Math. Program., Ser. A (2016) 156:161–187
DOI 10.1007/s10107-015-0874-5

FULL LENGTH PAPER

A Lagrangian–DNN relaxation: a fast method for computing tight lower bounds for a class of quadratic optimization problems

Sunyoung Kim · Masakazu Kojima · Kim-Chuan Toh

Received: 9 October 2013 / Accepted: 9 February 2015 / Published online: 21 February 2015

Abstract We propose an efficient computational method for linearly constrained quadratic optimization problems (QOPs) with complementarity constraints based on their Lagrangian and doubly nonnegative (DNN) relaxation and first-order algorithms. The simplified Lagrangian–completely positive programming (CPP) relaxation of such QOPs proposed by Arima, Kim, and Kojima in 2012 takes one of the simplest forms, an unconstrained conic linear optimization problem with a single Lagrangian parameter in a CPP matrix variable with its upper-left element fixed to 1. Replacing the CPP matrix variable by a DNN matrix variable, we derive the Lagrangian–DNN relaxation, and establish the equivalence between the optimal value of the DNN relaxation of the original QOP and that of the Lagrangian–DNN relaxation. We then propose an efficient numerical method for the Lagrangian–DNN relaxation using a bisection method

Sunyoung Kim's research was supported by NRF 2012-R1A1A2-038982 and NRF 2014-R1A2A1A11049618. Masakazu Kojima's research was supported by the Japan Science and Technology Agency (JST), the Core Research of Evolutionary Science and Technology (CREST) research project. Kim-Chuan Toh's research was supported in part by Ministry of Education Academic Research Fund R-146-000-168-112.

S. Kim (✉)
Department of Mathematics, Ewha W. University, 52 Ewhayeodae-gil,
Sudaemoon-gu, Seoul 120-750, Korea
e-mail: skim@ewha.ac.kr

M. Kojima
Research and Development Initiative, Chuo University,
1-13-27, Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
e-mail: kojima@is.titech.ac.jp

K.-C. Toh
Department of Mathematics, National University of Singapore,
10 Lower Kent Ridge Road, Singapore 119076, Singapore
e-mail: mattohkc@nus.edu.sg

combined with the proximal alternating direction multiplier and the accelerated proximal gradient methods. Numerical results on binary QOPs, quadratic multiple knapsack problems, maximum stable set problems, and quadratic assignment problems illustrate the superior performance of the proposed method for attaining tight lower bounds in shorter computational time.

Keywords Linearly constrained quadratic optimization problems with complementarity constraints · The Lagrangian–conic relaxation · Bisection method · Iterative solver

Mathematics Subject Classification 90C20 · 90C25 · 90C26

1 Introduction

We consider a linearly constrained quadratic optimization problem (QOP) with complementarity constraints:

$$\text{minimize } \left\{ \boldsymbol{u}^T \boldsymbol{Q}\boldsymbol{u} + 2\boldsymbol{c}^T\boldsymbol{u} \;\middle|\; \begin{array}{l} \boldsymbol{u} \in \mathbb{R}^m_+, \; \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} = 0, \\ u_i u_j = 0 \; ((i,j) \in \mathcal{E}) \end{array} \right\} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{q \times m}$, $\boldsymbol{b} \in \mathbb{R}^q$, $\boldsymbol{c} \in \mathbb{R}^m$, $\boldsymbol{Q} \in \mathbb{S}^m$, and $\mathcal{E} \subset \{(i,j) : 1 \leq i < j \leq m\}$ are given data. Note that the binary constraint $u_i(1-u_i) = 0$ can be converted to a complementarity constraint $u_i v_i = 0$ with a slack variable $v_i = 1 - u_i \geq 0$; thus QOP (1) can model nonconvex quadratic problems with linear, binary and complementarity constraints. We assume that the linear constraint set $\{\boldsymbol{u} \in \mathbb{R}^m_+ : \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{0}\}$ is bounded. The QOP model (1) satisfying this assumption includes various combinatorial optimization problems, for instance, the binary integer quadratic problem, the maximum stable set problem, the quadratic multiple knapsack problem, and the quadratic assignment problem [10,26,28].

The completely positive programming (CPP) relaxation of linearly constrained QOPs in binary and continuous variables by Burer [9] has gained considerable attention since 2009. Extending his work, theoretical results for a more general class of QOPs were established by Eichfelder and Povh [13,16] and by Arima et al. [1]. They showed that the exact optimal values of QOPs in their classes coincided with the optimal values of their CPP relaxation problems.

Such CPP relaxations are numerically intractable since the problem of determining whether a given matrix lies in the completely positive cone or the copositive cone is a NP-complete or co-NP-complete problem as shown in [24]. Replacing the completely positive cone by doubly nonnegative (DNN) cone and solving the resulting problem by a primal–dual interior-point method has been a popular approach [20,36]. The computational cost of this approach, however, is very expensive as the number of nonnegative constraints for the variables grows rapidly with the size of the problem.

Another approach to approximately solve the CPP relaxation of (1) was developed in [8] via adaptive polyhedral inner and outer approximations of the copositive cone. Impressive numerical results were reported that for certain randomly generated nonconvex QOPs over the standard simplex, the adaptive linear approximation method can solve problems with $n = 2000$ in about 30 s on the average. However, the paper [8] also reported disappointing news that the linear approximation method was not able to generate good bounds (even given substantial computing time) for more realistic QOPs such as those arising from quadratic assignment problems or the maximum stable set problems.

Recently, Arima et al. [2] introduced the simplified Lagrangian–CPP relaxation of a linearly constrained QOP in continuous and binary variables with a single parameter $\lambda \in \mathbb{R}$. It was derived by reducing the original QOP to an equivalent QOP with a single quadratic equality constraint in nonnegative variables, and applying the Lagrangian relaxation to the resulting QOP. As a result, an unconstrained QOP with a Lagrangian multiplier $\lambda \in \mathbb{R}$ in nonnegative variables was obtained. From the computational point of view, this Lagrangian relaxation is one of the simplest forms to handle with a solution method. Applying the CPP relaxation to the unconstrained QOP with λ lead to the Lagrangian–CPP relaxation. It was shown in [2] that the optimal values of the Lagrangian relaxation as well as its CPP relaxation monotonically converge to the exact optimal value of the original QOP as λ tends to ∞.

The main goals of this paper are twofold. First, we propose an efficient and effective numerical method for the QOP model (1) by extending the framework of the Lagrangian–CPP relaxation to the one including the Lagrangian–DNN relaxation of (1). Second, generalizing the brief discussion on such an extension in [2], we present a theoretical framework for the Lagrangian–conic relaxation of (1) that covers both Lagrangian–CPP and Lagrangian–DNN relaxations. We use the primal–dual pair of the Lagrangian–DNN relaxation with a sufficiently large $\lambda \in \mathbb{R}$ to compute a tight lower bound for the optimal value of (1). The main features of the Lagrangian–DNN relaxation are:

- The primal is an unconstrained DNN problem with a matrix variable whose upper-left corner element is fixed to 1. Thus, its dual becomes a simple problem with just a single variable.
- The primal DNN problem is strictly feasible, i.e., the primal feasible region intersects with the interior of the DNN cone.
- A common optimal objective value, shared by the primal–dual pair with a parameter $\lambda > 0$, monotonically converges to the optimal objective value of the DNN relaxation of (1). Hence, a lower bound with almost the same quality as the one obtained from the DNN relaxation of (1) can be computed for the optimal objective value of (1) via the simple Lagrangian–DNN relaxation with a sufficiently large $\lambda \in \mathbb{R}$.

The computational efficiency for solving the Lagrangian–DNN relaxation can be expected from the first and second features mentioned above. If a primal–dual interior-point method [7,18,31,32] for SDPs is used to solve the Lagrangian–DNN relaxation, then the inequality constraints induced from the nonnegativity of all elements of the DNN matrix variable may incur prohibitive computational burden. More precisely, if the size of the DNN matrix is n, then the number of inequalities to be added in the SDP problem amounts to $n(n-1)/2$. Thus, the conversion of a DNN problem into a standard SDP is computationally inefficient when n is large. To avoid such inefficiency of using a primal–dual interior-point method, the numerical method proposed in this paper employs *first-order algorithms* [4] for the Lagrangian–DNN relaxation without converting it into a standard SDP. The first-order algorithms in our approach is used to find the projection onto the intersection of two cones, SDP and nonnegative matrix cones. The first and second features previously mentioned considerably increase the efficiency and numerical stability of our first-order algorithms, respectively. We should mention that for solving linear SDPs with doubly nonnegative cone constraints, Wen et al. [34] have applied a direct extension of the alternating direction method of multipliers (ADMM) to solve the dual problems. However, it has been shown recently in [11] that such a direct extension of the ADMM may not necessarily be convergent.

The numerical results in Sect. 5 show that the Lagrangian–DNN relaxation provides tighter lower bounds more efficiently than the DNN relaxation of the QOP (1). When the proposed method is experimented on the test problems such as the binary integer quadratic problem, the maximum stable set problem, the quadratic multiple knapsack problem, and the quadratic assignment problem, the quality of the lower bounds obtained from the proposed method is tighter, compared to the known optimal values. As mentioned in the third main feature, a sufficiently large λ results in a tight lower bound. The proposed method can also solve the problems efficiently, in particular, it obtains the lower bound for the quadratic assignment problem much faster than SDPNAL, which is an advanced large scale SDP solver [37], applied to the DNN relaxation of the problem.

In Sect. 2, we list notation and symbols used throughout the paper. The QOP (1) is extended to a general QOP model, from which effective conic and Lagrangian–conic relaxations are derived. In Sect. 3, the conic and Lagrangian–conic relaxations and their relations are discussed. In particular, the Lagrangian–DNN relaxation, a special case of the Lagrangian–conic relaxations, is used to obtain a tight lower bound for the optimal value of the general QOP. In Sect. 4, we presents a bisection method together with the proximal alternating direction multiplier method [17] and the accelerated proximal gradient method [4] for solving the Lagrangian–DNN relaxation of the general QOP. Numerical results on the binary integer quadratic problem, the maximum stable set problem, the quadratic multiple knapsack problem, and the quadratic assignment problem are presented in Sect. 5. Finally, we conclude in Sect. 6.

2 Preliminaries

2.1 Notation and symbols

Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}_+$ the set of nonnegative real numbers, $\mathbb{S}^n$ the space of $n \times n$ symmetric matrices, and $\mathbb{S}^n_+$ the cone of $n \times n$ symmetric positive semidefinite matrices. We let $\mathbb{C} = \left\{ \boldsymbol{A} \in \mathbb{S}^n : \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0 \text{ for all } \boldsymbol{x} \in \mathbb{R}^n_+ \right\}$ (the copositive cone), $\boldsymbol{\Gamma} = \left\{ \boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x} \in \mathbb{R}^n_+ \right\}$, $\mathbb{C}^* =$ the convex hull of $\boldsymbol{\Gamma}$ (the completely positive cone, the dual of $\mathbb{C}$), $\mathbb{N} =$ the cone of $n \times n$ symmetric matrices with nonnegative elements . $\boldsymbol{Y} \bullet \boldsymbol{Z}$ means trace of $\boldsymbol{Y}\boldsymbol{Z}$ for every $\boldsymbol{Y},\ \boldsymbol{Z} \in \mathbb{S}^n$.

The following relations for $\mathbf{\Gamma}$, $\mathbb{S}^n_+$, $\mathbb{C}$, $\mathbb{C}^*$ and $\mathbb{N}$ are well-known:

$$\mathbf{\Gamma} \subset \mathbb{C}^* \subset \mathbb{S}^n_+ \cap \mathbb{N} \subset \mathbb{S}^n_+ \subset \mathbb{S}^n_+ + \mathbb{N} \subset \mathbb{C},\ \mathbf{\Gamma} = \left\{ \boldsymbol{X} \in \mathbb{S}^n_+ \cap \mathbb{N} : \mathrm{rank}(\boldsymbol{X}) = 1 \right\},$$
$$\mathbb{S}^n_+ \cap \mathbb{N} = \left(\mathbb{S}^n_+ + \mathbb{N}\right)^*$$

We call $\mathbb{S}^n_+ \cap \mathbb{N}$ *the doubly nonnegative cone*.

For $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$. We use the notation $(\boldsymbol{t}, \boldsymbol{u}) \in \mathbb{R}^{\ell+m}$ for the $(\ell + m)$-dimensional column vector consisting of $\boldsymbol{t} \in \mathbb{R}^\ell$ and $\boldsymbol{u} \in \mathbb{R}^m$. In the subsequent discussions, the quadratic form $\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x}$ associated with a matrix $\boldsymbol{Q} \in \mathbb{S}^n$ is represented as $\boldsymbol{Q} \bullet \boldsymbol{x}\boldsymbol{x}^T$ to suggest that $\boldsymbol{Q} \bullet \boldsymbol{x}\boldsymbol{x}^T$ with $\boldsymbol{x} \in \mathbb{R}^n_+$ is relaxed to $\boldsymbol{Q} \bullet \boldsymbol{X}$ with $\boldsymbol{X} \in \mathbb{C}^*$, $\boldsymbol{X} \in \mathbb{S}^n_+ \cap \mathbb{N}$ or $\boldsymbol{X} \in \mathbb{S}^n_+$.

2.2 A quadratic optimization model

We introduce a QOP of the form

$$\zeta := \inf \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T \;\middle|\; \begin{array}{l} \boldsymbol{x} \in \mathbb{R}^n_+,\ \boldsymbol{H}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T = 1, \\ \boldsymbol{Q}_p \bullet \boldsymbol{x}\boldsymbol{x}^T = 0\ (p = 1, 2, \ldots, q) \end{array} \right\}, \tag{2}$$

where $\boldsymbol{H}_0 \in \mathbb{S}^n$ and $\boldsymbol{Q}_p \in \mathbb{S}^n$ $(p = 0, 1, \ldots, q)$, to describe a class of conic relaxations and their further Lagrangian relaxations in Sect. 3. This form of QOP was introduced in [1] to establish an equivalence to its CPP relaxation under a set of conditions. If we are concerned with only the CPP relaxation among the conic relaxations, basic theoretical results presented in Sect. 3 can be obtained from [1,2]. Our main emphasis here is on extending their framework to a larger class of conic and Lagrangian–conic relaxations. In particular, the class includes a Lagrangian–DNN relaxation of QOP (2), which is shown to work very effectively and efficiently for large-scale QOPs in Sect. 5 with the first order methods described in Sect. 4.

