THEOREM 3.1. *The sequence of cones $\mathcal{K}_r$ satisfies*

$$\mathcal{K}_r \subseteq \mathcal{K}_{r+1} \subseteq \cdots \subseteq \mathcal{P}_2(H \cap S) \text{ and } \operatorname{int}(\mathcal{P}_2(H \cap S)) \subseteq \bigcup_{r=0}^{\infty} \mathcal{K}_r \subseteq \mathcal{P}_2(H \cap S).$$

Hence, $\lambda^*_{BQPP_{\mathcal{K}_r}} \uparrow z^*_{BQPP}$.

The size of the relaxations produced in the previous theorem grows exponentially in r. For this reason, instead of looking at the hierarchy of relaxations, we will concentrate on the first and simplest relaxation where $r = 0$,

$$\begin{aligned}\mathcal{K}_0 =&\Psi_2 + \sum_i (1 - x_i^2)\mathbf{R}_0 + \sum_j (b_j - a_j^T x)\mathbf{R}_1[x] + \sum_j (d_j - c_j^T x)\mathbf{R}_0^+\\ &+ \sum_j (k_j - x^T F_j x - e_j^T x)\mathbf{R}_0 + \sum_j (l_j - x^T G_j x - h_j^T x)\mathbf{R}_0^+.\end{aligned}$$

We study how to improve the approximation of $\mathcal{P}_2(H \cap S)$ using variations of the cone $\mathcal{K}_0$. The fundamental tool that we use to construct such inner approximations of $\mathcal{P}_2(H \cap S)$ is Lemma 2.2, a representation theorem for nonnegative linear polynomials over $\mathcal{B}$ which results in SOC conditions. These yield stronger approximations than $\mathcal{K}_0$ with an insignificant impact on the computational time.

3.2. New conic relaxations of BQPP. In this section, we present three relaxations for the BQPP problem. Two of these relaxations are based on SOC and SDP, and the final relaxation is solely based on SOC programming.

3.2.1. SOC-SDP-based relaxations of BQPP. Recall the previous polynomial formulation of the BQPP. First, notice that $x \in H$ implies $\|x\|_2^2 = n$. Therefore, $S \cap H \subseteq \mathcal{B}$, and by defining $\bar{\mathcal{K}}_0$ as

$$\begin{aligned}\bar{\mathcal{K}}_0 = \mathcal{P}_2(\mathcal{B}) &+ \sum_i (1+x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1-x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1-x_i^2)\mathbf{R}_0 \\ &+ \sum_j (b_j - a_j^T x)\mathbf{R}_1[x] + \sum_j (d_j - c_j^T x)\mathcal{P}_1(\mathcal{B}) + \sum_j (k_j - x^T F_j x - e_j^T x)\mathbf{R}_0 \\ &+ \sum_j (l_j - x^T G_j x - h_j^T x)\mathbf{R}_0^+,\end{aligned}$$

we have $\bar{\mathcal{K}}_0 \subseteq \mathcal{P}_2(S \cap \mathcal{B} \cap H) = \mathcal{P}_2(S \cap H)$.

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

$$\begin{aligned}\lambda - q(x) = s(x) &+ \sum_i (1+x_i)\alpha_i(x) + \sum_i (1-x_i)\beta_i(x) + \sum_i \gamma_i (1-x_i^2) \\ &+ \sum_j \delta_j(x)(b_j - a_j^T x) + \sum_j \eta_j(x)(d_j - c_j^T x) + \sum_j \theta_j (k_j - x^T F_j x - e_j^T x) \\ &+ \sum_j \xi_j (l_j - x^T G_j x - h_j^T x)\end{aligned}$$

with $s(x) = \begin{pmatrix}1 & x^T\end{pmatrix} S \binom{1}{x}$ and $S \in \mathcal{S}_+^{n+1}$, $\alpha_i(x) = \alpha_i^T \binom{\sqrt{n}}{x}$, $\beta_i(x) = \beta_i^T \binom{\sqrt{n}}{x}$, and $\eta_j(x) = \eta_j^T \binom{\sqrt{n}}{x}$, where $\alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}$, $\delta_j(x) \in R_1[x]$, $\gamma_i, \theta_j \in \mathbb{R}$, and $\xi_j \in \mathbb{R}_+$.

We then obtain the following relaxation of (BQPP):

$$\begin{aligned}(\mathbf{BQPP_{ss}})\ \min\ & \lambda \\ \text{s.t. } \lambda - q(x) = & \begin{pmatrix}1 & x^T\end{pmatrix} S \begin{pmatrix}1 \\ x\end{pmatrix} + \sum_i (1+x_i)\alpha_i^T \begin{pmatrix}\sqrt{n} \\ x\end{pmatrix} \\ &+ \sum_i (1-x_i)\beta_i^T \begin{pmatrix}\sqrt{n} \\ x\end{pmatrix} + \sum_i \gamma_i (1-x_i^2) \\ &+ \sum_j \delta_j(x)(b_j - a_j^T x) + \sum_j (d_j - c_j^T x)\eta_j^T \begin{pmatrix}\sqrt{n} \\ x\end{pmatrix} \\ &+ \sum_j \theta_j (k_j - x^T F_j x - e_j^T x) \\ &+ \sum_j \xi_j (l_j - x^T G_j x - h_j^T x), \\ S \in \mathcal{S}_+^{n+1}, \qquad & \alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \qquad \gamma_i, \theta_j \in \mathbb{R}, \qquad \xi_j \in \mathbb{R}_+.\end{aligned}$$

To strengthen this relaxation we can add valid inequalities to the original problem (BQPP) which is equivalent to adding more variables to the relaxation due to the next lemma.

LEMMA 3.2. *For any S, d, and $f \in \mathbf{R}_d[x]$,*

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

Notice that products of linear constraints, such as $(d_k - c_k^T x)(1+x_i)$, $(d_k - c_k^T x)(1-x_i)$, $(d_k - c_k^T x)(d_l - c_l^T x)$, $(1-x_j)(1-x_i)$, $(1+x_j)(1+x_i)$, and $(1-x_j)(1+x_i)$, are also considered as valid inequalities and can be added to ($\mathrm{BQPP}_{\mathrm{SS}}$) to further strengthen the relaxation. Hence, we obtain

$$
\begin{aligned}
(\mathbf{BQPP}_{\mathbf{ss^+}})\ \min\ & \lambda \\
\text{s.t. } \lambda - q(x) = & \begin{pmatrix}1 & x^T\end{pmatrix} S \begin{pmatrix}1\\ x\end{pmatrix} + \sum_i (1+x_i)\alpha_i^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} \\
& + \sum_i (1-x_i)\beta_i^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_i \gamma_i(1-x_i^2) \\
& + \sum_j \delta_j(x)(b_j - a_j^T x) + \sum_j (d_j - c_j^T x)\eta_j^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} \\
& + \sum_j \theta_j(k_j - x^T F_j x - e_j^T x) + \sum_j \xi_j(l_j - x^T G_j x - h_j^T x) \\
& + \sum_{i,k} \sigma_{ik}(d_k - c_k^T x)(1+x_i) + \sum_{i,k} \mu_{ik}(d_k - c_k^T x)(1-x_i) \\
& + \sum_{k\le l} \nu_{kl}(d_k - c_k^T x)(d_l - c_l^T x) + \sum_{i\le j} \tau_{ij}(1-x_i)(1-x_j) \\
& + \sum_{i\le j} \omega_{ij}(1+x_i)(1+x_j) + \sum_{i,j} \phi_{ij}(1-x_i)(1+x_j), \\
& S \in \mathcal{S}_+^{n+1}, \qquad \alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \qquad \gamma_i, \theta_j \in \mathbb{R}, \\
& \xi_j, \sigma_{ik}, \mu_{ik}, \nu_{kl}, \tau_{ij}, \omega_{ij}, \phi_{ij} \in \mathbb{R}_+.
\end{aligned}
$$

3.2.2. Pure SOC-based relaxations of BQPP. The relaxation ($\mathrm{BQPP}_{\mathrm{SS}}$) can be further relaxed by removing the positive semidefinite variable leading to the following relaxation:

$$
\begin{aligned}
\min\ & \lambda \\
\text{s.t. } \lambda - q(x) = & \sum_i (1+x_i)\alpha_i^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_i (1-x_i)\beta_i^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} \\
& + \sum_i \gamma_i(1-x_i^2) + \sum_j \delta_j(x)(b_j - a_j^T x) \\
& + \sum_j (d_j - c_j^T x)\eta_j^T \left(\sqrt{n}x\right) + \sum_j \theta_j(k_j - x^T F_j x - e_j^T x) \\
& + \sum_j \xi_j(l_j - x^T G_j x - h_j^T x), \\
& \alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \qquad \gamma_i, \theta_j \in \mathbb{R}, \qquad \xi_j \in \mathbb{R}_+.
\end{aligned}
$$

One type of valid inequalities that we consider for BQPP is:

$$
-1 \le x_i x_j \le 1. \tag{3.7}
$$

These inequalities are not violated in the presence of the SDP term. However, once the SDP term is removed, these constraints are no longer satisfied, and adding them will strengthen the SOC relaxation.

