

A Newton-bracketing method for a simple conic optimization problem

Sunyoung Kim*, Masakazu Kojima†, Kim-Chuan Toh‡

May 30, 2019

Abstract

For the Lagrangian-DNN relaxation of quadratic optimization problems (QOPs), we propose a Newton-bracketing method to improve the performance of the bisection-projection method implemented in BBCPOP [to appear in ACM Tran. Softw., 2019]. The relaxation problem is converted into the problem of finding the largest zero y^* of a continuously differentiable (except at y^*) convex function $g : \mathbb{R} \rightarrow \mathbb{R}$ such that $g(y) = 0$ if $y \leq y^*$ and $g(y) > 0$ otherwise. In theory, the method generates lower and upper bounds of y^* both converging to y^* . Their convergence is quadratic if the right derivative of g at y^* is positive. Accurate computation of $g'(y)$ is necessary for the robustness of the method, but it is difficult to achieve in practice. As an alternative, we present a secant-bracketing method. We demonstrate that the method improves the quality of the lower bounds obtained by BBCPOP and SDPNAL+ for binary QOP instances from BIQMAC. Moreover, new lower bounds for the unknown optimal values of large scale QAP instances from QAPLIB are reported.

Key words. Nonconvex quadratic optimization problems, conic relaxations, robust numerical algorithms, Newton-bracketing method, secant-bracketing method for generating valid bounds.

AMS Classification. 90C20, 90C22, 90C25,

1 Introduction

We consider a class of (nonconvex) quadratic optimization problems (QOPs) with linear and complementarity constraints in nonnegative continuous variables and binary vari-

*Department of Mathematics, Ewha W. University, 52 Ewhayeodae-gil, Sudaemoon-gu, Seoul 120-750, Korea (skim@ewha.ac.kr). The research was supported by NRF 2017-R1A2B2005119.

†Department of Industrial and Systems Engineering, Chuo University, Tokyo 192-0393, Japan (kojima@is.titech.ac.jp). This research was supported by Grant-in-Aid for Scientific Research (A) 19H00808

‡Department of Mathematics, and Institute of Operations Research and Analytics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg). This research is supported in part by the Ministry of Education, Singapore, Academic Research Fund (Grant number: R-146-000-257-112).

ables. We are particularly interested in efficient and robust numerical algorithms for solving the class of QOPs. This class includes many important combinatorial optimization problems such as binary integer QOPs, max-cut problems, maximum stable set problems, quadratic multi-knapsack problems and quadratic assignment problems. Solving a QOP in the class is known to be NP-hard in general, and various (convex) conic optimization problem relaxations including popular semidefinite programming (SDP) relaxations have been studied by many researchers from both theoretical and practical perspectives. See [1, 8, 18, 19, 21].

One of notable developments in the theory of conic relaxations for QOPs is Burer’s work in [7], where a completely positive programming (CPP) reformulation of a QOP in the class was proposed under an additional assumption on the feasible region of the QOP. This reformulation is very powerful in the sense that it attains the exact optimal value of the QOP. But unfortunately the CPP relaxation problem is numerically intractable.

Our focus in this paper is on tractable numerical methods. In particular, we are concerned with a Lagrangian doubly nonnegative (DNN) relaxation, which was proposed by Kim, Kojima and Toh [13] as a numerically tractable relaxation of the simplified Lagrangian-CPP reformulation [2] of a QOP in the class. The distinctive feature of their Lagrangian-DNN relaxation problem is the representation of the constraint set, which consists of a single linear equality constraint in a DNN variable matrix. They further reduced the optimal value of the dual of the relaxation problem to the largest zero y^* of an equation $g(y) = 0$ in a single variable y . Here $g : \mathbb{R} \rightarrow \mathbb{R}$ satisfies the following properties: (a) g is convex on \mathbb{R} . (b) $g(y) = 0$ if $y \leq y^*$ and $g(y) > 0$ otherwise. (c) Although g is not explicitly represented, the function value $g(y)$ at each $y \in \mathbb{R}$ can be computed through the metric projection onto the DNN cone. See Figure 1 (i). The properties (a) and (b) naturally lead to the bisection algorithm for approximately computing y^* . Step 0: Choose lb and ub such that $\text{lb} < y^* < \text{ub}$ and a sufficiently small positive $\epsilon > 0$. Step 1: Let $y = (\text{lb} + \text{ub})/2$. Step 2: If $g(y) < \epsilon$ then $\text{lb} = y$. Otherwise $\text{ub} = y$. Go to Step 1. Based on this idea, they proposed the bisection-projection (BP) method for approximating the dual optimal value y^* of the Lagrangian-DNN relaxation problem, and showed through numerical results that their method could efficiently compute high quality lower bounds for the optimal values of various QOP instances in the class.

Their simplified Lagrangian-DNN relaxation model and the BP method have been studied further to handle polynomial optimization problems (POPs) with binary, box and complementarity (BBC constraints), and extended to the Lagrangian conic optimization problem relaxation model in [3, 4, 5, 14], where the DNN cone is replaced with a more general convex cone. Recently, Ito, Kim, Kojima, Takeda and Toh [11] released a software package BBCPOP for solving QOPs and POPs with BBC constraints. It was demonstrated in [11] that BBCPOP can efficiently compute high quality lower bounds for optimal values of large scale QOPs and POPs with BBC constraints.

The main purpose of this paper is to improve the performance of BBCPOP by incorporating the 1-dimensional Newton method for approximating the largest zeros y^* of the equation $g(y) = 0$. See Figure 1 (ii). This idea was originally presented in [4], which showed that (d) $g : (y^*, \infty) \rightarrow \mathbb{R}$ is continuously differentiable, in addition to (a), (b) and (c) mentioned above. However, neither the explicit algebraic representation of the derivative $g'(y)$ at $y \in \mathbb{R}$ nor computing its exact value is possible, although the deriva-

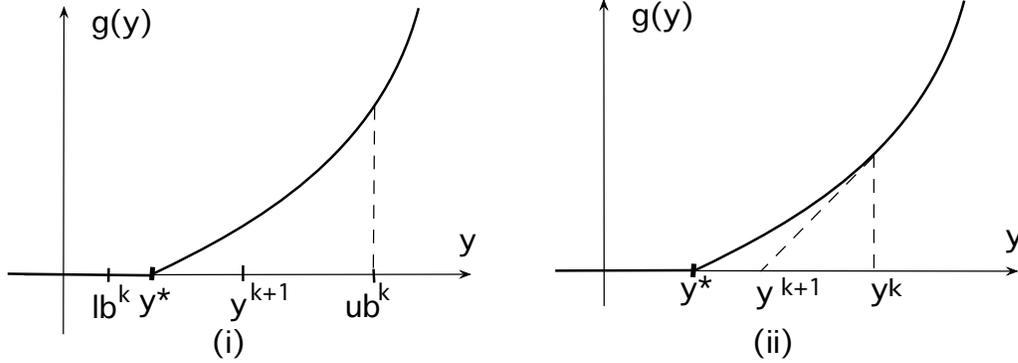


Figure 1: (i) The function $g : \mathbb{R} \rightarrow \mathbb{R}$ and the bisection algorithm. (ii) The Newton iteration.

tive $g'(y)$ at $y \in \mathbb{R}$ can be approximately computed as a by-product of the approximation of $g(y)$ through the metric projection onto the DNN cone. Due to this main difficulty, the 1-dimensional Newton was not incorporated in BBCPOP.

Here we propose a Newton-bracketing method for the conic optimization problem (COP) converted from a given QOP using the technique proposed in [4, 13]. More precisely, we pay attention to the fact that the sequence $\{y^k\}$ generated by the Newton iteration $y^{k+1} = y^k - g(y^k)/g'(y^k)$ ($k = 0, 1, \dots$), starting from y^0 satisfying $g(y^0) > 0$, monotonically converges to y^* from the above under the assumption that $g(y^k)$ and $g'(y^k)$ are exact. By itself, however, it does not generate any lower bound for y^* . By employing the technique proposed in [3], we generate a valid lower bound $\tilde{\text{lb}}^k$ at any iterate y^k and the upper bound is updated by y^{k+1} . Thus the Newton-bracketing method (Algorithm 3.2) proposed in this paper generates both upper bound y^k and lower bound $\tilde{\text{lb}}^k$ for y^* at each iteration, and the upper and lower bounds converge to y^* monotonically. Furthermore, we show that their convergence is locally quadratic if the right derivative of g at y^* is positive (See Theorem 3.3).

Despite the nice theoretical properties of the Newton-bracketing method mentioned above, its implementation can be very expensive computationally in practice since the accurate computation of the derivative $g'(y^k)$ is difficult. To avoid this difficulty, we present the damped secant-bracketing method (Algorithm 4.1) by replacing the derivative by the secant $(g(y^k) - g(y^{k-1})) / (y^k - y^{k-1})$. The proposed secant-bracketing method is implementable as the accurate computation of the function value is more manageable than the accurate evaluation of the derivative. We present some preliminary numerical results on the method applied to binary QOP instances from [20] and quadratic assignment problem (QAP) instances from [9]. We observe that the lower bounds obtained for the optimal values of the binary QOP instances by the new method are tighter than those by BBCPOP and SDPNAL+. Also new lower bounds obtained by the method for large scale QAP instances (whose optimal values are not known) are reported.

