Doubly Nonnegative Relaxations for Quadratic and Polynomial Optimization Problems with Binary and Box Constraints

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

Jan, 2017
Revised September, 2017

Abstract

We propose a doubly nonnegative (DNN) relaxation for polynomial optimization problems (POPs) with binary and box constraints. This work is an extension of the work by Kim, Kojima and Toh in 2016 from quadratic optimization problems (QOPs) to POPs. The dense and sparse DNN relaxations are reduced to a simple conic optimization problem (COP) to which an accelerated bisection and projection (BP) algorithm is applied. The COP involves a single equality constraint in a matrix variable which is restricted to the intersection of the positive semidefinite cone and a polyhedral cone representing the algebraic properties of binary and box constraints as well as the sparsity structure of the objective polynomial. Our DNN relaxation serves as a variant of the standard Lasserre moment-SOS hierarchy for binary and box constrained POPs as the size of the cones is gradually increased to compute tighter lower bounds for their optimal values. A significant part of the paper is devoted to deriving and analyzing a class of polyhedral cones which strengthen the moment-SOS relaxation and to efficient computation of the metric projections onto these cones in the accelerated BP algorithm. Using the basic DNN relaxation framework, we also show why tight lower bounds of binary and box constrained QOPs were obtained numerically by their Lagrangian-DNN relaxation in the work by the authors in 2016. Numerical results on large scale POPs are provided to demonstrate the efficiency of the proposed method for attaining tight bounds.

Key words. Polynomial optimization problems with nonnegative variables, doubly nonnegative relaxations, a class of polyhedral cones, the bisection and projection algorithm, computational efficiency and tight bounds.

AMS Classification. 90C22, 90C25, 90C26.

[*]Department of Mathematics, Ewha W. University, 52 Ewhayeodae-gil, Sudaemoon-gu, Seoul 120-750, Korea (skim@ewha.ac.kr). The research was supported by NRF 2014-R1A2A1A11049618 and 2017-R1A2B2005119.

[†]Department of Industrial and Systems Engineering, Chuo University, Tokyo 192-0393, Japan (masakazukojima@mac.com). This research was supported by Grant-in-Aid for Scientific Research (A) 26242027 and the Japan Science and Technology Agency (JST), the Core Research of Evolutionary Science and Technology (CREST) research project.

[‡]Department of Mathematics, and Institute of Operations Research and Analytics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg).

1 Introduction

Consider a polynomial optimization problem (POP) that minimizes a real-valued polynomial $f(\boldsymbol{x})$ in $\boldsymbol{x} \in \mathbb{R}^n$ over a basic semi-algebraic subset of $\mathbb{R}^n$. The problem serves as a fundamental nonconvex model in global optimization, notably, quadratic optimization problems (QOPs) with continuous and binary variables are its special cases. As many problems from applications, including combinatorial optimization, can be formulated as POPs with nonnegative variables, our focus is on designing efficient convex relaxation methods and algorithms for solving such POPs.

As a relaxation method for QOPs with nonnegative variables, the semidefinite programming (SDP) relaxation with the additional nonnegative constraint on the variable matrix, known as the doubly nonnegative (DNN) relaxation, was used in [9, 12, 31]. This is a natural approach in the sense that it strengthens the SDP relaxations [10, 24], which on their own have been already proved to be very successful in solving various QOPs. The DNN relaxation approach for QOPs, however, suffers from computational inefficiency if the SDP relaxation is solved by a primal-dual interior-point method [19, 23]. The inefficiency arises mainly from the rapidly increasing sizes of the positive semidefinite matrix variables and polyhedral constraints in the DNN relaxations. More precisely, when the DNN relaxation of a QOP is converted to an SDP in the standard form, the single nonnegative constraint imposed on the DNN matrix variable must be explicitly expressed as nonnegative constraints on the matrix elements, which makes the size of the SDP grow quadratically with the size of the DNN matrix variable.

The SDP relaxations for QOPs can be regarded as a special case of Lasserre's SDP relaxation [21] with the lowest hierarchy applied to QOPs. The hierarchy of SDP relaxations for POPs by Lasserre [21] is a powerful method supported by the convergence result to the optimal values of POPs under moderate assumptions. However, the size of Lasserre's SDP relaxation increases exponentially with the size of a given POP and/or the degree of polynomials. An approach to mitigate this difficulty is by exploiting the sparsity in the SDP relaxations, as proposed in [11, 18, 27]. A dynamic inequality generation scheme is proposed in [13] as an efficient method for solving polynomial programs. However, solving large-sized DNN relaxations of QOPs and POPs still remains a very challenging problem.

Recently, a bisection and projection (BP) algorithm was proposed by Kim, Kojima and Toh [16] (see also [4]) for a Lagrangian and doubly nonnegative (Lagrangian-DNN) relaxation, which was developed as a numerically tractable relaxation of the completely positive programming (CPP) reformulation [1, 7] of a class of QOPs with linear equality, binary and complementarity constraints in nonnegative variables. In their Lagrangian-DNN relaxation, a QOP in the class is relaxed into a simple DNN problem which minimizes a linear function in the matrix variable $\boldsymbol{X} \in \mathbb{S}^n$ subject to the DNN constraint $\boldsymbol{X} \in \mathbb{S}^n_+ \cap \mathbb{N}^n$ and a single linear equality constraint $X_{11} = 1$, where $\mathbb{S}^n$ denotes the linear space of $n \times n$ symmetric matrices, $\mathbb{S}^n_+$ the cone of positive semidefinite matrices in $\mathbb{S}^n$ and $\mathbb{N}^n$ the cone of matrices in $\mathbb{S}^n$ with nonnegative elements. It was demonstrated through numerical results that the BP algorithm could efficiently solve large-scale Lagrangian-DNN relaxation problems. Here we extend the application of the BP algorithm [16] to DNN relaxations of POPs in this paper. Extending the basic template of the BP algorithm to solve those DNN relaxations is, in principle, straightforward. However, the task of constructing suitable DNN relaxations requires substantial innovations, especially to design relaxations that are capable of exploiting the underlying problem structures and computationally can be solved efficiently.

We are mainly concerned with a POP in the following form:

$$\min \left\{ f(\boldsymbol{x}) : \boldsymbol{x} \in S \right\}, \tag{1}$$

where $S = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \; x_i \in \{0,1\} \; (i \in I_{\text{bin}}), \; x_i \in [0,1] \; (i \in I_{\text{box}}) \; \right\}$, with I_{bin} and I_{box} being a partition of $\{1 \ldots, n\}$ such that $I_{\text{bin}} \cup I_{\text{box}} = \{1 \ldots, n\}$ and $I_{\text{bin}} \cap I_{\text{box}} = \emptyset$. *Binary* variables are denoted by x_i $(i \in I_{\text{bin}})$, and *box constrained* variables by x_j $(j \in I_{\text{box}})$. Obviously, a QOP with binary and box constraints is a special case of (1), where the degree of the objective polynomial function $f(\boldsymbol{x})$ is 2.

The purpose of this paper is to develop a basic framework to derive DNN relaxations of POP (1) that are efficiently solvable by the BP algorithm for computing tight bounds of (1). The DNN relaxations of POP (1) that will be derived in this paper include (a) the standard dense and sparse DNN relaxations of QOPs with binary and box constraints, (b) the Lagrangian-DNN relaxation of a class of QOPs with binary and box constraints, and (c) hierarchies of dense and sparse DNN relaxations of POPs with binary and box constraints. The last two relaxations may be regarded as variants of the hierarchy of SDP relaxation of POPs [21] and its sparse version [27], respectively. Our hierarchy of DNN relaxations for a binary POP in fact strengthens the Lasserre-hierarchy of SDP relaxations [21] and hence it enjoys the same asymptotic convergence to the optimal value of the POP as in the case of SDP relaxations. But our DNN-relaxation hierarchy for a box constrained POP is different from the Lasserre's SDP-relaxation hierarchy because for the former, the algebraic properties of the box constraint are implicitly encoded in a class of polyhedral cones which we will construct later whereas for the latter, the box constraint must be expressed explicitly as constraints in the POP before deriving the SDP relaxations.

All the DNN relaxations are further reduced to a simple conic optimization problem (COP) of minimizing a linear function in $\boldsymbol{X} \in \mathbb{V}$ subject to $\boldsymbol{X} \in \mathbb{K}^1 \cap \mathbb{K}^2$ and a single linear equality constraint $\langle \boldsymbol{H}^0, \boldsymbol{X} \rangle = 1$. Here $\mathbb{V}$ is a finite dimensional linear space endowed with the inner product $\langle \cdot, \cdot \rangle$, $\mathbb{K}^1$ and $\mathbb{K}^2$ are closed convex cones in $\mathbb{V}$, and $\boldsymbol{H}^0 \in \mathbb{V}$. The resulting simple COP satisfies a few additional properties: $\mathbb{V}$ is the Cartesian product of some linear spaces of symmetric matrices, $\mathbb{K}^1$ is the Cartesian product of cones of some positive semidefinite matrices in $\mathbb{V}$, $\mathbb{K}^2$ is a polyhedral cone in $\mathbb{V}$, and $\boldsymbol{H}^0 \neq \boldsymbol{O}$ lies in the dual of $\mathbb{K}^1 \cap \mathbb{K}^2$. Relying on these properties, the BP algorithm can be applied to the simple COP.