Hence, we obtain our proposed SOC-based relaxation:

$$
\begin{aligned}
(\mathbf{BQPP_{soc}})\ \min\ & \lambda \\
\text{s.t. } \lambda - q(x) = {} & \sum_i (1+x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1-x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
& + \sum_i \gamma_i (1-x_i^2) + \sum_j \delta_j(x)(b_j - a_j^T x) \\
& + \sum_j (d_j - c_j^T x)\eta_j^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_j \theta_j (k_j - x^T F_j x - e_j^T x) \\
& + \sum_j \xi_j (l_j - x^T G_j x - h_j^T x) + \sum_{i<j} h_{ij}^+ (1 + x_i x_j) \\
& + \sum_{i<j} h_{ij}^- (1 - x_i x_j), \\
& \alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \qquad \gamma_i, \theta_j \in \mathbb{R}, \qquad \xi_j, h_{ij}^+, h_{ij}^- \in \mathbb{R}_+.
\end{aligned}
$$

By construction we have the following theorem relating the three presented relaxations.

THEOREM 3.3. *Let $\lambda^*_{BQPP_{SOC}}$, $\lambda^*_{BQPP_{SS}}$, and $\lambda^*_{BQPP_{SS^+}}$ be the optimal solution value of ($BQPP_{SOC}$), ($BQPP_{SS}$), and ($BQPP_{SS^+}$), respectively; then*

$$\lambda^*_{BQPP_{SOC}} \ge \lambda^*_{BQPP_{SS}} \ge \lambda^*_{BQPP_{SS^+}} \ge z^*_{BQPP}.$$

4. Applications. In this section, we apply our proposed framework to the following classes of constrained BQPPs:

- General quadratic polynomial problems;
- Quadratic linear constrained problems;
- Quadratic knapsack problems.

First, we start with the most general class of BQPPs where we have quadratic and linear constraints. Then we consider the special case with only linear constraints, and finally we consider problems with a single linear constraint. We re-derive existing relaxations that have been proposed in the literature for each of these problems and theoretically compare our proposed two SOC-SDP-based relaxations to them. We show theoretically that we obtain stronger relaxations based on applying the methodology of section 3. In addition, in section 5 we compare the relaxations computationally for each of these three classes of binary quadratic problems. Our computational results show that more efficient relaxations are obtained if the SDP term is omitted.

4.1. General quadratic polynomial problems. In this section, we consider the general BQPP. Lasserre [18] introduced SDP relaxations for binary polynomial programs by approximating $\mathcal{P}_2(H \cap S)$ using the cone

$$
\begin{aligned}
\Gamma_r := \Bigg(& \Psi_{r+2} + \sum_i (1-x_i^2)\Psi_r + \sum_i (x_i^2 - 1)\Psi_r + \sum_i (b_i - a_i^T x)\Psi_r \\
& + \sum_i (a_i^T x - b_i)\Psi_r + \sum_i (d_i - c_i^T x)\Psi_r + \sum_i (k_i - x^T F_i x - e_i^T x)\Psi_r \\
& + \sum_i (x^T F_i x + e_i^T x - k_i)\Psi_r + \sum_i (l_i - x^T G_i x - h_i^T x)\Psi_r \Bigg) \cap \mathbf{R}_2[x]
\end{aligned}
$$

for even $r \ge 0$.

LEMMA 4.1. $\mathit{\Gamma}_r \subseteq \mathcal{K}_r$.

Notice that from the definition of Ψ_r, $\mathit{\Gamma}_r = \mathit{\Gamma}_{r-1}$ for odd r. Also, $\mathcal{K}_r = \mathit{\Gamma}_r$ for even r, since $\mathbf{R}_r[x] = \Psi_r - \Psi_r$. However, when r is odd, $\Psi_r = \Psi_{r-1}$ so that $\mathbf{R}_r[x] \supsetneq \Psi_{r-1} - \Psi_{r-1}$, and thus $\mathit{\Gamma}_r \subsetneq \mathcal{K}_r$.

Taking $r = 0$, we obtain the Lasserre relaxation of order 1 (L1) for (BQPP):

$$\textbf{(BQPP}_{\textbf{L1}}\textbf{)} \ \min \lambda \quad \text{s.t. } \lambda - q(x) \in \mathit{\Gamma}_0.$$

Theorem 4.2 shows that ($\mathrm{BQPP}_{\mathrm{SS}^+}$) provides the best bound for the BQPP while ($\mathrm{BQPP}_{\mathrm{SS}}$) has a better bound than Lasserre's relaxation of order 1.

THEOREM 4.2. *Let $\lambda^*_{BQPP_{L1}}$, $\lambda^*_{BQPP_{SS}}$, and $\lambda^*_{BQPP_{SS^+}}$ be the optimal solution value of ($BQPP_{L1}$), ($BQPP_{SS}$), and ($BQPP_{SS^+}$), respectively; then*

$$\lambda^*_{BQPP_{L1}} \geq \lambda^*_{BQPP_{SS}} \geq \lambda^*_{BQPP_{SS^+}} \geq z^*_{BQPP}.$$

Proof. Define

$$\begin{aligned}
\mathcal{H}_1 =& \Psi_2 + \sum_i (1 - x_i^2)\mathbf{R}_0 + \sum_i (b_i - a_i^T x)\mathbf{R}_1[x] + \sum_i (d_i - c_i^T x)\mathcal{P}_1(\mathcal{B}) \\
&+ \sum_i (k_i - x^T F_i x - e_i^T x)\mathbf{R}_0 + \sum_i (l_i - x^T G_i x - h_i^T x)\mathbf{R}_0^+, \\
\mathcal{H}_2 =& \mathcal{H}_1 + (d_i - c_i^T x)\mathcal{P}_1(\mathcal{B}) + \sum_i (1 + x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1 - x_i)\mathcal{P}_1(\mathcal{B}) = \bar{\mathcal{K}}_0, \\
\mathcal{H}_3 =& \mathcal{H}_2 + \sum_{i,k} \Psi_0 (1 + x_i)(d_k - c_k^T x) + \sum_{i,k} \Psi_0 (1 - x_i)(d_k - c_k^T x) \\
&+ \sum_{k \leq l} \Psi_0 (d_k - c_k^T x)(d_l - c_l^T x) + \sum_{i \leq j} \Psi_0 (1 + x_i)(1 + x_j) \\
&+ \sum_{i \leq j} \Psi_0 (1 - x_i)(1 - x_j) + \sum_{i,j} \Psi_0 (1 + x_i)(1 - x_j).
\end{aligned}$$

We have

$$\Psi_1 = \mathbf{R}_0^+ \subseteq \mathcal{P}_1(\mathcal{B}) \Rightarrow \mathcal{K}_0 \subseteq \mathcal{H}_1.$$

In addition, from Lemma 4.1, by setting r to zero we have $\mathit{\Gamma}_0 \subseteq \mathcal{K}_0$ and, therefore,

$$\mathit{\Gamma}_0 \subseteq \mathcal{K}_0 \subseteq \mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \mathcal{H}_3. \qquad \square$$

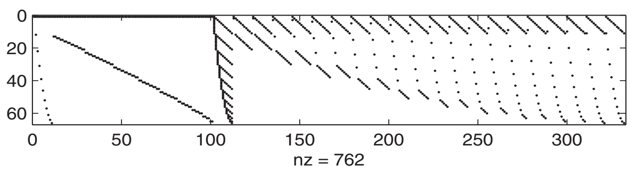
We now compare the relaxations in terms of computational complexity. Table 4.1 summarizes the number of variables (and for SDPs, the dimension) for each of the resulting optimization problems. Recall that the BQPP has t linear equalities, u linear inequalities, v quadratic equalities, w quadratic inequalities, and n binary variables.

While the complexities of ($\mathrm{BQPP}_{\mathrm{SS}^+}$), ($\mathrm{BQPP}_{\mathrm{SS}}$), and ($\mathrm{BQPP}_{\mathrm{L1}}$) are similar, we see that ($\mathrm{BQPP}_{\mathrm{SOC}}$) trades off an $(n+1)\times(n+1)$ SDP matrix variable for $n(n-1)$ linear nonnegative variables. If one applies an interior-point method to solve the relaxations, it is not immediately clear that ($\mathrm{BQPP}_{\mathrm{SOC}}$) is computationally cheaper. To see this, recall that the most time-consuming step for an interior-point algorithm is solving the Schur complement equation at each iteration. For this task, sparsity of the constraint

TABLE 4.1
Problem dimension for various BQPP relaxations.

