Research Reports on Mathematical and Computing Sciences

A Robust Lagrangian-DNN Method for
a Class of Quadratic Optimization Problems

Naohiko Arima, Sunyoung Kim,
Masakazu Kojima, and Kim-Chuan Toh

February 2016, B–482

B-482   A robust Lagrangian-DNN method
         for a class of quadratic optimization problems

Naohiko Arima[⋆], Sunyoung Kim[†], Masakazu Kojima[‡], and Kim-Chuan Toh[♯]

February, 2016

Abstract.   The Lagrangian-doubly nonnegative (DNN) relaxation has recently been shown to provide effective lower bounds for a large class of nonconvex quadratic optimization problems (QOPs) using the bisection method combined with first-order methods by Kim, Kojima and Toh in 2016. While the bisection method has demonstrated the computational efficiency, determining the validity of a computed lower bound for the QOP depends on a prescribed parameter $\epsilon > 0$. To improve the performance of the bisection method for the Lagrangian-DNN relaxation, we propose a new technique that guarantees the validity of the computed lower bound at each iteration of the bisection method for any choice of $\epsilon > 0$. It also accelerates the bisection method. Moreover, we present a method to retrieve a primal-dual pair of optimal solutions of the Lagrangian-DNN relaxation using the primal-dual interior-point method. As a result, the method provides a better lower bound and substantially increases the robustness as well as the effectiveness of the bisection method. Computational results on the binary QOPs, the multiple knapsack problems, the maximal stable set problems, and the quadratic assignment problems (QAPs) illustrate the robustness of the proposed method. In particular, a tight bound for QAPs with size $n = 50$ could be obtained.

Key words.   Nonconvex quadratic optimization problems with nonnegative variables, the Lagrangian-DNN relaxation, improved bisection method, the validity of lower bounds.

AMS Classification.   90C20, 90C25 90C26

[⋆]   Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan (`nao_arima@me.com`).

[†]   Department of Mathematics, Ewha W. University, 11-1 Dahyun-dong, Sudaemoon-gu, Seoul 120-750 Korea.   The research was supported by NRF 2014-R1A2A1A11049618 (`skim@ewha.ac.kr`)

[‡]   Department of Industrial and Systems Engineering, Chuo University, Tokyo 112-8551 Japan. This research was supported by Grant-in-Aid for Scientific Research (A) 26242027 and the Japan Science and Technology Agency (JST), the Core Research of Evolutionary Science and Technology (CREST) research project (`masakazukojima@mac.com`)

[♯]   Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076. Research supported in part by Ministry of Education Academic Research Fund R-146-000-194-112 (`mattohkc@nus.edu.sg`).

1 Introduction

We consider NP-hard, nonconvex quadratic optimization problems (QOPs) that can be formulated as

$$\varphi = \inf \left\{ \langle \boldsymbol{Q}^0, \, \boldsymbol{x}\boldsymbol{x}^T \rangle \mid \boldsymbol{x} \in \mathbb{R}^n_+, \, \langle \boldsymbol{H}^0, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = 1, \, \langle \boldsymbol{H}^1, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = 0 \right\}, \tag{1}$$

where $\boldsymbol{Q}^0$, $\boldsymbol{H}^0$, and $\boldsymbol{H}^1$ are $n \times n$ symmetric matrices $\mathbb{S}^n$, and $\langle \boldsymbol{A}, \, \boldsymbol{X} \rangle$ denotes the inner product of $\boldsymbol{A} \in \mathbb{S}^n$ and $\boldsymbol{X} \in \mathbb{S}^n$ defined by $\sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} X_{ij}$. Thus, $\langle \boldsymbol{A}, \, \boldsymbol{x}\boldsymbol{x}^T \rangle$ denotes a quadratic form $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ in $\boldsymbol{x} \in \mathbb{R}^n$. The formulation of nonconvex QOPs considered in this paper in the form of (1) was introduced in [1, 14].

A QOP expressed in the form of (1) consists of a quadratic form of objective function $\langle \boldsymbol{Q}^0, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = \boldsymbol{x}^T \boldsymbol{Q}^0 \boldsymbol{x}$ to be minimized and three constraints, a nonnegativity constraint $\boldsymbol{x} \geq \boldsymbol{0}$, an inhomogeneous quadratic equality constraint $\langle \boldsymbol{H}^0, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = \boldsymbol{x}^T \boldsymbol{H}^0 \boldsymbol{x} = 1$, and a homogeneous quadratic equality constraint $\langle \boldsymbol{H}^1, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = \boldsymbol{x}^T \boldsymbol{H}^1 \boldsymbol{x} = 0$. Both coefficient matrices $\boldsymbol{H}^0$ and $\boldsymbol{H}^1$ are assumed to be the sum of a positive semidefinite matrix and a nonnegative matrix. Although QOP (1) has a very small number of simple constraints and the assumption seems restrictive, many QOPs can be converted to a QOP in the class. In particular, QOP (1) includes QOPs with linear equality, binary and complementarity constraints in nonnegative vector variable and various combinatorial optimization problems. See Section 5 of [1] and Section 5 of [14]. Arima, Kim and Kojima [1, 2] reformulated the class of QOPs as an equivalent completely positive programming problem (CPP), a linear optimization problem over the completely positive cone. This result is an extension of Burer's CPP reformulation [7] of a class of QOPs with linear constraints in both binary and nonnegative continuous variables. These CPP reformulation of QOPs are mainly for theoretical interests since CPP problems are numerically intractable [17].

Doubly nonnegative (DNN) relaxations of QOPs and DNN relaxations of CPP reformulation of QOPs have been studied as numerically tractable alternatives [8, 22]. A computational approach to solve the DNN relaxation problem is to convert it to a semidefinite program (SDP) and apply the primal-dual interior point method [6, 11, 19, 20] to the resulting SDP [23, 12]. In this approach, the single nonnegative constraint imposed on the DNN matrix variable becomes nonnegative inequality constraints on the elements of the matrix variable in the SDP. Thus, the computational cost of this approach is very expensive as the number of nonnegative constraints in the SDP grows quadratically with the size of the DNN matrix variable.

Kim, Kojima and Toh [14] proposed a Lagrangian-DNN relaxation method for computing lower bounds of the minimum values of a class of QOPs, essentially the equivalent class of QOPs considered in this paper. The simple structure of the problem and the aforementioned assumption are keys to their approach. By applying the Lagrangian relaxation to the QOP, they derived a simpler QOP problem with a Lagrangian multiplier λ associated with the equality constraint $\langle \boldsymbol{H}^1, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = 0$:

$$\varphi(\lambda) = \inf \left\{ \langle \boldsymbol{Q}^0 + \lambda \boldsymbol{H}^1, \, \boldsymbol{x}\boldsymbol{x}^T \rangle \mid \boldsymbol{x} \in \mathbb{R}^n_+, \, \langle \boldsymbol{H}^0, \, \boldsymbol{x}\boldsymbol{x}^T \rangle = 1 \right\}. \tag{2}$$

Since $\langle \boldsymbol{H}^1, \, \boldsymbol{x}\boldsymbol{x}^T \rangle \geq 0$ for every $\boldsymbol{x} \geq \boldsymbol{0}$ by the assumption on $\boldsymbol{H}^1$, $\lambda \langle \boldsymbol{H}^1, \, \boldsymbol{x}\boldsymbol{x}^T \rangle$ added to the objective function serves as a penalty term, and the optimal value $\varphi(\lambda)$ of the

Lagrangian relaxation (2) of QOP (1) converges to the optimal value φ of the original QOP as λ tends to ∞ under an additional assumption requiring the boundedness of the feasible region of the QOP. Then, the DNN relaxation was applied to the Lagrangian relaxation (2) of QOP (1), which was called the Lagrangian-DNN relaxation. To solve the resulting problem with a sufficiently large $\lambda > 0$ fixed, they proposed a numerical method using a bisection method combined with the proximal alternating direction method of multiplier [10] and the accelerated proximal gradient method [5]. Their bisection method was shown to be computationally very efficient through numerical results on binary QOPs, quadratic multi-knapsack problems, maximum stable set problems, and quadratic assignment problems.

For the bisection method in [14], we cannot avoid dealing with the computational issue of guaranteeing the validity of the computed lower bounds. More precisely, the validity of the lower bound generated for the optimal value $\eta^d(\lambda)$ of the Lagrangian-DNN relaxation problem is dependent on a parameter, say $\epsilon > 0$. Here $\eta^d(\lambda)$ provides a lower bound for the optimal value $\varphi(\lambda)$ of QOP (2) and the optimal value φ of QOP (1). In the bisection method, a continuous function $f_\lambda : \mathbb{R} \to [0, \infty)$ is introduced such that $f_\lambda(y_0) = 0$ if $y_0 \leq \eta^d(\lambda)$ and $f_\lambda(y_0) > 0$ otherwise, where $f_\lambda(y_0)$ is induced from a norm of a symmetric matrix function of y_0. While it is straightforward in theory to judge whether a given y_0 is a lower or upper bound of $\eta^d(\lambda)$ according to whether $f_\lambda(y_0) = 0$ or $f_\lambda(y_0) > 0$, respectively, the evaluation of $f_\lambda(y_0)$ in computation is not exact and its numerical value remains positive even when $y_0 \leq \eta^d(\lambda)$ due to numerical errors. For the bisection method, we choose a parameter $\epsilon > 0$ to decide whether the pth iterate $y_0^p \in \mathbb{R}$ provides either a lower bound ℓ^{p+1} or an upper bound u^{p+1} for $\eta^d(\lambda)$ according to $f_\lambda(y_0) \leq \epsilon$ or $f_\lambda(y_0) > \epsilon$, respectively, and either $u^{p+1} = u^p$ or $\ell^{p+1} = \ell^p$ is assigned, respectively. Then the next iterate y_0^{p+1} is computed as $(\ell^{p+1} + u^{p+1})/2$. Thus the length of the interval $[\ell^p, u^p]$ is reduced by half at each iteration. If $\epsilon > 0$ (or nearly 0) is chosen to be too small, then $y_0^p < \eta^d(\lambda)$ is incorrectly assigned to an upper bound u^{p+1} when the numerical value of $f_\lambda(y_0^p)$ is greater than ϵ. As a result, the final lower bound will be much smaller than $\eta^d(\lambda)$. On the other hand, as a larger $\epsilon > 0$ is used, the final lower bound is expected to increase, even to the value larger than $\eta^d(\lambda)$, resulting in an invalid lower bound for $\eta^d(\lambda)$. Therefore, an appropriate choice of $\epsilon > 0$ is crucial to the successful implementation of the bisection method for a valid and tight lower bound for $\eta^d(\lambda)$.

The purpose of this paper is to increase the robustness and effectiveness of the Lagrangian-DNN relaxation method. First, we address the issue of numerical difficulty involving the parameter ϵ to improve the performance of the bisection method. Second, we present a method to approximate a primal-dual pair of optimal solutions of the Lagrangian-DNN relaxation problem so as to compute a further improved lower bound of the optimal value φ of QOP (1).

With the proposed technique to resolve the parameter dependency for determining the validity of lower bound, each iterate always generates a valid lower bound $\underline{\ell}^p$ of $\eta^d(\lambda)$ for any choice of $\epsilon > 0$ in theory. Thus, the modified bisection method implementing the proposed technique is less dependent on the choice of $\epsilon > 0$. We also show that the new technique frequently reduces the length of interval $[\ell^p, u^p]$ by the ratio of less than 0.5 at each iteration, making the modified bisection method more efficient.

The construction of an approximate optimal solution of the original QOP, which was not dealt with in [14], is an important issue. If the bisection method is applied to the dual of the Lagrangian-DNN relaxation problem as in [14], it provides an approximate optimal solution $(y_0^*, \boldsymbol{W}^*, \boldsymbol{Z}^*)$ of the dual of the Lagrangian-DNN relaxation problem, where $\boldsymbol{W}^*$ denotes a symmetric nonnegative matrix and $\boldsymbol{Z}^*$ a positive semidefinite matrix. In this case, however, any approximate optimal solution of the primal Lagrangian-DNN relaxation problem, which is necessary to retrieve an approximate optimal solution of the original QOP, is not available. To overcome this drawback of the bisection method, we propose to apply the primal-dual interior-point method [6, 11, 19, 20] to an SDP associated with the Lagrangian-DNN relaxation problem with the dual nonnegative matrix variable fixed to $\boldsymbol{W}^*$. As a result, we not only have an approximate optimal solution of the primal Lagrangian-DNN relaxation problem but also a better lower bound for the optimal value of the QOP.

The paper is organized as follows: Section 2 presents the class of QOPs and Lagrangian-DNN relaxations proposed in [14], and the bisection method for the Lagrangian-DNN relaxation. Section 3 is the main part of the paper. We first present the new technique for computing valid lower bounds of QOP (1) and incorporate it into an accelerated bisection method. We then introduce techniques to improve the quality of lower bounds by the primal-dual interior-point methods. Section 4 addresses the technical issues on the choice of the parameters λ and ϵ. In Section 5, we first show how we transform a class of QOPs with linear equalities, 0-1 and complementarity constraints into QOP (1), and then consider binary QOPs, quadratic multiple knapsack problems, maximum stable set problems and QAPs as special cases. We also present numerical results on those problems. Finally, we conclude in Section 6.