Two main issues, which are closely related, should be dealt with immediately in order for us to be able to solve the DNN relaxations of POP (1) efficiently. The first one is the construction of the polyhedral cone $\mathbb{K}^2$ for the simple COP. The second is the efficient computation of the metric projection from $\mathbb{V}$ onto $\mathbb{K}^2$, which is a very important step of the BP algorithm for the overall computational efficiency. To handle these two issues for the various DNN relaxations in a unified manner, we introduce a class of polyhedral cones in $\mathbb{V}$ onto which the metric projection can efficiently be computed. Then, each cone is discussed as a special case of the class of polyhedral cones, and as a result, various DNN relaxations can be derived by employing different cones. The BP algorithm applied to the DNN relaxations of POP (1) is shown to perform (a) much more efficiently than the primal-dual interior-point method SDPT3 [26] in solving the standard DNN relaxation, and (b) more robustly and efficiently than the recently developed semismooth Newton-CG augmented Lagrangian solver for large scale SDPs, SDPNAL+ [30], when it is tested on randomly generated POPs with binary and box constraints and the maximum complete satisfiability problem [14]. The next contribution we made in the paper is that by using the new framework in deriving the class of polyhedral cones, we are able to show why the Lagrangian-DNN relaxation developed in [2, 16] provided a tighter bound than the standard DNN relaxation.

The rest of the paper is organized as follows. In Section 2, we define the simple COP in a precise form, and describe the BP and accelerated BP algorithm [4] for solving the COP. We also review additional concepts necessary for the subsequent discussions. Section 3 includes the main results of this paper. We define a class of polyhedral cones (motivated by the algebraic properties of binary and box constraints) in $\mathbb{V}$ onto which the metric projection can be computed efficiently, and study its properties. In Sections 4 and 5, the dense and sparse DNN relaxations of POP (1) are derived, respectively, based on the results established in Section 3. Section 6 introduces stronger DNN relaxations of a binary and box constrained QOP, a full-DNN relaxation and a twin DNN relaxation. In particular, we show that the Lagrangian-DNN relaxation developed in [16] are at least as effective as these two DNN relaxations. In Section 7, we present some numerical results on dense and sparse DNN relaxations of randomly generated binary and box POPs with degrees 3, 4, 5 and 6. We first show that large scale instances cannot be solved by SeDuMi [25] and SDPT3 [26] based on the primal-dual interior-point method, while they can be solved by the accelerated BP algorithm and SDPNAL+ [30], which is currently the most advanced solver (based on a majorized semismooth Newton-CG augmented Lagrangian method) for large scale SDPs with bound constraints. Then the accelerated BP algorithm and SDPNAL+ are compared on large scale sparse binary and box POP instances with dimension up to a few thousands. We observe that SDPNAL+ is unstable for ill-conditioned and/or large scale instances although it attains very accurate lower bounds when it converges. It often provides invalid lower bounds if the iteration is terminated before convergence. The accelerated BP algorithm, on the other hand, is robust when dealing with ill-conditioned instances. Moreover, it can always generate valid lower bounds in less execution time than SDPNAL+ for large scale instances, although the quality of the lower bounds is slightly worse than that obtained by SDPNAL+ when the latter is successful in solving the instances. The Lagrangian-DNN relaxation of binary QOP mentioned in Section 6 is compared with its standard DNN relaxation mentioned in Section 4 to demonstrate the theoretical advantage of the Lagrangian-DNN relaxation. Finally, we conclude the paper in Section 8.

2 Preliminaries

2.1 A simple conic optimization problem

Let $\mathbb{V}$ be a finite dimensional vector space endowed with an inner product $\langle\cdot,\,\cdot\rangle$ and its induced norm $\|\cdot\|$ such that $\|\boldsymbol{X}\| = (\langle\boldsymbol{X},\,\boldsymbol{X}\rangle)^{1/2}$ for every $\boldsymbol{X} \in \mathbb{V}$. Let $\mathbb{K}^1$ and $\mathbb{K}^2$ be closed convex cones in $\mathbb{V}$ satisfying $(\mathbb{K}^1 \cap \mathbb{K}^2)^* = (\mathbb{K}^1)^* + (\mathbb{K}^2)^*$, $\boldsymbol{Q}^0 \in \mathbb{V}$ and $\boldsymbol{O} \neq \boldsymbol{H}^0 \in (\mathbb{K}^1)^* + (\mathbb{K}^2)^*$, where $\mathbb{K}^* = \{\boldsymbol{Y} \in \mathbb{V} : \langle\boldsymbol{X},\,\boldsymbol{Y}\rangle \geq 0 \text{ for all } \boldsymbol{X} \in \mathbb{K}\}$ denotes the dual cone of a cone $\mathbb{K} \subset \mathbb{V}$.

We introduce a simple conic optimization problem (COP):

$$\varphi^* = \min\left\{\langle\boldsymbol{Q}^0,\,\boldsymbol{X}\rangle : \langle\boldsymbol{H}^0,\,\boldsymbol{X}\rangle = 1,\ \boldsymbol{X} \in \mathbb{K}^1 \cap \mathbb{K}^2\right\}, \tag{2}$$

and its dual

$$\tau^* = \max\left\{\tau : \boldsymbol{Q}^0 - \boldsymbol{H}^0\tau - \boldsymbol{W} = \boldsymbol{Y},\ \boldsymbol{Y} \in (\mathbb{K}^1)^*,\ \boldsymbol{W} \in (\mathbb{K}^2)^*\right\}. \tag{3}$$

The primal-dual pair of COPs (2) and (3) plays a crucial role throughout the paper. All the dense and sparse DNN relaxations for QOPs and POPs with binary and box constraints discussed in Sections 4, 5 and 6 are eventually reduced to COP (2). Then, the accelerated bisection and projection algorithm [4], which is described as Algorithm 2.1, is applied to the primal-dual pair of COPs to solve the relaxations.

We remark that a stanform form SDP:

$$\min\left\{\langle\boldsymbol{C},\,\boldsymbol{U}\rangle : \langle\boldsymbol{A}_p,\,\boldsymbol{U}\rangle = b_p\ (p = 1,\ldots,m),\ \boldsymbol{U} \in \mathbb{S}^{n-1}\right\}$$

can always be rewritten as COP (2), where $\boldsymbol{C},\ \boldsymbol{A}_p \in \mathbb{S}^{n-1}$, $b_p \in \mathbb{R}$ $(p = 1,\ldots,m)$. In fact, let

$$\begin{aligned}
&\boldsymbol{Q}^0 = \begin{pmatrix} 0 & \boldsymbol{0}^t \\ \boldsymbol{0} & \boldsymbol{C} \end{pmatrix} \in \mathbb{S}^n,\\
&\boldsymbol{H}^0 = \text{the } n \times n \text{ matrix with } H^0_{ij} = 0 \text{ for all } i, j \text{ except } H^0_{11} = 1,\\
&\mathbb{K}^1 = \mathbb{S}^n_+,\ \mathbb{K}^2 = \left\{\begin{pmatrix} x_0 & \boldsymbol{0}^t \\ \boldsymbol{0} & \boldsymbol{U} \end{pmatrix} \in \mathbb{S}^n : \langle\boldsymbol{A}_p,\,\boldsymbol{U}\rangle - b_p x_0 = 0\right\}.
\end{aligned}$$

Then the SDP is equivalent to COP (2). Note that $\mathbb{K}^2$ is a linear subspace of $\mathbb{S}^n$.

2.2 The accelerated bisection and projection algorithm

The bisection and projection (BP) algorithm was originally proposed in [16] as a numerical method for solving the Lagrangian-DNN relaxation of a class of QOPs with linear equality, binary and complementarity constraints in nonnegative variables. While the special case where $\mathbb{V} = \mathbb{S}^n$ (the linear space of $n \times n$ symmetric matrices), $\mathbb{K}^1 = \mathbb{S}^n_+$ (the cone of positive semidefinite matrices in $\mathbb{S}^n$) and $\mathbb{K}^2 = \mathbb{N}^n$ (the cone of nonnegative matrices in $\mathbb{S}^n$) was dealt with in [16], the algorithm in [16] could be extended to the more general COP (2) as in [3].