Relaxation	SDP	SOC	Linear Nonnegative	Linear-free
$(\mathrm{BQPP}_{\mathrm{SS}^+})$	$1,\ (n+1)\times(n+1)$	$(2n+u)\ ,\ (n+1)$	$w+2tn+\binom{t+1}{2}+2\binom{n+1}{2}+n^2$	$n+(n+1)t+v$
$(\mathrm{BQPP}_{\mathrm{SS}})$	$1,\ (n+1)\times(n+1)$	$(2n+u)\ ,\ (n+1)$	w	$n+(n+1)t+v$
$(\mathrm{BQPP}_{\mathrm{L1}})$	$1,\ (n+1)\times(n+1)$	-	$2n+2t+2v+u+w$	-
$(\mathrm{BQPP}_{\mathrm{SOC}})$	-	$(2n+u)\ ,\ (n+1)$	$w+n(n-1)$	$n+(n+1)t+v$

matrices is key, and in this regard the trade-off is to the advantage of $(\mathrm{BQPP}_{\mathrm{SOC}})$. This is because in SDP, the Schur complement matrix is typically dense even when the constraint matrix is sparse, and hence computing the Cholesky factorization remains expensive. In contrast, for linear programming, a sparse constraint matrix results in a sparse Schur complement matrix, and this sparsity property can be exploited to speed up computation of the Cholesky factorization [9]. To illustrate that sparsity is present in our relaxations, we present in Figure 4.1 the nonzero elements of the constraint matrices of $(\mathrm{BQPP}_{\mathrm{SOC}})$ and $(\mathrm{BQPP}_{\mathrm{SS}})$ for an instance of the QKP with $n = 10$.



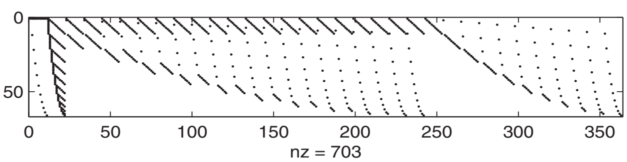


FIG. 4.1. *Sparsity of the constraint matrices for $(BQPP_{SOC})$ (top) and $(BQPP_{SS})$ (bottom).*

4.2. QLCPs. Without loss of generality, we formulate the binary QLCP as

$$\textbf{(QLCP)}\ \max\ x^TQx + p^Tx \quad \text{s.t. } a_j^Tx \le b_j \quad \forall j \in \{1,\cdots,m\},\quad x \in \{-1,1\}^n.$$

Specializing the results of section 3.2 to (QLCP), we obtain the following polynomial programming relaxations:

$$\textbf{(QLCP}_{\mathbf{ss}}\textbf{)}\ \min\ \lambda$$

$$\text{s.t. } \lambda - q(x) = \begin{pmatrix}1 & x\end{pmatrix} S \begin{pmatrix}1\\ x\end{pmatrix} + \sum_{j=1}^{m}(b_j - a_j^Tx)d_j^T\begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_{i=1}^{n}(1+x_i){f_i}^T\begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_{i=1}^{n}(1-x_i){g_i}^T\begin{pmatrix}\sqrt{n}\\ x\end{pmatrix}$$

$$+\sum_{i=1}^{n} c_i(1-x_i^2),$$

$$c_i \in \mathbb{R}, \qquad f_i, g_i, d_j \in \mathcal{L}^{n+1}, \qquad S \in \mathcal{S}_+^{n+1};$$

$(\mathbf{QLCP_{ss^+}})$ min λ

$$\begin{aligned}
\text{s.t. } \lambda - q(x) = {} & \begin{pmatrix}1 & x\end{pmatrix} S \begin{pmatrix}1\\ x\end{pmatrix} + \sum_{j=1}^{m}(b_j - a_j^T x) d_j^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix}\\
& + \sum_{i=1}^{n}(1+x_i){f_i}^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_{i=1}^{n}(1-x_i){g_i}^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix}\\
& + \sum_{i=1}^{n}\sum_{k=1}^{m} \alpha_{ik}(1+x_i)(b_k - a_k^T x)\\
& + \sum_{i=1}^{n}\sum_{k=1}^{m} \beta_{ik}(1-x_i)(b_k - a_k^T x)\\
& + \sum_{i=1}^{n}\sum_{j=i}^{n} \gamma_{ij}(1+x_i)(1+x_j)\\
& + \sum_{i=1}^{n}\sum_{j=i}^{n} \delta_{ij}(1-x_i)(1-x_j) + \sum_{i=1}^{n}\sum_{j=1}^{n} \zeta_{ij}(1+x_i)(1-x_j)\\
& + \sum_{k=1}^{m}\sum_{l=k}^{m} \eta_{kl}(b_k - a_k^T x)(b_l - a_l^T x) + \sum_{i=1}^{n} c_i(1-x_i^2),
\end{aligned}$$

$$c_i \in \mathbb{R}, \qquad \alpha_{ik}, \beta_{ik}, \gamma_{ij}, \delta_{ij}, \zeta_{ij}, \eta_{kl} \in \mathbb{R}_+^n,$$
$$f_i, g_i, d_j \in \mathcal{L}^{n+1}, \qquad S \in \mathcal{S}_+^{n+1};$$

$(\mathbf{QLCP_{soc}})$ min λ

$$\begin{aligned}
\text{s.t. } \lambda - q(x) = {} & \sum_{j=1}^{m}(b_j - a_j^T x) d_j^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_{i=1}^{n}(1+x_i){f_i}^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix}\\
& + \sum_{i=1}^{n}(1-x_i){g_i}^T \begin{pmatrix}\sqrt{n}\\ x\end{pmatrix} + \sum_{i=1}^{n} c_i(1-x_i^2)\\
& + \sum_{i<j} h_{ij}^+(1+x_i x_j) + \sum_{i<j}^{n} h_{ij}^-(1-x_i x_j),
\end{aligned}$$

$$c_i \in \mathbb{R}, \qquad h_{ij}^+, h_{ij}^- \in \mathbb{R}^+, \qquad f_i, g_i, d_j \in \mathcal{L}^{n+1}.$$

4.2.1. The relaxation of Burer and Lovász–Schrijver. Burer [6] presented an SDP-based relaxation for the QLCP where the variables are 0-1. We introduce the following relaxation that is at least as strong as the relaxation presented by Burer [39]:

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

where m is the number of linear constraints. Further, ($\text{QLCP}_{\text{Burer'}}$) is equivalent to

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

which can be written as

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

Notice that ($\text{QLCP}_{\text{Burer'}}$) reduces to the N^+ relaxation of Lovász and Schrijver [23] by setting the variables $\gamma_{ij}, \delta_{ij}, \zeta_{ij}$, and η_{kl} to zero. That is, N^+ is equivalent to the following relaxation:

$$
\begin{aligned}
(\mathbf{QLCP}_{N^+})\ \min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = \begin{pmatrix} 1 & x \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + \sum_{i=1}^{n} c_i x_i (1-x_i) \\
& \qquad + \sum_{i=1}^{n} \sum_{k=1}^{m} \alpha_{ik} x_i (b_k - a_k^T x) + \sum_{i=1}^{n} \sum_{k=1}^{m} \beta_{ik} (1-x_i)(b_k - a_k^T x), \\
& c_i \in \mathbb{R}, \qquad \alpha_{ik}, \beta_{ik} \in \mathbb{R}_+, \qquad S \in \mathcal{S}_+^{n+1}.
\end{aligned}
$$

4.2.2. Comparing relaxations for QLCP. We can prove the following result.

THEOREM 4.3. *Let $\lambda^*_{QLCP_{N^+}}$, $\lambda^*_{QLCP_{Burer'}}$, and $\lambda^*_{QLCP_{SS^+}}$ be the optimal solution value of ($QLCP_{N^+}$),($QLCP_{Burer'}$), and ($QLCP_{SS^+}$), respectively; then*

$$\lambda^*_{QLCP_{N^+}} \geq \lambda^*_{QLCP_{Burer'}} \geq \lambda^*_{QLCP_{SS^+}} \geq z^*_{QLCP}.$$

Proof. Define

$$\begin{aligned}
\mathcal{H}_4 =& \Psi_2 + \sum_{i,k} \Psi_0(1+x_i)(b_k - a_k^T x) + \sum_{i,k} \Psi_0(1-x_i)(b_k - a_k^T x) + \sum_i (1-x_i^2)\mathbf{R}_0, \\
\mathcal{H}_5 =& \mathcal{H}_4 + \sum_{i\leq j} \Psi_0(1+x_i)(1+x_j) + \sum_{i\leq j} \Psi_0(1-x_i)(1-x_j) + \sum_{i,j} \Psi_0(1+x_i)(1-x_j) \\
& + \Psi_0 \sum_{k\leq l} (b_k - a_k^T x)(b_l - a_l^T x), \\
\mathcal{H}_6 =& \mathcal{H}_5 + \sum_j (b_j - a_j^T x)\mathcal{P}_1(\mathcal{B}) + \sum_i (1+x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1-x_i)\mathcal{P}_1(\mathcal{B}).
\end{aligned}$$

Hence,

$$\mathcal{H}_4 \subseteq \mathcal{H}_5 \subseteq \mathcal{H}_6.$$

After a simple change of variables from $\{-1,1\}$ to $\{0,1\}$, $\mathcal{H}_4$ and $\mathcal{H}_5$ correspond to the representations ($QLCP_{N^+}$) and ($QLCP_{Burer'}$), respectively, while $\mathcal{H}_6$ corresponds to the representation ($QLCP_{SS^+}$). ∎

Table 4.2 lists the number of variables required to formulate the various relaxations for the QLCP of Theorem 4.3, in addition to ($QLCP_{SS}$) and ($QLCP_{SOC}$) where we have m linear constraints and n binary variables. While ($QLCP_{Burer'}$) and ($QLCP_{SS^+}$) have the same computational complexity, ($QLCP_{SS^+}$) provides the best bounds as shown in Theorem 4.3 and confirmed by the computational results of section 5.