2 Preliminaries

2.1 Notation and symbols

We use the following notation and symbols throughout the paper:

$$
\begin{aligned}
\mathbb{R}^n &= \text{the space of } n\text{-dimensional column vectors,} \\
\mathbb{R}_+^n &= \text{the nonnegative orthant of } \mathbb{R}^n, \\
\mathbb{S}^n &= \text{the space of } n \times n \text{ symmetric matrices,} \\
\mathbb{S}_+^n &= \text{the cone of } n \times n \text{ symmetric positive semidefinite matrices,} \\
\mathbb{N}^n &= \text{the cone of } n \times n \text{ symmetric nonnegative matrices,} \\
\langle \boldsymbol{A}, \boldsymbol{X} \rangle &= \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} X_{ij} \text{ (the inner product of } \boldsymbol{A} \in \mathbb{S}^n \text{ and } \boldsymbol{X} \in \mathbb{S}^n).
\end{aligned}
$$

For every $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x}^T$ denotes the transposition of $\boldsymbol{x}$; $\boldsymbol{x}^T$ is an n-dimensional row vector. We use the notation $(\boldsymbol{v}, \boldsymbol{w}) \in \mathbb{R}^{k+\ell}$ for the $(k + \ell)$-dimensional column vector consisting of $\boldsymbol{v} \in \mathbb{R}^k$ and $\boldsymbol{w} \in \mathbb{R}^\ell$. The quadratic form $\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$ associated with a matrix $\boldsymbol{Q} \in \mathbb{S}^n$ is represented as $\langle \boldsymbol{Q}, \boldsymbol{x}\boldsymbol{x}^T \rangle$ for every $\boldsymbol{x} \in \mathbb{R}^n$. In the subsequent discussions, $\langle \boldsymbol{Q}, \boldsymbol{x}\boldsymbol{x}^T \rangle$ is used to suggest that $\langle \boldsymbol{Q}, \boldsymbol{x}\boldsymbol{x}^T \rangle$ with $\boldsymbol{x} \in \mathbb{R}_+^n$ is relaxed to $\langle \boldsymbol{Q}, \boldsymbol{X} \rangle$ with $\boldsymbol{X} \in \mathbb{N}^n \cap \mathbb{S}_+^n$.

2.2 A class of simple quadratic optimization problems and their Lagrangian-DNN relaxations

Recently, Arima, Kim, Kojima, and Toh [3] proposed a unified framework for Lagrangian-conic relaxations of polynomial optimization problems over a closed (not necessarily convex) cones $\mathbb{K}$ in a finite dimensional space $\mathbb{V}$. This framework is based on the CPP and DNN relaxations of a class of QOPs studied in the authors' previous work [1, 2, 14]. In this section, we derive a Lagrangian-DNN relaxation of a class of QOPs and discuss its fundamental properties by specifying $\mathbb{V} = \mathbb{S}^n$, $\boldsymbol{\Gamma} = \{\boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x} \in \mathbb{R}^n_+\}$ and $\mathbb{K} = \mathbb{N}^n \cap \mathbb{S}^n_+$ in the unified framework. We note that $\boldsymbol{\Gamma}$ is a closed nonconvex cone contained in $\mathbb{N}^n \cap \mathbb{S}^n_+$.

For every $S \subset \mathbb{S}^n$, let

$$F(S) \;=\; \left\{ \boldsymbol{X} \in S \mid \langle \boldsymbol{H}^0, \boldsymbol{x}\boldsymbol{x}^T \rangle = 1, \; \langle \boldsymbol{H}^1, \boldsymbol{x}\boldsymbol{x}^T \rangle = 0 \right\}.$$

Then we can rewrite QOP (1) as

$$\varphi = \inf \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in F(\boldsymbol{\Gamma}) \right\}. \tag{3}$$

We may regard QOP (3) as a conic optimization problem over the closed nonconvex cone $\boldsymbol{\Gamma}$ in the unified framework [3]. By replacing $\boldsymbol{\Gamma}$ with its convex hull co$\boldsymbol{\Gamma}$, we obtain the CPP relaxation of QOP (3):

$$\underline{\varphi} = \inf \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in F(\text{co}\boldsymbol{\Gamma}) \right\}. \tag{4}$$

Obviously, $\underline{\varphi} \leq \varphi$ in general. Arima, Kim, Kojima, and Toh [3] provided the characterization for $\underline{\varphi}$ to attain φ (Theorem 3.1 of [3]). See also [1].

Since (4) is not numerically tractable, co$\boldsymbol{\Gamma}$ is further replaced by $\mathbb{N}^n \cap \mathbb{S}^n_+$ for a DNN relaxation of QOP (3) and its dual:

$$\begin{aligned}
\zeta^p &= \inf \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in F(\mathbb{N}^n \cap \mathbb{S}^n_+) \right\} \\
&= \inf \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in \mathbb{N}^n \cap \mathbb{S}^n_+, \; \langle \boldsymbol{H}^0, \boldsymbol{X} \rangle = 1, \; \langle \boldsymbol{H}^1, \boldsymbol{X} \rangle = 0 \right\},
\end{aligned} \tag{5}$$

$$\zeta^d = \sup \left\{ y_0 \mid \boldsymbol{W} \in \mathbb{N}^n, \; \boldsymbol{Q}^0 - \boldsymbol{H}^0 y_0 + \boldsymbol{H}^1 y_1 - \boldsymbol{W} = \boldsymbol{Y} \in \mathbb{S}^n_+ \right\}. \tag{6}$$

Applying a Lagrangian relaxation to DNN problem (5), we obtain the Lagrangian-DNN relaxation of QOP (3) and its dual:

$$\eta^p(\lambda) = \inf \left\{ \langle \boldsymbol{Q}^0 + \lambda \boldsymbol{H}^1, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in \mathbb{N}^n \cap \mathbb{S}^n_+, \; \langle \boldsymbol{H}^0, \boldsymbol{X} \rangle = 1 \right\}, \tag{7}$$

$$\eta^d(\lambda) = \sup \left\{ y_0 \mid \boldsymbol{W} \in \mathbb{N}^n, \; \boldsymbol{Q}^0 + \lambda \boldsymbol{H}^1 - \boldsymbol{H}^0 y_0 - \boldsymbol{W} = \boldsymbol{Y} \in \mathbb{S}^n_+ \right\}. \tag{8}$$

Here $\lambda \in \mathbb{R}$ denotes a Lagrangian multiplier associated with the equality constraint $\langle \boldsymbol{H}^1, \boldsymbol{X} \rangle = 0$. We could first apply the Lagrangian relaxation to QOP (1) or (3), and then the DNN relaxation, as mentioned in the Introduction. But, the resulting Lagrangian-DNN relaxation is exactly the same as (7).

Throughout the paper, we assume that

Condition (A) The feasible region $F(\boldsymbol{\Gamma})$ of QOP (3) is nonempty.

Condition (B) $\boldsymbol{O} \neq \boldsymbol{H}^0 \in \mathbb{N}^n + \mathbb{S}^n_+$ and $\boldsymbol{O} \neq \boldsymbol{H}^1 \in \mathbb{N}^n + \mathbb{S}^n_+$.

Condition (C) The feasible region $F(\mathbb{N}^n \cap \mathbb{S}^n_+)$ of DNN problem (5) is bounded.

We note that Condition (C) is equivalent to

Condition (C') $\boldsymbol{D} = \boldsymbol{O}$ if $\boldsymbol{D} \in \mathbb{N}^n \cap \mathbb{S}^n_+$, $\langle \boldsymbol{H}^0, \boldsymbol{D} \rangle = 0$ and $\langle \boldsymbol{H}^1, \boldsymbol{D} \rangle = 0$.

Conditions (A), (B) and (C') correspond to Conditions (a), (b) and (c) in [14], respectively. Obviously, the feasible region of (3) is a subset of the feasible region of (5). Thus, both feasible regions are nonempty and bounded by Conditions (A) and (C). By Condition (B), we have $\langle \boldsymbol{H}^1, \boldsymbol{X} \rangle \geq 0$ for every $\boldsymbol{X} \in \mathbb{N}^n \cap \mathbb{S}^n_+$. Hence, if we take $\lambda > 0$, $\lambda \langle \boldsymbol{H}^1, \boldsymbol{X} \rangle$ added to the objective function of (5) serves as a penalty term such that $\lambda \langle \boldsymbol{H}^1, \boldsymbol{X} \rangle \to \infty$ as $\lambda \to \infty$ for any feasible solution $\boldsymbol{X}$ of (7) not satisfying $\langle \boldsymbol{H}^1, \boldsymbol{X} \rangle = 0$. Furthermore, we have the following relationships.

Theorem 2.1.

(i) $\eta^d(\lambda_1) = \eta^p(\lambda_1) \leq \eta^p(\lambda_2) \leq \zeta^d = \zeta^p \leq \varphi$ if $0 \leq \lambda_1 \leq \lambda_2$.

(ii) $\eta^d(\lambda) \to \zeta^d$ as $\lambda \to \infty$.

Proof. See Theorem 2.1 of [3].

2.3 A bisection method

The class of QOPs in the form of (3) and Conditions (A), (B) and (C) are essentially equivalent to the class of QOPs studied in [14] and the conditions assumed there, respectively. In this section, we briefly explain the bisection method in [14] for solving (8).

Let $\lambda > 0$ be fixed. For every y_0, define

$$\begin{aligned}
\boldsymbol{G}_\lambda(y_0) &= \boldsymbol{Q}^0 + \lambda \boldsymbol{H}^1 - \boldsymbol{H}^0 y_0, \\
g_\lambda(y_0) &= \min \left\{ \|\boldsymbol{G}_\lambda(y_0) - \boldsymbol{Z}\| : \boldsymbol{Z} \in \mathbb{N}^n + \mathbb{S}^n_+ \right\}.
\end{aligned}$$

Then, y_0 is a feasible solution of (8) if and only if $g_\lambda(y_0) = 0$. Therefore, (8) is equivalent to the problem of maximizing y_0 subject to $g_\lambda(y_0) = 0$; $\eta^d(\lambda) = \sup \left\{ y_0 : g_\lambda(y_0) = 0 \right\}$. By definition, $g_\lambda(y_0) \geq 0$ for every $y_0 \in \mathbb{R}$. Recall that $\boldsymbol{H}^0 \in \mathbb{N}^n + \mathbb{S}^n_+$. Hence, if $g_\lambda(\bar{y}_0) = 0$ or equivalently $\boldsymbol{G}_\lambda(\bar{y}_0) \in \mathbb{N}^n + \mathbb{S}^n_+$, then

$$\boldsymbol{G}_\lambda(y_0) = \boldsymbol{G}_\lambda(\bar{y}_0) + (\bar{y}_0 - y_0)\boldsymbol{H}^0 \in \mathbb{N}^n + \mathbb{S}^n_+ \text{ or equivalently } g_\lambda(y_0) = 0$$

for every $y_0 \leq \bar{y}_0$. Thus, $g_\lambda(y_0) = 0$ for every $y_0 \in (-\infty, \eta^d(\lambda)]$. Furthermore, by Lemma 4.1 of [3], g_λ is continuous, convex and monotonically increasing on $[\eta^d(\lambda), +\infty)$.

For every $\boldsymbol{G} \in \mathbb{S}^n$, let $\Pi(\boldsymbol{G})$ and $\Pi^*(\boldsymbol{G})$ denote the metric projection of $\boldsymbol{G}$ onto the cone $\mathbb{N}^n \cap \mathbb{S}^n_+$ and its dual cone $(\mathbb{N}^n \cap \mathbb{S}^n_+)^* = \mathbb{N}^n + \mathbb{S}^n_+$, respectively.

$$\begin{aligned}
\Pi(\boldsymbol{G}) &= \operatorname{argmin} \left\{ \|\boldsymbol{G} - \boldsymbol{X}\| : \boldsymbol{X} \in \mathbb{N}^n \cap \mathbb{S}^n_+ \right\}, \\
\Pi^*(\boldsymbol{G}) &= \operatorname{argmin} \left\{ \|\boldsymbol{G} - \boldsymbol{Z}\| : \boldsymbol{Z} \in \mathbb{N}^n + \mathbb{S}^n_+ \right\}.
\end{aligned}$$

6

By the decomposition theorem of Moreau [16], we know that

$$\boldsymbol{G}_\lambda(y_0) \;=\; \Pi^*(\boldsymbol{G}_\lambda(y_0)) - \Pi(-\boldsymbol{G}_\lambda(y_0)).$$

Let $\widehat{\boldsymbol{X}}_\lambda(y_0)$ denote $\Pi(-\boldsymbol{G}_\lambda(y_0)) \in \mathbb{N}^n \cap \mathbb{S}_+^n$. Since $\Pi^*(\boldsymbol{G}_\lambda(y_0)) \in \mathbb{N}^n + \mathbb{S}_+^n$, we can take $\widehat{\boldsymbol{W}}_\lambda(y_0) \in \mathbb{N}^n$ and $\widehat{\boldsymbol{Y}}_\lambda(y_0) \in \mathbb{S}_+^n$ such that $\Pi^*(\boldsymbol{G}_\lambda(y_0)) = \widehat{\boldsymbol{W}}_\lambda(y_0) + \widehat{\boldsymbol{Y}}_\lambda(y_0)$. (Such $\widehat{\boldsymbol{W}}(y_0) \in \mathbb{N}^n$ and $\widehat{\boldsymbol{Y}}_\lambda(y_0) \in \mathbb{S}_+^n$ may not be unique.) Consequently,

$$g_\lambda(y_0) \;=\; \|\boldsymbol{G}_\lambda(y_0) - \Pi^*(\boldsymbol{G}_\lambda(y_0))\| = \|-\Pi(-\boldsymbol{G}_\lambda(y_0))\| = \left\|\widehat{\boldsymbol{X}}_\lambda(y_0)\right\|.$$

In the bisection method in [14], the accelerated proximal gradient method in [5] was applied to compute a decomposition $\boldsymbol{G}_\lambda(y_0) = \widehat{\boldsymbol{W}}_\lambda(y_0) + \widehat{\boldsymbol{Y}}_\lambda(y_0) - \widehat{\boldsymbol{X}}_\lambda(y_0)$. See Section 4.2 and Algorithm C of [14]. In view of numerical computation, the projection $\widehat{\boldsymbol{X}}_\lambda(y_0) = \Pi(-\boldsymbol{G}_\lambda(y_0))$ cannot be obtained exactly for a given $y_0 \in \mathbb{R}$, thus, the numerical value of $g_\lambda(y_0) = \left\|\widehat{\boldsymbol{X}}_\lambda(y_0)\right\|$ usually remains positive even when $y_0 \leq \eta^d(\lambda)$. Therefore, it is reasonable to determine whether y_0 is greater than $\eta^d(\lambda)$ according to whether the numerical value of the relative magnitude $f_\lambda(y_0) = \left\|\widehat{\boldsymbol{X}}_\lambda(y_0)\right\| / \max\{1, \|\boldsymbol{G}_\lambda(y_0)\|\}$ is greater than a sufficiently small positive number ϵ. To distinguish the exact value and the numerical value of $f_\lambda(y_0)$, we denote the latter by $h_\lambda(y_0)$.