The paper is organized as follows. In Section 2.1, we introduce a primal-dual pair of conic optimization problems, COPs (1) and (2), which serve as the aforementioned DNN relaxation of a QOP in the class and its dual. In Section 2.2, we present some fundamental

properties (Lemma 2.1) of their Lagrangian relaxations described as COPs (3) and (4). In Section 2.3, the BP method is described. We present the Newton-bracketing method in Section 3, the secant-bracketing method in Section 4, and how a QOP in the class can be reduced to COPs (1) and (2). Section 5. Some preliminary numerical results on the secant-bracketing method applied to binary QOP instances from [20] and large scale QAP instances from [9] are given in Section 6. We conclude in Section 7.

2 Preliminaries

2.1 A primal-dual pair of simple conic optimization problems

Let \mathbb{R} denote the set of real numbers and \mathbb{R}_+ the set of nonnegative real numbers. We use the following notation and symbols throughout the paper.

$$\begin{aligned} \mathbb{V} &= \text{a finite dimensional vector space endowed with an inner product} \\ &\quad \langle \mathbf{X}, \mathbf{Y} \rangle \text{ and a norm } \|\mathbf{X}\| = \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle} \text{ for every } \mathbf{X}, \mathbf{Y} \in \mathbb{V}, \\ \mathbb{K}_i &= \text{a nonempty closed convex cone in } \mathbb{V} \text{ (} i = 1, 2\text{)}, \\ \mathbb{K} &= \mathbb{K}_1 \cap \mathbb{K}_2. \end{aligned}$$

For every nonempty closed convex cone \mathbb{J} , its dual is denoted by \mathbb{J}^* , *i.e.*, $\mathbb{J}^* = \{\mathbf{Y} \in \mathbb{V} : \langle \mathbf{X}, \mathbf{Y} \rangle \geq 0 \text{ for every } \mathbf{X} \in \mathbb{J}\}$. In general, \mathbb{K}^* coincides with the closure of $\mathbb{K}_1^* + \mathbb{K}_2^*$. Throughout the paper, we assume that $\mathbb{K}^* = \mathbb{K}_1^* + \mathbb{K}_2^*$.

Let $\mathbf{H}^0, \mathbf{H}^1 \in \mathbb{V}$. We consider the following pair of primal-dual COPs.

$$\eta^p = \inf \{ \langle \mathbf{Q}^0, \mathbf{X} \rangle : \mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2, \langle \mathbf{H}^0, \mathbf{X} \rangle = 1, \langle \mathbf{H}^1, \mathbf{X} \rangle = 0 \}, \quad (1)$$

$$\eta^d = \sup \{ y_0 : \mathbf{Q}^0 - \mathbf{H}^0 y_0 + \mathbf{H}^1 y_1 = \mathbf{Y}_1 + \mathbf{Y}_2, \mathbf{Y}_1 \in \mathbb{K}_1^*, \mathbf{Y}_2 \in \mathbb{K}_2^* \}. \quad (2)$$

We assume the following condition.

Condition (I) COP (1) is feasible, and $\mathbf{H}^0, \mathbf{H}^1 \in \mathbb{K}_1^* + \mathbb{K}_2^*$.

2.2 A Lagrangian relaxation of COP (1) and its dual

If the Lagrangian relaxation is applied to COP (1), the following COP and its dual are obtained.

$$\eta_\lambda^p = \inf \{ \langle \mathbf{Q}^0 + \lambda \mathbf{H}^1, \mathbf{X} \rangle : \mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2, \langle \mathbf{H}^0, \mathbf{X} \rangle = 1 \}, \quad (3)$$

$$\eta_\lambda^d = \sup \{ y_0 : \mathbf{Q}^0 + \lambda \mathbf{H}^1 - \mathbf{H}^0 y_0 = \mathbf{Y}_1 + \mathbf{Y}_2, \mathbf{Y}_1 \in \mathbb{K}_1^*, \mathbf{Y}_2 \in \mathbb{K}_2^* \}, \quad (4)$$

where $\lambda \in \mathbb{R}$ denotes the Lagrangian multiplier for the homogeneous equality $\langle \mathbf{H}^1, \mathbf{X} \rangle = 0$ in (1). Since $\mathbf{H}^1 \in \mathbb{K}_1^* + \mathbb{K}_2^*$ by Condition (I), the term $\lambda \langle \mathbf{H}^1, \mathbf{X} \rangle$ added to the objective function is nonnegative for every $\mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2$ and $\lambda \geq 0$. As a result, the term serves as a penalty term for the violation of $\langle \mathbf{H}^1, \mathbf{X} \rangle = 0$ such that if $\mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2$ and $\langle \mathbf{H}^1, \mathbf{X} \rangle \neq 0$ then $\lambda \langle \mathbf{H}^1, \mathbf{X} \rangle \rightarrow \infty$ as $\lambda \rightarrow \infty$. Furthermore, the following lemma holds.

Lemma 2.1. [4, Lemmas 2.3 and 2.5] Suppose that Conditions (I) is satisfied. Then, the following assertions hold.

- (i) $\eta_\lambda^d = \eta_\lambda^p$ for every $\lambda \in \mathbb{R}$. Moreover, if η_λ^p is finite, then (4) has an optimal solution with the objective value $\eta_\lambda^d = \eta_\lambda^p$.
- (ii) $(\eta_\lambda^d = \eta_\lambda^p) \uparrow = \eta^d$. Here $\uparrow = \eta^d$ means “increases monotonically and converges to η^d as $\lambda \rightarrow \infty$ ”.
- (iii) Assume in addition that $\{\mathbf{X} \in F : \langle \mathbf{Q}^0, \mathbf{X} \rangle \leq \bar{\eta}\}$ is bounded for some $\bar{\eta}$. Then $\eta^d = \eta^p$.

In the remaining of the paper, we impose the following condition on the space \mathbb{V} and the cone $\mathbb{K} = \mathbb{K}_1 \cap \mathbb{K}_2$.

Condition (II) \mathbb{V} is the Cartesian product of symmetric matrix spaces, \mathbb{K}_1 the cone of consisting of the positive semidefinite matrices in \mathbb{V} , and \mathbb{K}_2 a closed convex cone in \mathbb{V} such that $(\mathbb{K}_1 \cap \mathbb{K}_2)^* = \mathbb{K}_1^* + \mathbb{K}_2^*$.

In this case, we know that $\mathbb{K}_1^* = \mathbb{K}_1$. By Lemma 2.1, we can obtain an accurate lower bound η_λ^d for the optimal value η^d of COP (2) by solving COP (4) for a sufficiently large positive λ . For simplicity of notation, we fix λ to be a sufficiently large positive number, and we rewrite the primal-dual pair of COPs (3) and (4) as

$$\varphi^* = \inf \{ \langle \mathbf{Q}, \mathbf{X} \rangle : \mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2, \langle \mathbf{H}, \mathbf{X} \rangle = 1 \}, \quad (5)$$

$$y^* = \sup \{ y : \mathbf{Q} - \mathbf{H}y = \mathbf{Y}_1 + \mathbf{Y}_2, \mathbf{Y}_1 \in \mathbb{K}_1^*, \mathbf{Y}_2 \in \mathbb{K}_2^* \}. \quad (6)$$

To solve COP (6), we present the bisection-projection (BP) method (Algorithm 2.3) in Section 2.3, the Newton-bracketing method (Algorithm 3.2) in Section 3 and the secant-bracketing method (Algorithm 4.1) in Section 4.

2.3 The bisection-projection method for COP (6)

We describe the bisection-projection (BP) method [3, 4, 11, 13, 14] for COP (6). For every $y \in \mathbb{R}$, define $\mathbf{G}(y) = \mathbf{Q} - \mathbf{H}y$ and

$$\begin{aligned} g(y) &= \min \{ \|\mathbf{G}(y) - (\mathbf{Y}_1 + \mathbf{Y}_2)\| : \mathbf{Y}_1 \in \mathbb{K}_1^*, \mathbf{Y}_2 \in \mathbb{K}_2^* \} \\ &= \|\mathbf{G}(y) - (\hat{\mathbf{Y}}_1(y) + \hat{\mathbf{Y}}_2(y))\|. \end{aligned} \quad (7)$$

Here $(\hat{\mathbf{Y}}_1(y), \hat{\mathbf{Y}}_2(y)) \in \mathbb{K}_1^* \times \mathbb{K}_2^*$ denotes an optimal solution. For every $y \in \mathbb{R}$, we have that $g(y) \geq 0$, and

$$\begin{aligned} g(y) = 0 &\Leftrightarrow \mathbf{G}(y) = \mathbf{Q} - \mathbf{H}y \in \mathbb{K}_1^* + \mathbb{K}_2^* \\ &\Leftrightarrow y \text{ is a feasible solution of COP (6)}. \end{aligned}$$

Thus, COP (6) can be rewritten as

$$y^* = \sup \{ y \in \mathbb{R} : g(y) = 0 \}. \quad (8)$$

From $\mathbf{H} \in \mathbb{K}_1^* + \mathbb{K}_2^*$ by Condition (I), we see that $g(y) = 0$ (i.e., $\mathbf{Q} - \mathbf{H}y \in \mathbb{K}_1^* + \mathbb{K}_2^*$) if $y \leq y^*$ and $g(y) > 0$ (i.e., $\mathbf{Q} - \mathbf{H}y \notin \mathbb{K}_1^* + \mathbb{K}_2^*$) otherwise.