We assume the following three conditions throughout the paper. These conditions are stronger than the ones assumed in [1,2], because the framework includes not only the CPP relaxation but also the DNN relaxation.

Condition (a) The feasible region

$$\left\{ \boldsymbol{x} \in \mathbb{R}^n_+ : \boldsymbol{H}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T = 1,\ \boldsymbol{Q}_p \bullet \boldsymbol{x}\boldsymbol{x}^T = 0\ (p = 1, 2, \ldots, q) \right\} \text{ is nonempty.} \tag{3}$$

Condition (b) $\boldsymbol{O} \neq \boldsymbol{H}_0 \in \mathbb{S}^n_+ + \mathbb{N}$ and $\boldsymbol{O} \neq \boldsymbol{Q}_p \in \mathbb{S}^n_+ + \mathbb{N}$ $(p = 1, 2 \ldots, q)$.

Condition (c) $\boldsymbol{D} = \boldsymbol{O}$ if $\boldsymbol{D} \in \mathbb{S}^n_+ \cap \mathbb{N}$, $\boldsymbol{H}_0 \bullet \boldsymbol{D} = 0$ and $\boldsymbol{Q}_p \bullet \boldsymbol{D} = 0$ $(p = 1, 2, \ldots, q)$.

Notice that if $\boldsymbol{H}_0 = \boldsymbol{O}$ then QOP (2) is infeasible, and if $\boldsymbol{Q}_p = \boldsymbol{O}$ for some p then the redundant constraint $\boldsymbol{Q}_p \bullet \boldsymbol{x}\boldsymbol{x}^T = 0$ can be eliminated. Thus, $\boldsymbol{H}_0 \neq \boldsymbol{O}$ and $\boldsymbol{Q}_p \neq \boldsymbol{O}$ $(p = 1, 2, \ldots, q)$ in Condition (b) can be assumed without loss of generality. We note that Condition (c) together with Condition (a) require that the feasible region is nonempty and bounded. For the proof of this fact, see (i) of Lemma 2 and its proof. Hence “inf” in (2) can be replaced by “min”. We let $\boldsymbol{x}^*$ be an optimal solution of QOP (2) with the finite optimal value ζ.

We can easily transform QOP (1) with linear and complementarity constraints to a QOP of the form (2) satisfying Conditions (a), (b) and (c) as follows. Define

$$\begin{aligned}
&\boldsymbol{x} = (x_1, \boldsymbol{u}) \in \mathbb{R}^n \text{ with } n = 1 + m,\\
&\boldsymbol{Q}_0 = \begin{pmatrix} 0 & \boldsymbol{c}^T \\ \boldsymbol{c} & \boldsymbol{Q} \end{pmatrix} \in \mathbb{S}^n, \quad \boldsymbol{H}_0 = \begin{pmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{O} \end{pmatrix} \in \mathbb{S}^n, \quad \boldsymbol{Q}_{01} = \begin{pmatrix} \boldsymbol{b}^T\boldsymbol{b} & \boldsymbol{b}^T\boldsymbol{A} \\ \boldsymbol{A}^T\boldsymbol{b} & \boldsymbol{A}^T\boldsymbol{A} \end{pmatrix} \in \mathbb{S}^n,\\
&\boldsymbol{C}_{ij} = \begin{matrix}\text{the } m \times m \text{ matrix with } (i,j)\text{th component } 1/2,\\ \text{and } 0 \text{ elsewhere } ((i,j) \in \mathcal{E}),\end{matrix} \quad \boldsymbol{Q}_{ij} = \begin{pmatrix} 0 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{C}_{ij} + \boldsymbol{C}_{ij}^T \end{pmatrix} \in \mathbb{S}^n.
\end{aligned}$$

Then QOP (1) is equivalent to

$$\text{minimize } \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T \,\middle|\, \begin{matrix} \boldsymbol{x} \in \mathbb{R}^n_+, \ \boldsymbol{H}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T = 1, \\ \boldsymbol{Q}_{01} \bullet \boldsymbol{x}\boldsymbol{x}^T = 0, \ \boldsymbol{Q}_{ij} \bullet \boldsymbol{x}\boldsymbol{x}^T = 0 \ ((i,j) \in \mathcal{E}) \end{matrix} \right\} \tag{4}$$

in the sense that $\boldsymbol{u} \in \mathbb{R}^m$ is a feasible solution of (1) with the objective value $\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u} + 2\boldsymbol{c}^T\boldsymbol{u}$ if and only if $\boldsymbol{x} = (1, \boldsymbol{u}) \in \mathbb{R}^n$ is a feasible solution of (4) with the same objective value $\boldsymbol{Q}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T$.

Lemma 1 *Suppose that the feasible region of QOP* (1) *is nonempty and that the polyhedral set,* $\left\{\boldsymbol{u} \in \mathbb{R}^m_+ : \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{0}\right\}$, *is bounded. Then QOP* (4) *induced from QOP* (1) *satisfies Conditions (a), (b) and (c). Here we assume that the subscripts of the matrices* $\boldsymbol{Q}_{01}$ *and* $\boldsymbol{Q}_{ij}$ $((i,j) \in \mathcal{E})$ *have been renumbered to* $1, 2, \ldots, q$ *for some* q.

Proof We only prove Condition (c) because Conditions (a) and (b) are obvious. Assume that $\boldsymbol{H}_0 \bullet \boldsymbol{D} = 0$, $\boldsymbol{Q}_{01} \bullet \boldsymbol{D} = 0$ and $\boldsymbol{Q}_{ij} \bullet \boldsymbol{D} = 0$ $(i,j) \in \mathcal{E}$ for some $\boldsymbol{D} \in \mathbb{S}^{1+m}_+ \cap \mathbb{N}$. Then, we see that

$$0 = \boldsymbol{H}_0 \bullet \boldsymbol{D} = D_{11}, \quad \text{and } 0 = \boldsymbol{Q}_{01} \bullet \boldsymbol{D} = \begin{pmatrix} \boldsymbol{b}^T\boldsymbol{b} & \boldsymbol{b}^T\boldsymbol{A} \\ \boldsymbol{A}^T\boldsymbol{b} & \boldsymbol{A}^T\boldsymbol{A} \end{pmatrix} \bullet \boldsymbol{D}. \tag{5}$$

Now we write $\boldsymbol{D} \in \mathbb{S}^{1+m}_+ \cap \mathbb{N}$ as $\begin{pmatrix} D_{11} & \boldsymbol{D}_{12} \\ \boldsymbol{D}_{12}^T & \boldsymbol{D}_{22} \end{pmatrix}$. From $0 = D_{11}$ and $\boldsymbol{D} \in \mathbb{S}^{1+m}_+$, we get $\boldsymbol{D}_{12} = \boldsymbol{0}$. As a result, the last relation in (5) implies that $\boldsymbol{A}^T\boldsymbol{A} \bullet \boldsymbol{D}_{22} = \boldsymbol{O}$. Since $\boldsymbol{A}^T\boldsymbol{A} \in \mathbb{S}^m_+$ and $\boldsymbol{D}_{22} \in \mathbb{S}^m_+$, we obtain $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{D}_{22} = \boldsymbol{O}$ and $\boldsymbol{A}\boldsymbol{D}_{22} = \boldsymbol{O}$. On the other hand, by the assumption of the lemma, there does not exist nonzero $\boldsymbol{d} \in \mathbb{R}^m$ such that $\boldsymbol{d} \geq \boldsymbol{0}$ and $-\boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}$. By applying Tucker's theorem of the alternative [33], we can take a $\boldsymbol{y} \in \mathbb{R}^m$ such that $\boldsymbol{A}^T\boldsymbol{y} > \boldsymbol{0}$. Multiplying $\boldsymbol{y}^T$ to the identity $\boldsymbol{A}\boldsymbol{D}_{22} = \boldsymbol{O}$, we obtain that $(\boldsymbol{y}^T\boldsymbol{A})\boldsymbol{D}_{22} = \boldsymbol{0}^T$, $(\boldsymbol{y}^T\boldsymbol{A}) > \boldsymbol{0}^T$ and $\boldsymbol{D}_{22} \in \mathbb{N}$, which implies $\boldsymbol{D}_{22} = \boldsymbol{O}$. Thus we have shown that $\boldsymbol{D} = \boldsymbol{O}$. □

Condition (b) implies that the inequalities $\boldsymbol{Q}_p \bullet \boldsymbol{x}\boldsymbol{x}^T \geq 0$ $(p = 1, 2, \ldots, q)$ hold for any $\boldsymbol{x} \in \mathbb{R}^n_+$. Hence, the set of equalities $\boldsymbol{Q}_p \bullet \boldsymbol{x}\boldsymbol{x}^T = 0$ $(p = 1, 2, \ldots, q)$ in QOP

(2) can be combined into a single equality $\boldsymbol{H}_1 \bullet \boldsymbol{x}\boldsymbol{x}^T = 0$, where $\boldsymbol{H}_1 = \sum_{p=1}^{q} \boldsymbol{Q}_p$. Consequently, we obtain a simplified QOP:

$$\zeta := \min \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T \mid \boldsymbol{x} \in \mathbb{R}^n_+,\ \boldsymbol{H}_0 \bullet \boldsymbol{x}\boldsymbol{x}^T = 1,\ \boldsymbol{H}_1 \bullet \boldsymbol{x}\boldsymbol{x}^T = 0 \right\}, \tag{6}$$

which is equivalent to QOP (2). Specifically, (2) and (6) share a common feasible region (3) and a common optimal solution $\boldsymbol{x}^*$ with the optimal value ζ. We note that QOP (4) (hence (1) is converted to QOP (6) if we define $\boldsymbol{H}_1 = \boldsymbol{Q}_{01} + \sum_{(i,j)\in\mathcal{E}} \boldsymbol{Q}_{ij}$.

3 Main results

We present a class of conic relaxations of QOP (6), which includes the CPP and DNN relaxations of QOP (6), in Sect. 3.1, and their further Lagrangian relaxations, called Lagrangian–conic relaxations, in Sect. 3.2. From the theoretical results shown in Lemmas 2, 3 and 4, we can conclude that the Lagrangian–DNN relaxation of QOP (6) is almost as effective as the DNN relaxation applied to the original QOP (2), and that solving the Lagrangian–DNN relaxation is expected to be more efficient and stable than solving the direct DNN relaxation of (2).

3.1 A class of conic relaxations

We rewrite QOP (2) as

$$\zeta := \min \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{X} \,\middle|\, \begin{array}{l} \boldsymbol{X} \in \boldsymbol{\Gamma},\ \boldsymbol{H}_0 \bullet \boldsymbol{X} = 1, \\ \boldsymbol{Q}_p \bullet \boldsymbol{X} = 0\ (p = 1, 2, \ldots, q) \end{array} \right\}$$

and QOP (6) as

$$\eta := \min \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{X} \,|\boldsymbol{H}_0 \bullet \boldsymbol{X} = 1,\ \boldsymbol{H}_1 \bullet \boldsymbol{X} = 0,\ \boldsymbol{X} \in \boldsymbol{\Gamma} \right\}.$$

Recall that $\boldsymbol{\Gamma} = \left\{ \boldsymbol{x}\boldsymbol{x}^T : \boldsymbol{x} \in \mathbb{R}^n_+ \right\}$. If $\boldsymbol{\Gamma}$ is replaced by a closed convex cone $\mathbb{K}$ in $\mathbb{S}^n$ satisfying $\boldsymbol{\Gamma} \subset \mathbb{K}$, then the following convex relaxations of QOPs (2) and (6) are obtained:

$$\zeta(\mathbb{K}) := \inf \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{X} \,\middle|\, \begin{array}{l} \boldsymbol{X} \in \mathbb{K},\ \boldsymbol{H}_0 \bullet \boldsymbol{X} = 1, \\ \boldsymbol{Q}_p \bullet \boldsymbol{X} = 0\ (p = 1, 2, \ldots, q) \end{array} \right\} \tag{7}$$

$$\eta(\mathbb{K}) := \inf \left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{X} \mid \boldsymbol{H}_0 \bullet \boldsymbol{X} = 1,\ \boldsymbol{H}_1 \bullet \boldsymbol{X} = 0,\ \boldsymbol{X} \in \mathbb{K} \right\}. \tag{8}$$

Note that the dual problem of (8) is given by

$$\eta^d(\mathbb{K}) := \sup \left\{ y_0 \mid \boldsymbol{Z} + y_0 \boldsymbol{H}_0 + y_1 \boldsymbol{H}_1 = \boldsymbol{Q}_0, \boldsymbol{Z} \in \mathbb{K}^*,\ \boldsymbol{y} = (y_0, y_1) \in \mathbb{R}^2 \right\}. \tag{9}$$

When $\mathbb{K}$ is chosen to be $\mathbb{C}^*$, $\mathbb{S}^n_+ \cap \mathbb{N}$ or $\mathbb{S}^n_+$, problem (7) [or problem (8)] is known as a CPP relaxation, a DNN relaxation or an SDP relaxation of QOP (2) [or QOP

(6)], respectively. As the CPP relaxation attains the exact optimal value ζ of QOP (2) [or QOP (6)] (see Theorem 3.5 of [1]), it is theoretically the most important among the three relaxations. It is, however, numerically intractable while the DNN and SDP relaxations are numerically implementable. From $\boldsymbol{\Gamma} \subset \mathbb{C}^* \subset \mathbb{S}^n_+ \cap \mathbb{N} \subset \mathbb{S}^n_+$ and

$$\zeta(\mathbb{S}^n_+) \leq \zeta(\mathbb{S}^n_+ \cap \mathbb{N}) \leq \zeta(\mathbb{C}^*) = \zeta(\boldsymbol{\Gamma}) = \zeta,$$

the DNN relaxation of QOP (2) provides a lower bound for the minimum value ζ of (2) at least as effectively as the SDP relaxation. Furthermore, under Conditions (a), (b) and (c), the Lagrangian–DNN relaxation of (6) [for which (2) is equivalent to] satisfies additional properties which are conducive to numerically solving the problem, especially with first-order methods [4,37].