TABLE 4.2
Problem dimension for various QLCP relaxations.

Relaxation	SDP	SOC	Linear Nonnegative	Linear-free
($QLCP_{SS^+}$)	1, $(n+1)\times(n+1)$	$(2n+m)$, $(n+1)$	$2nm + n^2 + 2\binom{n+1}{2} + \binom{m+1}{2}$	n
($QLCP_{Burer'}$)	1, $(n+1)\times(n+1)$	-	$2nm + n^2 + 2\binom{n+1}{2} + \binom{m+1}{2}$	n
($QLCP_{SS}$)	1, $(n+1)\times(n+1)$	$(2n+m)$, $(n+1)$	-	n
($QLCP_{N^+}$)	1, $(n+1)\times(n+1)$	-	$2nm$	n
($QLCP_{SOC}$)	-	$(2n+m)$, $(n+1)$	$n(n-1)$	n

Remark 4.4. We are unable to compare theoretically ($QLCP_{SS}$) with ($QLCP_{N^+}$) and ($QLCP_{Burer'}$). However, in our computational experiments in section 5.2, ($QLCP_{SS}$) always provides a strictly better bound than ($QLCP_{N^+}$), while ($QLCP_{Burer'}$) provides a strictly better bound than ($QLCP_{SS}$).

4.3. QKP. In this section, we consider the QKP which is the particular case of QLCP where $m = 1$. The QKP was introduced by Gallo, Hammer, and Simeone [10] and is NP-hard. The QKP can be interpreted as follows: we are given n items with a nonnegative weight w_i assigned to item i, and an $(n+1)\times(n+1)$ symmetric matrix Q with real entries. 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. Introducing the binary variable x_i such that

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

the problem may be formulated as:

$$\begin{aligned} \textbf{(QKP-P)} \max\ & q(x) = \begin{pmatrix} 1 & x \end{pmatrix} Q \begin{pmatrix} 1 \\ x \end{pmatrix} \\ \text{s.t. } & w^T x \leq c, \\ & x \in \{-1, 1\}^n. \end{aligned}$$

The QKP is a generalization of the linear knapsack problem (where the objective function is linear). As in the case of the linear knapsack problem, the QKP often appears as a subproblem to other complex problems such as the graph partitioning problem described in Johnson, Mehrotra, and Nemhauser [13]. Since the QKP is a constrained version of the binary quadratic problem, all valid inequalities for the unconstrained BQPP are also valid for the QKP, and hence they can be used to tighten bounds for this problem. Using the same approach as in section 3.2, we obtain the following relaxations of (QKP-P):

$$\begin{aligned} (\mathbf{QKP_{ss}}) \min\ & \lambda \\ \text{s.t. } \lambda - q(x) = {} & \begin{pmatrix} 1 & x \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + (c - w^T x) d^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\ & + \sum_{i=1}^{n} (1 + x_i) {f_i}^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_{i=1}^{n} (1 - x_i) {g_i}^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\ & + \sum_{i=1}^{n} c_i (1 - x_i^2), \\ & c_i \in \mathbb{R}, \qquad f_i, g_i, d \in \mathcal{L}^{n+1}, \qquad S \in \mathcal{S}_+^{n+1}; \end{aligned}$$

$$\begin{aligned} (\mathbf{QKP_{ss^+}}) \min\ & \lambda \\ \text{s.t. } \lambda - q(x) = {} & \begin{pmatrix} 1 & x \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + (c - w^T x) d^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\ & + \sum_{i=1}^{n} (1 + x_i) {f_i}^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_{i=1}^{n} (1 - x_i) {g_i}^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\ & + \sum_{i=1}^{n} \alpha_i (1 + x_i)(c - w^T x) + \sum_{i=1}^{n} \beta_i (1 - x_i)(c - w^T x) \\ & + \sum_{i \leq j} \gamma_{ij} (1 + x_j)(1 + x_i) + \sum_{i \leq j} \delta_{ij} (1 - x_j)(1 - x_i) \\ & + \sum_{i,j} \zeta_{ij} (1 - x_j)(1 + x_i) + \sum_{i=1}^{n} c_i (1 - x_i^2), \\ & c_i \in \mathbb{R}, \qquad \alpha_i, \beta_i, \gamma_{ij}, \delta_{ij}, \zeta_{ij} \in \mathbb{R}^+, \qquad f_i, g_i, d \in \mathcal{L}^{n+1}, \\ & S \in \mathcal{S}_+^{n+1}; \end{aligned}$$

$$
\begin{aligned}
\textbf{(QKP}_{\textbf{soc}}\textbf{)}\ \min\ & \lambda \\
\text{s.t. } & \lambda - q(x) = (c - w^T x) d^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_{i=1}^n (1+x_i) {f_i}^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
& \quad + \sum_{i=1}^n (1-x_i) {g_i}^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_{i=1}^n c_i (1 - x_i^2) \\
& \quad + \sum_{i<j} h_{ij}^+ (1 + x_i x_j) + \sum_{i<j} h_{ij}^- (1 - x_i x_j), \\
& c_i \in \mathbb{R}, \qquad h_{ij}^+, h_{ij}^- \in \mathbb{R}^+ \qquad f_i, g_i, d \in \mathcal{L}^{n+1}.
\end{aligned}
$$

4.3.1. Helmberg–Rendl–Weismantel QKP relaxation. Helmberg, Rendl, and Weismantel [12] presented four SDP-based relaxations for the QKP where the discrete set is $\{0,1\}^n$. These relaxations are obtained by considering the semidefinite matrix $X = xx^T$. In particular, they studied the relaxation

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

where $\bar{c} = \frac{1}{2}(\sum_i w_i - c)$, P is an $n \times n$ matrix with entries $P_{ij} = 4Q_{ij}$ (for $i \ne j$) and $P_{ii} = 4Q_{ii} - 4\sum_j Q_{ij} + 4Q_{0i}$, and $\text{cst} = Q_{00} - 2\sum_i Q_{0i} + \sum_{i,j} Q_{ij}$ are obtained by mapping the variables from $\{-1,1\}$ to $\{0,1\}$. Helmberg, Rendl, and Weismantel [12] showed that the optimal objective value of (QKP_{HRW4}), $\lambda^*_{\text{QKP}_{\text{HRW4}}}$, 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, Rendl, and Weismantel [12] strengthen these proposed relaxations by using cutting planes that are valid for BQPP. To illustrate the quality of these SDP relaxations and of the cutting planes, Helmberg, Rendl, and Weismantel [12] present computational results on instances with up to 61 items.

4.3.2. Comparing relaxations for QKP. In this section, we compare (QKP_{HRW4}) and our proposed relaxation. First we rederive (QKP_{HRW4}) in a different way by considering the problem

$$
\begin{aligned}
\textbf{(QKP-D)}\ \min\ & \lambda \\
\text{s.t. } & \lambda - p(x) \in \mathcal{P}_2(\{0,1\}^n \cap \{x : (\bar{c} - w^T x) \ge 0\}),
\end{aligned}
$$

where $p(x) = \sum_{i,j} P_{ij} x_i x_j + \text{cst}$. This problem can be relaxed using

$$
\mathcal{P}_2(\{0,1\}^n \cap \{x : (\bar{c} - w^T x) \ge 0\}) \supseteq \Psi_2 + \sum_i \Psi_0 x_i (\bar{c} - w^T x) + \sum_i x_i (1 - x_i) \mathbf{R}_0,
$$

obtaining

$$
\begin{aligned}
\min\ & \lambda \\
\text{s.t. } & \lambda - p(x) = \begin{pmatrix} 1 & x \end{pmatrix} S \begin{pmatrix} 1 \\ x \end{pmatrix} + \sum_i d_i x_i (\bar{c} - w^T x) + \sum_i c_i x_i (1 - x_i),
\end{aligned}
$$