In addition to Conditions (I) and (II), we assume the following condition in the subsequent discussion.

Condition (III) (Condition (D) of [3]) A positive number ρ such that $\langle \mathbf{I}, \mathbf{X} \rangle \leq \rho$ for all feasible solutions \mathbf{X} of (6) is known, where \mathbf{I} denotes the identity matrix in \mathbb{V} . Then, the primal-dual pair of COPs (5) and (6) are equivalent to

$$\varphi^* = \min \{ \langle \mathbf{Q}, \mathbf{X} \rangle : \langle \mathbf{H}, \mathbf{X} \rangle = 1, \langle \mathbf{I}, \mathbf{X} \rangle \leq \rho, \mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2 \}, \quad (9)$$

and its dual

$$y^* = \max \{ y + \rho t : \mathbf{G}(y) - \mathbf{I}t - \mathbf{Y}_2 = \mathbf{Y}_1 \in \mathbb{K}_1^*, \mathbf{Y}_2 \in \mathbb{K}_2^*, t \leq 0 \}, \quad (10)$$

respectively. For every $y \in \mathbb{R}$ and $\mathbf{Y}_2 \in \mathbb{K}_2^*$, let

$$\begin{cases} \tilde{t}(y, \mathbf{Y}_2) = \min\{0, \lambda_{\min}(\mathbf{G}(y) - \mathbf{Y}_2)\}, \\ \tilde{\mathbf{Y}}_1(y, \mathbf{Y}_2) = \mathbf{G}(y) - \mathbf{I}\tilde{t}(y, \mathbf{Y}_2) - \mathbf{Y}_2, \end{cases} \quad (11)$$

where $\lambda_{\min}(\mathbf{A})$ denotes the minimum eigenvalue of $\mathbf{A} \in \mathbb{V}$.

Lemma 2.2. (Lemma 3.1 of [3]) Let $y \in \mathbb{R}$ and $\mathbf{Y}_2 \in \mathbb{K}_2^*$. Then, $y + \rho\tilde{t}(y, \mathbf{Y}_2) \leq y^*$.

Proof. By the construction (11) of $\tilde{t}(y, \mathbf{Y}_2)$ and $\tilde{\mathbf{Y}}_1(y, \mathbf{Y}_2)$, we see that

$$\mathbf{G}(y) - \mathbf{I}\tilde{t}(y, \mathbf{Y}_2) - \mathbf{Y}_2 = \tilde{\mathbf{Y}}_1(y, \mathbf{Y}_2) \in \mathbb{K}_1^*,$$

which, together with $\mathbf{Y}_2 \in \mathbb{K}_2^*$, implies that $(t, y, \mathbf{Y}_1, \mathbf{Y}_2)$ with $t = \tilde{t}(y, \mathbf{Y}_2)$ and $\mathbf{Y}_1 = \tilde{\mathbf{Y}}_1(y, \mathbf{Y}_2)$ is a feasible solution of (10) with the objective value $y + \rho\tilde{t}(y, \mathbf{Y}_2)$. Hence, the inequality $y + \rho\tilde{t}(y, \mathbf{Y}_2) \leq y^*$ follows. \square

In Algorithm 2.3 described below, we utilize Lemma 2.2 to generate a valid lower bound $\tilde{\text{lb}}$ for y^* under the assumption that the computation of the minimal eigenvalue of $\lambda_{\min}(\mathbf{A})$ is accurate for every $\mathbf{A} \in \mathbb{V}$.

Let $y \in \mathbb{R}$. Then, every optimal solution $(\mathbf{Y}_1, \mathbf{Y}_2) = (\hat{\mathbf{Y}}_1(y), \hat{\mathbf{Y}}_2(y))$ of the minimization problem (7) is characterized by the KKT condition: there exists an $\mathbf{X} = \hat{\mathbf{X}}(y)$ such that

$$\mathbf{G}(y) = \mathbf{Y}_1 + \mathbf{Y}_2 - \mathbf{X}, \mathbf{Y}_1 \in \mathbb{K}_1^*, \mathbf{Y}_2 \in \mathbb{K}_2^*, \quad (12)$$

$$\mathbf{X} \in \mathbb{K}_1 \cap \mathbb{K}_2, \langle \mathbf{X}, \mathbf{Y}_1 \rangle = 0 \text{ and } \langle \mathbf{X}, \mathbf{Y}_2 \rangle = 0. \quad (13)$$

Therefore, we obtain

$$g(y) = \|\mathbf{G}(y) - (\hat{\mathbf{Y}}_1(y) + \hat{\mathbf{Y}}_2(y))\| = \|\hat{\mathbf{X}}(y)\|. \quad (14)$$

Algorithm 2.3. (the BP method [3, Algorithm 3.2])

Step 0: Let δ and ϵ be sufficiently small positive numbers. Here $\delta > 0$ determines the length of the target interval $[\text{lb}, \text{ub}]$ and $\epsilon > 0$ is used to determines whether $g(y)$ attains 0 numerically with $\|\mathbf{X}\| < \epsilon$ in Step 3. See(14). For instance, we can use $\delta = 1.0\text{e-}4$ and $\epsilon = 1.0\text{e-}12$ in the double precision arithmetic. Choose $\tilde{\text{lb}}$, lb and ub such that $\tilde{\text{lb}} = \text{lb} < y^* < \text{ub}$, where $\tilde{\text{lb}} = \text{lb}$ can be $-\infty$.

Step 1: If $\text{ub} - \text{lb} < \delta$, then output $\tilde{\text{lb}}$ as a valid lower bound for y^* and $[\text{lb}, \text{ub}]$ as an interval which expects to contain y^* .

Step 2: Let $y = (\text{lb} + \text{ub})/2$.

Step 3: Compute $(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2)$ which satisfies accurately (12) and approximately (13) by applying the accelerated proximal gradient (APG) method [6] to the problem (7). Let $\tilde{\text{lb}} = \max\{\tilde{\text{lb}}, y + \rho\tilde{t}(y, \mathbf{Y}_2)\}$. Let $\text{lb} = y$ if $\|\mathbf{X}\| < \epsilon$, and $\text{ub} = y$ otherwise (*i.e.*, $\|\mathbf{X}\| \geq \epsilon$). Go to Step 1.

At Step 3, $\mathbf{Y}_2 \in \mathbb{K}_2^*$ is obtained. As a result, Lemma 2.2 guarantees that $\tilde{\text{lb}}$ is a valid lower bound for y^* regardless of the choice of a positive ϵ . On the other hand, $y^* \in [\text{lb}, \text{ub}]$ is not guaranteed because of the numerical error.

The APG method [6] used in Step 3 is a first order iteration method that employs the metric projection Π_i^* onto \mathbb{K}_i^* ($i = 1, 2$). The BP method combined with the APG method has been implemented recently as a software package BBCPOP [10]. It employs the following error criterion

$$\kappa(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) = \max \left\{ \begin{array}{l} \frac{\langle \mathbf{X}, \mathbf{Y}_1 \rangle}{1 + \|\mathbf{X}\| + \|\mathbf{Y}_1\|}, \frac{\langle \mathbf{X}, \mathbf{Y}_2 \rangle}{1 + \|\mathbf{X}\| + \|\mathbf{Y}_2\|}, \\ \frac{\|\Pi_1^*(-\mathbf{X})\|}{1 + \|\mathbf{X}\|}, \frac{\|\Pi_2^*(-\mathbf{X})\|}{1 + \|\mathbf{X}\|} \end{array} \right\} \quad (15)$$

to decide whether the iterate $(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2)$ of the APG method approximately satisfies (13). We note that (12) is maintained throughout the iterations, and that $\Pi_i^*(-\mathbf{X}) = \mathbf{X} - \Pi_i(\mathbf{X})$ ($i = 1, 2$) holds for every $\mathbf{X} \in \mathbb{V}$ by Moreau's decomposition theorem [16]. Here $\Pi_i(\mathbf{X})$ ($\Pi_i^*(\mathbf{X})$, respectively) denotes the metric projection of $\mathbf{X} \in \mathbb{V}$ onto \mathbb{K}_i (\mathbb{K}_i^* , respectively) ($i = 1, 2$). If $\|\mathbf{X}\| < \epsilon$ or $\|\mathbf{X}\| \geq \epsilon$ and $\kappa(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) < \text{tol}$, then the iteration stops, where ϵ and tol are sufficiently small positive numbers, such as $\epsilon = \text{tol} = 1.0\text{e-}12$ in the double precision arithmetic. See Sections of 2.2 and 4.2 of [11] for more details.