We present three lemmas to show those properties in the remainder of Sect. 3 for the general case $\mathbb{K}$ where $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$. They are significant in their own right, although a numerical method is proposed only for $\mathbb{K} = \mathbb{S}^n_+ \cap \mathbb{N}$ in Sect. 4.

Lemma 2 *Suppose that $\mathbb{K}$ is a closed (not necessarily convex) cone in $\mathbb{S}^n$ satisfying $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$.*

(i) The feasible region of problem (7) with $\mathbb{K}$ is nonempty and bounded; hence $\zeta(\mathbb{K}) > -\infty$.
(ii) $\eta(\mathbb{K}) = \zeta(\mathbb{K}) \leq \zeta$.

Proof By Condition (a), we know that $\boldsymbol{x}^*(\boldsymbol{x}^*)^T$ is a feasible solution of problem (7) with $\mathbb{K}$. For assertion (i), it suffices to show that the feasible region of problem (7) with $\mathbb{K}$ is bounded. Assume on the contrary that there exists a sequence $\left\{\boldsymbol{X}^k \in \mathbb{K}\right\}$ with $\lim_{k\to\infty} \|\boldsymbol{X}^k\| = \infty$ such that

$$\boldsymbol{H}_0 \bullet \boldsymbol{X}^k = 1 \text{ and } \boldsymbol{Q}_p \bullet \boldsymbol{X}^k = 0 \quad (p = 1, 2, \ldots, q).$$

We may assume without loss of generality that $\boldsymbol{X}^k/\|\boldsymbol{X}^k\|$ converges to some $\boldsymbol{D} \in \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$. Dividing the identities above by $\|\boldsymbol{X}^k\|$ and taking their limit as $k \to \infty$, we have that

$$\boldsymbol{O} \neq \boldsymbol{D} \in \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}, \quad \boldsymbol{H}_0 \bullet \boldsymbol{D} = 0 \text{ and } \boldsymbol{Q}_p \bullet \boldsymbol{D} = 0 \quad (p = 1, 2, \ldots, q).$$

This contradicts the given Condition (c), and we have shown assertion (i).

By the assumption, $\boldsymbol{\Gamma} \subset \mathbb{K}$. Hence $\zeta(\mathbb{K}) \leq \zeta(\boldsymbol{\Gamma}) = \zeta$. Since $\boldsymbol{H}_1 = \sum_{p=1}^q \boldsymbol{Q}_p$, if $\boldsymbol{X} \in \mathbb{K}$ is a feasible solution of (7) with the objective value $\boldsymbol{Q}_0 \bullet \boldsymbol{X}$, then it is a feasible solution of (8) with the same objective value. Thus, $\eta(\mathbb{K}) \leq \zeta(\mathbb{K})$ follows. To prove the converse inequality, suppose that $\boldsymbol{X} \in \mathbb{K}$ is a feasible solution of (8) with $\mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$. By Condition (b), we know $\boldsymbol{Q}_p \in \mathbb{S}^n_+ + \mathbb{N}$ $(p = 1, 2, \ldots, q)$. Hence $\boldsymbol{Q}_p \bullet \boldsymbol{X} \geq 0$ $(p = 1, 2, \ldots, q)$, which together with $0 = \boldsymbol{H}_1 \bullet \boldsymbol{X} = \sum_{p=1}^q (\boldsymbol{Q}_p \bullet \boldsymbol{X})$ imply that $\boldsymbol{Q}_p \bullet \boldsymbol{X} = 0$ $(p = 1, 2, \ldots, q)$. Therefore, $\boldsymbol{X}$ is a feasible solution of (7) with the objective value $\boldsymbol{Q}_0 \bullet \boldsymbol{X}$, and the inequality $\eta(\mathbb{K}) \geq \zeta(\mathbb{K})$ follows. □

Observe that if the relaxation technique discussed above is directly applied to QOP (1), then we have the following problem:

$$\text{minimize} \left\{ \boldsymbol{Q} \bullet \boldsymbol{U} + 2\boldsymbol{c}^T\boldsymbol{u} \;\middle|\; \begin{array}{l} \boldsymbol{X} = \begin{pmatrix} 1 & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix} \in \mathbb{K},\ \boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{0}, \\ \boldsymbol{Q}_{ij} \bullet \boldsymbol{X} = 0\ ((i,j) \in \mathcal{E}) \end{array} \right\}. \tag{10}$$

For $\mathbb{K} = \boldsymbol{\Gamma}$, problems (7) induced from (4) and (10) induced from (1) are equivalent, and both problems represent QOP (1). If we choose a closed convex cone $\mathbb{K}$ with $\boldsymbol{\Gamma} \subset \mathbb{K}$, both problems serve as convex relaxations of QOP (1), but they are not equivalent in general. The essential difference lies in $\boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{0}$ and $\boldsymbol{Q}_{01} \bullet \boldsymbol{X} = 0$. Suppose that $\boldsymbol{X} = \begin{pmatrix} 1 & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix} \in \mathbb{K}$ is a feasible solution of problem (7) with $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+$. Then it satisfies

$$0 = \boldsymbol{Q}_{01} \bullet \begin{pmatrix} 1 & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}^T\boldsymbol{b} & \boldsymbol{b}^T\boldsymbol{A} \\ \boldsymbol{A}^T\boldsymbol{b} & \boldsymbol{A}^T\boldsymbol{A} \end{pmatrix} \bullet \begin{pmatrix} 1 & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix}.$$

Since $\boldsymbol{Q}_{01} \in \mathbb{S}^n_+$ and $\boldsymbol{X} \in \mathbb{S}^n_+$, we see that $\boldsymbol{Q}_{01}\boldsymbol{X} = \boldsymbol{O}$. It follows that

$$\boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} = \boldsymbol{0} \text{ and } \boldsymbol{b}\boldsymbol{u}^T + \boldsymbol{A}\boldsymbol{U} = \boldsymbol{O}.$$

From the first equality above, we know that $\boldsymbol{X}$ is a feasible solution of problem (10); hence problem (7) [hence (8)] provides a convex relaxation at least as good as problem (10). Furthermore, the second equality, which is not involved in (10) unless $\text{rank}(\boldsymbol{X}) = 1$, often contributes to strengthening the relaxation.

3.2 A class of Lagrangian–conic relaxations

For each closed cone $\mathbb{K}$ (not necessarily convex), we consider a Lagrangian relaxation of problem (8) and its dual.

$$\eta^p(\lambda, \mathbb{K}) := \inf\left\{ \boldsymbol{Q}_0 \bullet \boldsymbol{X} + \lambda \boldsymbol{H}_1 \bullet \boldsymbol{X} \mid \boldsymbol{H}_0 \bullet \boldsymbol{X} = 1,\ \boldsymbol{X} \in \mathbb{K} \right\}, \tag{11}$$

$$\eta^d(\lambda, \mathbb{K}) := \sup\left\{ y_0 \mid \boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1 - y_0 \boldsymbol{H}_0 \in \mathbb{K}^* \right\}, \tag{12}$$

where $\lambda \in \mathbb{R}$ denotes a Lagrangian parameter. We call either of (11) or (12) a Lagrangian–conic relaxation of QOP (6), and particularly a Lagrangian–DNN relaxation when $\mathbb{K} = \mathbb{S}^n_+ \cap \mathbb{N}$. (We use these names for simplicity, although (12) is precisely the dual of Lagrangian–conic or Lagrangian–DNN relaxation.) It is easily verified that the weak duality relation $\eta^d(\lambda, \mathbb{K}) \leq \eta^p(\lambda, \mathbb{K}) \leq \eta(\mathbb{K})$ hold for every $\lambda \in \mathbb{R}$. Note that by Condition (b), problem (11) is strictly feasible when $\mathbb{K}$ includes the completely positive cone $\mathbb{C}^*$ with nonempty interior w.r.t. $\mathbb{S}^n$ (see, for example, [15]).

Suppose that $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$. Then we see by Condition (b) that $\boldsymbol{H}_1 \in \mathbb{S}^n_+ + \mathbb{N}$. Hence $\boldsymbol{H}_1 \bullet \boldsymbol{X} \geq 0$ for every $\boldsymbol{X} \in \mathbb{K}$. Thus, the second term $\lambda \boldsymbol{H}_1 \bullet \boldsymbol{X}$ of the objective function of (11) serves as a penalty function for the equality constraint $\boldsymbol{H}_1 \bullet \boldsymbol{X} = 0$ such that for each $\boldsymbol{X} \in \mathbb{K}$.

$$\boldsymbol{H}_1 \bullet \boldsymbol{X} \geq 0 \text{ and } \lambda \boldsymbol{H}_1 \bullet \boldsymbol{X} \to \infty \text{ as } \lambda \to \infty \quad \text{if and only if } \boldsymbol{H}_1 \bullet \boldsymbol{X} \neq 0. \tag{13}$$

By Lemma 2, we also know that $-\infty < \eta(\mathbb{K}) = \zeta(\mathbb{K}) \leq \zeta$. Using these relations, we establish the following result.

Lemma 3 *Suppose that $\mathbb{K}$ is a closed cone (not necessarily convex) in $\mathbb{S}^n$ satisfying $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$. Then the following statements hold.*

(i) $\eta^p(\lambda_1, \mathbb{K}) \leq \eta^p(\lambda_2, \mathbb{K}) \leq \eta(\mathbb{K})$ *if* $\lambda_1 < \lambda_2$.
(ii) $\eta^d(\lambda_1, \mathbb{K}) \leq \eta^d(\lambda_2, \mathbb{K}) \leq \eta(\mathbb{K})$ *if* $\lambda_1 < \lambda_2$.
(iii) $\eta^p(\lambda, \mathbb{K})$ *converges to* $\eta(\mathbb{K})$ *as* $\lambda \to \infty$.
(iv) Moreover, if $\mathbb{K}$ is convex, then both problems (11) and (12) have optimal solutions with a common optimal value $\eta^p(\lambda, \mathbb{K}) = \eta^d(\lambda, \mathbb{K})$ for every sufficiently large $\lambda \in \mathbb{R}$. Thus, "sup" can be replaced by "maximize".

Proof Assertion (i) follows from the first relation of (13) and the weak duality of the Lagrangian relaxation. Assertion (ii) follows from the fact that $\boldsymbol{H}_1 \in \mathbb{S}^n_+ + \mathbb{N} \subset \mathbb{K}^* \subset \boldsymbol{\Gamma}^* = \mathbb{C}$. To prove (iii), define the level set with the objective value $\eta(\mathbb{K}) > -\infty$ by

$$L(\lambda, \mathbb{K}) = \left\{ \boldsymbol{X} \in \mathbb{K} : \boldsymbol{H}_0 \bullet \boldsymbol{X} = 1, \ \boldsymbol{Q}_0 \bullet \boldsymbol{X} + \lambda \boldsymbol{H}_1 \bullet \boldsymbol{X} \leq \eta(\mathbb{K}) \right\}$$

for problem (11) with each $\lambda \in \mathbb{R}$. Then $L(\lambda, \mathbb{K})$ contains an optimal solution $\widetilde{\boldsymbol{X}}$ of (8) for every $\lambda \in \mathbb{R}$, and $L(\lambda_1, \mathbb{K}) \supset L(\lambda_2, \mathbb{K})$ if $0 < \lambda_1 < \lambda_2$. We will show that $L(\lambda, \mathbb{K})$ is bounded for a sufficiently large $\lambda \in \mathbb{R}$. Assume on the contrary that there exists a sequence $\left\{(\lambda^k, \boldsymbol{X}^k) \in \mathbb{R}_+ \times \mathbb{K}\right\}$ such that $\boldsymbol{X}^k \in L(\lambda^k, \mathbb{K})$, $0 < \lambda^k \to \infty$ and $0 < \|\boldsymbol{X}^k\| \to \infty$ as $k \to \infty$. Then, we have

$$\frac{\boldsymbol{X}^k}{\|\boldsymbol{X}^k\|} \in \mathbb{K}, \quad \boldsymbol{H}_1 \bullet \frac{\boldsymbol{X}^k}{\|\boldsymbol{X}^k\|} \geq 0 \text{ (by } \boldsymbol{H}_1 \in \mathbb{S}^n_+ + \mathbb{N}), \quad \boldsymbol{H}_0 \bullet \frac{\boldsymbol{X}^k}{\|\boldsymbol{X}^k\|} = \frac{1}{\|\boldsymbol{X}^k\|}.$$

We may assume without loss of generality that $\boldsymbol{X}^k / \|\boldsymbol{X}^k\|$ converges to a nonzero $\boldsymbol{D} \in \mathbb{K}$. By taking the limit as $k \to \infty$, we obtain that

$$\boldsymbol{O} \neq \boldsymbol{D} \in \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}, \quad \boldsymbol{H}_0 \bullet \boldsymbol{D} = 0, \quad \boldsymbol{H}_1 \bullet \boldsymbol{D} = 0.$$

This contradicts the given Condition (c). Therefore, we have shown that $L(\bar{\lambda}, \mathbb{K})$ is bounded for some sufficiently large $\bar{\lambda} > 0$ and $\widetilde{\boldsymbol{X}} \in L(\lambda, \mathbb{K}) \subset L(\bar{\lambda}, \mathbb{K})$ for every $\lambda \geq \bar{\lambda} > 0$.