Algorithm 2.2. (Bisection Method)

Step 0. Choose $\lambda > 0$ sufficiently large (e.g., $1.0e3 \leq \lambda \leq 1.0e6$), and positive numbers ϵ and δ sufficiently small (e.g., $\epsilon = 1.0e\text{-}11$ and $\delta = 1.0e\text{-}6$). Here δ determines the target length of an interval $[\ell^p, u^p] \subset \mathbb{R}$ which contains an approximation of $\eta^d(\lambda)$. Let $p = 0$.

Step 1. Find $\ell^0, u^0 \in \mathbb{R}$ such that $\ell^0 < u^0$ and $h_\lambda(\ell^0) \leq \epsilon < h_\lambda(u^0)$. Let $y_0^0 = (\ell^0 + u^0)/2$.

Step 2. If $u^p - \ell^p < \delta \max\{1, |\ell^p|, |u^p|\}$, output $\ell(\lambda, \epsilon, \delta) = \ell^p$ as a lower bound for $\eta^d(\lambda)$. Otherwise, compute a decomposition $\boldsymbol{G}_\lambda(y_0^p) = \widehat{\boldsymbol{W}}_\lambda(y_0^p) + \widehat{\boldsymbol{Y}}_\lambda(y_0^p) - \widehat{\boldsymbol{X}}_\lambda(y_0^p)$.

Step 3. If $h_\lambda(y_0^p) \leq \epsilon$, then let $\ell^{p+1} = y_0^p$ and $u^{p+1} = u^p$. Otherwise, let $\ell^{p+1} = \ell^p$ and $u^{p+1} = y_0^p$.

Step 4. Let $y_0^{p+1} = (\ell^{p+1} + u^{p+1})/2$. Replace $p + 1$ by p and go to Step 2.

See Section 4 and Algorithm A of [14] for more details.

3 Main results

The main part of Algorithm 2.2 is Step 3 where the interval is updated using $\epsilon > 0$ given at Step 0. As $\epsilon > 0$ is increased, a larger value of y_0^p with $h_\lambda(y_0^p) < \epsilon$ is likely to be assigned to ℓ^{p+1} at Step 3. Then, the assignment may lead to an incorrect final result since y_0^p with $g_\lambda(y_0^p) > 0$ is assigned to a lower bound ℓ^{p+1} of $\eta^d(\lambda)$, providing an

overestimated $\ell(\lambda, \epsilon, \delta)$ such that $\ell(\lambda, \epsilon, \delta) > \eta^d(\lambda)$. This situation cannot be avoided even with very small ϵ. That is, no matter how small $\epsilon > 0$ is chosen, such an update of the interval with incorrect y_0^p could occur, even if we assume that the numerical value $h_\lambda(y_0^p)$ of $f_\lambda(y_0^p)$ is (almost) exact. Under the circumstance, the least we can do is to choose $\epsilon > 0$ sufficiently small to decrease the error $\ell^{p+1} - \eta^d(\lambda) \geq 0$ even when such an incorrect update occurs. However, if we choose $\epsilon > 0$ too small, then y_0^p with $h_\lambda(y_0^p) > \epsilon$ and $g_\lambda(y_0^p) = 0$ is incorrectly assigned to u^{p+1} at Step 3 due to possible numerical errors in the evaluation of $f_\lambda(y_0^p)$; the quality of the lower bound $\ell(\lambda, \epsilon, \delta)$ of $\eta^d(\lambda)$ deteriorates.

To overcome the aforementioned difficulty of Algorithm 2.2, we present a new technique in Section 3.1. This technique also works effectively to find an initial finite lower bound for $\eta^d(\lambda)$ and to accelerate Algorithm 2.2. In Sections 3.2 and 3.3, we propose to use the primal-dual interior-point method as a post-processing procedure of the accelerated bisection algorithm (Algorithm 3.2) for improving the quality of lower bounds and computing an approximate optimal solution of the primal Lagrangian-DNN relaxation problem (7).

In the subsequent sections, we assume

Condition (D) A positive number ρ such that $\langle \boldsymbol{I}, \boldsymbol{X} \rangle \leq \rho$ for all $\boldsymbol{X} \in F(\boldsymbol{\Gamma})$ is known

in addition to Conditions (A), (B) and (C). We note that the existence of such a ρ is guaranteed by Condition (C) and ρ is available in QOPs with binary variables and many combinatorial optimization problems. See Section 5.

3.1 Computing a valid lower bound for the optimal value ζ^d of (6) and accelerating the bisection

Under Conditions (D), QOP (3) is equivalent to

$$\varphi = \inf \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in F(\boldsymbol{\Gamma}), \langle \boldsymbol{I}, \boldsymbol{X} \rangle \leq \rho \right\},$$

and its DNN relaxation

$$\bar{\zeta}^p = \inf \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \mid \boldsymbol{X} \in F(\mathbb{N}^n \cap \mathbb{S}_+^n), \langle \boldsymbol{I}, \boldsymbol{X} \rangle \leq \rho \right\} \tag{9}$$

provides a lower bound for the optimal value φ of QOP (3). Obviously $\zeta^p \leq \bar{\zeta}^p \leq \varphi$. Let $\lambda > 0$. We introduce the following problems:

$$\bar{\zeta}^d = \sup \left\{ y_0 + \rho t \;\middle|\; \begin{array}{l} \boldsymbol{W} \in \mathbb{N}^n, \; t \leq 0, \\ \boldsymbol{Q}^0 - \boldsymbol{H}^0 y_0 + \boldsymbol{H}^1 y_1 - \boldsymbol{I}t - \boldsymbol{W} = \boldsymbol{Y} \in \mathbb{S}_+^n \end{array} \right\},$$

$$\bar{\eta}^d(\lambda) = \sup \left\{ y_0 + \rho t \;\middle|\; \begin{array}{l} \boldsymbol{W} \in \mathbb{N}^n, \; t \leq 0, \\ \boldsymbol{G}_\lambda(y_0) - \boldsymbol{I}t - \boldsymbol{W} = \boldsymbol{Y} \in \mathbb{S}_+^n \end{array} \right\}. \tag{10}$$

Then $\eta^d(\lambda) \leq \bar{\eta}^d(\lambda) \leq \bar{\zeta}^d \leq \bar{\zeta}^p \leq \varphi$.

Lemma 3.1. *Suppose that $y_0 \in \mathbb{R}$ and $\boldsymbol{W} \in \mathbb{N}^n$. Let $\mu_{\min}$ be the minimum eigenvalue of $\boldsymbol{G}_\lambda(y_0) - \boldsymbol{W}$, and $t = \min\{0, \mu_{\min}\}$. Then $y_0 + \rho t \leq \bar{\eta}^d(\lambda)$.*

Proof. By definition, $\boldsymbol{Y} \equiv \boldsymbol{G}_\lambda(y_0) - \boldsymbol{I}t - \boldsymbol{W} \in \mathbb{S}^n_+$. Hence $(t, y_0, \boldsymbol{W}, \boldsymbol{Y})$ is a feasible solution of (10). Therefore $y_0 + \rho t \leq \bar{\eta}^d(\lambda)$. $\qquad\square$

We can incorporate Lemma 3.1 into Step 3 of Algorithm 2.2 in Section 2.

Algorithm 3.2. (Accelerated Bisection Method)

Step 0. Choose $\lambda > 0$ sufficiently large (e.g., $1.0\,\mathrm{e}3 \leq \lambda \leq 1.0\,\mathrm{e}6$), and positive numbers ϵ and δ sufficiently small (e.g., $\epsilon = 1.0\,\mathrm{e}\text{-}11$ and $\delta = 1.0\,\mathrm{e}\text{-}6$). Here δ determines the target length of an interval $[\ell^p,\, u^p] \subset \mathbb{R}$ which contains an approximation of $\eta^d(\lambda)$. Let $p = 0$. (The same as Step 1 of Algorithm 2.2.)

Step 1'. Find a $u^0 \in \mathbb{R}$ such that $\eta^d(\lambda) \leq u^0$. Let $\ell^0 = \underline{\ell}^0 = -\infty$. Choose $y_0^0 \leq u^0$.

Step 2' If $u^p - \ell^p < \delta \max\{1, |\ell^p|, |u^p|\}$, output $\underline{\ell}(\lambda, \epsilon, \delta) = \underline{\ell}^p$ as a lower bound for $\bar{\eta}^d(\lambda)$. Otherwise, compute a decomposition $\boldsymbol{G}_\lambda(y_0^p) = \widehat{\boldsymbol{W}}_\lambda(y_0^p) + \widehat{\boldsymbol{Y}}_\lambda(y_0^p) - \widehat{\boldsymbol{X}}_\lambda(y_0^p)$. (The same as Step 2 of Algorithm 2.2 except $\ell(\lambda, \epsilon, \delta) = \ell^p$.)

Step 3'. Let $\mu_{\min}^p$ be the minimum eigenvalue of $\boldsymbol{G}_\lambda(y_0^p) - \widehat{\boldsymbol{W}}_\lambda(y_0^p)$, and $t^p = \min\{0,\, \mu_{\min}^p\}$. Let $\underline{\ell}^{p+1} = \max\{\underline{\ell}^p, y_0^p + \rho t^p\}$. If $h_\lambda(y_0^p) \leq \epsilon$, then let $\ell^{p+1} = y_0^p$ and $u^{p+1} = u^p$. Otherwise, let $\ell^{p+1} = \max\{\underline{\ell}^{p+1},\ \ell^p\}$ and $u^{p+1} = y_0^p$.

Step 4. Let $y_0^{p+1} = (\ell^{p+1} + u^{p+1})/2$. Replace $p + 1$ by p and go to Step 2. (The same as Step 4 of Algorithm 2.2.)

Step 3' always provides a (finite and valid) lower bound $\underline{\ell}^{p+1}$ for $\bar{\eta}^d(\lambda) \geq \eta^d(\lambda)$ whose validity is guaranteed by Lemma 3.1. The computed lower bound is valid even when $h_\lambda(y_0^p) > \epsilon$ or when the computation of the decomposition of $\boldsymbol{G}_\lambda(y_0^p)$ at Step 2' is inexact, under the assumption that $\widehat{\boldsymbol{W}}_\lambda(y_0^p) \in \mathbb{N}^n$ and that the minimum eigenvalue μ^p of $\boldsymbol{G}_\lambda(y_0^p) - \widehat{\boldsymbol{W}}_\lambda(y_0^p)$ is exact (or within the desired accuracy). This is an important feature of Algorithm 3.2.

Specifically, if $h_\lambda(y_0^p) = f_\lambda(y_0^p)$ attained the exact 0, then the lower bound $\underline{\ell}^{p+1} = y_0^p + \rho t^p$ for $\bar{\eta}^d(\lambda)$ and the lower bound $\ell^{p+1} = y_0^p$ for $\eta^d(\lambda)$ would coincide with each other since $t^p = \mu_{\min}^p = 0$. For the case that $0 < h_\lambda(y_0^p) \leq \epsilon$, we consider two possibilities: $\mu_{\min}^p \geq 0$ and $\mu_{\min}^p < 0$. If $\mu_{\min}^p \geq 0$, then $(y_0, \boldsymbol{W}, \boldsymbol{Y}) = (y_0^p, \widehat{\boldsymbol{W}}_\lambda(y_0^p), \boldsymbol{G}_\lambda(y_0^p) - \widehat{\boldsymbol{W}}_\lambda(y_0^p))$ is a feasible solution of (6). Thus $\ell^{p+1} = y_0^p = \underline{\ell}^{p+1} = y_0^p + \rho t^p$ is clearly a lower bound for $\eta^d(\lambda)$. If $\mu_{\min}^p < 0$, however, neither $(y_0, \boldsymbol{W}, \boldsymbol{Y}) = (y_0^p, \widehat{\boldsymbol{W}}_\lambda(y_0^p), \boldsymbol{G}_\lambda(y_0^p) - \widehat{\boldsymbol{W}}_\lambda(y_0^p))$ nor $(y_0, \boldsymbol{W}, \boldsymbol{Y}) = (y_0^p, \widehat{\boldsymbol{W}}_\lambda(y_0^p), \widehat{\boldsymbol{Y}}_\lambda(y_0^p))$ is a feasible solution of (6), thus, y_0^p is not guaranteed to be a lower bound for $\eta^d(\lambda)$. But $\underline{\ell}^{p+1} = y_0^p + \rho t^p$ provides a valid lower bound for $\bar{\eta}^d(\lambda)$ even in this case. We may regard ρt^p as a perturbation to restore the dual feasibility.

In many practical applications of QOP (3), an upper bound u^0 for its optimal value $\varphi \geq \zeta^d$ is known in advance, which may be obtained from a trivial feasible solution of the QOP or by a heuristic method applied to the QOP. In such a case, the process of finding a $u^0 \in \mathbb{R}$ such that $\eta^d(\lambda) \leq u^0$ can be removed at Step 1'. Then, Steps 2 and Step 3' will provide a finite lower bound $\ell^1 = y_0^0 + \rho t^0$ (if $h_\lambda(y_0^0) > \epsilon$) or $\ell^1 = y_0^0$ (otherwise).

Another important feature of Algorithm 3.2 is that it still reduces the interval $[\ell^p, u^p]$ by at least half at each iteration. Thus, the finite termination is guaranteed. Furthermore,

if $h_\lambda(y_0^p) > \epsilon$ and $\underline{\ell}^p \leq \ell^p < y_0^p + \rho t^p$ occur at Step 3', then

$$
\begin{aligned}
u^{p+1} - \ell^{p+1} &= y_0^p - \max\{\underline{\ell}^{p+1}, \ell^p\} \\
&= y_0^p - \max\{y_0^p + \rho t^p, \ell^p\} \\
&= y_0^p - \ell^p - (y_0^p + \rho t^p - \ell^p) \\
&= (u^p + \ell^p)/2 - \ell^p - (y_0^p + \rho t^p - \ell^p) \\
&= (u^p - \ell^p)/2 - (y_0^p + \rho t^p - \ell^p) < (u^p - \ell^p)/2.
\end{aligned}
$$

We notice that the bisection has been accelerated. Table 1 illustrates the effectiveness of the new technique (Steps 1' and 3') introduced in Algorithm 3.2. We observe that each iteration reduces the length of the interval $[\ell^p, u^p]$ by the ratio of less than 0.4 except the last two iterations.