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

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

where $\delta_{i=j}$ equals 1 if $i = j$, and 0 otherwise. Taking the dual of ($\text{QKP}_{\text{HRW4-D}}$), we obtain

$$\begin{aligned}
\max\ & \langle \bar{P}, \bar{X} \rangle \\
\text{s.t.}\ & \bar{X}_{00} = 1, && (4.1) \\
& \bar{X}_{ii} - \bar{X}_{i0} = 0, \quad 1 \le i \le n, && (4.2) \\
& \sum_{j=1}^{n} w_j \bar{X}_{ij} - \bar{c} \bar{X}_{ii} \le 0, \quad 1 \le i \le n, && (4.3) \\
& \bar{X} \succeq 0, && (4.4)
\end{aligned}$$

where $\bar{P} = \left(\begin{smallmatrix} \text{cst} & 0 \\ 0 & P \end{smallmatrix}\right)$. Since $X - \text{diag}(X)\text{diag}(X)^T \succeq 0$ is equivalent to $\bar{X} = \left(\begin{smallmatrix} 1 & \text{diag}(X)^T \\ \text{diag}(X) & X \end{smallmatrix}\right) \succeq 0$, the above problem is a reformulation of (QKP_{HRW4}). Taking $X = I$, $\bar{X}$ is strictly feasible for (QKP_{HRW4}); therefore Slater's constraint qualification is satisfied for (QKP_{HRW4}). In addition, $X - \text{diag}(X)\text{diag}(X)^T \succeq 0$ implies $-\frac{1}{8} \le X_{ij} \le 1$ [12]. 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. *Let $\lambda^*_{QKP_{HRW4\text{-}D}}$ and $\lambda^*_{QKP_{SS^+}}$ be the optimal solution values of ($QKP_{HRW4\text{-}D}$) and (QKP_{SS^+}), respectively; then*

$$\lambda^*_{QKP_{HRW4\text{-}D}} = \lambda^*_{QKP_{HRW4}} \ge \lambda^*_{QKP_{SS^+}} \ge z^*_{QKP}.$$

Proof. Define

$$\begin{aligned}
\mathcal{H}_7 =& \Psi_2 + \sum_i \Psi_0(1 + x_i)(c - w^T x) + \sum_i (1 - x_i^2)\mathbf{R}_0, \\
\mathcal{H}_8 =& \mathcal{H}_7 + \sum_i \Psi_0(1 - x_i)(c - w^T x) + \sum_{i \le j} \Psi_0(1 + x_i)(1 + x_j) \\
& + \sum_{i \le j} \Psi_0(1 - x_i)(1 - x_j) + \sum_{i,j} \Psi_0(1 + x_i)(1 - x_j) + (c - w^T x)\mathcal{P}_1(\mathcal{B}) \\
& + \sum_i (1 + x_i)\mathcal{P}_1(\mathcal{B}) + \sum_i (1 - x_i)\mathcal{P}_1(\mathcal{B}).
\end{aligned}$$

Hence,

$$\mathcal{H}_7 \subseteq \mathcal{H}_8.$$

After mapping the variables from $\{-1, 1\}$ to $\{0, 1\}$, $\mathcal{H}_7$ corresponds to the approximation of $\mathcal{P}_2(\{0,1\}^n \cap \{x : (\bar{c} - w^T x) \ge 0\})$ that is equivalent to ($\text{QKP}_{\text{HRW4-D}}$), and $\mathcal{H}_8$ corresponds to the representation (QKP_{SS^+}). ∎

Table 4.3 presents the number of variables for the relaxations ($\mathrm{QKP}_{\text{HRW4-D}}$), ($\mathrm{QKP}_{\text{SS+}}$), ($\mathrm{QKP}_{\text{SS}}$), and ($\mathrm{QKP}_{\text{SOC}}$). Notice that the first two relaxations have the same computational complexity. However, the ($\mathrm{QKP}_{\text{SS+}}$) relaxation provides the best bounds as shown in Theorem 4.5.

TABLE 4.3
Problem dimension for various QKP relaxations.

Relaxation	SDP	SOC	Linear Nonnegative	Linear-free
($\mathrm{QKP}_{\text{SS+}}$)	1, $(n+1)\times(n+1)$	$(2n+1)$, $(n+1)$	$2n+n^2+2\binom{n+1}{2}$	n
(QKP_{SS})	1, $(n+1)\times(n+1)$	$(2n+1)$, $(n+1)$	-	n
($\mathrm{QKP}_{\text{HRW4-D}}$)	1, $(n+1)\times(n+1)$	-	n	n
($\mathrm{QKP}_{\text{SOC}}$)	-	$(2n+1)$, $(n+1)$	$n(n-1)$	n

Remark 4.6. In some instances, even when using the weaker relaxation (QKP_{SS}), we obtain a strictly better bound than ($\mathrm{QKP}_{\text{HRW4}}$) as shown in section 5.3. For those instances, ($\mathrm{QKP}_{\text{SS+}}$) is also strictly better than ($\mathrm{QKP}_{\text{HRW4}}$).

5. Computational results. In this section, we present computational results obtained by implementing the proposed relaxations of section 3.2 to the three classes of BQPPs considered in section 4. We conduct comparisons based on computational time and on the quality of the bounds. The focus is on verifying the efficiency of the proposed SOC relaxations compared to the SOS/SDP-based relaxations. All relaxations were implemented with MATLAB 7.9.0 for constructing the problems, and SeDuMi solver version 1.3 [37] was used to solve the conic problems. The experiments were done on a 1200 MHz Sun Sparc machine, and the reported computational time is in cpu seconds.

5.1. General BQPPs computational results. In this section, we compare our proposed relaxations with Lasserre's relaxation of order 1 for solving general binary quadratic problems. We compare the following four relaxations:
($\mathbf{BQPP}_{\text{SS+}}$): the relaxation presented in section 3.2;
($\mathbf{BQPP}_{\text{SS}}$): the SOC-SDP-based relaxation presented in section 3.2;
($\mathbf{BQPP}_{\text{L1}}$): the relaxation presented in section 4.1;
($\mathbf{BQPP}_{\text{SOC}}$): the SOC relaxation presented in section 3.2.
We consider 100 randomly generated instances that vary in size, n, from 10 items up to 70 and density from 20% to 100%. Each instance has n quadratic equality constraints of the form $1-x_i^2=0$ to formulate the binary constraints. In addition, each instance has an equal number m of linear inequality constraints and of quadratic inequality constraints, with m varying from 1 to $\frac{n}{2}$. We implemented ($\mathrm{BQPP}_{\text{L1}}$) using our code. In Table 5.1, we report the average gap and the average computational time of all four relaxations (the average is computed over 5 instances for each combination of n and m). The gap (in %) is calculated as follows:

$$gap = 100 \times \frac{ub_{\text{relaxation}} - ub_{\text{best}}}{ub_{\text{best}}}\%,$$

where the best upper bound is the one obtained by the ($\mathrm{BQPP}_{\text{SS+}}$) relaxation.

The bound of ($\mathrm{BQPP}_{\text{SS+}}$) is the strongest among the four relaxations; therefore, we report the average gaps of ($\mathrm{BQPP}_{\text{SS}}$), ($\mathrm{BQPP}_{\text{L1}}$), and ($\mathrm{BQPP}_{\text{SOC}}$) relative to ($\mathrm{BQPP}_{\text{SS+}}$). Observe that ($\mathrm{BQPP}_{\text{SS}}$) provides better gaps than ($\mathrm{BQPP}_{\text{L1}}$) and ($\mathrm{BQPP}_{\text{SOC}}$) for all instances. To facilitate the comparison of ($\mathrm{BQPP}_{\text{L1}}$) and

Table 5.1
Computational results for the BQPP instances. The avg. gaps are with respect to ($BQPP_{SS+}$).

n	m	($\mathrm{BQPP}_{\mathrm{SS}+}$)	($\mathrm{BQPP}_{\mathrm{SS}}$)		($\mathrm{BQPP}_{\mathrm{L1}}$)		($\mathrm{BQPP}_{\mathrm{SOC}}$)	
		Time	Gap	Time	Gap	Time	Gap	Time
10	1	2.03	0.85	1.98	12.50	1.49	**2.40**	1.47
	5	1.99	2.15	1.71	32.40	1.27	**2.63**	1.07
20	1	5.42	0.24	5.36	7.96	4.18	**1.78**	1.81
	5	10.60	2.64	6.37	20.37	5.56	**16.59**	2.12
	10	14.96	4.26	6.66	72.30	5.36	**8.08**	2.42
30	1	22.33	1.09	16.39	**2.36**	10.32	28.11	7.69
	5	35.95	2.84	19.17	**23.88**	14.30	26.21	9.11
	15	73.29	10.74	22.94	**32.92**	17.02	53.21	11.19
40	1	78.18	1.66	56.30	34.89	34.59	**29.97**	33.07
	5	122.37	2.33	67.54	36.38	44.66	**28.18**	37.47
	20	306.31	5.71	88.80	50.60	48.27	**38.60**	44.11
50	1	268.93	0.68	179.74	**5.12**	112.49	15.16	48.72
	5	397.34	3.44	193.86	**17.71**	122.32	39.05	117.75
	25	1245.49	12.27	258.77	94.54	142.29	**43.08**	190.33
60	1	970.00	3.15	626.87	**19.61**	375.24	65.83	94.16
	5	1169.37	3.69	663.09	40.75	397.93	**39.75**	183.34
	30	5637.18	9.42	850.83	58.95	473.50	**52.10**	650.46
70	1	2793.31	0.93	2515.31	**29.44**	1214.23	31.51	165.63
	5	3848.18	2.50	2532.18	53.64	1245.09	**26.98**	549.22
	35	15420.53	14.85	2429.09	47.51	1446.99	**46.99**	1818.69
Avg.	-	-	4.27	-	34.69	-	30.06	

($\mathrm{BQPP}_{\mathrm{SOC}}$), we indicate the lower gap between them in bold. Notice that ($\mathrm{BQPP}_{\mathrm{SOC}}$) frequently has better gaps than ($\mathrm{BQPP}_{\mathrm{L1}}$).