Let $\left\{\lambda^k \geq \bar{\lambda}\right\}$ be a sequence that diverges to ∞. Since the nonempty level set $L(\lambda^k, \mathbb{K})$ is contained in a bounded set $L(\bar{\lambda}, \mathbb{K})$, problem (11) with each $\lambda = \lambda^k$ has an optimal solution $\boldsymbol{X}^k$ with the objective value $\eta^p(\lambda^k, \mathbb{K}) = \boldsymbol{Q}_0 \bullet \boldsymbol{X}^k + \lambda^k \boldsymbol{H}_1 \bullet \boldsymbol{X}^k$ in the level set $L(\lambda^k, \mathbb{K})$. We may assume without loss of generality that $\boldsymbol{X}^k$ converges to some $\overline{\boldsymbol{X}} \in L(\bar{\lambda}, \mathbb{K})$. It follows that

$$\begin{aligned}&\boldsymbol{H}_0 \bullet \boldsymbol{X}^k = 1, \quad \frac{\boldsymbol{Q}_0 \bullet \boldsymbol{X}^k}{\lambda^k} + \boldsymbol{H}_1 \bullet \boldsymbol{X}^k \leq \frac{\eta(\mathbb{K})}{\lambda^k}, \\ &\boldsymbol{Q}_0 \bullet \boldsymbol{X}^k \leq \eta(\mathbb{K}), \quad \boldsymbol{H}_1 \bullet \boldsymbol{X}^k \geq 0 \quad \text{[both by Condition (b)].}\end{aligned}$$

By taking the limit as $k \to \infty$, we obtain that

$$\overline{\boldsymbol{X}} \in \mathbb{K},\ \boldsymbol{H}_0 \bullet \overline{\boldsymbol{X}} = 1,\ \boldsymbol{H}_1 \bullet \overline{\boldsymbol{X}} = 0,\ \boldsymbol{Q}_0 \bullet \overline{\boldsymbol{X}} \leq \eta(\mathbb{K}).$$

This implies that $\overline{\boldsymbol{X}}$ is an optimal solution of problem (8). Hence, $\boldsymbol{Q}_0 \bullet \boldsymbol{X}^k$ converges to $\eta(\mathbb{K})$ as $k \to \infty$. We also see from

$$\boldsymbol{Q}_0 \bullet \boldsymbol{X}^k \leq \eta^p(\lambda^k, \mathbb{K}) = \boldsymbol{Q}_0 \bullet \boldsymbol{X}^k + \lambda^k \boldsymbol{H}_1 \bullet \boldsymbol{X}^k \leq \eta(\mathbb{K})$$

that $\eta^p(\lambda^k, \boldsymbol{X}^k)$ converges to $\eta(\mathbb{K})$. Thus, we have shown assertion (iii).

Finally, we prove assertion (iv). We first see that problem (11) has an interior feasible solution by Condition (b). Indeed, if $\widetilde{\boldsymbol{X}}$ is an interior point of $\mathbb{C}^*$, which also lies in the interior of any closed convex cone $\mathbb{K}$ satisfying $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$, then $\boldsymbol{H}_0 \bullet \widetilde{\boldsymbol{X}} > 0$. Hence $\widetilde{\boldsymbol{X}}/\big(\boldsymbol{H}_0 \bullet \widetilde{\boldsymbol{X}}\big)$ serves as an interior feasible solution of problems (11) with $\mathbb{K}$. On the other hand, we have observed in the proof of assertion (iii) above that problem (11) has optimal solutions if $\lambda > \bar{\lambda}$. Therefore, by the duality theorem for linear optimization problems over closed convex cones (see, for example, Theorem 4.2.1 of [25]), we obtain that $\eta^d(\lambda, \mathbb{K}) = \eta^p(\lambda, \mathbb{K})$ for every $\lambda \geq \bar{\lambda}$. □

Lemma 4 *Suppose that $\mathbb{K}$ is a closed convex cone in $\mathbb{S}^n$ satisfying $\boldsymbol{\Gamma} \subset \mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$.*

(i) Suppose $\eta^p(\lambda, \mathbb{K}) < \eta(\mathbb{K})$ for all λ, i.e., $\eta(\mathbb{K})$ is not attained by $\eta^p(\lambda^*, \mathbb{K})$ for any finite λ^*; then the optimal value of problem (9) is not attained at any feasible solution of the problem.

(ii) Suppose that $q \geq 1$; then the feasible set of (8), which is nonempty by Lemma 2, has no feasible point in the interior of $\mathbb{K}$.

Proof (i) We prove the result by contradiction. Suppose (9) attained the optimal value at a $\boldsymbol{y}^* = (y_0^*, y_1^*) \in \mathbb{R}^2$ with $y_0^* = \eta^d(\mathbb{K})$. By definition, it is clear that $\eta^d(-y_1^*, \mathbb{K}) \leq \eta^d(\mathbb{K})$. On the other hand, since y_0^* is feasible for (12) with parameter equal to $-y_1^*$, we have $\eta^d(-y_1^*, \mathbb{K}) \geq y_0^*$. Thus $y_0^* = \eta^d(\mathbb{K}) = \eta^d(-y_1^*, \mathbb{K})$. Now for $\lambda \geq -y_1^*$, we have

$$\eta^d(\mathbb{K}) = \eta^d(-y_1^*, \mathbb{K}) \leq \eta^d(\lambda, \mathbb{K}) \leq \eta^d(\mathbb{K}).$$

Hence $\eta^d(\mathbb{K}) = \eta^d(-y_1^*, \mathbb{K}) = \eta^d(\lambda, \mathbb{K}) = \eta^p(\lambda, \mathbb{K})$ for every sufficiently large $\lambda \geq -y_1^*$, where the last equality follows from assertion (iv) in Lemma 3. By letting $\lambda \to \infty$, we get by using assertion (iii) of Lemma 3 that $\eta^d(-y_1^*, \mathbb{K}) = \eta(\mathbb{K})$. Since $\eta^d(-y_1^*, \mathbb{K}) \leq \eta^p(-y_1^*, \mathbb{K}) \leq \eta(\mathbb{K})$, we deduce that $\eta^p(-y_1^*, \mathbb{K}) = \eta(\mathbb{K})$. However, this contradicts the condition that $\eta^p(\lambda, \mathbb{K}) < \eta(\mathbb{K})$ for all λ. Hence the optimal value $\eta^d(\mathbb{K})$ is not attained.

(ii) Let $\boldsymbol{X}$ be an interior point of $\mathbb{K}$. By Condition (b), we know that $\boldsymbol{O} \neq \boldsymbol{Q}_p \in \mathbb{S}^n_+ + \mathbb{N}$ $(p = 1, 2, \ldots, q)$. Since $\boldsymbol{X}$ lies in the interior of $\mathbb{K}$, there exists a positive number ϵ such that $\boldsymbol{X} - \epsilon \boldsymbol{Q}_p$ $(p = 1, , 2, \ldots, q)$ remain in $\mathbb{K} \subset \mathbb{S}^n_+ \cap \mathbb{N}$; hence $\boldsymbol{Q}_p \bullet (\boldsymbol{X} - \epsilon \boldsymbol{Q}_p) \geq 0$ $(p = 1, 2, \ldots, q)$. It follows that

$$\boldsymbol{Q}_p \bullet \boldsymbol{X} > \boldsymbol{Q}_p \bullet \boldsymbol{X} - \epsilon \boldsymbol{Q}_p \bullet \boldsymbol{Q}_p = \boldsymbol{Q}_p \bullet (\boldsymbol{X} - \epsilon \boldsymbol{Q}_p) \geq 0 \quad (p = 1, 2, \ldots, q).$$

Since $q \geq 1$ by the assumption, we obtain that $\boldsymbol{H}_1 \bullet \boldsymbol{X} = \sum_{p=1}^{q} \boldsymbol{Q}_p \bullet \boldsymbol{X} > 0$, and any interior point of $\mathbb{K}$ cannot be a feasible solution of (8). □

4 Algorithms

Here we design an efficient algorithm to solve problem (12) with $\mathbb{K} = \mathbb{S}^n_+ \cap \mathbb{N}$ [the Lagrangian–DNN relaxation of QOP (6)]. Before we present our algorithm, we should note that (12) with $\mathbb{K} = \mathbb{S}^n_+ \cap \mathbb{N}$ can actually be reformulated as the following standard linear SDP in dual form by introducing an auxiliary variable $\boldsymbol{U}$:

$$\sup\{y_0 \mid y_0 \boldsymbol{H}_0 + \boldsymbol{U} + \boldsymbol{S} = \boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1, -\boldsymbol{U} + \boldsymbol{Z} = \boldsymbol{0}, \boldsymbol{S} \in \mathbb{S}^n_+, \boldsymbol{Z} \in \mathbb{N}\}. \quad (14)$$

By considering $(y_0, \boldsymbol{U})$ and $(\boldsymbol{S}, \boldsymbol{Z})$ as two separate blocks, one can in theory apply the classical convergent 2-block ADMM, which was first introduced by Glowinski and Marrocco [22] and Gabay and Mercier [19], to solve (14). However, the practical performance of the classical 2-block ADMM in solving (14) is far from satisfactory. Typically it has great difficulty in obtaining a dual feasible solution even after a few thousand iterations. As an example, for the quadratic assignment problem `nug25` considered in Table 4, running 2000 ADMM iterations produced an approximate solution $(\hat{y}_0, \hat{\boldsymbol{U}}, \hat{\boldsymbol{S}}, \hat{\boldsymbol{Z}})$ with $\hat{y}_0 = 4.6578 \times 10^3$ and relative dual infeasibility $(\|\hat{\boldsymbol{U}} - \hat{\boldsymbol{Z}}\| + \|\hat{y}_0 \boldsymbol{H}_0 + \hat{\boldsymbol{U}} + \hat{\boldsymbol{S}} - (\boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1)\|) / \|\boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1\| = 1.07 \times 10^{-6}$. Comparing to the optimal value of 3.7440×10^3 for the combinatorial problem, the approximate bound $\hat{y}_0$ produced by the ADMM is not only an invalid bound, but the gap is also much worse than the valid lower bound of 3.6250×10^3 produced by the method we will design below. We should mention that using the convergent ADMM [35] designed to solve a problem of the form (12) also does not produce significant improvement. Finally we have also applied the SDPNAL algorithm [37] to solve (14). Although the performance of SDPNAL is generally much better than ADMM in terms of the achievable accuracy of the computed solution, the accuracy is still not enough to generate a good valid lower bound. For example, for the problem `nug25` just mentioned, SDPNAL took 418 s to generate an approximate solution $\hat{y}_0 = 3.8262 \times 10^3$ (which is again an invalid bound) with relative dual infeasibility of 8.95×10^{-8}. The unsatisfactory performance of the ADMM and SDPNAL in solving (14) thus shows the necessity to design an alternative algorithm to solve the problem.

For the subsequent discussion, we let $\mathbb{K}_1 = \mathbb{S}^n_+$ and $\mathbb{K}_2 = \mathbb{N}$, and hence $\mathbb{K}^* = \mathbb{K}^*_1 + \mathbb{K}^*_2$. We use $\Pi_{\mathbb{K}}(\cdot)$, $\Pi_{\mathbb{K}^*}(\cdot)$, $\Pi_i(\cdot)$ and $\Pi^*_i(\cdot)$ $(i = 1, 2)$ to denote the (metric) projections onto $\mathbb{K}$, $\mathbb{K}^*$, $\mathbb{K}_i$ and $\mathbb{K}^*_i$ $(i = 1, 2)$, respectively.

Suppose for the moment that we have an efficient algorithm to compute $\Pi_{\mathbb{K}^*}(\boldsymbol{G})$ for any given $\boldsymbol{G} \in \mathbb{S}^n$. (Note that to compute $\Pi_{\mathbb{K}^*}(\boldsymbol{G})$, we will present an accelerated proximal gradient method [4] described as Algorithm C in Sect. 4.1.) Then we can design a bisection method to solve (12), as we shall describe next.

For a sufficiently large and fixed $\lambda > 0$, define the function

$$g_\lambda(y_0) = \|\boldsymbol{G}_\lambda(y_0) - \Pi_{\mathbb{K}^*}(\boldsymbol{G}_\lambda(y_0))\| = \|\Pi_{\mathbb{K}}(-\boldsymbol{G}_\lambda(y_0))\|$$

where $\boldsymbol{G}_\lambda(y_0) = \boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1 - y_0 \boldsymbol{H}_0$ and $\| \cdot \|$ is the Frobenius norm induced from the inner product in $\mathbb{S}^n$. It is clear that problem (12) is equivalent to

$$\text{maximize } \{y_0 \mid g_\lambda(y_0) = 0\}.$$

We recall that problem (12) has an optimal solution for every sufficiently larger $\lambda \in \mathbb{R}$ by (iv) of Lemma 4 Thus we can solve (12) by the following simple bisection method.

Algorithm A: **A bisection method for solving (12)**
Choose a sufficiently large $\lambda > 0$, and a small tolerance $\varepsilon > 0$. Set $\boldsymbol{Y}_1^0 = \boldsymbol{0}$. Suppose that an interval $[a_0, b_0]$ has been determined such that $\eta^d(\lambda, \mathbb{K}) \in [a_0, b_0]$. Set $k = 1$. Then perform the following steps at the kth iteration:

```
Step 1. Set $y_0^k = (a_{k-1} + b_{k-1})/2$.
Step 2. Compute $\Pi_{\mathbb{K}^*}(\boldsymbol{G}_\lambda(y_0^k)) = \boldsymbol{Y}_1^k + \boldsymbol{Y}_2^k$ with $\boldsymbol{Y}_i^k \in \mathbb{K}_i^*, i = 1, 2,$ by Algorithm
        C using $\boldsymbol{Y}_1^{k-1}$ as the starting point.
Step 3. If $g_\lambda(y_0^k) < \varepsilon \max\{1, |y_0^k|\}$, set $a_k = y_0^k$; else, set $b_k = y_0^k$.
Step 4. If $b_k - a_k \leq 5 \times 10^{-4} \max\{|a_k|, |b_k|\}$, stop, end.
```

In our numerical experiments, we choose $\varepsilon = 10^{-12}$ in Algorithm A.

To determine an interval $[a_0, b_0]$ which contains $\eta^d(\lambda, \mathbb{K})$, we can loosely solve (12) by Algorithm B described in Sect. 4.1 to produce an $\boldsymbol{X}_0 \in \mathbb{K}$. Assume that $\boldsymbol{H}_0 \bullet \boldsymbol{X}_0 \neq 0$, then we can generate a feasible point $\hat{\boldsymbol{X}}$ for (11) by setting $\hat{\boldsymbol{X}} = \boldsymbol{X}_0/(\boldsymbol{H}_0 \bullet \boldsymbol{X}_0)$. As a result, we have

$$\eta^d(\lambda, \mathbb{K}) \leq \eta^p(\lambda, \mathbb{K}) \leq (\boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1) \bullet \hat{\boldsymbol{X}} =: b_{\text{test}}.$$

The following cases are considered in order to determine the interval $[a_0, b_0]$:

(i) If $b_{\text{test}} \leq -1$, then consider the set

$$\mathcal{J}_- = \left\{ l \mid \kappa^l b_{\text{test}} \text{ is feasible for (12)}, \ l \text{ is a positive integer} \right\}$$

where $\kappa > 1$ is a given constant, say 2. Let l^* be the smallest integer in $\mathcal{J}_-$. We can set $a_0 = \kappa^{l^*} b_{\text{test}}$ and $b_0 = \kappa^{l^*-1} b_{\text{test}}$. Numerically, we regard $\kappa^{-l} b_{\text{test}}$ as feasible if $g_\lambda(\kappa^{-l} b_{\text{test}}) < \varepsilon \max\{1, |\kappa^{-l} b_{\text{test}}|\}$.