Table 1: bqp100-1 (A binary quadratic program with dimension = 100 from BIQMAC Library [21].) $\lambda = 2.68\text{e}4$, $\epsilon = \sqrt{10}\,\text{e-}12$, $\delta = 1.0\,\text{e-}6$.

p	$\underline{\ell}^p$	ℓ^p	y_0^p	u^p	$u^p - \ell^p$	$\dfrac{u^p - \ell^p}{u^{p-1} - \ell^{p-1}}$
0	$-\infty$	$-\infty$	-3985.000	-3985.000	∞	
1	-9721.974	-9721.974	-6853.487	-3985.000	5736.974	0.00
2	-8448.959	-8448.959	-7651.223	-6853.487	1595.472	0.28
3	-8159.406	-8159.406	-7905.314	-7651.223	508.183	0.32
4	-8088.564	-8088.564	-7996.939	-7905.314	183.249	0.36
5	-8063.934	-8063.934	-8030.437	-7996.939	66.995	0.37
6	-8055.164	-8055.164	-8042.801	-8030.437	24.728	0.37
7	-8052.104	-8052.104	-8047.452	-8042.801	9.304	0.38
8	-8050.947	-8050.947	-8049.200	-8047.452	3.495	0.38
9	-8050.530	-8050.530	-8049.865	-8049.200	1.331	0.38
10	-8050.372	-8050.372	-8050.118	-8049.865	0.507	0.38
11	-8050.309	-8050.309	-8050.214	-8050.118	0.190	0.37
12	-8050.285	-8050.285	-8050.249	-8050.214	0.071	0.38
13	-8050.275	-8050.275	-8050.262	-8050.249	0.025	0.36
14	-8050.271	-8050.262	-8050.255	-8050.249	0.013	0.50
15	-8050.271	-8050.262	-8050.263	-8050.255	0.006	0.50

3.2 Improving the quality of lower bounds by the primal-dual interior point method

Let $(y_0^p, \underline{\ell}^p, \ell^p, u^p)$ $(p = 0, 1, \ldots, p^*)$ be the iterate of Algorithm 3.2 applied to the Lagrangian-DNN relaxation (10) of QOP (3), where p^* denotes the last iteration which returns a lower bound $\underline{\ell}(\lambda, \epsilon, \delta) = \underline{\ell}^{p^*}$ at Step 2. Then, it is possible to find a q such that $\underline{\ell}(\lambda, \epsilon, \delta) = \underline{\ell}^{q+1} = y_0^q + \rho t^q$ holds. By the definition of t^q,

$$
(t, y_0, \boldsymbol{W}, \boldsymbol{Y}) = (t^q, y_0^q, \widehat{\boldsymbol{W}}_\lambda(y_0^q), \boldsymbol{G}_\lambda(y_0^q) - \widehat{\boldsymbol{W}}_\lambda(y_0^q) - t^q \boldsymbol{I})
$$

10

is a feasible solution of (10) with the objective value $y_0^q + \rho t^q$. Fixing $\boldsymbol{W} = \widehat{\boldsymbol{W}}_\lambda(y_0^q)$, we consider a subproblem of (10).

$$\hat{\eta}^d(\lambda, \widehat{\boldsymbol{W}}_\lambda(y_0^q)) \;=\; \sup\left\{ y_0 + \rho t \;\middle|\; \boldsymbol{G}_\lambda(y_0) - \boldsymbol{I}t - \widehat{\boldsymbol{W}}_\lambda(y_0^q) \in \mathbb{S}_+^n,\ t \le 0 \right\}, \quad (11)$$

which is a simple dual form of SDP involving two variables t and y_0. Obviously, (t^q, y_0^q) is a feasible solution of (11) with the objective value $y_0^q + \rho t^q$, which indicates $y_0^q + \rho t^q \le \hat{\eta}^d(\lambda, \widehat{\boldsymbol{W}}_\lambda(y_0^q))$. Thus, we have shown that the lower bound obtained by Algorithm 3.2 can be improved by applying the primal-dual interior-point method to (11).

Let $(\hat{t}, \hat{y}_0)$ denote an approximate optimal solution of (11). If it is not a feasible solution within the desired accuracy, decrease $\hat{t}$ to t so that $(t, \hat{y}_0)$ becomes feasible. Therefore, we can assume without loss of generality that an approximate optimal solution of (11) is feasible, and $\hat{y}_0 + \rho\hat{t}$ provides a valid lower bound for $\hat{\eta}^d(\lambda)$, although $y_0^q + \rho t^q \le \hat{y}_0 + \rho\hat{t}$ may not be satisfied due to possible numerical errors.

3.3   Further improvement

In addition to Conditions (A), (B), (C) and (D), we assume

Condition (E) $\boldsymbol{H}^1 = \boldsymbol{Q}^1 + \boldsymbol{Q}^2$ for some $\boldsymbol{Q}^1 \in \mathbb{N}^n + \mathbb{S}_+^n$ and $\boldsymbol{Q}^2 \in \mathbb{N}^n + \mathbb{S}_+^n$.

When the binary quadratic program, the multiple quadratic knapsack problem, the maximum stable set problem, and the quadratic assignment problem are formulated as QOP (3), the constraint $\langle \boldsymbol{Q}^1, \boldsymbol{xx}^T \rangle = 0$ with $\boldsymbol{Q}^1 \in \mathbb{N}^n$ arises from the complementarity constraint (or 0-1 constraint) of the original problem, and $\langle \boldsymbol{Q}^2, \boldsymbol{xx}^T \rangle = 0$ with $\boldsymbol{Q}^2 \in \mathbb{S}_+^n$ from the linear equality constraint.

Under Condition (E), we consider

$$\begin{aligned} &\tilde{\eta}^d(\lambda, \widehat{\boldsymbol{W}}_\lambda(y_0^q)) \\ &= \sup\left\{ y_0 + \rho t \;\middle|\; \boldsymbol{Q}^0 + \boldsymbol{Q}^1 y_1 + \lambda\boldsymbol{Q}^2 - \boldsymbol{H}^0 y_0 - \boldsymbol{I}t - \widehat{\boldsymbol{W}}_\lambda(y_0^q) \in \mathbb{S}_+^n,\ t \le 0 \right\}, \end{aligned} \quad (12)$$

which is a dual form of SDP with three variables t, y_0 and y_1. If $\boldsymbol{Q}^1$ is the zero matrix, (12) coincides with (11). We also note that $\boldsymbol{Q}^2$ can be the zero matrix. By definition, the identity $\boldsymbol{G}_\lambda(y_0) = \boldsymbol{Q}^0 + \lambda\boldsymbol{H}^1 - \boldsymbol{H}^0 y_0 = \boldsymbol{Q}^0 + \lambda\boldsymbol{Q}^1 + \lambda\boldsymbol{Q}^2 - \boldsymbol{H}^0 y_0$ holds. If (t, y_0) is a feasible solution of (11), then (t, y_0, y_1) with $y_1 = \lambda$ is a feasible solution of (12). Hence $\hat{\eta}^d(\lambda, \widehat{\boldsymbol{W}}_\lambda(y_0^q)) \le \tilde{\eta}^d(\lambda, \widehat{\boldsymbol{W}}_\lambda(y_0^q))$ follows. Consequently, the lower bound $\hat{\eta}^d(\lambda, \widehat{\boldsymbol{W}}_\lambda(y_0^q))$ is further improved.

In Table 2, we compare the three lower bounds, which were obtained by Algorithm 3.2, the methods in Section 3.2 and in Section 3.3. The test problems are the BQPs (bqp100-1 through bqp250-2) [21], the maximum stable set problems (ldc.256,ldc.512,ldc.1024) [18], the quadratic multiple knapsack problems (qmk-bqp 100-1 through qmk bqp250-2) [14] and the QAPs (bur26a through tai30b) [13]. After applying Algorithm 3.2 with $\lambda = 7.2e5$, $\sqrt{10}$e-12 and $\delta = 1.0e-6$, we used SDPT3 [20] to compute approximate optimal solutions of SDPs (11) and (12). If the solution (t, y_0) of (11) (or (t, y_0, y_1) of (12)) by SDPT3 was not feasible, we decreased the value of t so that it could be feasible (see the

11

end of Section 3.2). The lower bounds by Algorithm 3.2, SDP (11) and SDP (12) are denoted by $\underline{\ell}(\lambda, \epsilon, \delta)$, $\hat{\ell}(\lambda, \epsilon, \delta)$ and $\tilde{\ell}(\lambda, \epsilon, \delta)$, respectively. Theoretically, we must have $\underline{\ell}(\lambda, \epsilon, \delta) \leq \hat{\ell}(\lambda, \epsilon, \delta) \leq \tilde{\ell}(\lambda, \epsilon, \delta)$. But, the inequalities were violated due to numerical errors in some cases.

Table 2: Comparison between the lower bound $\underline{\ell}(\lambda, \epsilon, \delta)$ by Algorithm 3.2, $\hat{\ell}(\lambda, \epsilon, \delta)$ from SDP (11) and $\tilde{\ell}(\lambda, \epsilon, \delta)$ from SDP (12). $\lambda = 7.2e5$, $\epsilon = \sqrt{10}e\text{-}12$ and $\delta = 1.0e\text{-}6$.

Problem	Opt.val	Lower bounds		
		Algorithm 3.2	+pdipm (SDPT3)	
		$\underline{\ell}(\lambda, \epsilon, \delta)$	$\hat{\ell}(\lambda, \epsilon, \delta)$	$\tilde{\ell}(\lambda, \epsilon, \delta)$
bqp100-1	-7.970000e3	-8.051341e3	<-8.051303e3	<-8.051280e3
bqp100-2	-1.103600e4	-1.104947e4	<-1.104940e4	<-1.104937e4
bqp250-1	-4.560700e4	-4.627062e4	<-4.627014e4	<-4.626993e4
bqp250-2	-4.481000e4	-4.560589e4	<-4.560536e4	<-4.560511e4
1dc.256	-3.000000e1	-3.000005e1	<-3.000005e1	>-3.000008e1
1dc.512	-5.200000e1	-5.271156e1	<-5.271119e1	>-5.288615e1
1dc.1024	-9.400000e1	-9.559922e1	<-9.559905e1	>-1.329568e2
qmk100-1		-3.669021e3	<-3.668980e3	<-3.668979e3
qmk100-2		-4.531051e3	<-4.531017e3	<-4.531017e3
qmk250-1		-1.907215e4	<-1.907200e4	<-1.907199e4
qmk250-2		-1.936524e4	<-1.936509e4	<-1.936509e4
bur26a	+5.426670e6	+5.425859e6	<+5.425906e6	>+5.411443e6
bur26b	+3.817852e6	+3.817059e6	<+3.817130e6	>+3.816977e6
chr15a	+9.896000e3	+9.895875e3	<+9.895877e3	<+9.895879e3
chr15b	+7.990000e3	+7.989815e3	<+7.989817e3	<+7.989817e3
nug20	+2.570000e3	+2.505842e3	<+2.505851e3	>+2.504587e3
nug25	+3.744000e3	+3.625063e3	<+3.625178e3	>+3.623816e3
tai30a	+1.818146e6	+1.706789e6	<+1.706819e6	>+1.674473e6
tai30b	+6.371171e8	+5.984202e8	<+5.984438e8	<+5.984462e8

4 Some technical issues on Algorithm 3.2

4.1 Selecting the Lagrangian multiplier parameter λ

As opposed to Theorem 2.1, we cannot make λ in Algorithm 3.2 as large as possible in numerical computation. An appropriately chosen λ in Algorithm 3.2 is essential for the successful implementation. The choice depends on the characteristics of a given QOP (3) such as the magnitudes of the data matrices $\boldsymbol{Q}^0$, $\boldsymbol{H}^0$, $\boldsymbol{H}^1$, its optimal value and solutions. To make the choice less dependent on each individual QOP, it is reasonable to scale the data matrices $\boldsymbol{Q}^0$ and $\boldsymbol{H}^1$ such that $\boldsymbol{Q}^0/\|\boldsymbol{Q}^0\|$ and $\boldsymbol{H}^1/\|\boldsymbol{H}^1\|$, where $\|\boldsymbol{A}\|$ denotes some matrix norm of a matrix $\boldsymbol{A} \in \mathbb{S}^n$. We take the Frobenius norm for $\|\boldsymbol{Q}^0\|$ and $\|\boldsymbol{H}^1\|$ in our numerical experiments whose numerical results are reported in Section 5. Note that $\boldsymbol{H}^0$ is the $n \times n$ matrix with 1 at the (n, n)th component and 0 elsewhere or the $n \times n$

identity matrix in many cases, so that it is already well-scaled. With this modification, the resulting Lagrangian relaxations corresponding to (7) and (8) turn out to be

$$\eta^p(\lambda) = \inf\left\{\langle \boldsymbol{Q}^0 + (\|\boldsymbol{Q}^0\|/\|\boldsymbol{H}^1\|)\lambda\boldsymbol{H}^1, \, \boldsymbol{X}\rangle \ \middle| \ \begin{array}{l} \boldsymbol{X} \in \mathbb{N}^n \cap \mathbb{S}_+^n, \\ \langle \boldsymbol{H}^0, \, \boldsymbol{X}\rangle = 1 \end{array}\right\} \quad \text{and}$$

$$\eta^d(\lambda) = \sup\left\{y_0 \ \middle| \ \begin{array}{l} \boldsymbol{W} \in \mathbb{N}^n, \\ \boldsymbol{Q}^0 + (\|\boldsymbol{Q}^0\|/\|\boldsymbol{H}^1\|)\lambda\boldsymbol{H}^1 - \boldsymbol{H}^0 y_0 - \boldsymbol{W} = \boldsymbol{Y} \in \mathbb{S}_+^n \end{array}\right\}, \quad (13)$$

respectively.

It is known from Theorem 2.1 that $\eta^d(\lambda_1) \le \eta^d(\lambda_2)$ if $0 < \lambda_1 < \lambda_2$; a larger $\lambda > 0$ is desirable for a tighter lower bound $\eta^d(\lambda)$ for the optimal value φ of QOP (1). This is, however, not necessarily true in the computation of $\eta^d(\lambda)$ using double precision floating point arithmetic. In particular, if we choose too large λ in (13), some significant digits in the values of coefficient matrix $\boldsymbol{Q}^0$ may be lost, and/or catastrophic numerical errors may occur.