In terms of computational cost, Table 5.1 and Figure 5.1 show that ($\mathrm{BQPP}_{\mathrm{SOC}}$) is the cheapest relaxation in most cases. When the number of linear constraints has a value of $\frac{n}{2}$, ($\mathrm{BQPP}_{\mathrm{L1}}$) is slightly cheaper, but for those cases the bounds provided by ($\mathrm{BQPP}_{\mathrm{L1}}$) are weaker than those provided by ($\mathrm{BQPP}_{\mathrm{SOC}}$). One can also observe that, for $n \geq 50$, higher computational times correspond to better bounds in most cases. However, this is misleading and is due to the averaging over 5 instances per line in the table. Looking at the detailed results for the 45 instances considered with $n \geq 50$, ($\mathrm{BQPP}_{\mathrm{L1}}$) is better than ($\mathrm{BQPP}_{\mathrm{SOC}}$) in terms of both time and bounds for 7 instances, and ($\mathrm{BQPP}_{\mathrm{SOC}}$) is better than ($\mathrm{BQPP}_{\mathrm{L1}}$) by both measures for 15 instances. Thus, higher times correspond to better bounds for roughly only half of the instances, and no clear conclusions can be drawn.

5.2. QLCP computational results. In this section, we compare our proposed relaxations of QLCP with the approach proposed by Burer [6] to solve BQPPs with linear constraints. Table 5.2 reports the average gap (in %) between each relaxation's upper bound and the optimal objective value (known a priori), as well as the average computational time. We compare five relaxations:

($\mathbf{QLCP}_{\mathbf{SS}+}$): the strengthened SDP relaxation presented in section 4.2;
($\mathbf{QLCP}_{\mathbf{Burer'}}$): the relaxation presented in section 4.2;
($\mathbf{QLCP}_{\mathbf{SS}}$): the SOC-SDP relaxation presented in section 4.2;
($\mathbf{QLCP}_{N+}$): the Lovász–Schrijver relaxation presented in section 4.2;
($\mathbf{QLCP}_{\mathbf{SOC}}$): the SOC relaxation presented in section 4.2.

We consider 732 instances that vary in size from 10 up to 50 items, and with density varying from 1% to 100%. The number of the linear constraints varies from 1 to 25. The data for the instances and their optimal objective values, as well as the

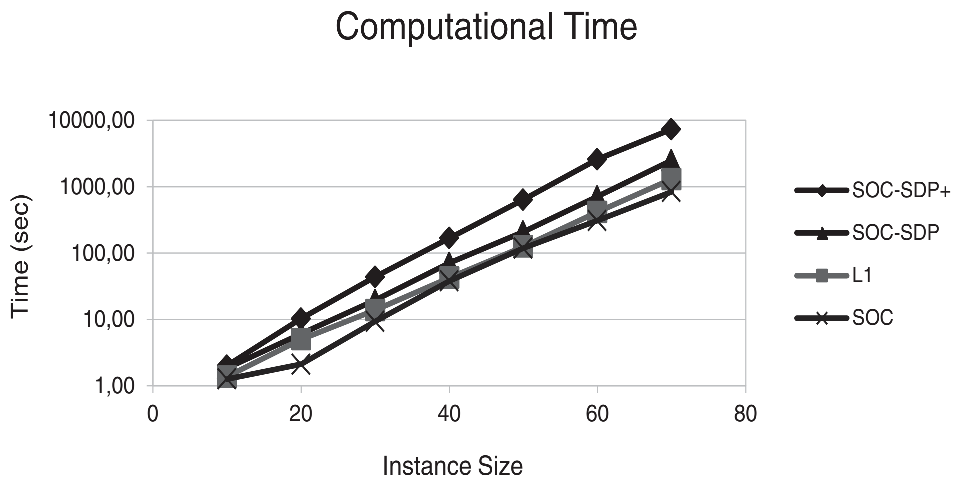


FIG. 5.1. *Computational time for BQPP (logarithmic scale).*

TABLE 5.2
Computational results for the QLCP instances.

n	m	$(\mathrm{QLCP}_{\mathrm{SS}+})$		$(\mathrm{QLCP}_{\mathrm{Burer'}})$			$(\mathrm{QLCP}_{\mathrm{SS}})$		(QLCP_{N+})		$(\mathrm{QLCP}_{\mathrm{SOC}})$	
		Gap	Time	Gap	Time1	Time2	Gap	Time	Gap	Time	Gap	Time
10	1	7.76	1.54	7.77	0.66	1.17	9.00	1.44	11.27	1.11	9.63	0.94
	5	11.28	2.82	11.30	0.72	2.18	16.60	1.77	21.59	1.63	18.73	1.15
20	1	3.72	6.90	3.75	1.24	5.95	4.94	5.19	7.89	4.40	7.84	1.80
	5	8.44	14.23	8.48	2.01	12.56	12.09	6.58	17.58	7.95	17.86	2.31
	10	10.62	20.10	10.64	2.22	17.53	15.70	7.11	21.86	11.63	23.19	2.45
30	1	1.74	20.52	1.80	2.17	17.64	2.32	13.66	3.40	12.00	5.94	8.50
	5	5.75	47.14	5.79	3.35	40.39	8.40	17.96	13.49	22.45	19.43	10.09
	15	10.09	76.75	10.14	4.76	64.28	15.41	20.52	22.66	34.63	29.21	11.84
40	1	1.26	72.38	1.31	3.76	63.50	1.84	41.09	2.91	34.29	8.86	29.63
	5	2.77	150.43	2.81	5.30	128.28	3.77	54.23	5.91	68.33	14.19	37.28
	20	9.94	297.52	10.01	10.21	245.19	16.03	66.26	26.89	120.24	33.11	42.54
50	1	1.07	222.98	1.11	4.96	200.60	1.31	134.42	2.07	104.82	4.79	32.99
	5	2.64	495.88	2.71	8.00	447.68	4.00	161.79	6.69	204.87	17.50	109.17
	25	9.57	1163.82	9.77	18.13	865.09	16.78	199.60	30.64	365.35	34.76	158.80
Avg.	-	6.19	-	6.24	-	-	9.16	-	13.92	-	17.50	-

upper bounds and computational time of Burer's specialized implementation, labeled as Time1 in Table 5.2, were all provided by Burer [6]. We also implemented Burer's relaxation using our code (as described in section 4.2), and we report the average computational time we obtained for it as Time2 in Table 5.2.

From Table 5.2, we see that Burer's relaxation is the most efficient in terms of computational time, but this is due to the fact that Burer's algorithm is specialized for solving problems of this form. However, in theory, it is an SDP-based relaxation, and thus the computational time has a higher order of complexity than the SOC-based relaxation, $(\mathrm{QLCP}_{\mathrm{SOC}})$. This can be seen when comparing Time2 with the computational time of the $(\mathrm{QLCP}_{\mathrm{SOC}})$ relaxation where the latter is on average 4 times more efficient for large n (see Figure 5.2). Among the four SDP-based relaxations, $(\mathrm{QLCP}_{\mathrm{SS}})$ is the most computationally efficient, as seen in Figure 5.2.

As shown in Theorem 4.3 and Table 4.2, $(\mathrm{QLCP}_{\mathrm{SS}+})$ provides the strongest bounds for the QLCP relaxation and has the same computational complexity as $(\mathrm{QLCP}_{\mathrm{Burer'}})$.

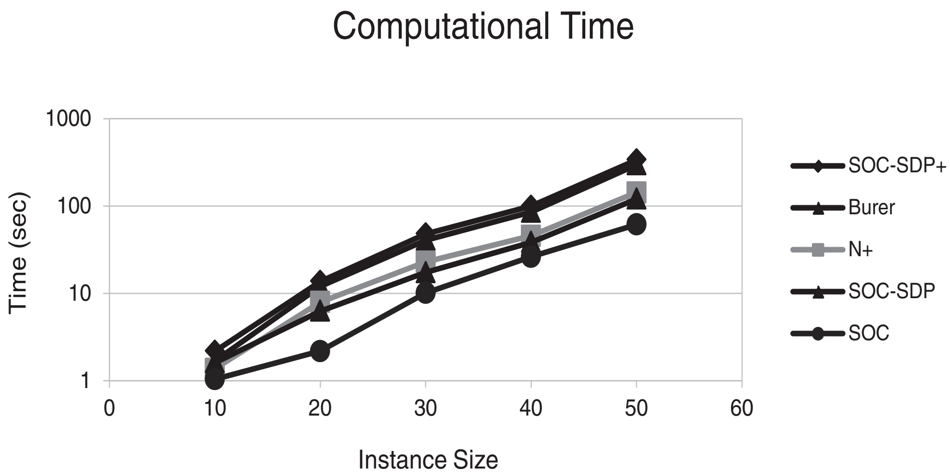


FIG. 5.2. *Computational time for QLCP (logarithmic scale).*

On the other hand, both (QLCP$_{N+}$) and (QLCP$_{\rm SS}$) are semidefinite-based relaxations but with less computational complexity than (QLCP$_{\rm SS+}$) and (QLCP$_{\rm Burer'}$). We notice that (QLCP$_{\rm SS}$) provides better bounds than (QLCP$_{N+}$) for all instances and is more computationally efficient. The average percentage gap for (QLCP$_{\rm SS}$) is 9.16% while that of (QLCP$_{N+}$) is 13.92%. In addition, (QLCP$_{\rm SOC}$) provides comparable bounds with (QLCP$_{N+}$) with an average percentage gap of 17.50%, but it is computationally the most efficient.