Define $\boldsymbol{G}(\tau) = \boldsymbol{Q}^0 - \boldsymbol{H}^0\tau$ and $g(\tau) = \min\left\{\|\boldsymbol{G}(\tau) - \boldsymbol{Z}\| : \boldsymbol{Z} \in (\mathbb{K}^1)^* + (\mathbb{K}^2)^*\right\}$ for every $\tau \in \mathbb{R}$. Then, τ is a feasible solution of (3) if and only if $g(\tau) = 0$. As a result, (3) is equivalent to $\max\{\tau : g(\tau) = 0\}$. Recall that $\boldsymbol{H}^0 \in (\mathbb{K}^1)^* + (\mathbb{K}^2)^*$. Hence, if $g(\bar{\tau}) = 0$, or equivalently, $\boldsymbol{G}(\bar{\tau}) \in (\mathbb{K}^1)^* + (\mathbb{K}^2)^*$ for some $\bar{\tau} \in \mathbb{R}$, then for every $\tau \leq \bar{\tau}$, $\boldsymbol{G}(\tau) = \boldsymbol{G}(\bar{\tau}) + (\bar{\tau} - \tau)\boldsymbol{H}^0 \in (\mathbb{K}^1)^* + (\mathbb{K}^2)^*$ or equivalently, $g(\tau) = 0$. Thus, $g(\tau) = 0$ for every $\tau \in (-\infty, \tau^*]$. Furthermore, by Lemma 4.1 of [3], g is continuously differentiable, convex and monotonically increasing on $[\tau^*, +\infty)$.

For every $\boldsymbol{G} \in \mathbb{V}$, let $\Pi(\boldsymbol{G})$ and $\Pi^*(\boldsymbol{G})$ denote the metric projection of $\boldsymbol{G}$ onto the cone $\mathbb{K}^1 \cap \mathbb{K}^2$ and its dual cone $(\mathbb{K}^1)^* + (\mathbb{K}^2)^*$, respectively, i.e.,

$$\begin{aligned}\Pi(\boldsymbol{G}) &= \operatorname{argmin}\left\{\|\boldsymbol{G} - \boldsymbol{X}\| : \boldsymbol{X} \in \mathbb{K}^1 \cap \mathbb{K}^2\right\},\\ \Pi^*(\boldsymbol{G}) &= \operatorname{argmin}\left\{\|\boldsymbol{G} - \boldsymbol{Z}\| : \boldsymbol{Z} \in (\mathbb{K}^1)^* + (\mathbb{K}^2)^*\right\}.\end{aligned}$$

By the decomposition theorem of Moreau [22], we know that $\boldsymbol{G}(\tau) = \Pi^*(\boldsymbol{G}(\tau)) - \Pi(-\boldsymbol{G}(\tau))$. Let $\widehat{\boldsymbol{X}}(\tau) := \Pi(-\boldsymbol{G}(\tau)) \in \mathbb{K}^1 \cap \mathbb{K}^2$. Then

$$g(\tau) = \|\boldsymbol{G}(\tau) - \Pi^*(\boldsymbol{G}(\tau))\| = \|-\Pi(-\boldsymbol{G}(\tau))\| = \left\|\widehat{\boldsymbol{X}}(\tau)\right\|.$$

Next, we describe the basic idea of the BP algorithm [16]. Given an initial interval $[\ell^0, u^0]$ of lower and upper bounds for τ^*, it starts with the iteration counter $p = 0$. First, set $\tau^{p+1} = (\ell^p + u^p)/2$. If $g(\tau^{p+1}) = \left\|\widehat{\boldsymbol{X}}(\tau^{p+1})\right\| > 0$, update the bounds such that $\ell^{p+1} = \ell^p$ and $u^{p+1} = \tau^{p+1}$. Otherwise, $g(\tau^{p+1}) = \left\|\widehat{\boldsymbol{X}}(\tau^{p+1})\right\| = 0$. Then, update the bounds such that $\ell^{p+1} = \tau^{p+1}$ and $u^{p+1} = u^p$. In either case, $\tau^* \in [\ell^{p+1}, u^{p+1}]$ holds. After replacing $p+1$ by p, the iteration continues until the length of the interval becomes sufficiently small.

For the BP algorithm in [16] to be used as a reliable solution method, a few numerical issues must be carefully addressed. The first issue is on the computation of the projection $\widehat{\boldsymbol{X}}(\tau) = \Pi(-\boldsymbol{G}(\tau))$. For this computation, the accelerated proximal gradient method [5] was applied in [16]; See Section 4.2 and Algorithm C of [16]. However, the projection $\widehat{\boldsymbol{X}}(\tau) = \Pi(-\boldsymbol{G}(\tau))$ cannot be obtained exactly for a given $\tau \in \mathbb{R}$ with double-precision floating point arithmetic, which frequently results in a positive numerical value of $g(\tau) = \left\|\widehat{\boldsymbol{X}}(\tau)\right\|$ even when $\tau \leq \tau^*$. The second numerical difficulty of the BP algorithm is that an initial interval $[\ell^0, u^0]$ containing τ^* is required. In many practical applications, u^0 is available but a reasonable ℓ^0 is not. Fortunately, one can always choose a sufficiently small test value ℓ^0 and check whether $g(\ell^0)$ is equal to 0 to pick a valid lower lower for τ^*.

For the aforementioned numerical difficulties, Arima, Kim, Kojima and Toh [4] proposed an improvement of the BP algorithm under the additional assumptions:

- We know an $\boldsymbol{I} \in \mathbb{V}$ which lies in the interior of $(\mathbb{K}^1)^*$ and a positive number ρ such that $\langle \boldsymbol{I}, \boldsymbol{X}\rangle \leq \rho$ holds for every feasible solution $\boldsymbol{X}$ of COP (2).
- For any $\boldsymbol{A} \in \mathbb{V}$, it is easy to compute $\mu(\boldsymbol{A}) = \sup\{\mu : \boldsymbol{A} - \mu\boldsymbol{I} \in (\mathbb{K}^1)^*\}$. (Since $\boldsymbol{I}$ lies in the interior of $(\mathbb{K}^1)^*$, $\mu(\boldsymbol{A})$ is finite for any $\boldsymbol{A} \in \mathbb{V}$).

The problem with $\mathbb{V} = \mathbb{S}^n$, $\mathbb{K}^1 = \mathbb{S}^n_+$, $\mathbb{K}^2 = \mathbb{N}^n$ and $\boldsymbol{I}$ = the $n \times n$ identity matrix was considered in [4], where $\mu(\boldsymbol{A})$ coincides with the minimum eigenvalue of $\boldsymbol{A} \in \mathbb{S}^n$. Under the assumptions previously mentioned, COP (2) is equivalent to

$$\varphi^* = \min\left\{\langle \boldsymbol{Q}^0,\, \boldsymbol{X}\rangle : \langle \boldsymbol{H}^0,\, \boldsymbol{X}\rangle = 1,\ \langle \boldsymbol{I},\, \boldsymbol{X}\rangle \leq \rho,\ \boldsymbol{X} \in \mathbb{K}^1 \cap \mathbb{K}^2\right\}.$$

For the computation of a valid lower bound for φ^*, we consider the dual:

$$\max\left\{\tau + \rho t : \boldsymbol{Q}^0 - \boldsymbol{H}^0\tau - \boldsymbol{I}t - \boldsymbol{W} = \boldsymbol{Y},\ \boldsymbol{Y} \in (\mathbb{K}^1)^*,\ \boldsymbol{W} \in (\mathbb{K}^2)^*,\ t \leq 0\right\}. \quad (4)$$

Suppose that a decomposition $\widehat{\boldsymbol{W}}(\tau^p) + \widehat{\boldsymbol{Y}}(\tau^p) - \widehat{\boldsymbol{X}}(\tau^p)$ of $\boldsymbol{G}(\tau^p)$ is computed approximately at the pth iteration of the BP algorithm, specifically that $\boldsymbol{G}(\tau^p) \approx \widehat{\boldsymbol{W}}(\tau^p) + \widehat{\boldsymbol{Y}}(\tau^p) - \widehat{\boldsymbol{X}}(\tau^p)$ with $\widehat{\boldsymbol{W}}(\tau^p) \in (\mathbb{K}^2)^*$. We note that neither $\widehat{\boldsymbol{Y}}(\tau^p) \in (\mathbb{K}^1)^*$ nor $\widehat{\boldsymbol{X}}(\tau^p) \in \mathbb{K}^1 \cap \mathbb{K}^2$ is required at this point. Let

$$\begin{aligned} t^p &= \min\{\mu(\boldsymbol{Q}^0 - \boldsymbol{H}^0\tau^p - \widehat{\boldsymbol{W}}(\tau^p)), 0\}, \\ \widetilde{\boldsymbol{Y}}^p &= \boldsymbol{Q}^0 - \boldsymbol{H}^0\tau^p - \widehat{\boldsymbol{W}}(\tau^p) - \boldsymbol{I}t^p. \end{aligned} \quad (5)$$

Then $t^p \leq 0$ and $\widetilde{\boldsymbol{Y}}^p \in (\mathbb{K}^1)^*$. As a result, $\tau^p + \rho t^p$ provides a valid lower bound for φ^* by the weak duality.