Figures 1 and 2 show how the lower bound $\underline{\ell}(\lambda, \epsilon, \delta)$ of $\bar{\eta}^d(\lambda)$ returned by Algorithm 3.2 changes as λ increases. The horizontal axis denotes $\log_{10} \lambda$ ($\lambda \in \{\lambda_1, \lambda_2, \ldots, \lambda_4\}$), where $\lambda_i = 1000\gamma^{i-1}$ and $\gamma^7 = 10^5$, $\epsilon = \sqrt{10}$e-12 and $\delta = 1.0$e-6. (We increased λ up to $1000\gamma^7 = 1.0e8$ in the numerical experiments, but for some of cases of $\lambda = 1000\gamma^5$, $1000\gamma^6$ and $1000\gamma^7$, differences among some lower bounds obtained are too large to be shown within the range of the horizontal axis in the figures.) The vertical axis denotes

$$\frac{\underline{\ell}(\lambda_i, \epsilon, \delta) - \underline{\ell}^*(\epsilon, \delta)}{|\underline{\ell}^*(\epsilon, \delta)|} \le 0, \ \text{where} \ \underline{\ell}^*(\epsilon, \delta) = \max\{\underline{\ell}(\lambda_i, \epsilon, \delta) : i = 1, 2, \ldots, 8\}.$$

From these figures, we observe that a suitable λ to obtain a tighter lower bound significantly differs among the instances.

In our numerical experiments in Section 5, Algorithm 3.2 is executed in parallel with $\lambda = \lambda_1, \lambda_2, \ldots, \lambda_8$ for $\underline{\ell}(\lambda_i, \epsilon, \delta)$.

4.2 Choices of the parameter ϵ

Although each iteration of Algorithm 3.2 is guaranteed to generate a valid lower bound $\underline{\ell}^{p+1}$ of $\bar{\eta}^d(\lambda)$ for any choice of the parameter $\epsilon > 0$, the quality of the final lower bound $\underline{\ell}(\lambda, \epsilon, \delta)$ computed with double precision floating point arithmetic may still be affected by ϵ. In computation, if we take $\epsilon > 0$ too small, then we may have $h_\lambda(y_0^p) > \epsilon$ for y_0^p much smaller than $\eta^d(\lambda)$ (hence $f_\lambda(y_0^p) = 0$) due to numerical errors. In this case, Step 3' is executed as $\ell^{p+1} = \max\{\underline{\ell}^{p+1}, \ell^p\}$ and $u^{p+1} = y_0^p < \eta^d(\lambda)$. We notice that the interval
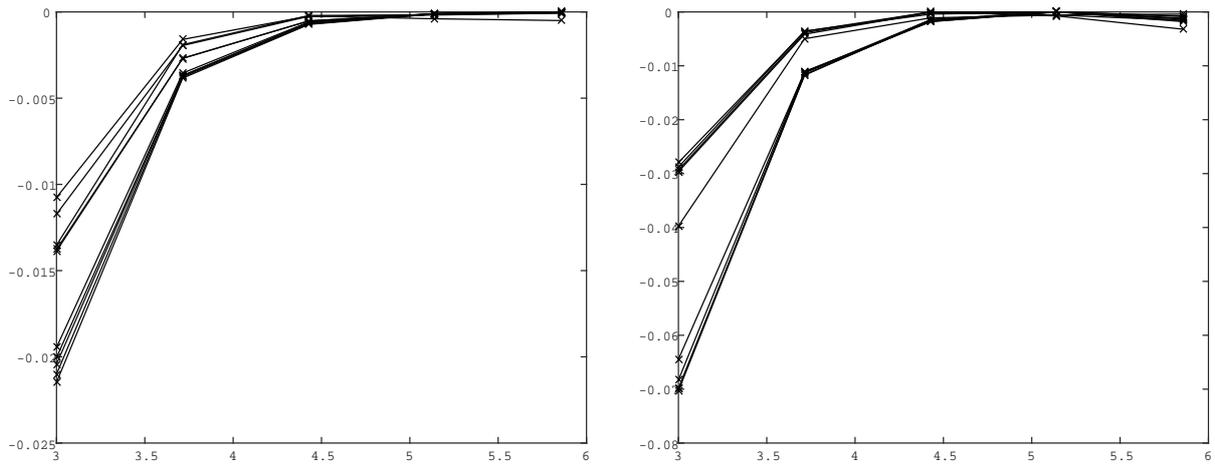
Figure 1: The sensitivity of the quality of lower bounds with respect to λ in Algorithm 3.2 ($\epsilon = \sqrt{10}$e-12 and $\delta = 1.0$e-6) for Billionnet-Elloumi binary quadratic program instances with dimension = 100 and 250 from BIQMAC library [21] on the left, and quadratic multi-knapsack problems with dimension = 100 and 250 [14] on the right. The best lower bounds obtained by Algorithm 3.2 for solving binary quadratic programs and quadratic multi-knapsack problems are shown in Tables 3 and 4, respectively.
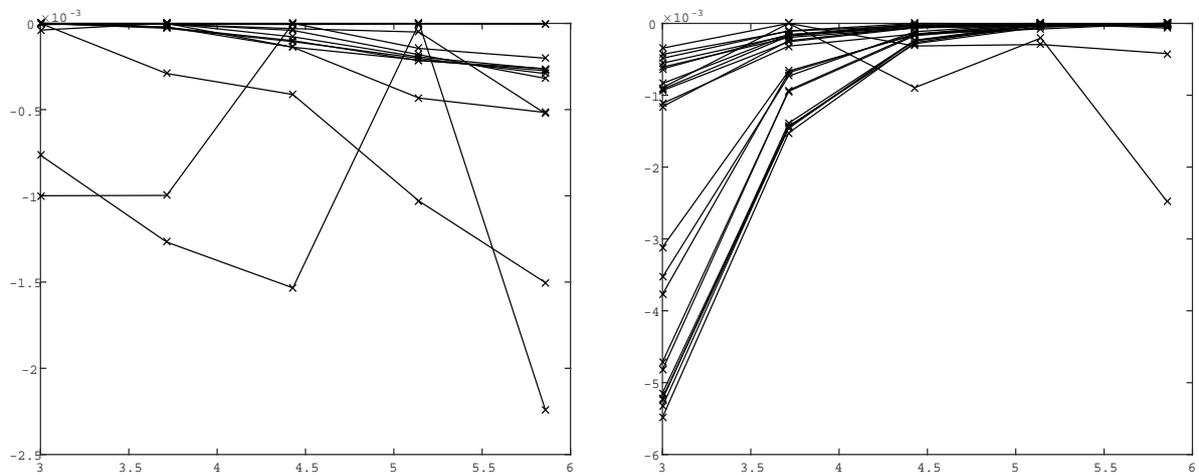
Figure 2: The sensitivity of the quality of lower bounds with respect to λ in Algorithm 3.2 ($\epsilon = \sqrt{10}$e-12 and $\delta = 1.0$e-6) applied to maximum stable set problems with dimensions = 256, 512 and 1024 from Sloane [18] on the left, and quadratic assignment problems with the size 15 - 35 from QAPLIB [13] on the right. The best lower bounds obtained by Algorithm 3.2 for solving maximum stable set problems and quadratic assignment problems are shown in Tables 5 and 6, respectively.

14
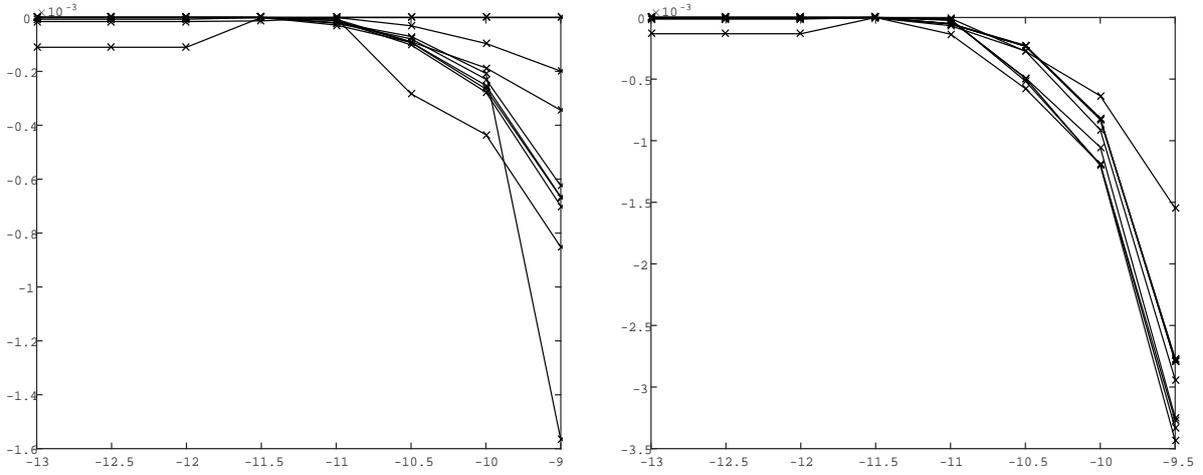
Figure 3: The sensitivity of the quality of lower bound $\underline{\ell}(\lambda, \epsilon, \delta)$ with respect to ϵ in Algorithm 3.2 ($\lambda = 1.0e5$ and $\delta = 1.0\text{e-}6$) applied to Billionnet-Elloumi binary quadratic program instances with dimension = 100 and 250 from BIQMAC library [21] on the left, and quadratic multi-knapsack problems with dimension = 100 and 250 from [14] on the right. The best lower bounds obtained by Algorithm 3.2 for these problems are shown in Tables 3 and 4.
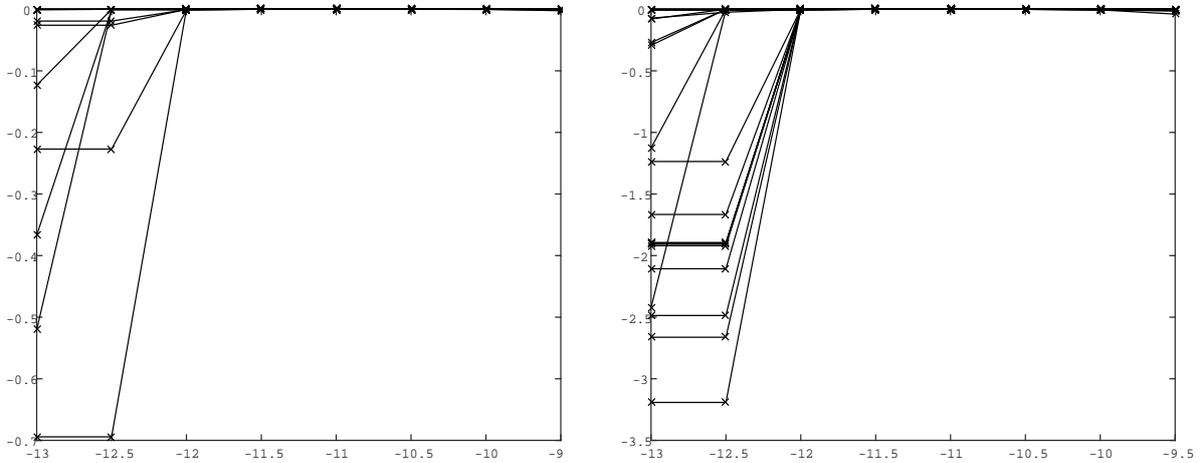
Figure 4: The sensitivity of the quality of lower bound $\underline{\ell}(\lambda, \epsilon, \delta)$ with respect to ϵ in Algorithm 3.2 ($\lambda = 1.0e5$ and $\delta = 1.0\text{e-}6$) applied to maximum stable set problems with dimensions = 256, 512 and 1024 from Sloane [18] on the left, and quadratic assignment problems with the size 15 - 35 from QAPLIB [13] on the right. The best lower bounds obtained by Algorithm 3.2 for these problems are shown in Tables 5 and 6 in Section 5.

15
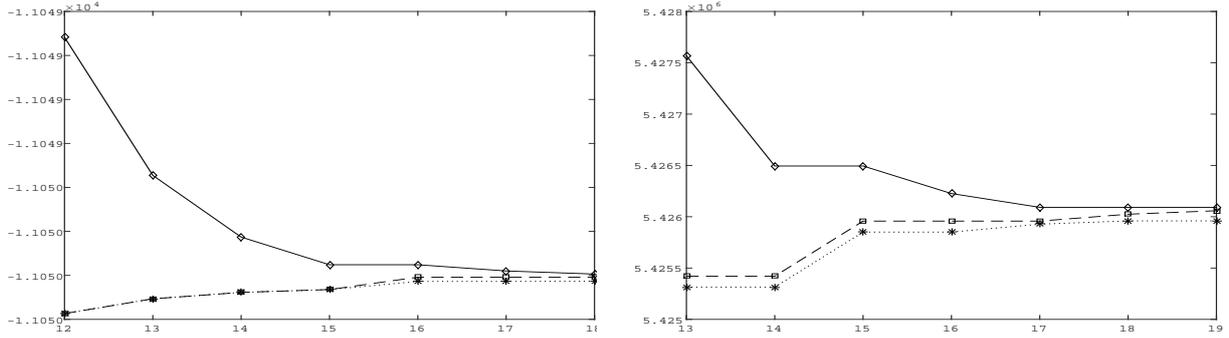
Figure 5: The graphs showing the changes of $\underline{\ell}^p$, ℓ^p and u^p at each iteration of Algorithm 3.2 for solving the binary quadratic optimization problem bpq100-2 from BIQMAC library [21] on the left and the quadratic assignment problem but26a from QAPLIB [13] on the right. $\epsilon = \sqrt{10}$e-12 was used. The numbers on the horizontal axis denote the iterations, and $\cdots * \cdots$ is the graph for the change of $\underline{\ell}^p$, - - -□- - - for ℓ^p and ——◇—— for u^p.