(ii) If $b_{\text{test}} \geq 1$, then consider the set

$$\mathcal{J}_+ = \left\{ l \mid \kappa^{-l} b_{\text{test}} \text{ is feasible for (13)}, \ l \text{ is a positive integer} \right\}.$$

If $\mathcal{J}_+$ is nonempty, let l^* be the smallest integer in $\mathcal{J}_+$. We can set $a_0 = \kappa^{-l^*} b_{\text{test}}$ and $b_0 = \kappa^{1-l^*} b_{\text{test}}$. On the other hand, if $\mathcal{J}_+$ is empty, then we know that $\eta^d(\lambda, \mathbb{K}) \leq 0$. Thus if $y_0 = -1$ is feasible for (12), we can set $a_0 = -1, b_0 = 0$. Otherwise, we know that $\eta^d(\lambda, \mathbb{K}) < -1$, and hence we can set $b_{\text{test}} = -1$ and determine a_0, b_0 as in case (i).

(iii) If $b_{\text{test}} \in (-1, 1)$, we can set $b_0 = b_{\text{test}}$ and $a_0 = -1$ if $y_0 = -1$ is feasible for (12). Otherwise, we set $b_{\text{test}} = -1$ and determine a_0, b_0 as in case (i).

4.1 A proximal alternating direction multiplier method for solving (12)

Here we design a proximal alternating direction multiplier (PADM) method [17] for solving the following problem:

$$\text{minimize } \left\{ -\boldsymbol{b}^T \boldsymbol{y} \mid \mathcal{A}^*(\boldsymbol{y}) + \boldsymbol{Z}_1 + \boldsymbol{Z}_2 = \boldsymbol{C},\ \boldsymbol{Z}_1 \in \mathbb{S}^n_+, \boldsymbol{Z}_2 \in \mathbb{N} \right\}, \tag{15}$$

where $\boldsymbol{C} \in \mathbb{S}^n$, $\boldsymbol{b} \in \mathbb{R}^m$ are given data and $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is a given linear map. Note that (12) is a special case of (15) with $\mathcal{A}^*(y_0) = y_0 \boldsymbol{H}_0$, $\boldsymbol{b} = 1$, and $\boldsymbol{C} = \boldsymbol{Q}_0 + \lambda \boldsymbol{H}_1$.

Consider the following augmented Lagrangian function associated with (15):

$$\begin{aligned} L(\boldsymbol{y}, \boldsymbol{Z}_1, \boldsymbol{Z}_2; \boldsymbol{X}) &= -\boldsymbol{b}^T \boldsymbol{y} + \boldsymbol{X} \bullet (\mathcal{A}^* \boldsymbol{y} + \boldsymbol{Z}_1 + \boldsymbol{Z}_2 - \boldsymbol{C}) + \frac{\sigma}{2} \|\mathcal{A}^* \boldsymbol{y} + \boldsymbol{Z}_1 + \boldsymbol{Z}_2 - \boldsymbol{C}\|^2 \\ &= -\boldsymbol{b}^T \boldsymbol{y} + \frac{\sigma}{2} \|\mathcal{A}^* \boldsymbol{y} + \boldsymbol{Z}_1 + \boldsymbol{Z}_2 + \frac{1}{\sigma} \boldsymbol{X} - \boldsymbol{C}\|^2 - \frac{1}{2\sigma} \|\boldsymbol{X}\|^2 \end{aligned} \tag{16}$$

where $\boldsymbol{X} \in \mathbb{S}^n$, $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{Z}_1 \in \mathbb{S}^n_+$, $\boldsymbol{Z}_2 \in \mathbb{N}$, and $\sigma > 0$ is the penalty parameter. Let $\mathcal{T}$ be a given self-adjoint positive semidefinite linear operator defined on $\mathbb{S}^n \times \mathbb{S}^n$. The PADM method solves (15) by performing the following steps at the kth iteration:

$$\begin{aligned} &\boldsymbol{y}^{k+1} = \operatorname{argmin} \left\{ L(\boldsymbol{y}, \boldsymbol{Z}_1^k, \boldsymbol{Z}_2^k; \boldsymbol{X}^k) \ \mid \ \boldsymbol{y} \in \mathbb{R}^m \right\} \\ &(\boldsymbol{Z}_1^{k+1}, \boldsymbol{Z}_2^{k+1}) \\ &\quad = \operatorname{argmin} \left\{ \begin{array}{l} L(\boldsymbol{y}^{k+1}, \boldsymbol{Z}_1, \boldsymbol{Z}_2; \boldsymbol{X}^k) \\ + \frac{\sigma}{2} \begin{pmatrix} \boldsymbol{Z}_1 - \boldsymbol{Z}_1^k \\ \boldsymbol{Z}_2 - \boldsymbol{Z}_2^k \end{pmatrix} \bullet \mathcal{T} \begin{pmatrix} \boldsymbol{Z}_1 - \boldsymbol{Z}_1^k \\ \boldsymbol{Z}_2 - \boldsymbol{Z}_2^k \end{pmatrix} \end{array} \middle| \boldsymbol{Z}_1 \in \mathbb{S}^n_+, \boldsymbol{Z}_2 \in \mathbb{N} \right\} \\ &\quad = \operatorname{argmin} \left\{ \begin{array}{l} 2\boldsymbol{R}_d^k \bullet (\boldsymbol{Z}_1 - \boldsymbol{Z}_1^k + \boldsymbol{Z}_2 - \boldsymbol{Z}_2^k) \\ + \begin{pmatrix} \boldsymbol{Z}_1 - \boldsymbol{Z}_1^k \\ \boldsymbol{Z}_2 - \boldsymbol{Z}_2^k \end{pmatrix} \bullet \left(\begin{pmatrix} \mathcal{I}\ \mathcal{I} \\ \mathcal{I}\ \mathcal{I} \end{pmatrix} + \mathcal{T} \right) \begin{pmatrix} \boldsymbol{Z}_1 - \boldsymbol{Z}_1^k \\ \boldsymbol{Z}_2 - \boldsymbol{Z}_2^k \end{pmatrix} \end{array} \middle| \boldsymbol{Z}_1 \in \mathbb{S}^n_+, \boldsymbol{Z}_2 \in \mathbb{N} \right\} \\ &\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \beta\sigma(\mathcal{A}^* \boldsymbol{y}^{k+1} + \boldsymbol{Z}_1^{k+1} + \boldsymbol{Z}_2^{k+1} - \boldsymbol{C}), \end{aligned}$$

where $\beta \in (0, \frac{1+\sqrt{5}}{2})$ is the step-length and $\boldsymbol{R}_d^k = \mathcal{A}^* \boldsymbol{y}^{k+1} + \boldsymbol{Z}_1^k + \boldsymbol{Z}_2^k + \sigma^{-1} \boldsymbol{X}^k - \boldsymbol{C}$. In our implementation, we fix $\beta = 1.618$.

By specifically choosing $\mathcal{T}$ to be $\mathcal{T} = \begin{pmatrix} \mathcal{I} & -\mathcal{I} \\ -\mathcal{I} & \mathcal{I} \end{pmatrix}$, the computation of $\boldsymbol{Z}_1^{k+1}, \boldsymbol{Z}_2^{k+1}$ above then reduces to computing the projections onto $\mathbb{S}^n_+$ and $\mathbb{N}$, respectively. As a result, we obtain the following efficient PADM method for solving (15). It can be shown that the PADM method converges; we refer the reader to [17] for the proof.

Algorithm B: A PADM method for solving (15)
Choose $\boldsymbol{Z}_1^0 = \boldsymbol{Z}_2^0 = \boldsymbol{X}^0 = \boldsymbol{0}$, and $\sigma > 0$, iterate the following steps:

Step 1. Compute $\boldsymbol{y}^{k+1}$ by solving the following linear system of equations:

$$\mathcal{A}\mathcal{A}^*\boldsymbol{y} = \frac{1}{\sigma}(\boldsymbol{b} - \mathcal{A}(\boldsymbol{X}^k)) - \mathcal{A}(\boldsymbol{Z}_1^k + \boldsymbol{Z}_2^k - \boldsymbol{C})$$

Step 2. Compute

$$\boldsymbol{Z}_1^{k+1} = \operatorname{argmin}\left\{\boldsymbol{R}_d^k \bullet (\boldsymbol{Z}_1 - \boldsymbol{Z}_1^k) + \|\boldsymbol{Z}_1 - \boldsymbol{Z}_1^k\|^2 \mid \boldsymbol{Z}_1 \in \mathbb{S}_+^n\right\} = \Pi_{\mathbb{S}_+^n}\left(\boldsymbol{Z}_1^k - \frac{1}{2}\boldsymbol{R}_d^k\right)$$
$$\boldsymbol{Z}_2^{k+1} = \operatorname{argmin}\left\{\boldsymbol{R}_d^k \bullet (\boldsymbol{Z}_2 - \boldsymbol{Z}_2^k) + \|\boldsymbol{Z}_2 - \boldsymbol{Z}_2^k\|^2 \mid \boldsymbol{Z}_2 \in \mathbb{N}\right\} = \Pi_{\mathbb{N}}\left(\boldsymbol{Z}_2^k - \frac{1}{2}\boldsymbol{R}_d^k\right).$$

Step 3. Compute

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \beta\sigma\left(\mathcal{A}^*\boldsymbol{y}^{k+1} + \boldsymbol{Z}_1^{k+1} + \boldsymbol{Z}_2^{k+1} - \boldsymbol{C}\right).$$

We can apply Algorithm B to (12) by taking $\boldsymbol{C} = \boldsymbol{Q}_0 + \lambda\boldsymbol{H}_1$, $\mathcal{A}(\boldsymbol{X}) = \boldsymbol{H}_0 \bullet \boldsymbol{X}$ and $\boldsymbol{b} = 1 \in \mathbb{R}$. Note that since the dual variable $\boldsymbol{y}$ is one-dimensional, solving the linear system of equation in Step 1 is very easy.

4.2 Computing $\Pi_{\mathbb{K}}(\boldsymbol{G})$ and $\Pi_{\mathbb{K}^*}(-\boldsymbol{G})$

Observe that checking whether a given scalar y_0 is feasible for (12) amounts to checking whether $\boldsymbol{G}_\lambda(y_0) = \boldsymbol{Q}_0 + \lambda\boldsymbol{H}_1 - y_0\boldsymbol{H}_0 \in \mathbb{K}^*$, i.e., whether $\Pi_{\mathbb{K}}(-\boldsymbol{G}_\lambda(y_0)) = \boldsymbol{O}$. To compute $\Pi_{\mathbb{K}}(\boldsymbol{G})$ for a given $\boldsymbol{G} \in \mathbb{S}^n$, we consider the following projection problem:

$$\text{minimize}\left\{\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{G}\|^2 \mid \boldsymbol{X} \in \mathbb{K}\right\} \tag{17}$$

where the unique solution gives the projection $\Pi_{\mathbb{K}}(\boldsymbol{G})$ of $\boldsymbol{G}$ onto $\mathbb{K}$. Note that for any $\boldsymbol{G} \in \mathbb{S}^n$, we have $\boldsymbol{G} = \Pi_{\mathbb{K}}(\boldsymbol{G}) - \Pi_{\mathbb{K}^*}(-\boldsymbol{G})$ by the decomposition theorem of Moreau [23]. Using this equality, we can prove that $\boldsymbol{X} = \Pi_{\mathbb{K}}(\boldsymbol{G})$, $\boldsymbol{Y}_1 + \boldsymbol{Y}_2 = \Pi_{\mathbb{K}^*}(-\boldsymbol{G})$, $\boldsymbol{Y}_1 \in \mathbb{K}_1^*$ and $\boldsymbol{Y}_2 \in \mathbb{K}_2^*$ if and only if

$$\begin{gathered}\boldsymbol{X} - \boldsymbol{G} - \boldsymbol{Y}_1 - \boldsymbol{Y}_2 = \boldsymbol{0}, \quad \boldsymbol{X} \bullet \boldsymbol{Y}_1 = 0, \ \boldsymbol{X} \bullet \boldsymbol{Y}_2 = 0, \\ \boldsymbol{X} \in \mathbb{K}_1, \ \boldsymbol{X} \in \mathbb{K}_2, \ \boldsymbol{Y}_1 \in \mathbb{K}_1^*, \ \boldsymbol{Y}_2 \in \mathbb{K}_2^*,\end{gathered} \tag{18}$$

which can also be derived as the KKT conditions for (17).

In Algorithm C described below, we will measure the accuracy of an approximation $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}_1, \hat{\boldsymbol{Y}}_2)$ of $(\boldsymbol{X}, \boldsymbol{Y}_1, \boldsymbol{Y}_2)$ satisfying (18) by computing the following residual:

$$\delta_{\mathbb{K}} = \frac{1}{1+\|\boldsymbol{G}\|} \max \left\{ \begin{array}{l} \|\hat{\boldsymbol{X}} - \boldsymbol{G} - \hat{\boldsymbol{Y}}_1 - \hat{\boldsymbol{Y}}_2\|, \frac{|\langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}_1 \rangle|}{1+\|\hat{\boldsymbol{Y}}_1\|}, \frac{|\langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}_2 \rangle|}{1+\|\hat{\boldsymbol{Y}}_2\|}, \\ \|\Pi_1^*(-\hat{\boldsymbol{X}})\|, \|\Pi_2^*(-\hat{\boldsymbol{X}})\|, \|\Pi_1(-\hat{\boldsymbol{Y}}_1)\|, \|\Pi_2(-\hat{\boldsymbol{Y}}_2)\| \end{array} \right\}. \tag{19}$$

It turns out that to solve (17) for a given $\boldsymbol{G} \in \mathbb{S}^n$, it is more efficient to consider the following problem for computing the projection of $-\boldsymbol{G}$ onto $\mathbb{K}^* = \mathbb{K}_1^* + \mathbb{K}_2^*$:

$$\text{minimize} \left\{ \text{minimize} \left\{ \frac{1}{2}\|\boldsymbol{G} + \boldsymbol{Y}_1 + \boldsymbol{Y}_2\|^2 \mid \boldsymbol{Y}_2 \in \mathbb{K}_2^* \right\} \mid \boldsymbol{Y}_1 \in \mathbb{K}_1^* \right\},$$

which can equivalently be formulated as follows:

$$\text{minimize} \left\{ f(\boldsymbol{Y}_1) := \frac{1}{2}\|\Pi_2(\boldsymbol{G} + \boldsymbol{Y}_1)\|^2 \mid \boldsymbol{Y}_1 \in \mathbb{K}_1^* \right\}. \tag{20}$$

It is easy to show that if $\boldsymbol{Y}_1^*$ is an optimal solution of (20), then $\boldsymbol{X}^* = \Pi_2(\boldsymbol{G} + \boldsymbol{Y}_1^*)$ is an optimal solution of (17). It can be shown that $f(\cdot)$ is continuously differentiable on $\mathbb{S}^n$ with $\nabla f(\boldsymbol{Y}_1) = \Pi_2(\boldsymbol{G} + \boldsymbol{Y}_1)$, and the gradient is Lipschitz continuous with modulus $L = 1$. However, note that $f(\cdot)$ is not twice continuously differentiable on $\mathbb{S}^n$.