The BP algorithm incorporating this technique can generate a valid lower bound $\underline{\ell}^{p+1} = \tau^p + \rho t^p$ for φ^* at each iteration, even if the algorithm starts with $\ell^0 = -\infty$. The numerical results in [4] showed that this technique also served to accelerate the BP algorithm. In the numerical results in Section 7, the accelerated BP algorithm will be used.

Let the numerical value of $\|\widehat{\boldsymbol{X}}(\tau)\|/\max\{1, \|\boldsymbol{G}(\tau)\|\}$ be denoted by $h(\tau)$. We are now ready to describe the accelerated BP algorithm.

Algorithm 2.1. (Accelerated BP Algorithm)

Step 0. Choose sufficiently small positive numbers ϵ and δ (e.g., $\epsilon = 10^{-11}$ and $\delta = 10^{-4}$). Here δ determines the target length of an interval $[\ell^p,\ u^p] \subset \mathbb{R}$ which is expected to contain τ^*. Let $p = 0$.

Step 1. Find a $u^0 \in \mathbb{R}$ such that $\tau^* \leq u^0$. Let $\ell^0 = \underline{\ell}^0 = -\infty$. Choose $\tau^0 \leq u^0$.

Step 2. If $u^p - \ell^p < \delta \max\{1, |\ell^p|, |u^p|\}$, output $\underline{\ell}^p$ as a lower bound for τ^*. Otherwise, compute a decomposition $\boldsymbol{G}(\tau^p) \approx \widehat{\boldsymbol{W}}(\tau^p) + \widehat{\boldsymbol{Y}}(\tau^p) - \widehat{\boldsymbol{X}}(\tau^p)$.

Step 3. Take $t^p \in \mathbb{R}$ and $\widetilde{\boldsymbol{Y}}^p \in \mathbb{V}$ as in (5). Let $\underline{\ell}^{p+1} = \max\{\underline{\ell}^p, \tau^p + \rho t^p\}$. If $h(\tau^p) \leq \epsilon$, then let $\ell^{p+1} = \tau^p$ and $u^{p+1} = u^p$. Otherwise, let $\ell^{p+1} = \max\{\underline{\ell}^{p+1},\ \ell^p\}$ and $u^{p+1} = \tau^p$.

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

See Section 3 of [4] and Section 4 of [16] for more details.

In the simple COP (2), $\mathbb{K}^1$ is usually the cone (or the Cartesian product of cones) of positive semidefinite matrices as in Sections 4–6, and the metric projection $\Pi_{\mathbb{K}^1}(\boldsymbol{Z})$ of each $\boldsymbol{Z} \in \mathbb{V}$ onto $\mathbb{K}^1$ can be computed by the eigenvalue decomposition. Thus the only remaining important step is the metric projection onto the cone $\mathbb{K}^2$, which should be carried out efficiently for the overall computational efficiency. In Section 3, we present a class of polyhedral cones onto which the metric projection can efficiently be computed.

2.3 QOPs with binary and box constraints and their standard DNN relaxations

We begin by noting that any quadratic function in $\boldsymbol{x} \in \mathbb{R}^n$ can be written as $\langle \boldsymbol{Q}^0,\, (1,\boldsymbol{x})^T(1,\boldsymbol{x})\rangle$, where $\boldsymbol{x}$ is an n-dimensional row vector and $\boldsymbol{Q}^0 \in \mathbb{S}^{1+n}$. As a special case of (1), we consider the QOP with binary and box constraints

$$\min \left\{ \langle \boldsymbol{Q}^0,\, (1,\boldsymbol{x})^T(1,\boldsymbol{x})\rangle : \boldsymbol{x} \in S \right\}. \tag{6}$$

By introducing a symmetric matrix variable $\begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n}$, and a subset

$$T \;=\; \left\{ \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n} : \boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^T,\ \boldsymbol{x} \in S,\ X_{00} = 1 \right\},$$

we can rewrite (6) as

$$\min \left\{ \left\langle \boldsymbol{Q}^0, \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \right\rangle : \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in T \right\}. \tag{7}$$

Thus the QOP (6) in $\mathbb{R}^n$ is *lifted* to the equivalent problem (7) in $\mathbb{S}^{1+n}$. Between a feasible solution $\boldsymbol{x} \in \mathbb{R}^n$ of (6) and a feasible solution $\begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n}$ of (7), the following correspondence holds:

$$1 \leftrightarrow X_{00},\ x_i \leftrightarrow x_i\ (i = 1,\ldots,n),\ x_i x_j \leftrightarrow X_{ij}\ (i = 1,\ldots,n, j = 1,\ldots,n). \tag{8}$$

To derive a DNN relaxation of QOP (6), we relax the feasible region T of the lifted problem (7) to a convex subset of $\mathbb{S}^{1+n}$, which is described as the intersection of a hyperplane and two closed convex cones $\mathbb{K}^1$, $\mathbb{K}^2$ as in (2). By construction, if $\begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in T$ then $\begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n}_+$, $\begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \geq \boldsymbol{O}$, $\left\langle \boldsymbol{H}^0, \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \right\rangle = X_{00} = 1$. Here $\boldsymbol{H}^0 \in \mathbb{S}^{1+n}$ has 1 as its $(0,0)$th element and 0 elsewhere. It follows from $\boldsymbol{x} \in S$ that the identity $x_i = x_i^2$ $(i \in I_{\rm bin})$ and the inequality $x_j \geq x_j^2$ $(j \in I_{\rm box})$ hold for every $\boldsymbol{x} \in S$. Since x_i^2 corresponds to X_{ii} $(i = 1,\ldots,n)$, the identity and inequality induce $x_i = X_{ii}$ $(i \in I_{\rm bin})$ and $x_j \geq X_{jj}$ $(j \in I_{\rm box})$, respectively. By defining

$$\mathbb{L} \;=\; \left\{ \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in \mathbb{N}^{1+n} : x_i = X_{ii}\ (i \in I_{\rm bin}),\ x_j \geq X_{jj}\ (j \in I_{\rm box}) \right\}, \tag{9}$$

which forms a polyhedral cone in $\mathbb{S}^{1+n}$, we see that every $\begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \in T$ lies in $\mathbb{L}$. Here $\mathbb{N}^{1+n}$ denotes the cone of $(1+n)\times(1+n)$ symmetric matrices with all their elements nonnegative. Therefore, COP (2) with $\mathbb{K}^1 = \mathbb{S}^{1+n}_+$ and $\mathbb{K}^2 = \mathbb{L}$ serves as the *standard DNN relaxation* of QOP (6).

2.4 Notation and symbols for POPs and their DNN relaxations

For the extension of the discussion in Section 2.3 to a general POP (1) with binary and box constraints and its dense and sparse DNN relaxations in the subsequent sections, we introduce the following notation and symbols.

Let $\mathbb{Z}^n_+$ denote the set of n-dimensional nonnegative integer vectors. For each $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n_+$, let $\boldsymbol{x}^{\boldsymbol{\alpha}} = x^{\boldsymbol{\alpha}_1} \cdots x^{\boldsymbol{\alpha}_n}$ denote a *monomial.* We call $\deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) = \max\{\alpha_i : i = 1, \ldots, n\}$ the *degree* of a monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$. Each polynomial $f(\boldsymbol{x})$ is represented as $f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{F}} c(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$ for some nonempty finite subset $\mathcal{F}$ of $\mathbb{Z}^n_+$ and $c(\boldsymbol{\alpha}) \in \mathbb{R}$ ($\boldsymbol{\alpha} \in \mathcal{F}$). We call $\mathrm{supp} f = \{\boldsymbol{\alpha} \in \mathcal{F} : c(\boldsymbol{\alpha}) \neq 0\}$ the *support* of $f(\boldsymbol{x})$; hence $f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathrm{supp} f} c(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$ is the minimal representation of $f(\boldsymbol{x})$. We call $\deg f = \max\{\deg(\boldsymbol{x}^{\boldsymbol{\alpha}}) : \boldsymbol{\alpha} \in \mathrm{supp} f\}$ the *degree* of $f(\boldsymbol{x})$.