5.3. QKP computational results. In this section, we compare the performance of our proposed relaxations for the QKP with the relaxation of Helmberg, Rendl, and Weismantel [12] presented in section 4.3.1. We generated test instances using the approach proposed in [33]. The P_{ij} and w_j values are discrete, taken from a uniform random distribution in $[1, 100]$ and $[1, 50]$, respectively. The capacity $\bar{c}$ is uniformly distributed in $[50, \sum_{j=1}^{n} w_j]$. The density d of the P matrix varies from 10% to 90%.

The presented computational results are based on the following four types of relaxations for the QKP:
(QKP$_{\rm SS+}$): the relaxation presented in section 4.3;
(QKP$_{\rm SS}$): the SOC-SDP relaxation presented in section 4.3;
(QKP$_{\rm HRW4}$): the Helmberg et al. SDP relaxation presented in section 4.3.1;
(QKP$_{\rm SOC}$): the SOC relaxation presented in section 4.3.
Table 5.3 reports results for 45 instances. These instances vary in size and density. The size varies from 20 to 100 items, and the density varies from 10% to 90% with a step size of 20%. For each instance, we report the upper bound and the solution time in seconds.

In terms of computational time, (QKP$_{\rm SOC}$) is the most computationally efficient for all instances. For example, for the largest instances ($n = 100$), the (QKP$_{\rm SOC}$) relaxation is on average 23 times faster than the (QKP$_{\rm SS+}$), 19 times faster than the (QKP$_{\rm SS}$) relaxation, and 10 times faster than the (QKP$_{\rm HRW4}$) relaxation (see Figure 5.3).

Table 5.3
Computational results for the QKP instances. The gaps are with respect to (QKP_{SS^+}).

n	d	(QKP_{SS^+})		(QKP_{SS})			(QKP_{HRW4})			(QKP_{SOC})		
		UB	Time	UB	Gap	Time	UB	Gap	Time	UB	Gap	Time
20	10	809.00	8.66	811.22	0.27	6.19	814.84	0.72	4.10	811.74	0.34	4.54
20	30	2617.50	4.01	2619.34	0.07	5.68	2623.98	0.25	3.10	2619.48	0.08	1.97
20	50	1120.90	7.52	1137.25	1.46	6.09	1175.07	4.83	4.14	1262.98	12.68	1.42
20	70	2340.94	5.53	2356.25	0.65	5.51	2397.20	2.40	4.14	2540.40	8.52	1.69
20	90	6082.09	5.61	6083.70	0.03	5.72	6086.12	0.07	3.85	6083.80	0.03	1.75
30	10	1011.34	20.83	1022.20	1.07	18.24	1044.39	3.27	9.31	1129.01	11.63	6.91
30	30	3451.65	24.15	3470.97	0.56	16.37	3511.30	1.73	9.77	3939.00	14.12	6.01
30	50	8116.24	17.14	8125.16	0.11	19.48	8142.11	0.32	12.25	8127.76	0.14	9.83
30	70	8042.65	15.01	8047.03	0.05	18.20	8073.14	0.38	10.78	8108.38	0.82	6.94
30	90	5114.00	15.96	5127.57	0.27	15.78	5150.78	0.72	9.35	5136.34	0.44	8.81
40	10	3845.33	51.43	3853.49	0.21	55.43	3864.51	0.50	38.72	3875.12	0.77	32.86
40	30	11807.67	40.09	11809.44	0.02	59.27	11828.42	0.18	32.40	11811.54	0.03	34.71
40	50	4298.30	93.95	4309.76	0.27	69.20	4365.56	1.56	34.06	5161.31	20.08	26.29
40	70	17415.63	76.24	17424.10	0.05	60.41	17446.14	0.18	35.92	17447.01	0.18	31.18
40	90	25599.30	64.70	25612.48	0.05	59.17	25630.04	0.12	39.15	25615.00	0.06	36.12
50	10	2316.83	274.29	2353.89	1.60	158.24	2412.48	4.13	96.62	2846.05	22.84	44.32
50	30	11414.34	186.91	11433.16	0.16	188.84	11485.59	0.62	114.12	12050.94	5.58	64.22
50	50	23823.61	270.40	23846.12	0.09	181.09	23863.04	0.17	116.33	23850.99	0.11	27.62
50	70	32567.32	133.12	32571.10	0.01	213.29	32626.49	0.18	113.25	32575.12	0.02	26.29
50	90	17658.96	167.46	17671.03	0.07	168.55	17682.63	0.13	91.98	17672.78	0.08	23.05
60	10	7173.33	705.30	7188.68	0.21	673.37	7215.96	0.59	394.70	7410.08	3.30	138.25
60	30	26403.91	552.28	26496.51	0.35	644.20	26530.82	0.48	312.36	26502.66	0.37	79.46
60	50	13853.47	726.82	13871.42	0.13	682.12	13895.51	0.30	355.53	14396.64	3.92	78.18
60	70	56556.58	663.95	56561.20	0.01	797.11	56583.48	0.05	343.42	56561.20	0.01	58.54
60	90	62009.00	357.10	62009.00	0.00	478.40	62015.61	0.01	391.59	62009.00	0.00	38.21
70	10	3961.79	2969.84	4036.66	1.89	1689.45	4109.61	3.73	954.02	5104.22	28.84	230.87
70	30	20191.73	2698.05	20208.57	0.08	2262.87	20275.13	0.41	1237.78	21826.79	8.10	296.70
70	50	45493.48	2760.52	45507.07	0.03	2407.57	45573.21	0.18	1224.95	45752.77	0.57	154.61
70	70	1621.19	2900.58	1631.57	0.64	2308.23	1882.75	16.13	1081.38	1737.92	7.20	143.12
70	90	32850.56	1777.27	32857.31	0.02	2574.93	32913.98	0.19	1157.09	32876.13	0.08	102.06
80	10	13062.74	4407.65	13074.13	0.09	5008.22	13118.78	0.43	2584.26	13506.53	3.40	564.75
80	30	1480.00	3327.65	1480.00	0.00	4388.67	1537.29	3.87	2143.41	1532.02	3.51	264.94
80	50	23126.43	6694.43	23141.40	0.06	4650.24	23220.33	0.41	2494.70	25240.44	9.14	365.01
80	70	58613.63	5422.86	58621.35	0.01	5419.69	58649.30	0.06	2979.25	59322.02	1.21	270.39
80	90	112167.40	4178.58	112184.20	0.01	5052.10	112202.99	0.03	2958.44	112184.53	0.02	185.92
90	10	6189.28	15610.86	6311.21	1.97	7057.15	6447.72	4.18	4818.87	8500.89	37.35	517.90
90	30	30656.56	16455.66	30710.62	0.18	9398.88	30829.68	0.56	5587.52	36535.46	19.18	740.54
90	50	81336.10	10319.41	81344.17	0.01	12623.02	81393.43	0.07	6233.23	81385.48	0.06	426.81
90	70	8004.38	12082.24	8014.95	0.13	11942.53	8312.97	3.86	4292.94	8297.26	3.66	458.36
90	90	55262.87	11603.07	55285.71	0.04	8883.04	55305.54	0.08	5640.32	55291.14	0.05	295.34
100	10	23941.78	23975.63	23951.45	0.04	18831.11	23977.05	0.15	9883.59	24021.78	0.33	1867.44
100	30	40216.48	31499.01	40257.87	0.10	17673.77	40370.14	0.38	9832.83	45597.97	13.38	973.49
100	50	11707.00	27958.75	11737.62	0.26	18867.58	11879.03	1.47	8553.04	13937.02	19.05	1308.24
100	70	122205.33	20428.73	122215.14	0.01	24684.33	122305.61	0.08	9482.50	122476.61	0.22	431.02
100	90	63378.00	12182.11	63378.00	0.00	14881.31	63411.61	0.05	10280.16	63378.00	0.00	484.25
Avg.		-	-	-	0.30	-	-	1.34	-	-	5.81	-

Further, for all the tested instances, the (QKP_{SS^+}) and (QKP_{SS}) bounds are strictly tighter than the ones provided by (QKP_{HRW4}), even though the bounds for the (QKP_{HRW4}) relaxation are known to be strong [12, 33]. In addition, we report the gap between the bounds of (QKP_{SS^+}), (QKP_{HRW4}), and (QKP_{SOC}) and the bound of (QKP_{SS^+}). Over all instances, the percentage gap of the (QKP_{SOC}) relaxation with respect to the (QKP_{HRW4}) relaxation ranges from -8% to around 31% with an