$[\ell^{p+1}, u^{p+1}]$ lies below $\eta^d(\lambda)$, which should be avoided since it means $\underline{\ell}(\lambda, \epsilon, \delta) \leq u^{p+1} = y_0^{p+1} < \eta^d(\lambda)$. On the other hand, as we take ϵ larger, $y_0^p > \eta^d(\lambda)$ (hence $f_\lambda(y_0^p) > 0$) but $h_\lambda(y_0^p) < \epsilon$ may occur. Then, at Step 3', the interval is updated as $\ell^{p+1} = y_0^p > \eta^d(\lambda)$ and $u^{p+1} = u^p$. Thus, $\eta^d(\lambda) < \ell^{p+1} \leq y_0^q$ holds at the subsequent iterations $q \geq p+1$. Although the relation $y_0^q + \rho t^q \leq \eta^d(\lambda)$ is guaranteed by Lemma 3.1, the quality of lower bounds $y_0^q + \rho t^q$ ($q \geq p+1$) (hence $\underline{\ell}(\lambda, \epsilon, \delta)$) may deteriorate if ℓ^{p+1} is much larger than $\eta^d(\lambda)$.

To search for an appropriate value of ϵ, we performed preliminary numerical experiments on ϵ. Figures 3 and 4 show the change of $\underline{\ell}(\lambda, \epsilon, \delta)$ as ϵ changes, where $\lambda = 1.0$e5 and $\delta = 1.0$e-6 were used. The horizontal axis denotes $\log_{10} \epsilon$ ($\epsilon \in \{\epsilon_1, \epsilon_2, \ldots, \epsilon_8\}$), where $\epsilon_i = 1.0$e-9α^{i-2} ($i = 1, 2, \ldots, 8$) and $\alpha = \sqrt{10}$. The vertical axis denotes

$$\frac{\underline{\ell}(\lambda, \epsilon_i, \delta) - \underline{\ell}^*(\lambda, \delta)}{|\underline{\ell}^*(\lambda, \delta)|} \leq 0, \text{ where } \underline{\ell}^*(\lambda, \delta) = \max\{\underline{\ell}(\lambda, \epsilon_i, \delta) : i = 1, 2, \ldots, 8\}.$$

From the figures, we observe that 1.0e-12 $\leq \epsilon \leq 1.0$e-11 provides a tighter lower bound than the other values.

Figure 5 displays the changes of $\underline{\ell}^p$, ℓ^p and u^p as the iteration of Algorithm 3.2 increases. At the last iteration, $u^p - \ell^p$ attains less than $\delta = 1.0$e-6, but there is a larger gap than δ between $\underline{\ell}^p$ and ℓ^p.

5   Applications

In Section 5.1, we show how a class of QOPs with linear equality, 0-1 and complementarity constraints, which covers various combinatorial optimization problems, can be converted into QOP (3). Then, we focus on the binary quadratic optimization problem in Section

16

5.2, the quadratic multiple knapsack problems in Section 5.3, the quadratic assignment problem in Section 5.5, and show the numerical results. The maximum stable set problem could be formulated in QOP (15), but it is directly reduced to QOP (3) in Section 5.4.

All the numerical experiments were performed with Matlab on a Mac Pro with Intel Xeon E5 8-core CPU (3.0 GHZ) and 64 GB memory, and SDPT3 [20] was used to solve SDP (12) described in Section 3.3. For the numerical results in the tables, the parameters $\epsilon = \sqrt{10}$e-12 and $\delta = 1.0$e-6 are fixed, and

$$\tilde{\lambda}^* = \arg \max\{\underline{\ell}(\lambda_i, \epsilon, \delta), \ \tilde{\ell}(\lambda_i, \epsilon, \delta) : \lambda_i = 1000\gamma^{i-1} \ (i = 1, 2, \ldots, 8)\}, \qquad (14)$$

where $\gamma^7 = 10^5$, $\underline{\ell}(\lambda_i, \epsilon, \delta)$ denotes the lower bound obtained by Algorithm 3.2 and $\tilde{\ell}(\lambda_i, \epsilon, \delta)$ the lower bound by SDPT3 applied to SDP (12). The computation of $\tilde{\ell}(\lambda_i, \epsilon, \delta)$ $(i = 1, 2, \ldots, 8)$ as well as $\underline{\ell}(\lambda_i, \epsilon, \delta)$ $(i = 1, 2, \ldots, 8)$ were executed in parallel.

The optimal values or the best known object values are available for our test problems selected from the sets of the binary quadratic optimization problems [21], the maximum stable set problems [18] and the quadratic assignment problems [13]. In our numerical experiments implementing Algorithm 3.2, initial points u^0 and y_0^0 at Step 1' for these test problems were taken as

$$u_0 = y_0^0 = \text{opt.val} + \beta \times \max\{|\text{opt.val}|, 1\}$$

in Tables 3, 5, 6, and 7, where opt.val denotes the optimal value or the best known objective value and $\beta = 0.5$. For the quadratic multiple knapsack test problems [14] whose optimal values are unknown, we used a trivial initial guess as $u_0 = y_0^0 = 0$ in Table 4. The test problems included in Tables 3, 4, 5, and 6 are the same as the ones in Tables 1, 2, 3, and 4 of [14], respectively. We note that the original bisection method, Algorithm 2.2, was applied to the problems in [14].

The aim of the numerical experiments in this paper is to find valid and tight lower bounds, rather than to achieve the computational efficiency of the proposed methods. Thus, CPU time shown in the tables is not as fast as in [14]. Shorter CPU time of the proposed methods can be obtained by loosening the parameters, which will result in valid but less tight lower bounds.

5.1 A class of QOPs with linear equality, 0-1 and complementarity constraints

We consider

$$\varphi = \min\left\{\langle \boldsymbol{Q}, \boldsymbol{u}\boldsymbol{u}^T\rangle \mid \boldsymbol{u} \in \mathbb{R}_+^m, \ \boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \ u_i u_j = 0 \ (i, j) \in \mathcal{E}\right\}, \qquad (15)$$

where $\boldsymbol{A}$ denotes an $k \times m$ matrix, $\boldsymbol{Q} \in \mathbb{S}^m$, $\boldsymbol{b} \in \mathbb{R}^k$ and $\mathcal{E}$ a subset of $\{(i, j) : 1 \le i < j \le m\}$. In this subsection, we convert QOP (15) to QOP (3) to apply the numerical methods described in Section 3. The 0-1 constraints $u_j = 0$ or 1 $(j \in J \subseteq \{1, 2, \ldots, m\})$ can be added as they can be converted into linear equalities $u_j + v_j = 1$ $(j \in J)$ and complementarity constraints $u_j v_j = 0$ $(j \in J)$ by introducing a slack variable $v_j \in \mathbb{R}_+$ $(j \in J)$.

Let $n = m + 1$. Define

$$\boldsymbol{Q}^0 \;=\; \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{pmatrix} \in \mathbb{S}^n, \; \boldsymbol{Q}^1 = \begin{pmatrix} \boldsymbol{A}^T \\ -\boldsymbol{b}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{A} & -\boldsymbol{b} \end{pmatrix} \in \mathbb{S}^n,$$

$$\boldsymbol{H}^0 \;=\; \text{the } n \times n \text{ symmetric positive semidefinite matrix}$$
$$\text{with 1 at the } (n, n)\text{th component and 0 elsewhere,}$$
$$\boldsymbol{Q}_{ij} \;=\; \text{the } n \times n \text{ symmetric nonnegative matrix with 1 at the } (i, j)\text{th}$$
$$\text{and the } (j, i)\text{th components and 0 elsewhere } ((i, j) \in \mathcal{E}). \qquad (16)$$

For every $\boldsymbol{x} = (\boldsymbol{u}, u_{m+1}) \in \mathbb{R}^n$, we notice that

$$\langle \boldsymbol{Q}^0, \boldsymbol{xx}^T \rangle \;=\; \langle \boldsymbol{Q}, \boldsymbol{uu}^T \rangle, \; \langle \boldsymbol{H}^0, \boldsymbol{xx}^T \rangle = u_{m+1}^2,$$
$$\langle \boldsymbol{Q}^1, \boldsymbol{xx}^T \rangle \;=\; \|\boldsymbol{Au} - \boldsymbol{b}u_{m+1}\|^2, \; \langle \boldsymbol{Q}_{ij}, \boldsymbol{xx}^T \rangle = 2u_i u_j \; ((i, j) \in \mathcal{E}).$$

Thus, we can rewrite QOP (15) as

$$\varphi \;=\; \left\{ \langle \boldsymbol{Q}, \boldsymbol{xx}^T \rangle \; \middle| \; \begin{array}{l} \boldsymbol{x} \in \mathbb{R}^n_+, \; \langle \boldsymbol{H}^0, \boldsymbol{xx}^T \rangle = 1, \; \langle \boldsymbol{Q}^1, \boldsymbol{xx}^T \rangle = 0, \\ \langle \boldsymbol{Q}_{ij}, \boldsymbol{xx}^T \rangle = 0 \; (i, j) \in \mathcal{E} \end{array} \right\}. \qquad (17)$$

By construction, $\langle \boldsymbol{Q}^1, \boldsymbol{xx}^T \rangle \geq 0$ and $\langle \boldsymbol{Q}_{ij}, \boldsymbol{xx}^T \rangle \geq 0 \; (i, j) \in \mathcal{E}$ for every $\boldsymbol{x} \in \mathbb{R}^n_+$. Letting $\boldsymbol{Q}^2 = \sum_{(i,j) \in \mathcal{E}} \boldsymbol{Q}_{ij} \in \mathbb{N}^n$ and $\boldsymbol{H}^1 = \boldsymbol{Q}^1 + \boldsymbol{Q}^2$, we see that

$$\langle \boldsymbol{Q}_{ij}, \boldsymbol{xx}^T \rangle = 0 \; ((i, j) \in \mathcal{E}) \;\; \Leftrightarrow \;\; \langle \boldsymbol{Q}^2, \boldsymbol{xx}^T \rangle = 0,$$
$$\langle \boldsymbol{Q}^1, \boldsymbol{xx}^T \rangle = 0 \;\; \text{and} \;\; \langle \boldsymbol{Q}^2, \boldsymbol{xx}^T \rangle = 0 \;\; \Leftrightarrow \;\; \langle \boldsymbol{H}^1, \boldsymbol{xx}^T \rangle = 0$$

for every $\boldsymbol{x} \in \mathbb{R}^n_+$. Therefore, QOP (17) (hence QOP (15)) is equivalent to QOP (3):

$$\begin{aligned} \varphi \;=\; & \min \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{xx}^T \rangle \; \middle| \; \begin{array}{l} \boldsymbol{x} \in \mathbb{R}^n_+, \; \langle \boldsymbol{H}^0, \boldsymbol{xx}^T \rangle = 1, \\ \langle \boldsymbol{Q}^1, \boldsymbol{xx}^T \rangle = 0, \; \langle \boldsymbol{Q}^2, \boldsymbol{xx}^T \rangle = 0 \end{array} \right\} \\ =\; & \min \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \; \middle| \; \boldsymbol{X} \in \boldsymbol{\Gamma}, \; \langle \boldsymbol{H}^0, \boldsymbol{X} \rangle = 1, \; \langle \boldsymbol{H}^1, \boldsymbol{X} \rangle = 0 \right\} \\ =\; & \min \left\{ \langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle \; \middle| \; \boldsymbol{X} \in F(\boldsymbol{\Gamma}) \right\}. \end{aligned}$$

Lemma 5.1. *If the feasible region of QOP* (15) *is nonempty and* $\{\boldsymbol{u} \in \mathbb{R}^m_+ : \boldsymbol{Au} = \boldsymbol{b}\}$ *is bounded, then QOP* (3) *induced from QOP* (15) *satisfies Conditions (A), (B) and (C). Furthermore, if* $\sum_{i=1}^m u_i^2 \leq \bar{\rho}$ *holds for every* $\boldsymbol{u} \in \{\boldsymbol{u} \in \mathbb{R}^m_+ : \boldsymbol{Au} = \boldsymbol{b}\}$, *then* $\langle \boldsymbol{I}, \boldsymbol{X} \rangle \leq \bar{\rho} + 1$ *holds for every* $\boldsymbol{X} \in F(\boldsymbol{\Gamma})$, *where* $\boldsymbol{I}$ *denotes the* $n \times n$ *identity matrix.*

Proof. QOP (3) induced from QOP (15) apparently satisfies Conditions (A) and (B) by construction. Condition (C) (or equivalent Condition (C')) follows from Lemma 1 of [14]. To prove the last assertion, assume that $\boldsymbol{x} = (\boldsymbol{u}, u_{m+1})$ satisfies $\boldsymbol{xx}^T \in F(\boldsymbol{\Gamma})$. Then $\boldsymbol{u} \in \{\boldsymbol{u} \in \mathbb{R}^m_+ : \boldsymbol{Au} = \boldsymbol{b}\}$ and $u_{m+1} = 1$ by construction. Hence $\langle \boldsymbol{I}, \boldsymbol{xx}^T \rangle = \sum_{i=1}^m u_i^2 + u_{m+1}^2 \leq \bar{\rho} + 1$ follows.

5.2 Binary quadratic programs

The binary quadratic program (BQP) is formulated as

$$\varphi \;=\; \min\left\{\langle \overline{\boldsymbol{Q}}, \boldsymbol{u}\boldsymbol{u}^T\rangle \;\mid\; \boldsymbol{u} \in \{0,\,1\}^k\right\},$$

where $\overline{\boldsymbol{Q}} \in \mathbb{S}^k$. Introducing a slack variable vector $\boldsymbol{v} \in \mathbb{R}^k$, we reformulate the problem as

$$\varphi \;=\; \min\left\{\langle \overline{\boldsymbol{Q}}, \boldsymbol{u}\boldsymbol{u}^T\rangle \;\mid\; \begin{array}{l}(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2k}_+,\; \boldsymbol{u} + \boldsymbol{v} = \boldsymbol{e},\\ u_i v_i = 0 \;(i = 1, 2, \ldots, k)\end{array}\right\},$$

where $\boldsymbol{e}$ denotes the k-dimensional column vector of 1's. Let $m = 2k$, and let $\boldsymbol{I}$ denote the $k \times k$ identity matrix. By defining

$$\boldsymbol{Q} \;=\; \begin{pmatrix} \overline{\boldsymbol{Q}} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \in \mathbb{S}^m,\; \boldsymbol{A} = (\boldsymbol{I}\ \boldsymbol{I}),\; \boldsymbol{b} = \boldsymbol{e},\; \mathcal{E} = \{(i, i+k) : i = 1, 2, \ldots, k\},$$

the binary quadratic program is represented as QOP (15). From the construction, every feasible solution $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^m$ of the formulated problem satisfies $\sum_{i=1}^m x_i^2 = k = m/2$. Thus, we can take $\rho = k + 1$ for the construction of DNN problem (9) from QOP (15).