In the APG method, the metric projections from \mathbb{V} onto the cones \mathbb{K}_i ($i = 1, 2$) and their duals play an essential role. We have been dealing with the case where $\mathbb{K}_1 = \mathbb{K}_1^*$ is the cone of symmetric matrices in \mathbb{V} , thus the metric projection of $\mathbf{A} \in \mathbb{V}$ onto $\mathbb{K}_1 = \mathbb{K}_1^*$ is carried out efficiently and accurately via the eigenvalue decomposition of \mathbf{A} . For the efficient and accurate metric projection onto the cone \mathbb{K}_2 , we need to restrict \mathbb{K}_2 to a class of polyhedral cones onto which the accurate metric projection can be efficiently computed. Such a class of polyhedral cones were studied in [14]. The class includes the cone of nonnegative matrices in \mathbb{V} , polyhedral cones induced from SDP relaxations of binary and complementarity constraints, and their intersections. Those cones are described in Section 5 where applications of COP (1) to doubly nonnegative relaxations of combinatorial QOPs are discussed.

3 The Newton-bracketing method

We use the same notation and symbols as in Section 2, specifically $(\widehat{\mathbf{Y}}_1(y), \widehat{\mathbf{Y}}_2(y))$ denotes an optimal solution of the minimization problem (7) and $(\widehat{\mathbf{X}}(y), \widehat{\mathbf{Y}}_1(y), \widehat{\mathbf{Y}}_2(y)) \in (\mathbb{K}_1 \cap \mathbb{K}_2) \times \mathbb{K}_1^* \times \mathbb{K}_2^*$ satisfies the KKT condition (12) and (13); See also (14). To describe the 1-dimensional Newton method for computing y^* , we need the following lemma, which shows some fundamental properties of the function $g : \mathbb{R} \rightarrow \mathbb{R}_+$.

Lemma 3.1. (*[4, Lemma 4.1]*) Assume that y^* is finite.

- (i) $g : \mathbb{R} \rightarrow \mathbb{R}_+$ is continuous and convex.
- (ii) If $y > y^*$, then $\langle \mathbf{H}, \widehat{\mathbf{X}}(y) \rangle > 0$ and $\widehat{\mathbf{X}}(y)/\langle \mathbf{H}, \widehat{\mathbf{X}}(y) \rangle$ is a feasible solution of the primal COP (5).
- (iii) If $y > y^*$, then $g'(y) = dg(y)/dy = \langle \mathbf{H}, \widehat{\mathbf{X}}(y) \rangle / g(y) > 0$; hence $g : (y^*, \infty) \rightarrow \mathbb{R}$ is continuously differentiable and strictly increasing.
- (iv) Assume that $\mathbf{G}(\bar{z})$ lies in the interior of $\mathbb{K}_1^* \times \mathbb{K}_2^*$ for some \bar{z} . Then $\frac{g(y) - g(y^*)}{y - y^*}$ converges to a positive value as $y \downarrow y^*$; hence the right derivative $g'_+(y^*)$ of $g(y)$ at $y = y^*$ is positive.

Suppose that $g(\bar{y}) > 0$, i.e., $\bar{y} > y^*$ for some $\bar{y} \in \mathbb{R}$. Then the Newton iteration for computing y^* is given by

$$\begin{aligned}
\bar{y}^+ &= \bar{y} - \frac{g(\bar{y})}{g'(\bar{y})} = \bar{y} - \frac{\langle \widehat{\mathbf{X}}(\bar{y}), \widehat{\mathbf{X}}(\bar{y}) \rangle}{\langle \mathbf{H}, \widehat{\mathbf{X}}(\bar{y}) \rangle} \\
&= \bar{y} - \frac{\langle \widehat{\mathbf{Y}}_1(\bar{y} + \widehat{\mathbf{Y}}_2(\bar{y})) - \mathbf{G}(\bar{y}), \widehat{\mathbf{X}}(\bar{y}) \rangle}{\langle \mathbf{H}, \widehat{\mathbf{X}}(\bar{y}) \rangle} \quad (\text{since } \widehat{\mathbf{X}}(\bar{y}) = \widehat{\mathbf{Y}}_1(\bar{y}) + \widehat{\mathbf{Y}}_2(\bar{y}) - \mathbf{G}(\bar{y}) \text{ by (12)}) \\
&= \bar{y} + \frac{\langle \mathbf{G}(\bar{y}), \widehat{\mathbf{X}}(\bar{y}) \rangle}{\langle \mathbf{H}, \widehat{\mathbf{X}}(\bar{y}) \rangle} \quad (\text{since } \langle \widehat{\mathbf{X}}(\bar{y}), \widehat{\mathbf{Y}}_1(\bar{y}) \rangle = \langle \widehat{\mathbf{X}}(\bar{y}), \widehat{\mathbf{Y}}_2(\bar{y}) \rangle = 0 \text{ by (13)}) \\
&= \bar{y} + \frac{\langle \mathbf{Q} - \mathbf{H}\bar{y}, \widehat{\mathbf{X}}(\bar{y}) \rangle}{\langle \mathbf{H}, \widehat{\mathbf{X}}(\bar{y}) \rangle} \quad (\text{by the definition of } \mathbf{G}(\bar{y})) \\
&= \langle \mathbf{Q}, \widetilde{\mathbf{X}}(\bar{y}) \rangle \geq \varphi^*,
\end{aligned}$$

where $\widetilde{\mathbf{X}}(\bar{y}) = \widehat{\mathbf{X}}(\bar{y})/\langle \mathbf{H}, \widehat{\mathbf{X}}(\bar{y}) \rangle$ denotes a feasible solution of the primal COP (5). It should be noted that the sequence $\{y^k\}$ generated by the Newton iteration from any initial iterate $y^0 > y^*$ monotonically decreases and converges to y^* by (i), (ii) and (iii) of Lemma 3.1.

Now we combine the Newton iteration with Lemma 2.2 to design Newton-bracketing method for solving COP (6).

Algorithm 3.2.

Step 0: Let $\widetilde{\text{lb}}^0 = -\infty$ and $k = 0$. Choose a $y^0 > y^*$.

Step 1: Compute $(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) \in (\mathbb{K}_1 \cap \mathbb{K}_2) \times \mathbb{K}_1^* \times \mathbb{K}_2^*$ which satisfies the KKT condition (12) and (13) with $y = y^k$; hence (14) holds with $(\widehat{\mathbf{X}}(y), \widehat{\mathbf{Y}}_1(y), \widehat{\mathbf{Y}}_2(y)) = (\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2)$ and $y = y^k$. Let $g(y^k) = \|\mathbf{X}\|$.

Step 2: (Application of Lemma 2.2 to update $\widetilde{\text{lb}}^k$) Let $\widetilde{\text{lb}}^{k+1} = \max\{\widetilde{\text{lb}}^k, y^k + \rho \tilde{t}(y^k, \mathbf{Y}_2)\}$.

Step 3: If $g(y^k) = 0$ then $y^k = y^*$ and stop the iteration.

Step 4: (Newton step to update y^k) Let $g'(y^k) = \langle \mathbf{H}, \mathbf{X} \rangle / g(y^k)$, and $y^{k+1} = y^k - g(y^k)/g'(y^k)$.

Step 5: Replace k by $k + 1$, and go to Step 1.

Theorem 3.3.

- (i) $y^* \in [\tilde{\text{lb}}^k, y^k]$ for every $k = 1, 2, \dots$, $y^k \downarrow = y^*$ and $\tilde{\text{lb}}^k \uparrow = y^*$. Here $\uparrow = y^*$ (or $\downarrow = y^*$) means “increases (or decreases) monotonically and converges to y^* as $k \rightarrow \infty$ ”
- (ii) Assume that $\mathbf{G}(\bar{z}_0)$ lies in the interior of $\mathbb{K}_1^* + \mathbb{K}_2^*$ for some \bar{z}_0 . Then the convergence of y^k to y^* as well as the convergence of $\tilde{\text{lb}}^k$ to y^* are quadratic.

To prove the theorem, we need the following lemma.

Lemma 3.4. Assume that $g(y) > 0$ for some $y \in \mathbb{R}$. Then $y - \rho g(y) \leq y + \rho \tilde{t}(y, \widehat{\mathbf{Y}}_2(y)) \leq y^*$.