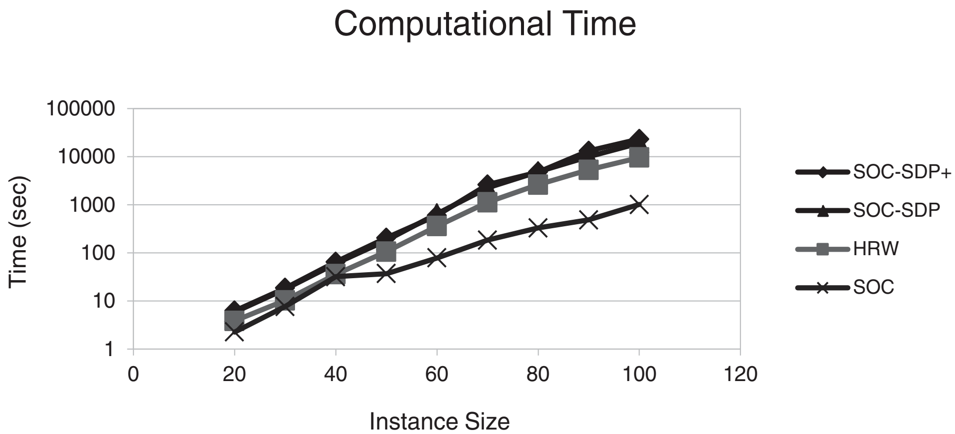


FIG. 5.3. *Computational time for QKP (logarithmic scale).*

average of 4.39%, where a negative sign implies that the (QKP_{SOC}) relaxation is better. Notice that (QKP_{SOC}) performs particularly well for instances with high density. In particular, (QKP_{SOC}) obtains better bounds than (QKP_{HRW4}) for all the instances with $d = 90\%$.

6. Conclusion and future work. In this research we used polynomial programming approaches to produce tractable relaxations for general binary quadratic polynomial optimization problems. These approximations utilize linear, second-order, and semidefinite cones over which it is known how to optimize efficiently. We proposed an SOC relaxation for the general BQPP and applied it to several BQPPs. When compared to SDP-based relaxations, these SOC-based relaxations are significantly more computationally efficient with only a small degradation of bounds.

For the general BQPP, we proposed two SOC-SDP-based relaxations and compared them theoretically and experimentally with Lasserre's relaxation of order 1. By exploiting the linear constraints using SOCs we were able to obtain a stronger relaxation than Lasserre's. We also conducted computational results on binary QLCPs and showed that the quality of the bounds provided by our SOC-SDP-based relaxation is competitive with those from the very recent specialized relaxation of Burer for this problem [6]. Finally, for the QKP, we showed that the two proposed SOC-SDP-based relaxations are a strict improvement on the best relaxation in the literature. Theoretical results as well as computational experiments show that our SOC-SDP-based relaxation outperforms the relaxation of Helmberg, Rendl, and Weismantel [12] in terms of bound while both relaxations are comparable in terms of computational time. We also relaxed our proposed relaxation to obtain a weaker SOC-only relaxation that is computationally more efficient while still providing comparable bounds to [12], and for problems with high density it provides better bounds.

The main objective of our research is to develop an exact algorithm for solving general BQPPs. Our SOC relaxations show strong potential, both in terms of bounds and of computational time, to be used in an exact algorithm scheme to find optimal solutions for large instances of such problems in a reasonable time. Future research will investigate the use of SOC-based relaxations with additional valid inequalities. In particular, we are developing nonlinear cuts based on polynomial programming to further strengthen the proposed relaxations.

Acknowledgments. The authors thank Sam Burer for providing us with the QLCP instances and their optimal values. The authors also thank the anonymous referees for their comments and suggestions that helped improve the presentation of the paper.

REFERENCES

[1] B. Alidaee, G. Kochenberger, and A. Ahmadian, 0-1 *quadratic programming approach for the optimal solution of two scheduling problems*, Internat. J. Systems Sci., 25 (1994), pp. 401–408.

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

[3] F. Barahona, M. Jünger, and G. Reinelt, *Experiments in quadratic* 0-1 *programming*, Math. Program., 44 (1989), pp. 127–137.

[4] E. Boros and P. Hammer, *Pseudo-boolean optimization*, Discrete Appl. Math., 123 (2002), pp. 155–225.

[5] E. Boros and A. Prekopa, *Probabilistic bounds and algorithms for the maximum satisfiability problem*, Ann. Oper. Res., 21 (1989), pp. 109–126.

[6] S. Burer, *private communication*, 2009.

[7] S. Burer, R. Monteiro, and Y. Zhang, *Rank-two relaxation heuristics for max-cut and other binary quadratic programs*, SIAM J. Optim., 12 (2002), pp. 503–521.

[8] E. de Klerk and D. Pasechnik, *Approximation of the stability number of a graph via copositive programming*, SIAM J. Optim., 12 (2002), pp. 875–892.

[9] K. Fujisawa, M. Kojima, and K. Nakata, *Exploiting sparsity in primal-dual interior-point methods for semidefinite programming*, Math. Program., 79 (1997), pp. 235–253.

[10] G. Gallo, P. Hammer, and B. Simeone, *Quadratic knapsack problems*, European J. Oper. Res., 12 (1980), pp. 132–149.

[11] M. Grötschel, M. Jünger, and G. Reinelt, *An application of combinatorial optimization to statistical physics and circuit layout design*, Oper. Res., 36 (1988), pp. 493–513.

[12] C. Helmberg, F. Rendl, and R. Weismantel, *A semidefinite programming approach to the quadratic knapsack problem*, J. Comb. Optim., 4 (2000), pp. 197–215.

[13] E. L. Johnson, A. Mehrotra, and G. L. Nemhauser, *Min-cut clustering*, Math. Program., 62 (1993), pp. 133–151.

[14] S. Kim and M. Kojima, *Second-order cone programming relaxation of nonconvex quadratic optimization problems*, Optim. Methods Softw., 15 (2001), pp. 201–224.

[15] S. Kim and M. Kojima, *Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations*, Comput. Optim. Appl., 26 (2003), pp. 143–154.

[16] S. Kim, M. Kojima, and M. Yamashita, *Second order cone programming relaxation of a positive semidefinite constraint*, Optim. Methods Softw., 18 (2003), pp. 535–541.

[17] M. Kojima, S. Kim, and H. Waki, *A general framework for convex relaxation of polynomial optimization problems over cones*, J. Oper. Res. Soc. Japan, 46 (2003), pp. 125–144.

[18] J. Lasserre, *An explicit equivalent positive semidefinite program for nonlinear* 0-1 *programs*, SIAM J. Optim., 12 (2001), pp. 756–769.

[19] J. Lasserre, *Global optimization problems with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[20] D. Laughhunn, *Quadratic binary programming with application to capital-budgeting problems*, Oper. Res., 18 (1970), pp. 454–461.

[21] M. Laurent, *Semidefinite representations for finite varieties*, Math. Program., 109 (2007), pp. 1–26.

[22] E. Lawler, *The quadratic assignment problem*, Management Sci., 9 (1963), pp. 586–599.

[23] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

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

[25] Y. Nesterov, *Semidefinite relaxation and non-convex quadratic optimization*, Optim. Methods Softw., 12 (1997), pp. 1–20.

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

[27] P. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, 2000.
[28] P. Parrilo, *An Explicit Construction of Distinguished Representations of Polynomials Nonnegative over Finite Sets*, Technical report, IfA Technical Report AUT02-02, Zurich, Switzerland, 2002.
[29] P. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Program., 96 (2003), pp. 293–320.
[30] P. Parrilo and B. Sturmfels, *Minimizing polynomial functions, algorithmic and quantitative real algebraic geometry*, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 60 (2003), pp. 83–89.
[31] J. F. Peña, J. C. Vera, and L. F. Zuluaga, *Exploiting equalities in polynomial programming*, Oper. Res. Lett., 36, 2008.
[32] A. Phillips and J. Rosen, *A quadratic assignment formulation of the molecular conformation problem*, J. Global Optim., 4 (1994), pp. 229–241.
[33] D. Pisinger, *The quadratic knapsack problem—a survey*, Discrete Appl. Math., 155 (2007), pp. 623–648.
[34] I. Pólik and T. Terlaky, *A survey of the $\mathcal{S}$-lemma*, SIAM Rev., 49 (2007), pp. 371–418.
[35] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.
[36] N. Shor, *A class of global minimum bounds of polynomial functions*, Cybernetics, 23 (1987), pp. 731–734.
[37] J. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 1999, pp. 11-12.
[38] B. Sturmfels, J. Demmel, and J. Nie, *Minimizing polynomials via sum of squares over the gradient ideal*, Math. Program., 106 (2006), pp. 587–606.
[39] J. C. Vera, L. Zuluaga, and J. Peña, *Positive polynomials on equality constrained domains*, working paper.
[40] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of Semidefinite Programming—Theory, Algorithms, and Applications*, Springer, New York, 2000.
[41] L. Zuluaga, J. C. Vera, and J. Peña, *LMI approximations for cones of positive semidefinite forms*, SIAM J. Optim., 16, 2006, pp. 1076–1091.