Let $\mathcal{A}$ be a nonempty finite subset of $\mathbb{Z}^n_+$ with cardinality $|\mathcal{A}|$, and let $\mathbb{S}^{\mathcal{A}}$ denote the linear space of $|\mathcal{A}| \times |\mathcal{A}|$ symmetric matrices whose rows and columns are indexed by $\mathcal{A}$. The $(\boldsymbol{\alpha}, \boldsymbol{\beta})$th component of each $\boldsymbol{X} \in \mathbb{S}^{\mathcal{A}}$ is written as $X_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ ($(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$). The *inner product* of $\boldsymbol{X}$, $\boldsymbol{Y} \in \mathbb{S}^{\mathcal{A}}$ is defined by $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \sum_{\boldsymbol{\beta}\in\mathcal{A}} X_{\boldsymbol{\alpha}\boldsymbol{\beta}} Y_{\boldsymbol{\alpha}\boldsymbol{\beta}}$, and the *norm* of $\boldsymbol{X} \in \mathbb{S}^{\mathcal{A}}$ is defined by $\|\boldsymbol{X}\| = (\langle \boldsymbol{X}, \boldsymbol{X} \rangle)^{1/2}$. Assuming that the elements of $\mathcal{A}$ are enumerated in an appropriate order, we denote a $|\mathcal{A}|$-dimensional row vector of monomials $\boldsymbol{x}^{\boldsymbol{\alpha}}$ ($\boldsymbol{\alpha} \in \mathcal{A}$) by $\boldsymbol{x}^{\mathcal{A}}$, and a $|\mathcal{A}| \times |\mathcal{A}|$ symmetric matrix $(\boldsymbol{x}^{\mathcal{A}})^T(\boldsymbol{x}^{\mathcal{A}})$ of monomials $\boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}}$ ($(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$) by $\boldsymbol{x}^{\square\mathcal{A}} \in \mathbb{S}^{\mathcal{A}}$. We call $\boldsymbol{x}^{\square\mathcal{A}}$ a *moment matrix.*

For a pair of subsets $\mathcal{A}$ and $\mathcal{B}$ of $\mathbb{Z}^n_+$, let $\mathcal{A} + \mathcal{B} = \{\boldsymbol{\alpha} + \boldsymbol{\beta} : \boldsymbol{\alpha} \in \mathcal{A},\ \boldsymbol{\beta} \in \mathcal{B}\}$ denote their Minkowski sum. Let $\mathbb{S}^{\mathcal{A}}_+$ denote the cone of positive semidefinite matrices in $\mathbb{S}^{\mathcal{A}}$, and $\mathbb{N}^{\mathcal{A}}$ the cone of nonnegative matrices in $\mathbb{S}^{\mathcal{A}}$. By construction, $\boldsymbol{x}^{\square\mathcal{A}} \in \mathbb{S}^{\mathcal{A}}_+$ for every $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{x}^{\square\mathcal{A}} \in \mathbb{N}^{\mathcal{A}}$ for every $\boldsymbol{x} \in \mathbb{R}^n_+$.

2.5 Lifting POP (1) in $\mathbb{R}^n$ to the space of higher dimensional symmetric matrices

Define $\boldsymbol{r} : \mathbb{Z}^n_+ \to \mathbb{Z}^n_+$ by

$$r_i(\boldsymbol{\alpha}) \ = \ \begin{cases} 0 & \text{if } i \in I_{\mathrm{bin}} \text{ and } \alpha_i = 0, \\ 1 & \text{if } i \in I_{\mathrm{bin}} \text{ and } \alpha_i > 0, \\ \alpha_i & \text{otherwise } (\textit{i.e.}, i \in I_{\mathrm{box}}). \end{cases} \tag{10}$$

Then, for each monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$ with $\boldsymbol{\alpha} \in \mathbb{Z}^n_+$ and $\boldsymbol{x} \in S$, we observe that

$$\boldsymbol{x}^{\boldsymbol{\alpha}} \ = \ \Pi_{i\in I_{\mathrm{bin}}} x_i^{\alpha_i} \Pi_{j\in I_{\mathrm{box}}} x_j^{\alpha_j} \ = \ \Pi_{i\in I_{\mathrm{bin}}} x_i^{r_i(\boldsymbol{\alpha})} \Pi_{j\in I_{\mathrm{box}}} x_j^{\alpha_j} \ = \ \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\alpha})}.$$

Thus, each monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$ in the objective function $f(\boldsymbol{x})$ of POP (1) with binary and box constraints may be replaced by $\boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\alpha})}$. We may assume without loss of generality that $\mathrm{supp}(f) = \boldsymbol{r}(\mathrm{supp}(f))$.

To construct a doubly nonnegative (DNN) relaxation of POP (1), we first choose a nonempty finite subset $\mathcal{A}$ of $\mathbb{Z}^n_+$ satisfying the condition that

$$\boldsymbol{0} \in \mathcal{A} = \boldsymbol{r}(\mathcal{A}) \text{ and } \mathrm{supp} f = \boldsymbol{r}(\mathrm{supp}(f)) \subset \mathcal{A} + \mathcal{A}. \tag{11}$$

By choosing a $|\mathcal{A}| \times |\mathcal{A}|$ matrix $\boldsymbol{Q}^0 \in \mathbb{S}^{\mathcal{A}}$ such that $f(\boldsymbol{x}) = \langle \boldsymbol{Q}^0, \boldsymbol{x}^{\square\mathcal{A}} \rangle$, we rewrite POP (1) as

$$\min\{\langle \boldsymbol{Q}^0, \boldsymbol{X} \rangle : \boldsymbol{X} \in T\}, \tag{12}$$

where $T = \left\{\boldsymbol{x}^{\square\mathcal{A}} \in \mathbb{S}^{\mathcal{A}} : \boldsymbol{x} \in S\right\}$. As mentioned in Section 2.4, we relax T to a convex subset of $\mathbb{S}^{\mathcal{A}}$, which is the intersection of a hyperplane and two cones $\mathbb{K}^1$, $\mathbb{K}^2$ in $\mathbb{S}^{\mathcal{A}}$. We discuss on the cones in detail in Section 4.

3 A class of polyhedral cones and the metric projection onto them

Before describing the class of polyhedral cones we are interested in, we first discuss why the cones become essential for the subsequent discussions. Recall that POP (1) in the n-dimensional space $\mathbb{R}^n$ has been lifted to the problem (12) in the symmetric matrix space $\mathbb{S}^{\mathcal{A}}$. The two problems are equivalent under the correspondence $\mathbb{S}^{\mathcal{A}} \ni \boldsymbol{x}^{\square\mathcal{A}} \leftrightarrow \boldsymbol{X} \in \mathbb{S}^{\mathcal{A}}$, or componentwisely $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha\beta}} = \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \leftrightarrow X_{\boldsymbol{\alpha\beta}}$ $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$. In addition to that, many identities and inequalities hold among the elements $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha\beta}} = \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}}$ of $\boldsymbol{x}^{\square\mathcal{A}}$ $((\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A})$ for all $\boldsymbol{x} \in S$, which are then translated to equalities and inequalities in $X_{\boldsymbol{\alpha\beta}}$ $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$.

In particular, if the relation $\boldsymbol{r}(\boldsymbol{\alpha} + \boldsymbol{\beta}) = \boldsymbol{r}(\boldsymbol{\gamma} + \boldsymbol{\delta})$ holds, then $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha\beta}} = \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} = (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\gamma\delta}} = \boldsymbol{x}^{\boldsymbol{\gamma}+\boldsymbol{\delta}}$. Thus, the relation $\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta}) = \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})$ naturally induces an equivalence relation $\sim$ in $\mathcal{A} \times \mathcal{A}$ such that $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim (\boldsymbol{\gamma}, \boldsymbol{\delta})$ if and only if $\boldsymbol{r}(\boldsymbol{\alpha} + \boldsymbol{\beta}) = \boldsymbol{r}(\boldsymbol{\gamma} + \boldsymbol{\delta})$ holds. With this equivalence relation $\sim$, a common nonnegative value can be assigned to $X_{\boldsymbol{\alpha\beta}}$ for all $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in each equivalence class. The use of the equivalence class in this way considerably simplifies the description of the polyhedral cone $\mathbb{L}$ used for $\mathbb{K}^2$ in the DNN relaxation (2).

Translating the inequalities in $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha\beta}}$ $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$ to the ones in $X_{\boldsymbol{\alpha\beta}}$ $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$ for the construction of $\mathbb{L}$ is not as straightforward as in the case for equalities. In fact, only a subset of the inequalities in $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha\beta}}$ is translated to the ones for $\mathbb{L}$. If all the inequalities from $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha\beta}}$ $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A}$ are included in the resulting cone $\mathbb{L}$, then the computation of the metric projection from $\mathbb{S}^{\mathcal{A}}$ onto $\mathbb{L}$ may be neither efficient nor accurate. Thus, the resulting cone $\mathbb{L}$ should be constructed with an appropriate structure conducive for the efficient and accurate computation of the metric projection.

For this purpose, a chain of equivalence classes $[(\boldsymbol{\alpha}^1, \boldsymbol{\beta}^1)], \ldots, [(\boldsymbol{\alpha}^m, \boldsymbol{\beta}^m)]$ along which the chain of inequalities $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}_1\boldsymbol{\beta}_1} \geq \cdots \geq (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}_m\boldsymbol{\beta}_m}$ is satisfied for every $\boldsymbol{x} \in S$ is defined by introducing a preorder $\succeq$ in $\mathcal{A} \times \mathcal{A}$. The inequalities generated from a family of disjoint chains are used to construct $\mathbb{L}$. The cone $\mathbb{L}$ constructed this way is essential for the algorithm in Section 3.3, as the most crucial step of the algorithm is in computing the metric projection onto $\mathbb{L}$.