We observe in Table 3 that the lower bounds $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ obtained by Algorithm 3.2 are better than the ones reported in [14] except for bqp100-1 and bqp100-2. Although $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ can not exceed $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ in theory, $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ frequently becomes smaller than $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ in computation due to numerical errors from SDPT3. Notice that the number of iterations in the column of Algorithm 3.2 is at most 20 by accelerating the bisection method. The CPU time for implementing the technique in Section 3.3 is very short as shown in the last column of Table 3.

5.3 Quadratic multiple knapsack problems

We let $\overline{\boldsymbol{A}}$ be an $k \times m$ matrix, $\bar{\boldsymbol{b}} \in \mathbb{R}^m$ and $\overline{\boldsymbol{Q}} \in \mathbb{S}^m$ in this subsection. The quadratic multiple knapsack problem is expressed as

$$\varphi \;=\; \min\left\{\langle \overline{\boldsymbol{Q}}, \boldsymbol{u}\boldsymbol{u}^T\rangle \;\mid\; \overline{\boldsymbol{A}}\boldsymbol{u} + \boldsymbol{s} = \bar{\boldsymbol{b}},\; \boldsymbol{u} \in \{0,\,1\}^m,\; \boldsymbol{s} \geq \boldsymbol{0}\right\}. \tag{18}$$

We can convert (18) into QOP (15), and (3) as described in the previous section.

Table 4 shows the numerical results on the same quadratic multiple knapsack problems as in Table 2 of [14] where the original bisection method, Algorithm 2.2, was applied to the problems. As mentioned previously, the optimal values of the test problems in Table 4 are not known. As a result, larger values obtained for the lower bounds cannot be regarded as better bounds than smaller values without considering the validity of the lower bounds obtained. If we compare the results with the ones in Table 2 of [14], we notice that the lower bound $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ is slightly worse than the one reported there in all cases. Since the optimal value is not known, we cannot discuss the quality of $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ using the optimal value. What we can say is that $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ is a valid lower bound, while the validity of the lower bounds obtained by the original bisection method (Algorithm 2.2) cannot be guaranteed to be valid as described in Section 3.1.

We see in Table 4 that the number of iterations of Algorithm 3.2 is small and the lower bounds $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ were further improved by the technique in Section 3.3 for all test problems.

Table 3: Billionnet-Elloumi BQP instances with dimension = 100 and 250 from BIQMAC library [21]. $\epsilon = \sqrt{10}$e-12, $\delta = 1.0$e-6 and $u_0 = y_0^0 =$ Opt.val $+ 0.5 \times \max\{|\text{Opt.val}|, 1\}$ were used. See (14) for the definition of $\tilde{\lambda}^*$. Numerical results on the same set of BQP instances were reported in Table 1 of [14]. The asterisk * in bqp100-1* and bqp100-2* indicates that both lower bounds are slightly worse than the one obtained in [14], but for all other problems, at least one of them is better than the lower bound obtained by [14]. † means SDPT3 termination code.

Problem	$\tilde{\lambda}^*$	Opt.val	Lower bounds	
			Algorithm 3.2 $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$	+pdipm $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$
			(#iter, CPU)	(SDPT3 CPU)
bqp100-1*	2.68e4	-7.970000e3	-8.050271e3(15,3.5e1)	< -8.048805e3(2.2e0)†-5
bqp100-2*	1.93e7	-1.103600e4	-1.104795e4(19,4.1e1)	> -1.104802e4(1.7e0)†-5
bqp100-3	1.00e8	-1.272300e4	-1.272300e4(9,1.2e1)	> -1.272419e4(2.2e0)†-5
bqp100-4	1.00e8	-1.036800e4	-1.036800e4(10,1.3e1)	> -1.037406e4(2.1e0)†-7
bqp100-5	1.93e7	-9.083000e3	-9.087234e3(19,4.1e1)	> -9.087295e3(1.6e0)†-5
bqp250-1	3.73e6	-4.560700e4	-4.626981e4(13,3.1e2)	< -4.626926e4(8.9e0)†-5
bqp250-2	7.20e5	-4.481000e4	-4.560589e4(17,3.9e2)	< -4.560511e4(6.5e0)
bqp250-3	7.20e5	-4.903700e4	-4.948116e4(17,3.9e2)	< -4.948032e4(7.1e0)†-5
bqp250-4	7.20e5	-4.127400e4	-4.202922e4(12,2.9e2)	< -4.202858e4(7.6e0)†-5
bqp250-5	7.20e5	-4.796100e4	-4.845482e4(16,3.7e2)	< -4.845401e4(8.1e0)†-5

Table 4: Quadratic multiple knapsack problems with dimension = 100 and 250 [14]. $\epsilon = \sqrt{10}$e-12, $\delta = 1.0$e-6 and $u_0 = y_0^0 = 0$ were used. See (14) for the definition of $\tilde{\lambda}^*$. † means SDPT3 termination code.

Problem	$\tilde{\lambda}^*$	Lower bounds	
		Algorithm 3.2 $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$	+pdipm $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$
		(#iter, CPU)	(SDPT3 CPU)
qmk100-1	2.68e4	-3.657909e3(15,4.1e1)	< -3.657907e3(1.5e0)
qmk100-2	2.68e4	-4.521207e3(14,3.4e1)	< -4.521206e3(1.6e0)
qmk100-3	1.39e5	-4.638951e3(16,5.7e1)	< -4.638942e3(1.5e0)
qmk100-4	1.39e5	-4.695456e3(14,3.5e1)	< -4.695452e3(2.0e0)
qmk100-5	2.68e4	-3.835542e3(15,4.5e1)	< -3.835541e3(1.5e0)
qmk250-1	1.39e5	-1.905591e4(12,3.6e2)	< -1.905590e4(1.1e1)
qmk250-2	1.39e5	-1.934490e4(16,4.4e2)	< -1.934484e4(8.7e0)
qmk250-3	1.39e5	-1.983744e4(12,3.6e2)	< -1.983743e4(1.0e1)
qmk250-4	1.39e5	-1.843393e4(16,4.4e2)	< -1.843388e4(8.6e0)
qmk250-5	1.39e5	-1.948604e4(12,3.6e2)	< -1.948603e4(1.1e1)

5.4 Maximum stable set problems

Given a graph G with n nodes and edge set $\mathcal{E}$, the stability number $\alpha(G)$ is defined as the cardinality of a maximum stable set of G.

$$-\alpha(G) \;=\; \min\left\{-\boldsymbol{e}^T\boldsymbol{u} \mid u_i u_j = 0, \; ((i,j) \in \mathcal{E}), \boldsymbol{u} \in \{0,1\}^n\right\},$$

where $\boldsymbol{e} = (1,1,\ldots,1)^T \in \mathbb{R}^n$. As a lower bound for $\alpha(G)$, we introduce a QOP [9]:

$$\varphi \;=\; \min\left\{\langle -\boldsymbol{e}\boldsymbol{e}^T, \boldsymbol{x}\boldsymbol{x}^T\rangle \mid \begin{array}{l} \boldsymbol{x} \in \mathbb{R}^n_+, \; \langle \boldsymbol{I}, \boldsymbol{x}\boldsymbol{x}^T\rangle = 1, \\ x_i x_j = 0 \; ((i,j) \in \mathcal{E}) \end{array}\right\}.$$

Note that $-\alpha(G) \geq \varphi$ follows by letting $\boldsymbol{x} = \boldsymbol{u}/\sqrt{\boldsymbol{e}^T\boldsymbol{u}}$. Define $\boldsymbol{Q}_{ij} \in \mathbb{S}^n_+$ $((i,j) \in \mathcal{E})$ by (16), and let $\boldsymbol{Q}^0 = -\boldsymbol{e}\boldsymbol{e}^T$, $\boldsymbol{H}^0 = \boldsymbol{I}$ and $\boldsymbol{H}^1 = \boldsymbol{Q}^2 = \sum_{(i,j)\in\mathcal{E}} \boldsymbol{Q}_{ij}$. Then, we can rewrite the QOP as (3). Obviously, Conditions (A), (B), (C) and (D) are all satisfied. In this case, we can take $\rho = 1$ for DNN problem (9) which is equivalent to DNN relaxation (5) of QOP (3).

We display the numerical results in Table 5. If the values of $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ and $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ are compared to the numerical results in Table 3 of [14], we observe that at least one of the lower bounds is better than the one in [14] for all test problems except for 1tc.512 and 1tc.1024. The number of iterations taken by Algorithm 3.2 is at most 20 and the CPU time spent by SDPT3 for further improvement ranges from 1.4e0 to 3.8e1 seconds.

Table 5: Maximum stable set problems with $n = 256, 512$ and 1024 from [18]. $\epsilon = \sqrt{10}$e-12, $\delta = 1.0$e-6 and $u_0 = y_0^0 = $ Opt.val $+ 0.5 \times \max\{|\text{Opt.val}|, 1\}$ were used. See (14) for the definition of $\tilde{\lambda}^*$. The asterisk * in 1tc.256* and 1tc.1024* indicates that both lower bounds are slightly worse than the one obtained in [14], but for all other problems, at least one of them is better than the lower bound obtained by [14]. † indicates SDPT3 termination code.

Problem	$\tilde{\lambda}^*$	Opt.val	Lower bounds	
			Algorithm 3.2 $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$	+pdipm $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$
			(#iter, CPU)	(SDPT3 CPU)
1dc.256	1.00e8	-3.000000e1	-3.000000e1(11,1.6e1)	> -2.324768e2(1.4e0)†-7
1et.256	1.00e3	-5.000000e1	-5.448141e1(14,4.9e1)	< -5.448140e1(1.5e0)
1tc.256*	1.00e8	-6.300000e1	-6.328793e1(13,5.4e1)	> -4.341231e4(8.9e-01)†-7
1zc.256	1.00e3	-3.600000e1	-3.733345e1(19,2.1e1)	< -3.733338e1(1.2e0)
1dc.512	1.39e5	-5.200000e1	-5.270561e1(20,2.7e2)	> -5.270580e1(6.1e0)†-7
1et.512	2.68e4	-1.000000e2	-1.035688e2(8,2.1e2)	> -1.035689e2(5.9e0)†-7
1tc.512	1.39e5	-1.100000e2	-1.126771e2(11,3.0e2)	> -1.126773e2(6.3e0)†-7
1zc.512	5.18e3	-6.200000e1	-6.800684e1(14,1.3e2)	< -6.800644e1(8.1e0)†-7
1dc.1024	1.00e3	-9.400000e1	-9.556077e1(19,2.1e3)	> -9.556079e1(3.4e1)†-7
1et.1024	1.00e3	-1.710000e2	-1.821063e2(16,2.2e3)	< -1.821059e2(3.0e1)†-7
1tc.1024*	1.00e3	-1.960000e2	-2.048393e2(14,2.9e3)	> -2.048393e2(3.8e1)†-7
1zc.1024	1.00e3	-1.120000e2	-1.278913e2(20,1.4e3)	> -1.375008e2(3.0e1)†-7

5.5 Quadratic assignment problems

Given $m \times m$ matrices $\boldsymbol{B}$ and $\boldsymbol{C}$, the QAP is described as

$$\varphi = \left\{ \left\langle \boldsymbol{U}, \boldsymbol{B}\boldsymbol{U}\boldsymbol{C}^T \right\rangle \mid \boldsymbol{U} \text{ is an } m \times m \text{ permutation matrix} \right\}.$$

We characterize the permutation matrix condition on $\boldsymbol{U}$ as

$$\boldsymbol{U} \geq \boldsymbol{O}, \ \left\langle \boldsymbol{e}\boldsymbol{e}_j^T, \boldsymbol{U} \right\rangle = \left\langle \boldsymbol{e}_j\boldsymbol{e}^T, \boldsymbol{U} \right\rangle = 1 \ (j = 1, \ldots, m),$$
$$U_{ik}U_{jk} = U_{ki}U_{kj} = 0 \ (k = 1, \ldots, m, \ i \neq j).$$

Identifying the $m \times m$ matrix variable $\boldsymbol{U} = (\boldsymbol{u}_1 \ \boldsymbol{u}_2 \ \ldots \ \boldsymbol{u}_m)$ with the m^2-dimensional column vector variable $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m)$ concatenating all the column vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m$ vertically, we can rewrite the objective quadratic form of the QAP and the condition as

$$\left\langle \boldsymbol{U}, \boldsymbol{B}\boldsymbol{U}\boldsymbol{C}^T \right\rangle = \left\langle \boldsymbol{C} \otimes \boldsymbol{B}, \boldsymbol{u}\boldsymbol{u}^T \right\rangle \text{ and}$$
$$\boldsymbol{u} \in \mathbb{R}_+^{m^2}, \ (\boldsymbol{e} \otimes \boldsymbol{e}_j^T)\boldsymbol{u} = (\boldsymbol{e}_j \otimes \boldsymbol{e}^T)\boldsymbol{u} = 1 \ (j = 1,,\ldots, m),$$
$$u_{ik}u_{jk} = u_{ki}u_{kj} = 0 \ (k = 1, \ldots, m, i \neq j),$$

respectively, where $u_{ik} = U_{ik}$. We note that the derived objective function is a quadratic form in $\boldsymbol{u} \in \mathbb{R}^{m^2}$ and the conditions on the vector variable $\boldsymbol{u} \in \mathbb{R}_+^{m^2}$ consist of equality and complementarity constraints. Therefore, we can express the QAP in the form of QOP (15) by defining a matrix $\boldsymbol{A}$, a vector $\boldsymbol{b}$ and a set $\mathcal{E}$ appropriately. By construction, we know that $\sum_{i=1}^{m} \sum_{j=1}^{m} u_{ij}^2 = m$ for every feasible solution $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m)$ of the QAP. Hence, by Lemma 5.1, we can take $\rho = m + 1$ for the construction of DNN problem (9) from QOP (15).