The KKT conditions for (20) are given by:

$$\Pi_2(\boldsymbol{G} + \boldsymbol{Y}_1) - \boldsymbol{X} = \boldsymbol{0}, \quad \boldsymbol{Y}_1 \bullet \boldsymbol{X} = 0, \quad \boldsymbol{Y}_1 \in \mathbb{K}_1^*, \ \boldsymbol{X} \in \mathbb{K}_1.$$

To solve problem (20), we can either use a projected gradient method [5] or the accelerated proximal gradient (APG) method in [4]. We prefer the latter because of its superior iteration complexity. The details of the APG method is described next.

Algorithm C: An accelerated proximal gradient method for solving (20)

Choose $\bar{\boldsymbol{Y}}_1^0 = \boldsymbol{Y}_1^0 \in \mathbb{S}^n$ and a small tolerance $\epsilon > 0$; set $t_0 = 1$. Iterate the following steps:

```
Step 1. Compute $\nabla f(\bar{\boldsymbol{Y}}_1^k) = \Pi_2(\boldsymbol{G} + \bar{\boldsymbol{Y}}_1^k) = \max\{\boldsymbol{G} + \bar{\boldsymbol{Y}}_1^k, \boldsymbol{O}\}$.
Step 2. Set $\boldsymbol{H}^k = \bar{\boldsymbol{Y}}_1^k - L^{-1}\nabla f(\bar{\boldsymbol{Y}}_1^k)$. Compute $\boldsymbol{Y}_1^{k+1} = \Pi_1^*(\boldsymbol{H}^k)$.
Step 3. Set $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$ and $\bar{\boldsymbol{Y}}_1^{k+1} = \boldsymbol{Y}_1^{k+1} + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{Y}_1^{k+1} - \boldsymbol{Y}_1^k)$.
Step 4. Set $(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}_1, \hat{\boldsymbol{Y}}_2) = (\Pi_2(\boldsymbol{G} + \boldsymbol{Y}_1^{k+1}), \boldsymbol{Y}_1^{k+1}, \Pi_2^*(-\boldsymbol{G} - \boldsymbol{Y}_1^{k+1}))$. Compute
        the residual $\delta_{\mathbb{K}}$ defined in (19).
Step 5. If $\delta_{\mathbb{K}} \leq \epsilon$, stop, end.
```

We note that Algorithm C is called in Algorithm A with $\boldsymbol{G} = -\boldsymbol{G}_\lambda(y_0^k)$ to determine whether y_0^k is feasible for (12). It is also used to check whether $\kappa^l b_{test}$ and $\kappa^{-l} b_{test}$ are feasible for (12). As Algorithm C generates approximations of $\Pi_{\mathbb{K}}(\boldsymbol{G})$ and $\Pi_{\mathbb{K}^*}(-\boldsymbol{G})$, it is important to pick a small tolerance ϵ (10^{-12} in our numerical experiments) in the algorithm in order to determine the feasibility of y_0^k unambiguously.

We close this section with the following iteration complexity result for Algorithm C.

Theorem 1 *Let $\{\boldsymbol{Y}_1^k\}_{k=0}^{\infty}$ be generated by Algorithm C. Then for any $k \geq 1$, we have*

$$0 \leq f(\boldsymbol{Y}_1^k) - f(\boldsymbol{Y}_1^*) \leq \frac{2\|\boldsymbol{Y}_1^k - \boldsymbol{Y}_1^0\|^2}{(k+1)^2}.$$

Proof It follows from [4, Theorem 4.1] by noting that $\nabla f(\cdot)$ is Lipschitz continuous with modulus $L = 1$. □

5 Numerical experiments

To apply the numerical methods presented in the previous section, we take $\mathbb{K} = \mathbb{S}_+^n \cap \mathbb{N}$. We demonstrate the efficiency and effectiveness of solving the Lagrangian–DNN relaxation (12) using Algorithm A by comparing with the DNN relaxation (7) of (6) in terms of the quality of lower bounds and CPU time.

The test problems include the binary integer quadratic problems, the quadratic multiple knapsack problems, the maximum stable set problems, and the quadratic assignment problems. All the experiments were performed in MATLAB on a Dell Precision T3500 Desktop with Intel Xeon quad-core CPU (2.80 GHZ) and 24 GB memory.

5.1 Binary integer quadratic problems

The binary integer quadratic (BIQ) problem is described as

$$v^* = \min \{\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} \mid \boldsymbol{x} \in \{0,1\}^m\}, \tag{21}$$

where $\boldsymbol{Q}$ is an $m \times m$ symmetric matrix (not necessarily positive semidefinite). By [9], a natural DNN relaxation of problem (21) is given by:

$$v^{(0)} := \min \left\{ \boldsymbol{Q} \bullet \boldsymbol{X} \;\middle|\; \begin{array}{l} \mathrm{diag}(\boldsymbol{X}) - \boldsymbol{x} = 0, \\ \begin{bmatrix} 1 & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{bmatrix} \in \mathbb{S}_+^{m+1} \cap \mathbb{N}^{m+1} \end{array} \right\}. \tag{22}$$

Note that by introducing slack variables, $v_i = 1 - x_i$, $i = 1, \ldots, m$, (21) can be reformulated in the form (1) as follows:

$$v^* = \min \{\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} \mid \boldsymbol{x} + \boldsymbol{v} = \boldsymbol{e},\; \boldsymbol{x} \circ \boldsymbol{v} = 0,\; \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{v} \geq \boldsymbol{0}\}. \tag{23}$$

where $\circ$ denotes the operation of performing elementwise multiplication. Observe that the direct DNN relaxation of the form (10) for problem (23) is given by

$$v^{(1)} := \min \left\{ \boldsymbol{Q} \bullet \boldsymbol{X} \;\middle|\; \begin{array}{l} \operatorname{diag}(\boldsymbol{X}) - \boldsymbol{x} = 0,\ \operatorname{diag}(\boldsymbol{V}) - \boldsymbol{v} = 0,\ \operatorname{diag}(\boldsymbol{W}) = 0 \\ \boldsymbol{x} + \boldsymbol{v} = \boldsymbol{e},\ \begin{bmatrix} 1 & \boldsymbol{x}^T & \boldsymbol{v}^T \\ \boldsymbol{x} & \boldsymbol{X} & \boldsymbol{W}^T \\ \boldsymbol{v} & \boldsymbol{W} & \boldsymbol{V} \end{bmatrix} \in \mathbb{S}_+^{2m+1} \cap \mathbb{N}^{2m+1} \end{array} \right\}. \tag{24}$$

We can consider the relaxation of the form (7) for (23) with $\mathbb{K} = \mathbb{S}_+^{2m+1} \cap \mathbb{N}^{2m+1}$, and the subsequent Lagrangian–DNN relaxation (11) and its dual (12) to generate a lower bound $v^{(2)}$ for (21).

Table 1 presents the numerical results we obtain for solving (22), the direct DNN relaxation (24), and the Lagrangian–DNN relaxation of (23). The test problems are the Billionnet–Elloumi instances from BIQMAC library [6]. The lower bounds $v_{\text{LB}}^{(0)}$ obtained from (22) are the current state-of-the-art, and they are computed by the advanced large scale SDP solver, SDPNAL, originally developed in [37]. The stopping tolerance for SDPNAL is set to 10^{-5} rather than the default value of 10^{-6} since the former is more efficient for the purpose of computing a lower bound for (22). We should mention that as the solver SDPNAL only produces an approximately primal–dual feasible solution for (22), the procedure in [21] is used to generate a valid lower bound $v_{\text{LB}}^{(0)}$ for $v^{(0)}$ based on the approximately feasible solution.

From the table, we may observe that the Lagrangian–DNN relaxation (12) of (23) can produce much stronger lower bounds $v_{\text{LB}}^{(2)}$ than the bounds $v_{\text{LB}}^{(0)}$ generated from the standard DNN relaxation problem (22), while the CPU times taken to compute the bounds $v_{\text{LB}}^{(2)}$ by Algorithm A are at most 2.5 times that taken to compute $v_{\text{LB}}^{(0)}$ by SDPNAL. The fact that more time is needed to compute $v_{\text{LB}}^{(2)}$ should not come as a surprise since the matrix variable involved has dimension $2m + 1$ whereas the corresponding variable for $v_{\text{LB}}^{(0)}$ has dimension $m + 1$. But we have seen that the

Table 1 BIQ problems with $m = 100$ and $m = 250$: $\lambda = 10^6 \|\boldsymbol{Q}_0\| / \|\boldsymbol{H}_1\|$

Problem	Optimal value v^*	$v_{\text{LB}}^{(0)}$ (SDPNAL)	Time	$v_{\text{LB}}^{(1)}$ (SDPNAL)	Time	$v_{\text{LB}}^{(2)}$ (Algo. A)	Time
bqp100-1	−7.97000000e3	−8.380978995e3	27	−8.071032266e3	102	−8.04687500e3	29
bqp100-2	−1.10360000e4	−1.148973750e4	28	−1.108631912e4	77	−1.10449219e4	26
bqp100-3	−1.27230000e4	−1.315374085e4	36	−1.272419250e4	148	−1.27246094e4	29
bqp100-4	−1.03680000e4	−1.073250841e4	33	−1.036927798e4	147	−1.03710937e4	28
bqp100-5	−9.08300000e3	−9.487049678e3	41	−9.115484423e3	64	−9.08935547e3	30
bqp250-1	−4.56070000e4	−4.766681027e4	201	−4.637834459e4	621	−4.62695313e4	272
bqp250-2	−4.48100000e4	−4.722281064e4	202	−4.573238489e4	720	−4.56054688e4	377
bqp250-3	−4.90370000e4	−5.108018909e4	167	−4.959841323e4	685	−4.94921875e4	420
bqp250-4	−4.12740000e4	−4.331281079e4	271	−4.213361046e4	778	−4.20507812e4	303
bqp250-5	−4.79610000e4	−5.000450078e4	259	−4.857234124e4	671	−4.84570313e4	398

The values $v_{\text{LB}}^{(0)}$ are computed based on (22) using the SDPNAL algorithm in [37]. Similarly, the values $v_{\text{LB}}^{(1)}$ in (24) are computed using SDPNAL formulation (23) is able to produce much stronger lower bound than that produced by the standard DNN relaxation (22) of (21).

Next we test the strength of the Lagrangian–DNN relaxation (12) of (23) by comparing the lower bound $v_{\rm LB}^{(2)}$ with the lower bound $v_{\rm LB}^{(1)}$ produced by the direct DNN relaxation (24) of (23). Notice that the bounds $v_{\rm LB}^{(2)}$ are tighter than $v_{\rm LB}^{(1)}$ for almost all test problems, except for bqp100-3 and bqp100-4 where the differences between $v_{\rm LB}^{(2)}$ and $v_{\rm LB}^{(1)}$ are very small. The bounds $v_{\rm LB}^{(1)}$ are now much closer to $v_{\rm LB}^{(2)}$ compared to the differences between $v_{\rm LB}^{(0)}$ and $v_{\rm LB}^{(2)}$, however, computing $v_{\rm LB}^{(1)}$ takes much longer time than $v_{\rm LB}^{(2)}$ or $v_{\rm LB}^{(0)}$. Thus we see that solving the Lagrangian–DNN relaxation of (23) by our proposed Algorithm A can generate the tight lower bounds $v_{\rm LB}^{(2)}$ efficiently.

5.2 Quadratic multiple knapsack problems

The problem under consideration is the following:

$$v^* := \min \left\{ \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} \;\middle|\; \boldsymbol{A}\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{b},\ \boldsymbol{x} \in \{0, 1\}^m, \boldsymbol{s} \geq 0 \right\}, \tag{25}$$

where $\boldsymbol{A} \in \mathbb{R}^{q\times m}$ and $\boldsymbol{b} \in \mathbb{R}^q$ have their entries all being positive. The problem (25), studied in [10,29], is a generalization of the binary quadratic single knapsack problem. In our numerical experiments, we set $q = 10$, $m = 100$, and use the same matrix $\boldsymbol{Q}$ with $m = 100$ as in the BIQ problems in Sect. 5.1. The matrix $\boldsymbol{A}$ is generated randomly with its elements independently drawn from the uniformly distribution on the interval [0, 10]. The vector $\boldsymbol{b}$ is set to $\boldsymbol{b} = m\boldsymbol{e}$.