Suppose that $S = [0, 1]^2$. Then $x_1 \geq x_1^2 \geq x_1^2 x_2$, $x_2 \geq x_2^2 \geq x_1 x_2^2$, $x_1 x_2 \geq x_1^2 x_2^2$ form a family of three disjoint chains of inequalities, which can be translated to the corresponding inequalities in $X_{\boldsymbol{\alpha\beta}}$ $((\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A} \times \mathcal{A})$ to construct $\mathbb{L}$. But an inequality such as $x_1 x_2 \geq x_1 x_2^2$ should not be added since it is not disjoint with the last two chains.

We note that one can use a directed graph to represent a chain of inequalities such that the nodes are the exponents of the monomials and the directed edges denote the inequalities between the monomials. For example, the first chain of inequalities can be represented as $(1,0) \rightarrow (2,0) \rightarrow (2,1)$. As for the case when $x_1 \in \{0,1\}$ and $x_2 \in [0,1]$, equivalence classes should be considered more carefully. For example, $x_1^2 x_2$ and $x_1 x_2$ are monomials in an equivalence class since $\boldsymbol{r}((2,1)) = \boldsymbol{r}((1,1))$. Thus, the family of chains above itself is not disjoint.

Throughout this section, for simplicity, the symmetric matrix space $\mathbb{S}^{\mathcal{A}}$ is identified with the $|\mathcal{A}| \times |\mathcal{A}|$-dimensional Euclidean space. Although the symmetry is lost by this generalization, it does not affect the construction of the class of polyhedral cones. Indeed, this generalization enables us to apply the result and the method in this section directly to the sparse DNN relaxation of POP (1) in Section 5, and to the discussion related to the Lagrangian-DNN relaxation derived from the CPP reformulation of a class of QOPs with equality, binary and complementarity constraints [16] in Section 6. This is another motivation for the generalization.

3.1 Preorder, equivalence relations and a chain of equivalence classes

Let $\boldsymbol{\Theta}$ be a nonempty finite set, and $\mathbb{R}^{\boldsymbol{\Theta}}$ be the $|\boldsymbol{\Theta}|$-dimensional Euclidean space of row vectors $\boldsymbol{X}$ whose elements are indexed by $\boldsymbol{\Theta}$, i.e., the θth element of $\boldsymbol{X}$ is denoted by X_θ ($\theta \in \boldsymbol{\Theta}$). Let $\mathbb{R}^{\boldsymbol{\Theta}}_+$ denote the nonnegative orthant $\{\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{\Theta}} : X_{\boldsymbol{\theta}} \geq 0 \text{ for every } \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ of $\mathbb{R}^{\boldsymbol{\Theta}}$. We call a binary relation $\succeq$ in $\boldsymbol{\Theta}$ a *preorder* if it satisfies

$$\begin{aligned}
&\boldsymbol{\theta} \succeq \boldsymbol{\theta} \text{ for every } \boldsymbol{\theta} \in \boldsymbol{\Theta} \ (\textit{reflexive}), \\
&\boldsymbol{\theta}^1 \succeq \boldsymbol{\theta}^2 \text{ and } \boldsymbol{\theta}^2 \succeq \boldsymbol{\theta}^3 \Rightarrow \boldsymbol{\theta}^1 \succeq \boldsymbol{\theta}^3 \ (\textit{transitive}).
\end{aligned}$$

The preorder $\succeq$ induces a *strict preorder* $\succ$ and an *equivalence relation* $\sim$:

$$\begin{aligned}
\boldsymbol{\theta} \succ \boldsymbol{\eta} \ &\Leftrightarrow \ \boldsymbol{\theta} \succeq \boldsymbol{\eta} \text{ and } \boldsymbol{\eta} \not\succeq \boldsymbol{\theta}, \\
\boldsymbol{\theta} \sim \boldsymbol{\eta} \ &\Leftrightarrow \ \boldsymbol{\theta} \succeq \boldsymbol{\eta} \text{ and } \boldsymbol{\eta} \succeq \boldsymbol{\theta}.
\end{aligned} \tag{13}$$

Let $[\boldsymbol{\theta}]$ denote the *equivalence class* which contains $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, i.e., $[\boldsymbol{\theta}] = \{\boldsymbol{\eta} \in \boldsymbol{\Theta} : \boldsymbol{\eta} \sim \boldsymbol{\theta}\}$. We can consistently use $\succeq$ and $\succ$ in the family of equivalence classes $\{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ such that $[\boldsymbol{\theta}] \succeq [\boldsymbol{\eta}]$ (or $[\boldsymbol{\theta}] \succ [\boldsymbol{\eta}]$) if $\boldsymbol{\theta} \succeq \boldsymbol{\eta}$ (or $\boldsymbol{\theta} \succ \boldsymbol{\eta}$). In fact, $\succeq$ acts as a *partial order* in the family, which is reflexive, transitive, and *antisymmetric, i.e.*, $[\boldsymbol{\theta}] \succeq [\boldsymbol{\eta}]$ and $[\boldsymbol{\eta}] \succeq [\boldsymbol{\theta}] \Rightarrow [\boldsymbol{\theta}] = [\boldsymbol{\eta}]$. We would frequently denote an equivalence class by $E_{\boldsymbol{\sigma}}$, where $\boldsymbol{\sigma}$ means a representative element of the equivalence class or a symbol attached to the class.

A finite sequence of equivalence classes $E_{\boldsymbol{\sigma}^1}, \ldots, E_{\boldsymbol{\sigma}^\ell}$ for $\ell \geq 1$ is called a *chain* if $E_{\boldsymbol{\sigma}^1} \succ \cdots \succ E_{\boldsymbol{\sigma}^\ell}$. In particular, a single equivalence class itself is a chain. For simplicity of notation, each chain $E_{\boldsymbol{\sigma}^1}, \ldots, E_{\boldsymbol{\sigma}^\ell}$ is identified with the family of equivalence classes $\{E_{\boldsymbol{\sigma}^1}, \ldots, E_{\boldsymbol{\sigma}^\ell}\}$. A chain Γ is *maximal* if it is not a proper subfamily of any other chain. Two chains Γ^1 and Γ^2 are *disjoint* if $\Gamma^1 \cap \Gamma^2 = \emptyset$.

3.2 A class of polyhedral cones

Let $\{\Gamma_1, \ldots, \Gamma_r\}$ be a given family of chains of equivalence classes. Since $[\boldsymbol{\theta}]$ itself is a chain of equivalence classes ($\boldsymbol{\theta} \in \boldsymbol{\Theta}$), we may assume without loss of generality that the family $\{\Gamma_1, \ldots, \Gamma_r\}$ covers the family of equivalence classes $\{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ such that

$$\bigcup_{p=1}^{r} \Gamma_p = \{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}. \tag{14}$$

Define

$$\mathbb{L} \;=\; \left\{ \boldsymbol{X} \in \mathbb{R}_+^{\boldsymbol{\Theta}} : \begin{array}{l} X_{\boldsymbol{\theta}} = X_{\boldsymbol{\eta}} \text{ if } \boldsymbol{\theta} \sim \boldsymbol{\eta}, \\ X_{\boldsymbol{\theta}} \geq X_{\boldsymbol{\eta}} \text{ if } \Gamma_p \ni [\boldsymbol{\theta}] \succ [\boldsymbol{\eta}] \in \Gamma_p \ (p = 1, \ldots, r) \end{array} \right\}.$$

Obviously, $\mathbb{L}$ forms a polyhedral cone in $\mathbb{R}^{\boldsymbol{\Theta}}$.

Remark 3.1. As a special case of families $\{\Gamma_1, \ldots, \Gamma_r\}$ of chains of equivalence classes, we can consider "the finest family" $\{\{[\boldsymbol{\theta}], [\boldsymbol{\tau}]\} : \boldsymbol{\theta} \in \boldsymbol{\Theta},\ \boldsymbol{\tau} \in \boldsymbol{\Theta},\ \boldsymbol{\theta} \succ \boldsymbol{\tau}\}$ to impose all the inequalities induced from the preorder $\succeq$ on $\mathbb{L}$. In this case, the resulting "smallest" cone $\mathbb{L}^{\text{finest}}$ is of the form

$$\mathbb{L}^{\text{finest}} \;=\; \left\{ \boldsymbol{X} \in \mathbb{R}_+^{\boldsymbol{\Theta}} : \begin{array}{l} X_{\boldsymbol{\theta}} = X_{\boldsymbol{\eta}} \text{ if } \boldsymbol{\theta} \sim \boldsymbol{\eta}, \\ X_{\boldsymbol{\theta}} \geq X_{\boldsymbol{\eta}} \text{ if } [\boldsymbol{\theta}] \succ [\boldsymbol{\eta}] \end{array} \right\}.$$

The finest family, however, is not necessarily disjoint, and the computation of the metric projection $\Pi_{\mathbb{L}^{\text{finest}}}(\cdot)$ can be extremely expensive. In the next section, we shall impose a certain disjoint property on the family $\{\Gamma_1, \ldots, \Gamma_r\}$ at the expense of constructing a larger cone $\mathbb{L}$ so that the metric projection from $\mathbb{R}^{\boldsymbol{\Theta}}$ onto $\mathbb{L}$ can be computed efficiently.