Proof. By Lemma 2.2, it suffices to show that $-g(y) \leq \tilde{t}(y, \widehat{\mathbf{Y}}_2(y))$. It follows from (12) that

$$\begin{aligned} \lambda_{\min}(\mathbf{G}(y) - \widehat{\mathbf{Y}}_2(y)) &= \lambda_{\min}(\widehat{\mathbf{Y}}_1(y) - \widehat{\mathbf{X}}(y)) \\ &= \mathbf{u}^T(\widehat{\mathbf{Y}}_1(y) - \widehat{\mathbf{X}}(y))\mathbf{u} \text{ for some } \mathbf{u} \text{ with } \|\mathbf{u}\| = 1 \\ &\geq -\mathbf{u}^T \widehat{\mathbf{X}}(y)\mathbf{u} \text{ (since } \widehat{\mathbf{Y}}_1(y) \in \mathbb{K}_1^*) \\ &\geq -\lambda_{\max}(\widehat{\mathbf{X}}(y)) \\ &\geq -\|\widehat{\mathbf{X}}(y)\| = -g(y). \end{aligned}$$

Since $-g(y) < 0$, $-g(y) \leq \min\{0, \lambda_{\min}(\mathbf{G}(y) - \widehat{\mathbf{Y}}_2(y))\} = \tilde{t}(y, \widehat{\mathbf{Y}}_2(y))$ follows. □

Proof of Theorem 3.3. The assertions of (i) follows from Lemma 2.2, Lemma 3.1, and the construction of the sequences $\{\tilde{\text{lb}}^k\}$ and $\{y^k\}$ by Algorithm 3.2. To prove (ii), assume that $\mathbf{G}(\bar{z}_0)$ lies in the interior of $\mathbb{K}_1^* + \mathbb{K}_2^*$ for some \bar{z}_0 . By (iv) of Lemma 3.1, the right derivative $g'_+(y^*)$ of $g(y)$ at $y = y^*$ is positive. Define

$$h(y) = \begin{cases} g(y) & \text{if } y \geq y^*, \\ g'_+(y^*)(y - y^*) & \text{otherwise.} \end{cases}$$

Then $h : \mathbb{R} \rightarrow \mathbb{R}$ is continuously differentiable and convex, and $h'(y^*) = g'_+(y^*) > 0$. Since the application of Newton method to $h(y) = 0$ with the initial point y^0 satisfying $h(y^0) = g(y^0) > 0$ yields the same sequence as the Newton method applied to solve $g(y) = 0$, y^k converges quadratically to y^* (hence $g(y^k)$ converges to 0) as $k \rightarrow \infty$ by [17, Monotone Newton Theorem 13.3.4]. This implies the quadratic convergence of $\{g(y^k)\}$ to 0. By Lemma 3.4, $\tilde{\text{lb}}^k$ also converges quadratically to y^* . □

4 The damped secant-bracketing method

Although Algorithm 3.2 (the Newton-bracketing method) has nice theoretical properties as shown in Theorem 3.3, its accurate implementation is difficult. More precisely, the exact function value $g(y^k)$ and its derivative $g'(y^k)$ have been assumed for

each iterate y^k in Algorithm 3.2. However, when Algorithm 3.2 is implemented in practice, only approximate value $(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2)$ of $(\widehat{\mathbf{X}}(y^k), \widehat{\mathbf{Y}}_1(y^k), \widehat{\mathbf{Y}}_2(y^k))$ can be computed by the APG method. From those values, $g(y^k)$ and $g'(y^k)$ are approximated by $\|\mathbf{X}\|$ and $\langle \mathbf{H}, \mathbf{X} \rangle / \|\mathbf{X}\|$, respectively. In particular, the accurate computation of $g'(y^k) = \langle \mathbf{H}, \widehat{\mathbf{X}}(y^k) \rangle / \|\widehat{\mathbf{X}}(y^k)\|$ is difficult as the denominator $\|\widehat{\mathbf{X}}(y^k)\|$ as well as the numerator $\langle \mathbf{H}, \widehat{\mathbf{X}}(y^k) \rangle$ both converge to zero as y^k approaches to y^* . This computational issue should be carefully dealt with for the robustness of the Newton-bracketing method in practice.

To avoid such a difficulty and for the purpose of developing a practically robust method, we replace the derivative $g'(y^k)$ at the k th iterate y^k by the secant $(g(y^k) - g(y^{k-1})) / (y^k - y^{k-1})$ ($k = 1, 2, \dots$). Then the iterative formula turns out to be

$$\begin{aligned} y^{k+1} &= y^k - \alpha^k \frac{g(y^k)(y^k - y^{k-1})}{g(y^k) - g(y^{k-1})} \\ &= y^k - \alpha^k \frac{\|\widehat{\mathbf{X}}(y^k)\|(y^k - y^{k-1})}{\|\widehat{\mathbf{X}}(y^k)\| - \|\widehat{\mathbf{X}}(y^{k-1})\|} \quad (k = 1, 2, \dots). \end{aligned} \quad (16)$$

Here $\alpha^k \in (0, 1]$ is a damping factor, which is multiplied to the secant to avoid the occurrence of $y^{k+1} < y^*$ in any case, which might happen due to inaccurate computations of $g(y^{k-1})$ and $g(y^k)$. To initiate the first secant, we need to prepare two points $y^0 > y^1 > y^*$.

Algorithm 4.1. (The damped secant-bracketing method)

Step 0: Let δ and ϵ be sufficiently small positive numbers (for example, $\delta = 1.0\text{e-}4$ and $\epsilon = 1.0\text{e-}12$ in the double precision arithmetic). Choose y^0 and y^1 such that $y^* < y^1 < y^0$. Let $\tilde{\text{lb}}^0 = -\infty$ and $k = 0$.

Step 1: Compute $(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) = (\mathbf{X}^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k)$ which satisfies accurately (12) and approximately (13) with $y = y^k$ by the APG method.

Step 2: (Application of Lemma 2.2 to update $\tilde{\text{lb}}^k$) Let $\tilde{\text{lb}}^{k+1} = \max\{\tilde{\text{lb}}^k, y^k + \rho \tilde{t}(y^k, \mathbf{Y}_2^k)\}$. If $k=0$, then let $k=1$ and go to Step1.

Step 3: If $\|\mathbf{X}^k\| < \epsilon$ or $y^k - \tilde{\text{lb}}^{k+1} < \delta$, then stop the iteration, and output $\tilde{\text{lb}}^{k+1}$ as a valid lower bound for y^* and y^k as an approximation of y^* .

Step 4: (Damped secant iteration) Apply

$$y^{k+1} = y^k - \alpha^k \frac{\|\mathbf{X}^k\|(y^k - y^{k-1})}{\|\mathbf{X}^k\| - \|\mathbf{X}^{k-1}\|}$$

with some $\alpha^k \in (0, 1]$ to update y^k .

Step 5: Replace k by $k + 1$, and go to Step 1.

The use of the secant certainly mitigates the difficulty of computing the derivative $g'(y^k)$ accurately in the Newton method. Computing the secant by $(g(y^k) - g(y^{k-1})) / (y^k - y^{k-1})$ involves the computation of the denominator $y^k - y^{k-1}$, which is almost exact. To compute the numerator with high accuracy, we still need to compute the function value

$g(y^k)$ accurately at each iteration ($k = 0, 1, \dots$). The computations that are not carried out with sufficient accuracy may affect the entire iteration. In particular, if the computed secant is much smaller than the real secant or the computed value of $g(y^k)$ is much larger than its real value at the k th iteration, then the next iterate y^{k+1} could be smaller than y^* and Algorithm 4.1 cannot be continued in a consistent manner.

The stopping criteria in the APG method combined with Algorithm 4.1 should be carefully set up. We recall the discussion on the stopping criteria of the APG method combined with Algorithm 2.3 in Section 2.3. To approximate $(\widehat{\mathbf{X}}(y^k), \widehat{\mathbf{Y}}_1(y^k), \widehat{\mathbf{Y}}_2(y^k))$ with high accuracy by the APG method, we modify the stopping criteria mentioned there. We introduce two criteria which must be satisfied simultaneously for the APG method to stop the iteration. The first one is $\kappa(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2) < tol$.

For the second criterion, let $(\widetilde{\mathbf{X}}^p, \widetilde{\mathbf{Y}}_1^p, \widetilde{\mathbf{Y}}_2^p)$ denote the p th iterate of the APG method ($p = 1, 2, \dots$). We compute

$$z^p = y^k - \alpha^k \frac{\|\widetilde{\mathbf{X}}^p\| (y^k - y^{k-1})}{\|\widetilde{\mathbf{X}}^p\| - \|\mathbf{X}^{k-1}\|} \quad (p = 1, 2, \dots).$$

Here z^p becomes the next iterate y^{k+1} of Algorithm 4.1 if the APG method stops at iteration p . Through numerical experiments, we have observed that the sequence $\{z^p\}$ increased on average as p increase, and in many cases, it converged to some z^* . To check the convergence of the generated sequence numerically to some z^* , which is almost equal to z^p at the p th iteration, some variance of the subsequence $z^{q(p)+1}, z^{p(q)+2}, \dots, z^p$ and the slope of their linear interpolation are incorporated into the second criteria (for example, $q(p) = \max\{1, p - 100\}$). Our second criterion is: the variance and the slope are both sufficiently small. The details are omitted here.