By the formulation of Burer in [9], the natural DNN relaxation of (25) is given by

$$v^{(0)} := \min \left\{ \boldsymbol{Q} \bullet \boldsymbol{X} \;\middle|\; \begin{array}{l} \boldsymbol{A}\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{b},\ \mathrm{diag}(\boldsymbol{X}) - \boldsymbol{x} = 0 \\ \left(\begin{bmatrix} \boldsymbol{a}_i \\ \boldsymbol{e}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_i \\ \boldsymbol{e}_i \end{bmatrix}^T \right) \bullet \begin{bmatrix} \boldsymbol{X} & \boldsymbol{W}^T \\ \boldsymbol{W} & \boldsymbol{S} \end{bmatrix} = b_i^2 \ (i = 1, \ldots, q) \\ \begin{bmatrix} 1 & \boldsymbol{x}^T & \boldsymbol{s}^T \\ \boldsymbol{x} & \boldsymbol{X} & \boldsymbol{W}^T \\ \boldsymbol{s} & \boldsymbol{W} & \boldsymbol{S} \end{bmatrix} \in \mathbb{S}^{m+q+1} \cap \mathbb{N}^{m+q+1} \end{array} \right\} \tag{26}$$

where $\boldsymbol{a}_i^T$ denotes the ith row of $\boldsymbol{A}$, and $\boldsymbol{e}_i$ is the ith unit vector in $\mathbb{R}^q$. We can introduce slack variables, $v_i = 1 - x_i$, $i = 1, \ldots, m$, just as for the BIQ problems in Sect. 5.1, to reformulate (27) into the form (1) as follows:

$$v^* = \min \left\{ \boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} \;\middle|\; \begin{array}{l} \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{I}_m & 0 \\ A & 0 & \boldsymbol{I}_q \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{v} \\ \boldsymbol{s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{b} \end{bmatrix}, \\ \boldsymbol{x} \circ \boldsymbol{v} = 0,\ \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{v} \geq \boldsymbol{0}, \boldsymbol{s} \geq \boldsymbol{0} \end{array} \right\}. \tag{27}$$

Table 2 Quadratic multiple knapsack problems: $\lambda = 10^6 \|\boldsymbol{Q}_0\|/\|\boldsymbol{H}_1\|$

Problem	$v_{\rm LB}^{(0)}$ (SDPNAL)	Time	$v_{\rm LB}^{(1)}$ (SDPNAL)	Time	$v_{\rm LB}^{(2)}$ (Algo. A)	Time
qmk100-1	−3.839274027e3	53	−3.696358669e3	185	−3.64746094e3	34
qmk100-2	−4.700674021e3	44	−4.558158548e3	145	−4.51171875e3	36
qmk100-3	−4.800266728e3	50	−4.679702850e3	177	−4.62890625e3	71
qmk100-4	−4.876050590e3	61	−4.732551889e3	182	−4.67651367e3	54
qmk100-5	−4.009051102e3	44	−3.883856215e3	172	−3.82690430e3	53
qmk250-1	−1.944972732e4	152	−1.924816440e4	900	−1.90136719e4	447
qmk250-2	−1.978813105e4	141	−1.960199989e4	1005	−1.92968750e4	460
qmk250-3	−2.024463770e4	130	−2.003825141e4	846	−1.97949219e4	441
qmk250-4	−1.878872520e4	154	−1.862331681e4	1041	−1.83935547e4	485
qmk250-4	−1.984494800e4	136	−1.967123749e4	1223	−1.94433594e4	441

The values $v_{\rm LB}^{(0)}$ are computed based on (26) using the SDPNAL algorithm in [37]

In the numerical results presented in Table 2, we report the lower bound $v_{\rm LB}^{(0)}$ computed by SDPNAL for (26), the lower bound $v_{\rm LB}^{(1)}$ computed by SDPNAL for the DNN relaxation of (27) based on Burer's formulation in [9], and the lower bound $v_{\rm LB}^{(2)}$ computed by Algorithm A for the Lagrangian–DNN relaxation associated with (27).

As we can see from the numerical results that the lower bound $v_{\rm LB}^{(0)}$ is much weaker than $v_{\rm LB}^{(1)}$ and $v_{\rm LB}^{(2)}$; thus we will mainly compare the bounds $v_{\rm LB}^{(1)}$ and $v_{\rm LB}^{(2)}$. Table 2 shows that the bound $v_{\rm LB}^{(2)}$ based on the Lagrangian–DNN relaxation introduced in this paper can be computed much more efficiently than the lower bound $v_{\rm LB}^{(1)}$. Furthermore, $v_{\rm LB}^{(2)}$ is stronger than $v_{\rm LB}^{(1)}$.

5.3 Maximum stable set problems

For a graph G with m nodes and edge set $\mathcal{E}$, the stability number $\alpha(G)$ is the cardinality of a maximal stable set of G [12], and

$$-\alpha(G) := \min\,\{-\boldsymbol{e}^T\boldsymbol{x} \;\mid\; x_i x_j = 0 \;((i,j) \in \mathcal{E}),\; \boldsymbol{x} \in \{0,1\}^m\} \tag{28}$$

$$\geq\; \min\left\{ (-\boldsymbol{e}\boldsymbol{e}^T) \bullet (\boldsymbol{z}\boldsymbol{z}^T) \;\middle|\; \begin{array}{c} \sum_{i=1}^{n} z_i^2 = 1,\;\; z_i z_j = 0 \\ ((i,j) \in \mathcal{E}),\;\; \boldsymbol{z} \geq \boldsymbol{0} \end{array} \right\}. \tag{29}$$

Note that (29) is derived from (28) by setting $\boldsymbol{z} = \boldsymbol{x}/\sqrt{\boldsymbol{e}^T\boldsymbol{x}}$ for $0 \neq \boldsymbol{x} \in \{0,1\}^m$.

Let $\boldsymbol{E}_{ij}$ be the $m \times m$ symmetric matrix whose elements are all zeros, except the (i,j) and (j,i) elements which are equal to 1. By setting

$$\boldsymbol{Q}_0 = -\boldsymbol{e}\boldsymbol{e}^T, \quad \boldsymbol{H}_0 = \boldsymbol{I}, \quad \boldsymbol{H}_1 = \sum_{(ij)\in\mathcal{E}} \boldsymbol{E}_{ij} \tag{30}$$

Table 3 Maximum stable set problems with $n = 256, 512$ and 1024: $\lambda = 10^6\|\boldsymbol{Q}_0\|/\|\boldsymbol{H}_1\|$

Problem	Optimal value v^*	$v_{\text{LB}}^{(1)}$ (SDPNAL)	Time	$v_{\text{LB}}^{(2)}$ (Algo. A)	Time
1dc.256	−30	−3.002325924e1	38	−3.00048828e1	15
1et.256	−50	−5.447886964e1	20	−5.44921875e1	28
1tc.256	−63	−6.325525918e1	22	−6.32812500e1	34
1zc.256	−36	−3.733740409e1	8	−3.73413086e1	10
1dc.512	−52	−5.272842750e1	160	−5.27099609e1	125
1et.512	−100	−1.035793061e2	136	−1.03613281e2	200
1tc.512	−110	−1.125746351e2	371	−1.12841797e2	305
1zc.512	−62	−6.800808859e1	87	−6.80175781e1	158
1dc.1024	−94	−9.557482297e1	2570	−9.55810547e1	1182
1et.1024	−171	−1.820965698e2	1032	−1.82128906e2	1398
1tc.1024	−196	−2.042445571e2	4443	−2.04589844e2	1811
1zc.1024	−112	−1.280028889e2	514	−1.28051758e2	233

The values $v_{\text{LB}}^{(1)}$ are computed based on (31) using the SDPNAL algorithm in [37]

it is readily shown that (29) is equivalent to problem (8) with $\mathbb{K} = \boldsymbol{\Gamma}$. A well-known DNN relaxation of (29) is the following:

$$v^{(1)} = \min\left\{\boldsymbol{Q}_0 \bullet \boldsymbol{X} \;\middle|\; \boldsymbol{I} \bullet \boldsymbol{X} = 1,\ \boldsymbol{E}_{ij} \bullet \boldsymbol{X} = 0\ ((i,j) \in \mathcal{E}),\ \boldsymbol{X} \in \mathbb{S}^n_+ \cap \mathbb{N}\right\}. \quad (31)$$

Table 3 presents the comparison of the lower bounds and computation times for problem (29). The test problems are graph instances (arising from coding theory) collected by Sloane [30]. The lower bounds $v_{\text{LB}}^{(1)}$ we generated from (31) are the current state-of-the-art, and they are computed by the advanced large scale SDP solver, SDPNAL. The lower bounds $v_{\text{LB}}^{(2)}$ are computed for the problem (29) based on the Lagrangian–DNN relaxation (12) by Algorithm A. We can see that the difference between the bounds $v_{\text{LB}}^{(1)}$ and $v_{\text{LB}}^{(2)}$ is less than 10^{-2} for all test problems, and the CPU times to compute $v_{\text{LB}}^{(1)}$ and $v_{\text{LB}}^{(2)}$ are comparable, except for the problems 1dc.1024, 1tc.1024, 1zc.1024. For these problems, computing $v_{\text{LB}}^{(1)}$ by SDPNAL takes more than twice time computing $v_{\text{LB}}^{(2)}$ by Algorithm A.

We have also tested Algorithm A for solving the `Hamming 6-4` instance, a graph with 64 vertices and $\alpha(G) = 4$ from the Second DIMACS Challenge [14]. For this problem, it was reported in [8] that the linear approximation method took about 58 min to solve the problem. For our algorithm, the problem (12) was solved in less than 2 s and generated the lower bound of -4.00024414 for $-\alpha(G)$.

5.4 Quadratic assignment problems

We will identify a matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n\times n}$ with the n^2-vector $\boldsymbol{x} = \text{vec}(\boldsymbol{X}) = [\boldsymbol{x}_1; \ldots; \boldsymbol{x}_n]$. Given matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$, the quadratic assignment problem is:

$$\begin{aligned} v^* &:= \min \left\{ \boldsymbol{X} \bullet \boldsymbol{A}\boldsymbol{X}\boldsymbol{B}^T \;\middle|\; \begin{array}{l} \boldsymbol{e}^T \boldsymbol{X}\boldsymbol{e}_j = 1 = \boldsymbol{e}_j^T \boldsymbol{X}\boldsymbol{e} \\ (j = 1, \ldots, n),\;\; \boldsymbol{X} \in \{0, 1\}^{n \times n} \end{array} \right\} \\ &= \min \left\{ \boldsymbol{x}^T (\boldsymbol{B} \otimes \boldsymbol{A})\boldsymbol{x} \;\middle|\; \begin{array}{l} (\boldsymbol{e}_j^T \otimes \boldsymbol{e}^T)\boldsymbol{x} = 1 = (\boldsymbol{e}^T \otimes \boldsymbol{e}_j^T)\boldsymbol{x} \\ (j = 1, \ldots, n),\;\; \boldsymbol{x} \in \{0, 1\}^{n^2} \end{array} \right\}. \end{aligned} \tag{32}$$

Here $\boldsymbol{e} \in \mathbb{R}^n$ denotes the vector of ones, $\boldsymbol{e}_j \in \mathbb{R}^n$ the jth coordinate Unit vector, and $\otimes$ denotes the Kronecker product. We can express the above problem in the form of (1) by introducing an additional n vector variables $\boldsymbol{v}_i = \boldsymbol{e} - \boldsymbol{x}_i \; i = 1, \ldots, n$, but the resulting Lagrangian–DNN relaxation (12) of (1) will involve matrices with dimensions $2n^2 \times 2n^2$. From the computational point of view, such a doubling of matrix dimension is expensive, and our numerical experience also show that it is costly and difficult to solve (12) to high accuracy with such a formulation. For this reason, we will consider the following alternative reformulation of (32) introduced by Povh and Rendl [26]:

$$v^* := \min \left\{ \boldsymbol{X} \bullet (\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}^T) \;\mid\; \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I} = \boldsymbol{X}\boldsymbol{X}^T,\, \boldsymbol{X} \geq 0 \right\} \tag{33}$$

$$= \min \left\{ (\boldsymbol{B} \otimes \boldsymbol{A}) \bullet \boldsymbol{Y} \;\middle|\; \begin{array}{l} \sum_{i=1}^n \boldsymbol{Y}^{ii} = \boldsymbol{I},\; \boldsymbol{I} \bullet \boldsymbol{Y}^{ij} = \delta_{ij} \; (1 \leq i,\; j \leq n), \\ \boldsymbol{E} \bullet \boldsymbol{Y} = n^2,\; \boldsymbol{Y} \in \mathbb{C}^* \end{array} \right\}. \tag{34}$$

Here $\boldsymbol{Y}^{ij}$ corresponds to the (i, j)-block in the expansion of $\boldsymbol{x}\boldsymbol{x}^T$ as an $n^2 \times n^2$ matrix, and δ denotes Kronecker's delta. A natural DNN relaxation of (34) is following:

$$v^{(1)} := \min \left\{ (\boldsymbol{B} \otimes \boldsymbol{A}) \bullet \boldsymbol{Y} \;\middle|\; \begin{array}{l} \sum_{i=1}^n \boldsymbol{Y}^{ii} = \boldsymbol{I},\; \boldsymbol{I} \bullet \boldsymbol{Y}^{ij} = \delta_{ij} \; (1 \leq i,\; j \leq n), \\ \boldsymbol{E} \bullet \boldsymbol{Y} = n^2,\; \boldsymbol{Y} \in \mathbb{S}_+^{n^2} \cap \mathbb{N} \end{array} \right\}. \tag{35}$$

Using the fact that $\{\boldsymbol{X} \in \mathbb{R}^{n \times n} \mid \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_n, \boldsymbol{X} \geq 0\}$ completely characterizes the set of $n \times n$ permutation matrices, we can show that (33) can equivalently be formulated as follows:

$$v^* := \min \left\{ \boldsymbol{X} \bullet (\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}^T) \;\middle|\; \begin{array}{l} \boldsymbol{e}^T \boldsymbol{X}\boldsymbol{e}_i = 1 = \boldsymbol{e}_i^T \boldsymbol{X}\boldsymbol{e} \; (i = 1, \ldots, n), \\ X_{it}X_{jt} = 0 = X_{ti}X_{tj} \; (t = 1, \ldots, n,\; i \neq j), \\ \boldsymbol{X} \geq \boldsymbol{O} \end{array} \right\}. \tag{36}$$

By considering the Lagrangian–DNN relaxation of (11) and its dual (12) for (36), we can generate a lower bound for (36), denoted as $v_{\mathrm{LB}}^{(2)}$.