The cone $\mathbb{L}$ can be rewritten as

$$\mathbb{L} \;=\; \left\{ \boldsymbol{X} \in \mathbb{R}^{\boldsymbol{\Theta}} : \begin{array}{l} X_{\boldsymbol{\theta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ \text{ if } \boldsymbol{\theta} \in E_{\boldsymbol{\sigma}} \in \Gamma_p \ (p = 1, \ldots, r), \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} \text{ if } \Gamma_p \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_p \ (p = 1, \ldots, r) \end{array} \right\}. \tag{15}$$

If the chain Γ_p is represented as

$$\Gamma_p = \{E_{\boldsymbol{\sigma}^1}, \ldots, E_{\boldsymbol{\sigma}^\ell}\}, \text{ where } E_{\boldsymbol{\sigma}^1} \succ \cdots \succ E_{\boldsymbol{\sigma}^\ell} \tag{16}$$

for a fixed $p \in \{1, \ldots, r\}$, then the inequalities $\xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}}$ for $\Gamma_p \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_p$ are written as $\xi_{\boldsymbol{\sigma}^j} \geq \xi_{\boldsymbol{\sigma}^{j+1}}$ ($j = 1, \ldots, \ell - 1$). Specifically, when Γ_p consists of a single equivalence class $E_{\boldsymbol{\sigma}^1}$, these inequalities vanish.

3.3 The metric projection from $\mathbb{R}^{\boldsymbol{\Theta}}$ onto $\mathbb{L}$

Now we consider the metric projection $\Pi_{\mathbb{L}}$ from $\mathbb{R}^{\boldsymbol{\Theta}}$ onto $\mathbb{L}$. First, we show a fundamental property of $\Pi_{\mathbb{L}}$. Let $\boldsymbol{Z} \in \mathbb{R}^{\boldsymbol{\Theta}}$ and $\overline{\boldsymbol{X}} = \Pi_{\mathbb{L}}(\boldsymbol{Z})$. By definition, $\overline{\boldsymbol{X}}$ is the unique optimal solution of the strictly convex quadratic optimization problem:

$$\begin{aligned} &\min \left\{ \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} (X_{\boldsymbol{\theta}} - Z_{\boldsymbol{\theta}})^2 : \boldsymbol{X} \in \mathbb{L} \right\} \\ &= \min \left\{ \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} (\xi_{\boldsymbol{\sigma}} - Z_{\boldsymbol{\theta}})^2 : \begin{array}{l} X_{\boldsymbol{\theta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ \text{ if } \boldsymbol{\theta} \in E_{\boldsymbol{\sigma}} \in \Gamma_p \ (p = 1, \ldots, r), \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} \text{ if } \Gamma_p \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_p \ (p = 1, \ldots, r) \end{array} \right\}. \end{aligned} \tag{17}$$

To characterize $\overline{\boldsymbol{X}} = \Pi_{\mathbb{L}}(\boldsymbol{Z})$ in a compact way, we use the notation $\mu(\boldsymbol{Z}, E) = \left(\sum_{\boldsymbol{\theta}\in E} Z_{\boldsymbol{\theta}}\right) / |E|$ to denote the average of $Z_{\boldsymbol{\theta}}$ ($\boldsymbol{\theta} \in E$), and $\mu_+(\boldsymbol{Z}, E) = \max\{\mu(\boldsymbol{Z}, E), \boldsymbol{0}\}$ for every $\boldsymbol{Z} \in \mathbb{R}^{\boldsymbol{\Theta}}$ and $E \subset \boldsymbol{\Theta}$.

Lemma 3.2. *Let $\boldsymbol{Z} \in \mathbb{R}^{\boldsymbol{\Theta}}$ and $\overline{\boldsymbol{X}} = \Pi_{\mathbb{L}}(\boldsymbol{Z})$. Assume that the family of chains $\Gamma_1, \ldots, \Gamma_r$ is a partition of the family of equivalence classes $\{[\boldsymbol{\theta}] : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, i.e., assume that $\Gamma_i \bigcap \Gamma_j = \emptyset$ ($i \neq j$) in addition to (14). For an arbitrary fixed $p \in \{1, \ldots, r\}$, denote the chain Γ_p as in (16).*

- *(a) If $\ell = 1$ (i.e., Γ_p consists of a single equivalence set $E_{\boldsymbol{\sigma}^1}$) or $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^1}) \geq \cdots \geq \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^\ell})$, then $\overline{X}_{\boldsymbol{\theta}} = \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$ ($\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}, j = 1, \ldots, \ell$).*
- *(b) Suppose that $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k}) < \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^{k+1}})$ for some $k = 1, \ldots, \ell - 1$. Then $\overline{X}_{\boldsymbol{\theta}} = \overline{X}_{\boldsymbol{\eta}}$ for every $\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^k}$ and $\boldsymbol{\eta} \in E_{\boldsymbol{\sigma}^{k+1}}$.*

Proof. By the assumption, we can transform the problem (17) further into

$$\min\left\{\sum_{q=1}^{r}\sum_{E_{\boldsymbol{\sigma}}\in\Gamma_q}\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}}}(\xi_{\boldsymbol{\sigma}} - Z_{\boldsymbol{\theta}})^2 : \begin{array}{l} X_{\boldsymbol{\theta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ \text{ if } \boldsymbol{\theta} \in E_{\boldsymbol{\sigma}} \in \Gamma_q \ (q = 1, \ldots, r), \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} \text{ if } \Gamma_q \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_q \ (q = 1, \ldots, r) \end{array}\right\}$$
$$= \sum_{q=1}^{r}\min\left\{\sum_{E_{\boldsymbol{\sigma}}\in\Gamma_q}\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}}}(\xi_{\boldsymbol{\sigma}} - Z_{\boldsymbol{\theta}})^2 : \begin{array}{l} X_{\boldsymbol{\theta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ \text{ if } \boldsymbol{\theta} \in E_{\boldsymbol{\sigma}} \in \Gamma_q, \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} \text{ if } \Gamma_q \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_q \end{array}\right\}.$$

This implies that the computation of $\overline{\boldsymbol{X}} = \Pi_{\mathbb{L}}(\boldsymbol{Z})$ is divided into r subproblems

$$\min\left\{\sum_{E_{\boldsymbol{\sigma}}\in\Gamma_q}\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}}}(\xi_{\boldsymbol{\sigma}} - Z_{\boldsymbol{\theta}})^2 : \begin{array}{l} X_{\boldsymbol{\theta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ \text{ if } \boldsymbol{\theta} \in E_{\boldsymbol{\sigma}} \in \Gamma_q, \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} \text{ if } \Gamma_q \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_q \end{array}\right\} \tag{18}$$

($q = 1, \ldots, r$). By assumption, the pth problem to compute ($X_{\boldsymbol{\theta}}$ ($\theta \in E_{\boldsymbol{\sigma}} \in \Gamma_p$) is written as

$$\min\left\{\sum_{j=1}^{\ell}\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}^j}}(\xi_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}})^2 : \begin{array}{l} X_{\boldsymbol{\theta}} = \xi_{\boldsymbol{\sigma}^j} \in \mathbb{R}_+ \text{ if } \boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j} \ (j = 1, \ldots, \ell), \\ \xi_{\boldsymbol{\sigma}^j} \geq \xi_{\boldsymbol{\sigma}^{j+1}} \ \ (j = 1, \ldots, \ell - 1) \end{array}\right\}. \tag{19}$$

We can rewrite each term $\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}^j}}(\xi_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}})^2$ of the objective function of the pth subproblem (19) as $|E_{\boldsymbol{\sigma}^j}|\left(\xi_{\boldsymbol{\sigma}^j} - \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})\right)^2 + \sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}^j}} Z_{\boldsymbol{\theta}}^2 - |E_{\boldsymbol{\sigma}^j}|\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})^2$, which is a strictly convex quadratic function in a single variable $\xi_{\boldsymbol{\sigma}^j} \in \mathbb{R}$ with the unique minimizer $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$. Hence, if $\bar{\xi}_{\boldsymbol{\sigma}^j} > \hat{\xi}_{\boldsymbol{\sigma}^j} \geq \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$ or $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j}) \geq \hat{\xi}_{\boldsymbol{\sigma}^j} > \bar{\xi}_{\boldsymbol{\sigma}^j}$, then $\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}^j}}(\hat{\xi}_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}})^2 < \sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}^j}}(\bar{\xi}_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}})^2$. This fact is used to prove both assertions (a) and (b).