Table 6 shows the results for the same set of QAPs as in Table 4 of [14]. For all cases except for chr18b and chr25a, the lower bound $(\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ or $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta))$ is better than the one in [14]. The lower bounds $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ were further improved as shown in the last column of Table 6, except for 4 test problems. Algorithm 3.2 consumes at most 23 iterations. The CPU time spent by Algorithm 3.2 and the techniques in Section 3.3 for solving tai35b is around 3000 seconds.

Table 7 shows numerical results for larger-sized QAPs with n up to 50. Notice that the execution time ranges from 4.3e3 to 2.8e4 seconds for QAPs with $n = 50$, and that very tight lower bounds are obtained for lipa40a, lipa40b, lipa50a and lipa50b.

6 Concluding remarks

We have proposed methods to enhance the performance of the bisection method in [14] for solving the Lagrangian-DNN relaxation of a class of QOPs (3) with nonnegative variables. The proposed techniques strengthen the bisection method by removing irregular aspects of the numerical implementation regarding the parameter ϵ that decides the quality of lower bounds. The validity of the obtained lower bound has been ensured by the theoretical result of Lemma 3.1, and the bisection method has been accelerated. In addition, the results from the accelerated bisection method have further been improved

Table 6: The results on QAPs from QAPLIB [13]. $\epsilon = \sqrt{10}$e-12, $\delta = 1.0$e-6 and $u_0 = y_0^0 = $ Opt.val $+ 0.5 \times \max\{|\text{Opt.val}|, 1\}$ were used. See (14) for the definition of $\tilde{\lambda}^*$. Numerical results on the same set of QAP instances were reported in Table 4 of [14]. The asterisk * in chr18b* and chr25a* indicates that both lower bounds are slightly worse than the one obtained in [14], but for all other problems, at least one of them is better than the lower bound obtained by [14]. † denotes SDPT3 termination code.

Problem	$\tilde{\lambda}^*$	Opt.val	Lower bounds	
		(or the best	Algorithm 3.2 $\ell(\tilde{\lambda}^*, \epsilon, \delta)$	+pdipm $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$
		known obj.val	(#iter, CPU)	(SDPT3 CPU)
bur26a	1.39e5	+5.426670e6	+5.425904e6(22,8.3e2)	< +5.426016e6(1.6e1)†-5
bur26b	1.39e5	+3.817852e6	+3.817102e6(20,5.2e2)	< +3.817177e6(1.9e1)†-5
bur26c	1.39e5	+5.426795e6	+5.425551e6(20,5.2e2)	< +5.425677e6(2.0e1)†-5
bur26d	7.20e5	+3.821225e6	+3.819787e6(17,3.6e2)	> -8.451798e9(1.2e2)†-6
bur26e	1.39e5	+5.386879e6	+5.386490e6(21,5.8e2)	< +5.386603e6(1.8e1)†-5
chr15a	3.73e6	+9.896000e3	+9.895994e3(13,1.6e1)	< +9.895995e3(1.7e0)†-5
chr15b	1.93e7	+7.990000e3	+7.989994e3(13,1.4e1)	> -1.383629e+10(9.8e-01)†-7
chr15c	1.93e7	+9.504000e3	+9.503991e3(9,1.1e1)	> -1.226575e+10(1.4e0)
chr18a	3.73e6	+1.109800e4	+1.109797e4(11,5.5e1)	< +1.109797e4(3.6e0)†-5
chr18b*	1.00e3	+1.534000e3	+1.531530e3(14,7.1e1)	< +1.532315e3(2.8e0)
chr20a	7.20e5	+2.192000e3	+2.191865e3(15,1.2e2)	< +2.191883e3(5.6e0)†-5
chr20b	1.93e7	+2.298000e3	+2.297999e3(18,9.7e1)	> -1.905310e9(2.5e0)†-7
chr20c	7.20e5	+1.414200e4	+1.414185e4(11,4.4e1)	< +1.414186e4(6.3e0)†-5
chr22a	3.73e6	+6.156000e3	+6.155996e3(15,6.6e1)	< +6.155997e3(1.3e1)†-5
chr22b	1.39e5	+6.194000e3	+6.193929e3(15,1.7e2)	< +6.193960e3(9.7e0)†-5
chr25a*	1.39e5	+3.796000e3	+3.795839e3(17,2.0e2)	< +3.795896e3(1.6e1)†-5
nug20	1.39e5	+2.570000e3	+2.506036e3(20,1.3e2)	< +2.506095e3(4.9e0)†-5
nug25	2.68e4	+3.744000e3	+3.625001e3(18,4.2e2)	< +3.625437e3(1.3e1)†-5
nug30	2.68e4	+6.124000e3	+5.948060e3(23,1.1e3)	< +5.948817e3(3.0e1)†-5
tai30a	2.68e4	+1.818146e6	+1.706514e6(22,7.9e2)	< +1.706809e6(3.7e1)
tai30b	1.00e3	+6.371171e8	+5.978000e8(15,1.2e3)	< +5.984902e8(4.3e1)†-5
tai35a	2.68e4	+2.422002e6	+2.216156e6(23,1.9e3)	< +2.216553e6(8.2e1)†-5
tai35b	2.68e4	+2.833154e8	+2.695152e8(17,2.9e3)	< +2.695305e8(1.0e2)†-5

Table 7: The results on QAPs with size $40 \sim 50$ from QAPLIB [13]. $\epsilon = \sqrt{10}\text{e-}12$, $\delta = 1.0\text{e-}6$ and $u_0 = y_0^0 = \text{Opt.val} + 0.5 \times \max\{|\text{Opt.val}|, 1\}$ were used. See (14) for the definition of $\tilde{\lambda}^*$. † means SDPT3 termination code.

Problem	$\tilde{\lambda}^*$	Opt.val or the best known obj.val	Lower bounds	
			Algorithm 3.2 $\underline{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ (#iter, CPU)	+pdipm $\tilde{\ell}(\tilde{\lambda}^*, \epsilon, \delta)$ (SDPT3 CPU)
lipa40a	1.39e5	+3.153800e4	+3.153613e4(19,5.8e3)	< +3.153797e4(2.2e2)†-5
lipa40b	1.00e8	+4.765810e5	+4.765808e5(14,2.0e3)	> -6.981097e9(3.6e2)†-3
lipa50a	1.93e7	+6.209300e4	+6.209290e4(20,2.8e4)	> -1.903467e8(3.9e2)†-5
lipa50b	1.00e8	+1.210244e6	+1.210244e6(8,4.3e3)	> -1.430566e+10(5.1e2)†-5
sko42	2.68e4	+1.581200e4	+1.533057e4(20,1.0e4)	< +1.533305e4(2.1e2)†-5
sko49	2.68e4	+2.338600e4	+2.264641e4(21,2.1e4)	< +2.265046e4(5.8e2)†-5
tai40a	3.73e6	+3.139370e6	+2.843105e6(19,3.7e3)	> -3.543538e+10(1.7e2)†-5
tai40b	2.68e4	+6.372509e8	+6.087160e8(22,8.2e3)	< +6.087554e8(1.8e2)
tai50a	5.18e3	+4.938796e6	+4.385967e6(20,1.4e4)	< +4.390681e6(5.7e2)†-5
tai50b	2.68e4	+4.588215e8	+4.310475e8(21,3.0e4)	< +4.310796e8(7.3e2)†-5
tho40	2.68e4	+2.405160e5	+2.264708e5(21,5.7e3)	< +2.264956e5(2.0e2)
wil50	2.68e4	+4.881600e4	+4.810985e4(24,2.8e4)	< +4.812159e4(7.5e2)

by solving the SDP problem (12) induced from the Lagrangian-DNN relaxation (10) of the QOP (3) using a primal-dual interior-point method. The computational results in Section 5 demonstrate that the proposed methods work effectively on the test problems, including large-size QAPs, for instance, QAPs with $n = 50$, which are known to be difficult to solve.

For the computational efficiency, the techniques in [3] for exploiting the sparsity of QOPs can be incorporated into the method. While it is expected to reduce the computational time for solving the Lagrangian-DNN relaxation, the quality of the lower bounds may be affected.

If we replace $\mathbb{N}^n$ by a closed convex cone $\mathbb{K} \subset \mathbb{S}^n$ in (5), (6), (7) and (8), we have a primal-dual pair of general conic optimization problems and their Lagrangian relaxation in $\mathbb{S}^n$, respectively, where $(\mathbb{K} + \mathbb{S}_+^n)^* = \mathbb{K}^* + \mathbb{S}_+^n$ is assumed. Then, the idea of the methods presented in this paper can be applied in a straightforward manner to such a problem. In particular, consider an SDP

$$\zeta^p = \inf \left\{ \langle \boldsymbol{Q}, \boldsymbol{U} \rangle \mid \begin{array}{l} \boldsymbol{U} \in \mathbb{S}_+^m, \ \langle \boldsymbol{H}, \boldsymbol{U} \rangle = 0, \\ \langle \boldsymbol{A}_i, \boldsymbol{U} \rangle = b_i \ (i = 1, 2, \ldots, k) \end{array} \right\},$$

where $\boldsymbol{Q} \in \mathbb{S}^m$, $\boldsymbol{H} \in \mathbb{S}_+^m$, $\boldsymbol{A}_i \in \mathbb{S}^m$ and $b_i \in \mathbb{R}$ $(i = 1, 2, \ldots, k)$. If we take $\boldsymbol{H} = \boldsymbol{O}$, then the resulting SDP coincides with an equality standard form of SDP. Let $n = m + 1$. Define $\boldsymbol{Q}^0 \in \mathbb{S}_+^n$ and $\boldsymbol{H}^0 \in \mathbb{S}_+^n$ by (16), and let

$$\boldsymbol{H}^1 = \begin{pmatrix} \boldsymbol{H} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{pmatrix} \in \mathbb{S}^n,$$

$$\mathbb{L} = \left\{ \boldsymbol{X} = \begin{pmatrix} \boldsymbol{U} & \boldsymbol{0} \\ \boldsymbol{0}^T & x \end{pmatrix} \in \mathbb{S}^n : \langle \boldsymbol{A}_i, \boldsymbol{U} \rangle - b_i x = 0 \ (i = 1, 2, \ldots, k) \right\}.$$

We note that $\mathbb{L}$ forms a linear subspace of $\mathbb{S}^n$. Then we can rewrite the SDP as the conic optimization problem:

$$\zeta^p \;=\; \inf\left\{\langle \boldsymbol{Q}^0,\, \boldsymbol{X}\rangle \;\mid\; \boldsymbol{X} \in \mathbb{L} \cap \mathbb{S}^n_+,\; \langle \boldsymbol{H}^0,\, \boldsymbol{X}\rangle = 1,\; \langle \boldsymbol{H}^1,\, \boldsymbol{X}\rangle = 0\right\}. \qquad (19)$$

In [4], Arima, Kim, Kojima Toh formulated a hierarchy of SDP relaxations of polynomial optimization problems, a variation of Lasserre's hierarchy in [15], in the form of (19). It will be an interesting future subject to develop an efficient numerical method for solving the conic optimization problem based on this idea.

References

[1] N. Arima, S. Kim, and M. Kojima. A quadratically constrained quadratic optimization model for completely positive cone programming. *SIAM J. Optim.*, 23(4):2320–2340, 2013.

[2] N. Arima, S. Kim, and M. Kojima. Simplified copositive and lagrangian relaxations for linearly constrained quadratic optimization problems in continuous and binary variables. *Pacific J. of Optim.*, 10:437–451, 2014.

[3] N. Arima, S. Kim, M. Kojima, and K. C. Toh. Lagrangian-conic relaxations, part I: A unified framework and its applications to quadratic optimization problems. Research report B-475, Tokyo Institute of Technology, Department of Mathematical and Computing Sciences, Oh-Okayama, Meguro-ku, Tokyo 152-8552, January 2014.

[4] N. Arima, S. Kim, M. Kojima, and K. C. Toh. Lagrangian-conic relaxations, part II: Applications to polynomial optimization problems. Research report B-476, Tokyo Institute of Technology, Department of Mathematical and Computing Sciences, Oh-Okayama, Meguro-ku, Tokyo 152-8552, January 2014.

[5] A. Beck and Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2:183–202, 2009.

[6] B. Borcher. CSDP, a c library for semidefinite programming. *Optim. Methods and Softw.*, 11:613–623, 1999.

[7] S. Burer. On the copositive representation of binary and continuous non-convex quadratic programs. *Math. Program.*, 120:479–495, 2009.

[8] S. Burer. Optimizatin a polyhedral-semidefinite relaxation of completely positive programs. *Math. Prog. Comp.*, 2:1–19, 2010.

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

[10] M. Fazel, T. K. Pong, D. F. Sun, and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM J. Matrix Anal. Appl.*, 34:946–977, 2013.

[11] K. Fujisawa, M. Fukuda, M. Kobayashi, M. Kojima, K. Nakata, M. Nakata, and Yamashita. M. SDPA (SemiDefinite Programming Algorithm) user's manual – Version 7.0.5. Research report B-448, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152-8552, Japan, 2008.

[12] D. Ge and Y. Ye. On doubly positive semidefinite programming relaxations. http://www.optimization-online.org/DB_HTML/2010/08/2709.html, August 2010.

[13] P. Hahn and M. Anjos. QAPLIB – a quadratic assignment problem library. http://www.seas.upenn.edu/qaplib.

[14] S. Kim, M. Kojima, and K. C. Toh. A lagrangian-dnn relaxation: a fast method for computing tight lower bounds for a class of quadratic optimization problems. *Math. Program.*, 156:161–187, 2016.

[15] J. B. Lasserre. Global optimization with polynomials and the problems of moments. *SIAM J. Optim.*, 11:796–817, 2001.

[16] J. J. Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires. *C. R. Acad. Sci.*, 255(238-240), 1962.

[17] K. G. Murty and S. N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. *Math. Program.*, 39:117–129, 1987.

[18] N. Sloane. Challenge problems: Independent sets in graphs.

[19] J. F. Sturm. SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods and Softw.*, 11&12:625–653, 1999.

[20] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95:189–217, 2003.

[21] A. Wiegele. Biq mac library. http://www.biqmac.uni-klu.ac.at/biqmaclib.html, 2007.

[22] L. Q. Yang, D. F. Sun, and K. C. Toh. SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Prog. Comp.*, 7:331–366, 2015.

[23] A. Yoshise and Y. Matsukawa. On optimization over the doubly nonnegative cone. IEEE Multi-conference on Systems and Control, 2010.