The second criterion can be very strict for some problems. In fact, the APG method sometimes does not terminate with the two stopping criteria, even when p reaches a prescribed maximum number of iterations. From our numerical experience, we noticed that this usually occurs when the second criterion is too strict. For such cases, we allow Algorithm 4.1 to switch to the bisection temporarily. More precisely, we add $lb = \widetilde{lb}^0$ and $ub = y^0$ in Step 0 and insert the following steps between Steps 2 and 3.

Step 2.3: If $\|\mathbf{X}\| < \epsilon$, then let $lb = \max\{lb, y^k\}$. Otherwise let $ub = \min\{ub, y^k\}$.

If the APG method for computing $(\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2)$ at Step 1 terminates with the two stopping criteria, go to Step 3. Otherwise go to Step 2.6.

Step 2.6: Reset $y^k = (ub + lb)/2$. If $ub - lb < \delta$, then stop the iteration, and output \widetilde{lb}^{k+1} as a valid lower bound for y^* and y^k as an approximation of y^* . Otherwise go to Step 1.

This modification improves the robustness and stability of Algorithm 4.1.

5 Applications to DNN relaxations of Combinatorial Quadratic Optimization Problems

Let $\mathbf{C} \in \mathbb{S}^n$, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{\ell \times n}$, $\mathbf{b} \in \mathbb{R}^\ell$, $I_{\text{bin}} \subset \{1, \dots, n\}$ (the index set for binary variables), and $I_{\text{comp}} \subset \{(j, k) : 1 \leq j < k \leq n\}$ (the index set for pairs of complementary

variables). Consider a QOP of the following form:

$$\zeta_{\text{QOP}} = \inf \left\{ \mathbf{u}^T \mathbf{C} \mathbf{u} + 2\mathbf{c}^T \mathbf{u} : \begin{array}{l} \mathbf{u} \in \mathbb{R}_+^n, \mathbf{A} \mathbf{u} - \mathbf{b} = \mathbf{0}, \\ u_i(1 - u_i) = 0 \ (i \in I_{\text{bin}}), \\ u_j u_k = 0 \ ((j, k) \in I_{\text{comp}}) \end{array} \right\}. \quad (17)$$

We assume that the feasible region of QOP (17) is nonempty.

To derive a doubly nonnegative (DNN) relaxation for QOP (17) of the form COP (1), we introduce some notation and symbols. Let \mathbb{R}^{1+n} denote the $(1+n)$ -dimensional Euclidean space of column vectors $\mathbf{x} = (x_0, x_1, \dots, x_n)$, and $\mathbb{R}_+^{1+n} = \{\mathbf{x} \in \mathbb{R}^{1+n} : x_i \geq 0 \ (i = 0, \dots, n)\}$ (the nonnegative orthant of \mathbb{R}^{1+n}). Let \mathbb{S}^{1+n} denote the space of $(1+n) \times (1+n)$ symmetric matrices with row and column indices $i = 0, 1, \dots, n$, \mathbb{S}_+^{1+n} the cone of positive semidefinite matrices in \mathbb{S}^{1+n} , \mathbb{N}^{1+n} the cone of nonnegative matrices in \mathbb{S}^{1+n} and $\mathbb{D}^{1+n} = \mathbb{S}_+^{1+n} \cap \mathbb{N}^{1+n}$ (the *DNN cone*). By definition, $\mathbf{x} \mathbf{x}^T \in \mathbb{D}^{1+n}$ for every $\mathbf{x} = (x_0, \mathbf{u}) \in \mathbb{R}_+^{1+n}$.

Let

$$\left. \begin{aligned} \mathbb{V} &= \mathbb{S}^{1+n}, \mathbb{K}_1 = \mathbb{S}_+^{1+n}, \\ \mathbb{K}_2 &= \left\{ \mathbf{X} \in \mathbb{N}^{1+n} : \begin{array}{l} X_{0i} = X_{i0} = X_{ii} \ (i \in I_{\text{bin}}), \\ X_{jk} = X_{kj} = 0 \ ((j, k) \in I_{\text{comp}}) \end{array} \right\}, \\ \mathbb{K} &= \mathbb{K}_1 \cap \mathbb{K}_2 = \left\{ \mathbf{X} \in \mathbb{D}^{1+n} : \begin{array}{l} X_{0i} = X_{i0} = X_{ii} \ (i \in I_{\text{bin}}), \\ X_{jk} = X_{kj} = 0 \ ((j, k) \in I_{\text{comp}}) \end{array} \right\}, \\ \mathbf{H}^0 &= \text{the matrix in } \mathbb{S}^{1+n} \text{ with 1 at the } (0, 0) \text{ element and 0 elsewhere,} \\ \mathbf{Q}^0 &= \begin{pmatrix} 0 & \mathbf{c}^T \\ \mathbf{c} & \mathbf{C} \end{pmatrix}, \mathbf{H}^1 = (-\mathbf{b} \ \mathbf{A})^T (-\mathbf{b} \ \mathbf{A}) \in \mathbb{S}_+^{1+n} \subset \mathbb{K}_1^* + \mathbb{K}_2^*. \end{aligned} \right\} \quad (18)$$

Then, for every $\mathbf{x} = (x_0, \mathbf{u}) \in \mathbb{R}_+^{1+n}$,

$$\begin{aligned} x_0 = 1 &\Leftrightarrow \langle \mathbf{H}^0, \mathbf{x} \mathbf{x}^T \rangle = x_0^2 = 1, \\ \mathbf{A} \mathbf{u} - \mathbf{b} x_0 = \mathbf{0} &\Leftrightarrow \langle \mathbf{H}^1, \mathbf{x} \mathbf{x}^T \rangle = (\mathbf{A} \mathbf{u} - \mathbf{b} x_0)^T (\mathbf{A} \mathbf{u} - \mathbf{b} x_0) = 0, \\ \langle \mathbf{Q}^0, \mathbf{x} \mathbf{x}^T \rangle &= \mathbf{u}^T \mathbf{C} \mathbf{u} + 2x_0 \mathbf{c}^T \mathbf{u}. \end{aligned}$$

It is easy to verify that if $\mathbf{x} = (1, \mathbf{u}) \in \mathbb{R}_+^{1+n}$, then

$$\begin{aligned} [\mathbf{x} \mathbf{x}^T]_{0i} &= [\mathbf{x} \mathbf{x}^T]_{i0} = x_0 u_i = u_i, \quad [\mathbf{x} \mathbf{x}^T]_{ii} = u_i^2 \ (i \in I_{\text{bin}}), \\ [\mathbf{x} \mathbf{x}^T]_{jk} &= u_j u_k = u_k u_j = [\mathbf{x} \mathbf{x}^T]_{kj} \ ((j, k) \in I_{\text{comp}}); \end{aligned}$$

hence,

$$u_i(1 - u_i) = 0 \ (i \in I_{\text{bin}}) \ \text{and} \ u_j u_k = 0 \ ((j, k) \in I_{\text{comp}}) \Leftrightarrow \mathbf{x} \mathbf{x}^T \in \mathbb{K}_2$$

Therefore, QOP (17) is equivalent to

$$\zeta_{\text{QOP}} = \inf \left\{ \langle \mathbf{Q}^0, \mathbf{x} \mathbf{x}^T \rangle : \begin{array}{l} \mathbf{x} \mathbf{x}^T \in \mathbb{K}_1 \cap \mathbb{K}_2, \\ \langle \mathbf{H}^0, \mathbf{x} \mathbf{x}^T \rangle = 1, \langle \mathbf{H}^1, \mathbf{x} \mathbf{x}^T \rangle = 0 \end{array} \right\}. \quad (19)$$

Note that every feasible solution $\mathbf{u} \in \mathbb{R}^n$ of QOP (17) with the objective value $\mathbf{u}^T \mathbf{C} \mathbf{u} + 2\mathbf{c}^T \mathbf{u}$ corresponds to a feasible solution $\mathbf{x} = (1, \mathbf{u})$ of QOP (19) with the same objective value $\langle \mathbf{Q}^0, \mathbf{x} \mathbf{x}^T \rangle$. By construction, it is obvious that $\mathbf{H}^0, \mathbf{H}^1 \in \mathbb{S}_+^{1+n} \subset \mathbb{K}_1^* + \mathbb{K}_2^*$; hence Condition (I) is satisfied.

Now, by replacing $\mathbf{x} \mathbf{x}^T \in \mathbb{V}$ with a matrix variable $\mathbf{X} \in \mathbb{V}$, we obtain COP (1) with $\mathbb{V}, \mathbb{K}_1, \mathbb{K}_2, \mathbf{Q}^0, \mathbf{H}^0$ and \mathbf{H}^1 given in (18), which serves as a DNN relaxation of QOP (19). For the efficient computation of the metric projection onto \mathbb{K}_2 defined above, we refer to the paper [14].