Table 4 presents the lower bounds and computation times corresponding to the relaxation problems (35) and the Lagrangian–DNN relaxation of (36) with $\mathbb{K} = \mathbb{S}_+^{n^2} \cap \mathbb{N}$. The test problems were downloaded from [27]. We observe that the bounds $v_{\mathrm{LB}}^{(2)}$ are slightly closer to v^* for the problems `nug**`, `tai30*`, and are comparable for the other problems. However, the lower bounds $v_{\mathrm{LB}}^{(2)}$ can be computed much faster by Algorithm A than $v_{\mathrm{LB}}^{(1)}$ by SDPNAL.

Table 4 Quadratic assignment problems: $\lambda = 10^5\|\boldsymbol{Q}_0\|/\|\boldsymbol{H}_1\|$

Problem	Optimal value v^*	$v_{\rm LB}^{(1)}$ (SDPNAL)	Time	$v_{\rm LB}^{(2)}$ (Algo. A)	Time
bur26a	5.42667000e6	5.42508491e6	1485	5.42578125e6	261
bur26b	3.81785200e6	3.81569956e6	1200	3.81640625e6	212
bur26c	5.42679500e6	5.42409564e6	1853	5.42500000e6	237
bur26d	3.82122500e6	3.81843768e6	1483	3.81875000e6	156
bur26e	5.38687900e6	5.38585258e6	2241	5.38476563e6	187
chr15a	9.89600000e3	9.87953412e3	212	9.89453125e3	19
chr15b	7.99000000e3	7.98997388e3	160	7.98828125e3	21
chr15c	9.50400000e3	9.50397182e3	141	9.50390625e3	14
chr18a	1.10980000e4	1.10863272e4	281	1.10976563e4	43
chr18b	1.53400000e3	1.53351902e3	193	1.53281250e3	96
chr20a	2.19200000e3	2.19118820e3	578	2.19140625e3	78
chr20b	2.29800000e3	2.29783783e3	665	2.29785156e3	78
chr20c	1.41420000e4	1.41316846e4	714	1.41406250e4	92
chr22a	6.15600000e3	6.15574509e3	1290	6.15429688e3	90
chr22b	6.19400000e3	6.19362785e3	990	6.19335938e3	87
chr25a	3.79600000e3	3.79559572e3	1968	3.79589844e3	159
nug20	2.57000000e3	2.50185661e3	184	2.50500000e3	43
nug25	3.74400000e3	3.61940440e3	555	3.62500000e3	139
nug30	6.12400000e3	5.93960496e3	1209	5.94726563e3	306
tai30a	1.81814600e6[a]	1.70543609e6	1115	1.70625000e6	347
tai30b	6.37117113e8	5.94933253e8	2649	5.98242188e8	854
tai35a	2.42200200e6[a]	2.13062174e6	10,457	2.21640625e6	615
tai35b	2.83315445e8[a]	2.64225932e8	17,648	2.69433594e8	2259

The values $v_{\rm LB}^{(1)}$ are computed based on (35) using the SDPNAL algorithm in [37]

[a] The value is not known to be optimal

5.5 Comparison of DNN relaxation (9) and Lagrangina-DNN relaxation (12)

To give the reader an idea on how difficult it is to solve problem (9) with $\mathbb{K} = \mathbb{S}^n_+ \cap \mathbb{N}$ accurately, we apply Algorithm B to solve the problem, and the numerical results are presented in Table 5. In the table, the quantities R_p and R_d are the relative primal and dual infeasibilities for problems (8) and (9), respectively. As can be observed from the table, even after running Algorithm B for 10,000 iterations, the dual feasibility of the final iteration is still quite large except for the maximum stable set problems. As a result, the corresponding dual objective value does not provide a valid lower bound for (8). However, by using the computed dual variables $\hat{\boldsymbol{y}} = (\hat{y}_0, \hat{y}_1)$, we can attempt to generate a valid lower bound as follows. Let $\boldsymbol{G}(y_0) = \boldsymbol{Q}_0 - \hat{y}_1\boldsymbol{H}_1 - y_0\boldsymbol{H}_0$. Set

$$v_{\rm LB}^{(3)} = \begin{cases} \max\{y_0^j := \hat{y}_0(1 + 0.01 \times j) \mid \boldsymbol{G}(y_0^j) \in \mathbb{K}^*,\ j = 0, 1, \ldots\} & \text{if } \hat{y}_0 < 0 \\ \max\{y_0^j := \hat{y}_0(1 - 0.01 \times j) \mid \boldsymbol{G}(y_0^j) \in \mathbb{K}^*,\ j = 0, 1, \ldots\} & \text{if } \hat{y}_0 > 0 \\ \max\{y_0^j := -0.01 \times j \mid \boldsymbol{G}(y_0^j) \in \mathbb{K}^*,\ j = 0, 1, \ldots\} & \text{if } \hat{y}_0 = 0. \end{cases} \quad (37)$$

Table 5 Performance of PADM (Algorithm B with a maximum of 10,000 iterations) and SDPNAL in solving (9) with $\mathbb{K} = \mathbb{S}^n_+ \cap \mathbb{N}$

Problem	$v^{(2)}_{\mathrm{LB}}$ (Algo. A)	Time	$v^{(3)}_{\mathrm{LB}}$ (PADM)	Time|iter| $\max\{R_p, R_d\}$	$v^{(3)}_{\mathrm{LB}}$ (SDPNAL)	Time| $\max\{R_p, R_d\}$
bqp100-1	$-8.04687500e3$	29	$-8.21461716e3$	100|10000|1.62e−3	$-8.14239100e3$	218|5.08e−4
bqp100-2	$-1.10449219e4$	26	$-1.12193981e4$	102|10000|1.82e−3	$-1.11888727e4$	367|4.61e−4
bqp100-3	$-1.27246094e4$	29	$-1.29551863e4$	103|10000|1.80e−3	$-1.29000349e4$	170|1.45e−4
bqp100-4	$-1.03710937e4$	28	$-1.05647288e4$	114|10000|1.84e−3	$-1.29000349e4$	143|7.04e−5
bqp100-5	$-9.08935547e3$	30	$-9.23988363e3$	104|10000|1.81e−3	$-9.20423625e3$	236|8.51e−5
1dc.256	$-3.00048828e1$	15	$-3.03000421e1$	158|10000|1.57e−7	$-3.02999610e1$	22|8.60e−6
1et.256	$-5.44921875e1$	28	$-5.50096590e1$	125|10000|1.51e−7	$-5.50124260e1$	44|9.93e−6
1tc.256	$-6.32812500e1$	34	$-6.38727706e1$	127|10000|6.44e−7	$-6.38726877e1$	40|2.19e−6
1zc.256	$-3.73413086e1$	10	$-3.77066666e1$	23|2064|1.75e−9	$-3.77062792e1$	10|8.98e−6
chr15a	$9.89453125e3$	19	$9.44046443e3$	129|10000|1.53e−2	$9.78302925e3$	386|6.52e−4
chr15b	$7.98828125e3$	21	$7.51407756e3$	119|10000|2.19e−2	$7.89900041e3$	244|2.30e−4
chr15c	$9.50390625e3$	14	$9.03523147e3$	113|10000|2.40e−2	$9.39679398e3$	136|2.84e−4

As we can observe from Table 5, it is generally difficult to solve (9) to high accuracy by Algorithm B or the solver SDPNAL, except for the maximum stable set problems. This is not surprising since the problems generally do not attain the optimal values by Lemma 4(i). Moreover, the time taken to solve (9) and the subsequent generation of the valid lower bound $v_{\mathrm{LB}}^{(3)}$ by Algorithm B or SDPNAL is generally much longer than that taken to compute $v_{\mathrm{LB}}^{(2)}$ based on (12) by Algorithm A. Worse still, the lower bound $v_{\mathrm{LB}}^{(3)}$ generated is also inferior to $v_{\mathrm{LB}}^{(2)}$, except for the maximum stable set problems.

6 Concluding remarks

We have presented a theoretical framework for the Lagrangian–conic relaxation of the QOP (1) and a computational method for the Lagrangian–DNN relaxation to improve the effectiveness and efficiency of obtaining the lower bounds for (1).

The theoretical results on the equivalence between the optimal value of the DNN relaxation of (1) and that of the Lagrangian–DNN relaxation shown in Sect. 3 provide the theoretical support for the tight lower bounds obtained numerically in Sect. 5. The computational efficiency of the proposed method has been achieved by the simple bisection method combined with the efficient first order methods, the proximal alternating direction multiplier and the accelerated proximal gradient methods.

As shown in Sect. 5, the proposed method can compute the lower bounds of improved quality for the test problems efficiently. Specifically, the quadratic assignment problem, known to be very hard problem to solve, could be handled very efficiently.

The approach proposed in this paper for the QOP (1) can be further extended to solve a general class of polynomial optimization problems (POPs). Currently, the numerical methods based on SDPs for POPs suffer from the numerical inefficiency caused by high degrees and large dimensions of POPs. Some of these difficulties can be dealt with the proposed idea in this paper. See also [3]. We intend to work on this project in the future.

References

1. Arima, N., Kim, S., Kojima, M.: A quadratically constrained quadratic optimization model for completely positive cone programming. SIAM J. Optim. **23**, 2320–2340 (2013)
2. Arima, N., Kim, S., Kojima, M.: Simplified copositive and Lagrangian relaxations for linearly constrained quadratic optimization problems in continuous and binary variables. Pac. J. Optim. **10**, 437–451 (2013)
3. Arima, N., Kim, S., Kojima, M.: Extension of completely positive cone relaxation to polynomial optimization, Research report B-471, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Feb 2013
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
5. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
6. BIQMAC Library. http://www.biqmac.uni-klu.ac.at/biqmaclib.html
7. Borcher, B.: CSDP, a C library for semidefinite programming. Optim. Methods Softw. **11**, 613–623 (1999)

8. Bundfuss, S., Dür, M.: An adaptive linear approximation algorithm for copositive programs. SIAM J. Optim. **20**, 30–53 (2009)
9. Burer, S.: On the copositive representation of binary and continuous non-convex quadratic programs. Math. Program. **120**, 479–495 (2009)
10. Burer, S.: Optimizating a polyhedral-semidefinite relaxation of completely positive programs. Math. Program. Comput. **2**, 1–19 (2010)
11. Chen, C., He, B., Ye, Y., Yuan X.: The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. Math. Program. (2015, to appear)
12. de Klerk, E., Pasechnik, D.V.: Approximation of the stability number of a graph via copositive programming. SIAM J. Optim. **12**, 875–892 (2002)
13. Dickinson, P.J.C., Eichfelder, G., Povh, J.: Erratum to: "On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets" [Optim. Letters, 2012]. Optimization Online (2012). http://www.optimization-online.org/DB_HTML/2012/09/3598.html
14. DIMACS, Second DIMACS Challenge, Test instances available at: http://dimacs.rutgers.edu/Challenges/
15. Dür, M., Still, G.: Interior points of the completely positive cone. Electron. J. Linear Algebra **17**, 28–33 (2008)
16. Eichfelder, G., Povh, J.: On the set-semidefinite representation of nonconvex quadratic programs over arbitrary feasible sets. Optim. Lett. (2012). doi:10.1007/s11590-012-0450-3
17. Fazel, M., Pong, Ti. K., Sun, D., Tseng, P.: Hankel matrix rank minimization with applications to system identification and realization. SIAM J. Matrix Anal. A **34**, 946–977 (2013)
18. Fujisawa, K., Fukuda, M., Kobayashi, K., Kojima, M., Nakata, K., Nakata, M., Yamashita, M.: SDPA (semidefinite programming algorithm) user's manual—version 7.05. Research report B-448, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo (2008)
19. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Comput. Math. Appl. **2**, 17–40 (1976)
20. Ge, D., Ye. Y.: On doubly positive semidefinite programming relaxations. Optimization Online (2010). http://www.optimization-online.org/DB_HTML/2010/08/2709.html
21. Jansson, C., Keil, C.: Rigorous error bounds for the optimal value in semidefinite programming. SIAM J. Numer. Anal. **46**, 188–200 (2007)
22. Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéares. Revue Francaise d'Automatique, Informatique et Recherche Opérationelle **9**(R-2), 41–76 (1975)
23. Moreau, J.J.: Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires. C. R. Acad. Sci. **255**, 238–240 (1962)
24. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and non-linear programming. Math. Program. **39**, 117–129 (1987)
25. Nesterov, Y.E., Nemirovskii, A.: Interior Point Methods for Convex Programming. SIAM, Philadelphia (1994)
26. Povh, J., Rendl, F.: Copositive and semidefinite relaxations of the quadratic assignment problem. Discrete Optim. **6**, 231–224 (2009)
27. Quadratic assignment problems. http://www.seas.upenn.edu/qaplib
28. Rendl, F., Rinaldi, G., Wiegele, A.: Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Math. Program. **121**, 307–335 (2010)
29. Sarac, T., Sipahioglu, A.: A genetic algorithm for the quadratic multiple knapsack problem. In: Advances in Brain, Vision, and Artificial Intelligence, vol. 4729 of Lecture Notes in Computer Science, pp. 490–498. Springer, Heidelberg (2007)
30. Sloane, N.: Challenge problems: independent sets in graphs. http://neilsloane.com/doc/graphs.html
31. Strum, J.F.: SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11 & 12**, 625–653 (1999)
32. Toh, K., Todd, M.J., Tütüntü, R.H.: SDPT3—a MATLAB software package for semidefinite programming. Optim. Methods Softw. **11 & 12**, 545–581 (1999)
33. Tucker, A.W.: Dual systems of homogeneous linear relations. In: Kuhn, Tucker (eds.) Linear Inequalities and Related Systems, Annals of Mathematics Studies, No. 38. Princeton University Press, Princeton (1956)
34. Wen, Z., Goldfarb, D., Yin, W.: Alternating direction augmented Lagrangian methods for semidefinite programming. Math. Program. Comput. **2**, 203–230 (2010)

35. Sun, D.F., Toh, K.C., Yang, L.Q.: A convergent proximal alternating direction method of multipliers for conic programming with 4-block constraints (2014). arXiv:1404.5378
36. Yoshise, A., Matsukawa, Y.: On optimization over the doubly nonnegative cone. In: Proceedings of 2010 IEEE Multi-conference on Systems and Control, pp. 13–19 (2010)
37. Zhao, X.Y., Sun, D.F., Toh, K.C.: A Newton-CG augmented Lagrangian method for semidefinite programming. SIAM J. Optim. **20**, 1737–1765 (2010)