6 Preliminary numerical results

We implemented the damped secant-bracketing method (Algorithm 4.1) by modifying the software package BBCPOP [10], which is based on the BP method (Algorithm 2.3) for solving DNN relaxations of QOPs and polynomial optimization problems (POPs) with binary, box and complementarity (BBC) constraints. For numerical tests, we experimented with binary quadratic optimization problem (BQOP) instances from [20] in Section 6.1, and quadratic assignment problem (QAP) instances from [9] in Section 6.2. All the computations were performed in MATLAB 2018b on iMac Pro with Intel Xeon W CPU (3.2 GHZ) and 128 GB memory.

We compare our results from Algorithm 4.1 (the damped secant-bracketing method) with BBCPOP and SDPNAL+. BBCPOP is based on Algorithm 2.3 and SDPNAL+ [22] is a Matlab implementation of the majorized semismooth Newton-CG augmented Lagrangian method for large scale SDPs with bounded variables.

6.1 Binary quadratic optimization problems

We solved DNN relaxations of BQOP instances, bqp100-1, ..., bqp100-5, bqp500-1, ..., bqp500-5, from [20] by three methods, BBCPOP, Algorithm 4.1 and SDPNAL+ [22].

Each BQOP instance is of the following form:

$$\zeta_{\text{BQOP}} = \inf \left\{ \mathbf{v}^T \mathbf{F} \mathbf{v} : \mathbf{v} \in \mathbb{R}_+^r, v_i(1 - v_i) = 0 \ (i = 1, \dots, r) \right\} \quad (20)$$

with $r = 100$ or $r = 500$, where $\mathbf{F} \in \mathbb{S}^r$. The problem (20) is a special case of QOP (17), thus the discussion in the previous section could be directly applied to BQOP (20) for its DNN relaxation. To strengthen the relaxation [13, 12], instead of the direct application, we first reformulate BQOP (20) to the following QOP by introducing a slack variable vector $\mathbf{w} \in \mathbb{R}_+^r$.

$$\zeta_{\text{BQOP}} = \inf \left\{ \mathbf{v}^T \mathbf{F} \mathbf{v} : \begin{array}{l} \mathbf{v} \in \mathbb{R}_+^r, \mathbf{w} \in \mathbb{R}_+^r, \mathbf{v} + \mathbf{w} = \mathbf{e}, \\ v_i(1 - v_i) = 0, w_i(1 - w_i) = 1 \ (i = 1, \dots, r) \end{array} \right\}, \quad (21)$$

where \mathbf{e} denotes the column vector of ones in \mathbb{R}^r . Let $n = 2r$, $\ell = r$, $I_{\text{bin}} = \{1, \dots, n\}$, $I_{\text{comp}} = \emptyset$,

$$\mathbf{u} = \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} \in \mathbb{R}^n, \mathbf{C} = \begin{pmatrix} \mathbf{F} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix} \in \mathbb{S}^n, \mathbf{c} = \mathbf{0} \in \mathbb{R}^n, \mathbf{A} = (\mathbf{I} \ \mathbf{I}), \mathbf{b} = \mathbf{e},$$

where \mathbf{I} denotes the $r \times r$ identity matrix. Then BQOP (21) can be rewritten as QOP (17), and its DNN relaxation of the form COP (1) can be derived with $\mathbb{K}_1, \mathbb{K}_2, \mathbf{Q}^0, \mathbf{H}^0$ and \mathbf{H}^1 given in (18).

We applied BBCPOP and Algorithm 4.1 with the modification mentioned at the end of Section 4 to the Lagrangian relaxation of the resulting COP (1) and its dual (2), *i.e.*, the pair of COPs (3) and (4) with $\lambda = 10,000$. (See Section 2.2). Since \mathbb{K}_1 and \mathbb{K}_2 are given in (18) with $I_{\text{bin}} = \{1, \dots, n\}$ and $I_{\text{comp}} = \emptyset$, COP (1) is equivalent to the DNN problem

$$\eta^p = \inf \left\{ \langle \mathbf{Q}^0, \mathbf{X} \rangle : \begin{array}{l} \mathbf{X} \in \mathbb{D}^{1+n}, \langle \mathbf{H}^0, \mathbf{X} \rangle = 1, \langle \mathbf{H}^1, \mathbf{X} \rangle = 0, \\ X_{0i} = X_{i0} = X_{ii} \ (i = 1, \dots, n) \end{array} \right\}. \quad (22)$$

We also applied SDPNAL+ to this problem with the default parameters.

Table 1 presents the numerical results. We can observe two distinct features of Algorithm 4.1, which are inherited from the Newton-bracketing method to some extent, in comparison to BBCPOP. The first one is that ‘it’ in Algorithm 4.1 is smaller than ‘it’ in BBCPOP in most of the instances. This is due to the advantage of the secant iteration over the bisection iteration. We note that Algorithm 4.1 sometimes switched to the bisection (Steps 2.3 and 2.6 stated at the end of Section 4) in many instances, which increased ‘it’, but ‘it’ of Algorithm 4.1 still was smaller than ‘it’ of BBCPOP. However, ‘itAPG’ (the total number of iterations spent by the APG method) in Algorithm 4.1 is larger than ‘itAPG’ in BBCPOP. This is because Algorithm 4.1 requires more accuracy in the computation of $(\widehat{\mathbf{X}}(y^k), \widehat{\mathbf{Y}}_1(y^k), \widehat{\mathbf{Y}}_2(y^k))$ by the APG method as we discussed at the end of Section 4. As a result, the execution time of Algorithm 4.1 is longer than that of BBCPOP.

The other important feature of Algorithm 4.1 is the quality of lower bounds obtained. Indeed, the lower bounds obtained by Algorithm 4.1 for larger scale instances are obviously tighter than those obtained by the other methods.

6.2 Quadratic assignment problems

We identify a matrix $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r] \in \mathbb{R}^{r \times r}$ with the r^2 -dimensional column vector $\mathbf{u} = \text{vec}(\mathbf{W}) = [\mathbf{w}_1; \dots; \mathbf{w}_r] \in \mathbb{R}^{r^2}$. Given matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{r \times r}$, the quadratic assignment problem is stated as

$$\begin{aligned} \zeta_{\text{QAP}} &= \inf \left\{ \mathbf{W} \bullet \mathbf{A} \mathbf{W} \mathbf{B}^T : \begin{array}{l} \mathbf{e}^T \mathbf{W} \mathbf{e}_j = 1, \mathbf{e}_j^T \mathbf{W} \mathbf{e} = 1, \\ (j = 1, \dots, r), \mathbf{W} \in \{0, 1\}^{r \times r} \end{array} \right\} \\ &= \min \left\{ \mathbf{u}^T (\mathbf{B} \otimes \mathbf{A}) \mathbf{u} : \begin{array}{l} (\mathbf{e}_j^T \otimes \mathbf{e}^T) \mathbf{u} = 1, (\mathbf{e}^T \otimes \mathbf{e}_j^T) \mathbf{u} = 1 \\ (j = 1, \dots, r), u_i(1 - u_i) = 1 \ (i = 1, \dots, r^2) \end{array} \right\}. \end{aligned}$$

Here $\mathbf{e} \in \mathbb{R}^r$ denotes the vector of ones, $\mathbf{e}_j \in \mathbb{R}^r$ the j th coordinate unit vector, and \otimes denotes the Kronecker product. This problem is a special case of QOP (17), so its DNN relaxation of the form COP (1) can be derived.

Table 2 shows numerical results on large scale QAP instances from [9], whose optimal values are not known. COP (1) derived as the DNN relaxation of the instance sko81 in

Table 1: Binary QOP instances from [20]. LB denotes a valid lower bound for the optimal value. LB corresponds to $\tilde{\text{lb}}$ in Algorithms 4.1 and 2.3 (BBCPOP). We also applied Lemma 2.2 to the output of SDPNAL+ and computed a valid lower bound for the optimal value. ‘sec’ denotes the execution time, ‘itAPG’ the total number of iterations spent in the APG method in the case of Algorithms 4.1 and 2.3 and the total number of iterations in the case of SDPNAL+, and ‘it’ the total number of iterations in Algorithms 2.3 and 4.1. The bold digits in the column LB show the cases where Algorithm 4.1 computed tighter lower bounds than the other two methods.

Problem	Opt.Val.	Solver	LB(sec,itAPG/it)
bqp100-1	-7970	Algorithm 4.1	-8036 (1.1e2,14980/14)
bqp100-1		BBCPOP	-8039(5.7e1, 8453/20)
bqp100-1		SDPNAL+	-8050(5.4e1, 5741)
bqp100-2	-11036	Algorithm 4.1	-11036 (1.1e2,15310/14)
bqp100-2		BBCPOP	-11043(3.3e1, 5265/20)
bqp100-2		SDPNAL+	-11039(2.9e2,23868)
bqp100-3	-12723	Algorithm 4.1	-12723(5.4e1, 8455/11)
bqp100-3		BBCPOP	-12723(2.0e1, 3874/21)
bqp100-3		SDPNAL+	-12723(1.8e1, 3321)
bqp100-4	-10368	Algorithm 4.1	-10368(6.0e1, 9105/11)
bqp100-4		BBCPOP	-10368(2.2e1, 4124/14)
bqp100-4		SDPNAL+	-10369(2.3e1, 3991)
bqp100-5	-9083	Algorithm 4.1	-9083(1.0e2,13365/13)
bqp100-5		BBCPOP	-9083(4.4e1, 7183/21)
bqp100-5		SDPNAL+	-9083(2.0e2,21039)
bqp500-1	-116586	Algorithm 4.1	-122598 (1.3e4,35030/21)
bqp500-1		BBCPOP	-133277(6.1e2, 2104/21)
bqp500-1		SDPNAL+	-122633(3.0e3, 8783)
bqp500-2	-128223	Algorithm 4.1	-132730 (1.0e4,27840/17)
bqp500-2		BBCPOP	-142163(5.6e2, 1915/21)
bqp500-2		SDPNAL+	-132773(3.1e3, 7591)
bqp500-3	-130812	Algorithm 4.1	-134800 (1.2e4,31350/18)
bqp500-3		BBCPOP	-145715(7.0e2, 2323/22)
bqp500-3		SDPNAL+	-134841(3.1e3, 8656)
bqp500-4	-130097	Algorithm 4.1	-135485 (1.0e4,28235/17)
bqp500-4		BBCPOP	-147512(7.0e2, 2468/22)
bqp500-4		SDPNAL+	-135527(2.8e3, 7699)
bqp500-5	-125487	Algorithm 4.1	-130300 (8.1e3,28630/18)
bqp500-5		BBCPOP	-140946(7.1e2, 2355/22)
bqp500-5		SDPNAL+	-130404(1.6e3, 6194)

Table 2 involves a 6562×6562 dense variable matrix \mathbf{X} . Such a large scale DNN problem is extremely hard to solve by many of the existing software packages. To solve such a large scale QAP, it is meaningful to reduce the gap between the lower and upper bounds.

Recently, Mittelman [15] applied BBCPOP to large scale QAP instances with unknown optimal values, and computed their then-new lower bounds which had not been obtained before. His results demonstrated that BBCPOP is very powerful and effective for solving DNN problems from large scale QAP instances. His numerical experiments were conducted with MATLAB 2018a on 32GB Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz. To compare BBCPOP with Algorithm 4.1, we applied the two methods to some of the QAP instances on the same computing environment (MATLAB 2018b on iMac Pro with Intel Xeon W CPU (3.2 GHZ) and 128 GB memory). See the 3rd and 4th columns of Table 2. As observed in the numerical results on BQOP instances in Table 1, the lower bounds obtained by Algorithm 4.1 are of higher quality than those obtained by BBCPOP for all instances in Table 2, but Algorithm 4.1 is 2-5 times slower than BBCPOP. We also see that ‘it’ of Algorithm 4.1 is larger than ‘it’ of BBCPOP for instances tai35b – tai60b, tai80b and tho40. This is because the APG method often failed to terminate with the stopping criteria within the prescribed iteration upper bound, and switched to the bisection (Steps 2.3 and 2.6 stated at the end of Section 4) for those instances.

In [15], BBCPOP was applied to larger scale QAP instances in addition to the ones in Table 2, including tai100a, tai100b and wil100. But we have not applied Algorithm 4.1 to those larger instances because they are too time consuming.

7 Concluding remarks

While our focus has been on QOPs in this paper, the Lagrangian-DNN was extended to a class of polynomial optimization problems (POPs) with binary, box and complementarity constraints. In fact, BBCPOP was designed to solve such POPs, and the paper [11] reported better numerical results in terms of computational time (and/or quality of lower bounds in some instances) on large scale randomly generated instances from the class than SDPNAL+ which is regarded as a state-of-art software for solving DNN relaxation problems.

Theoretically, the extension of the Newton-bracketing method to the Lagrangian-DNN relaxation induced from the class of POPs is almost straightforward. As observed in Section 6, the quality of the optimal values of relaxation problems of large-scale QOP instances obtained by the secant-bracketing method are clearly better than those by BBCPOP. The secant-bracketing method is, however, more time-consuming than BBCPOP. As a result, the computational efficiency of the method should be improved before incorporating it into BBCPOP to handle large scale QOPs and POPs.

References

- [1] M. F. Anjos and J. B. Lasserre. *Handbook on Semidefinite, Conic and Polynomial Optimization*, volume 166 of *International Series in Operations Research and*

Table 2: QAP instances from [9]. LB denotes a valid lower bound for the optimal value. LB corresponds to $\tilde{\text{lb}}$ in Algorithms 2.3 and 4.1 (BBCPOP). ‘sec’ denotes the execution time, ‘itAPG’ the total number of iterations spent in the APG method, and ‘it’ the total number of iterations in Algorithms 4.1 and BBCPOP. The bold digits show differences between LBs obtained by Algorithm 4.1 and BBCPOP.

Problem	Best known upper bounds	Algorithm 4.1 LB(sec,itAPG/it)	BBCPOP [15] LB(sec,itAPG/it)
tai35b	283315445	269 754270 (4.0e3,17265/28)	269 532372 (1.3e3,6272/21)
tai40b	637250948	609242568 (9.6e3,19195/29)	608808404 (3.7e3,8168/21)
tai50b	458821517	431 245612 (2.4e4,15595/28)	431 090738 (7.3e3,5072/21)
tai60a	7205962	632 6094 (2.8e4,6485/23)	632 5979 (1.5e4,3710/22)
tai60b	608215054	592 829061 (6.6e4,16630/27)	592 371789 (3.5e4,9008/22)
tai80a	13499184	116570 55 (1.1e5,5280/21)	116570 14 (2.0e4,1091/22)
tai80b	818415043	786900529 (3.7e5,18070/33)	298474 (8.1e4,4017/22)
sko42	15812	1533 4 (5.7e3,9245/16)	1533 3 (2.1e3,3898/21)
sko49	23386	2265 3 (1.2e4,9170/19)	2265 1 (4.4e3,3383/21)
sko56	34458	3338 9 (2.3e4,8780/18)	3338 6 (8.7e3,3393/22)
sko64	48498	470 20 (5.6e4,8920/18)	470 18 (1.8e4,3001/22)
sko72	66256	644 61 (1.1e5,9565/17)	644 56 (5.3e4,4923/22)
sko81	90998	883 70 (2.0e5,9555/17)	883 63 (8.1e4,3740/23)
tho40	240516	226 516 (5.0e3,10065/24)	226 491 (2.2e3,4826/21)
wil50	48816	481 23 (1.3e4,8650/20)	481 22 (8.0e3,5453/21)

Management Science. Springer, 2012.

- [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. A robust Lagrangian-DNN method for a class of quadratic optimization problems. *Comput. Optim. Appl.*, 66(3):453–479, 2017.
- [4] 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. *Pacific J. of Optim.*, 14(1):161–192, 2018.
- [5] N. Arima, S. Kim, M. Kojima, and K. C. Toh. Lagrangian-conic relaxations, Part II: Applications to polynomial optimization problems. *To appear in Pacific J. of Optim.*
- [6] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2:183–202, 2009.
- [7] S. Burer. On the copositive representation of binary and continuous non-convex quadratic programs. *Math. Program.*, 120:479–495, 2009.
- [8] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.
- [9] P. Hahn and M. Anjos. QAPLIB – a quadratic assignment problem library. <http://www.seas.upenn.edu/qaplib>.
- [10] N. Ito, S. Kim, M. Kojima, A. Takeda, and K.C. Toh. Equivalences and differences in conic relaxations of combinatorial quadratic optimization problems. *J. Global Optim.*, 72(4):619–653, 2018.
- [11] N. Ito, S. Kim, M. Kojima, A. Takeda, and K.C. Toh. BBCPOP: A sparse doubly nonnegative relaxation of polynomial optimization problems with binary, box and complementarity constraints. *ACM Trans. Math. Softw.*, To appear, 2019.
- [12] S. Kim and M. Kojima. Binary quadratic optimization problems that are difficult to solve by conic relaxations. *Discrete Optim.*, 24:170–183, 2017.
- [13] 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.
- [14] S. Kim, M. Kojima, and K. C. Toh. Doubly nonnegative relaxations for quadratic and polynomial optimization problems with binary and box constraints. Research Rport B-483, Tokyo Institute of Technology, Department of Mathematical and Computing Sciences, Oh-Okayama, Meguro-ku, Tokyo 152-8552, July 2016.

- [15] H. D. Mittelmann. Improved QAPLIB lower bounds using BBCPOP. May 2018.
- [16] J. J. Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires. *C. R. Acad. Sci.*, 255:238–240, 1962.
- [17] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.
- [18] N. Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.
- [19] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.
- [20] A. Wiegele. Biq mac library. <http://www.biqmac.uni-klu.ac.at/biqmaclib.html>, 2007.
- [21] H. Wolkowicz, R. Saigal, and L. Vandenberghe. *Handbook of semidefinite programming: theory, algorithms, and applications*. Springer, 2000.
- [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.