(a) Let $\bar{\xi}_{\boldsymbol{\sigma}^j} = \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$ ($j = 1, \ldots, \ell$). From the assumption that $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^1}) \geq \cdots \geq \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^\ell})$, we see that $\bar{\xi}_{\boldsymbol{\sigma}^1} \geq \cdots \geq \bar{\xi}_{\boldsymbol{\sigma}^\ell} \geq 0$. Hence $\overline{X}_{\boldsymbol{\theta}} = \bar{\xi}_{\boldsymbol{\sigma}^j}$ ($\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}, j = 1, \ldots, \ell$) satisfies the constraints of the subproblem (19). By the fact previously mentioned, we also see that $\bar{\xi}_{\boldsymbol{\sigma}^j} = \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$ minimizes each term $\sum_{\boldsymbol{\theta}\in E_{\boldsymbol{\sigma}^j}}\left(\xi_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}}\right)^2$ in $\xi_{\boldsymbol{\sigma}^j} \geq 0$

$(j=1,\ldots,\ell)$. Therefore $\overline{X}_{\boldsymbol{\theta}} = \bar{\xi}_{\boldsymbol{\sigma}^j}$ $(\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}, j=1,\ldots,\ell)$ is the minimizer of the subproblem (19).

(b) Since $\overline{\boldsymbol{X}}$ is feasible, there exist $\bar{\xi}_j$ such that $\overline{X}_{\boldsymbol{\theta}} = \bar{\xi}_j \in \mathbb{R}_+$ if $\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}$ $(j=1,\ldots,\ell)$. Assume on the contrary that $\bar{\xi}_{\boldsymbol{\sigma}^k} > \bar{\xi}_{\boldsymbol{\sigma}^{k+1}}$. Consider $\hat{\xi}$ such that $\hat{\xi}_{\boldsymbol{\sigma}^j} = \bar{\xi}_{\boldsymbol{\sigma}^j}$ $(j \neq k,\ j \neq k+1)$. We consider two cases. First, suppose that $\mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k}) < \bar{\xi}_{\boldsymbol{\sigma}^k}$. Set $\hat{\xi}_{\boldsymbol{\sigma}^k} = \hat{\xi}_{\boldsymbol{\sigma}^{k+1}} = \max\{\mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k}), \bar{\xi}_{\boldsymbol{\sigma}^{k+1}}\}$. Then we observe that

$$\hat{\xi}_{\boldsymbol{\sigma}^1} \geq \cdots \geq \hat{\xi}_{\boldsymbol{\sigma}^{k-1}} \geq \bar{\xi}_{\boldsymbol{\sigma}^k} > \hat{\xi}_{\boldsymbol{\sigma}^k} = \hat{\xi}_{\boldsymbol{\sigma}^{k+1}} \geq \bar{\xi}_{\boldsymbol{\sigma}^{k+1}} \geq \hat{\xi}_{\boldsymbol{\sigma}^{k+2}} \geq \cdots \geq \hat{\xi}_{\boldsymbol{\sigma}^\ell}.$$

Hence

$$\sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^k}} (\hat{\xi}_{\boldsymbol{\sigma}^k} - Z_{\boldsymbol{\theta}})^2 < \sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^k}} (\bar{\xi}_{\boldsymbol{\sigma}^k} - Z_{\boldsymbol{\theta}})^2 \text{ (since } \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k}) \leq \hat{\xi}_{\boldsymbol{\sigma}^k} < \bar{\xi}_{\boldsymbol{\sigma}^k}),$$

$$\sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^{k+1}}} (\hat{\xi}_{\boldsymbol{\sigma}^{k+1}} - Z_{\boldsymbol{\theta}})^2 \begin{cases} < \displaystyle\sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^{k+1}}} (\bar{\xi}_{\boldsymbol{\sigma}^{k+1}} - Z_{\boldsymbol{\theta}})^2 & \text{if } \bar{\xi}_{\boldsymbol{\sigma}^{k+1}} < \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k}) \leq \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^{k+1}}), \\ = \displaystyle\sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^{k+1}}} (\bar{\xi}_{\boldsymbol{\sigma}^{k+1}} - Z_{\boldsymbol{\theta}})^2 & \text{otherwise.} \end{cases}$$

It follows that $(\widehat{X}_{\boldsymbol{\theta}} = \hat{\xi}_{\boldsymbol{\sigma}^j}$ $(\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j},\ j=1,\ldots,\ell))$ is a feasible solution of the subproblem (19), and that the objective value $\sum_{i=1}^{\ell} \sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}} (\hat{\xi}_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}})^2$ is smaller than the optimal value $\sum_{j=1}^{\ell} \sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}} (\bar{\xi}_{\boldsymbol{\sigma}^j} - Z_{\boldsymbol{\theta}})^2$. This is a contradiction. Next suppose that $\bar{\xi}_{\boldsymbol{\sigma}^k} \leq \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k})$. In this case, set $\hat{\xi}_{\boldsymbol{\sigma}^k} = \hat{\xi}_{\boldsymbol{\sigma}^{k+1}} = \bar{\xi}_{\boldsymbol{\sigma}^k}$. Then

$$\hat{\xi}_{\boldsymbol{\sigma}^1} \geq \cdots \geq \hat{\xi}_{\boldsymbol{\sigma}^{k-1}} \geq \bar{\xi}_{\boldsymbol{\sigma}^k} = \hat{\xi}_{\boldsymbol{\sigma}^k} = \hat{\xi}_{\boldsymbol{\sigma}^{k+1}} > \bar{\xi}_{\boldsymbol{\sigma}^{k+1}} \geq \hat{\xi}_{\boldsymbol{\sigma}^{k+2}} \geq \cdots \geq \hat{\xi}_{\boldsymbol{\sigma}^\ell},$$
$$\sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^{k+1}}} (\hat{\xi}_{\boldsymbol{\sigma}^{k+1}} - Z_{\boldsymbol{\theta}})^2 < \sum_{\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^{k+1}}} (\bar{\xi}_{\boldsymbol{\sigma}^{k+1}} - Z_{\boldsymbol{\theta}})^2 \text{ (since } \bar{\xi}_{\boldsymbol{\sigma}^{k+1}} < \bar{\xi}_{\boldsymbol{\sigma}^k} = \hat{\xi}_{\boldsymbol{\sigma}^{k+1}} \leq \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k})).$$

As we can similarly derive a contradiction, assertion (b) follows. □

Based on Lemma 3.2, we present an algorithm for computing the metric projection $\overline{\boldsymbol{X}} = \Pi_{\mathbb{L}}(\boldsymbol{Z})$ of a given $\boldsymbol{Z} \in \mathbb{S}^{\mathcal{A}}$.

Algorithm 3.3. For $p=1,\ldots,r$, execute the following steps.

Step 0. Represent Γ_p as in (16). Compute $\mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$ $(j=1,\ldots,\ell)$.

Step 1. If $\ell = 1$ or $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^1}) \geq \cdots \geq \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^\ell})$, output $\overline{X}_{\boldsymbol{\theta}} = \mu_+(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^j})$ $(\boldsymbol{\theta} \in E_{\boldsymbol{\sigma}^j}, j=1,\ldots,\ell)$.

Step 2. Otherwise, find $k \in \{1,\ldots,\ell-1\}$ such that $\mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^k}) < \mu(\boldsymbol{Z}, E_{\boldsymbol{\sigma}^{k+1}})$. Replace Γ_p by $\{E_{\boldsymbol{\sigma}^1},\ldots,E_{\boldsymbol{\sigma}^{k-1}}, E_{\boldsymbol{\sigma}^k} \cup E_{\boldsymbol{\sigma}^{k+1}}, E_{k+2},\ldots,E_{\boldsymbol{\sigma}^\ell}\}$, where $E_{\boldsymbol{\sigma}^k} \cup E_{\boldsymbol{\sigma}^{k+1}}$ is regarded as a single equivalence class, and go to Step 0.

4 Dense DNN relaxations of POP (1)

Recall that the POP (1) in $\mathbb{R}^n$ is first lifted in Section 2.5 to the problem (12) in $\mathbb{S}^{\mathcal{A}}$, where $\mathcal{A} \subset \mathbb{Z}^n_+$ satisfies (11). Then, the lifted problem (12) is relaxed to a COP of the form (2). The construction of the cone $\mathbb{L}_{\rm d}$ for $\mathbb{K}^2$ used in (2) is presented in Section 4.1 using the results in Section 3. For the computation of the metric projection $\Pi_{\mathbb{L}_{\rm d}}$, Lemma 3.2 and Algorithm 3.3 can be applied. We illustrate the computation with an example in Section 4.2. In Sections 4.3 and 4.4, we describe the standard choices of $\mathcal{A}$ for DNN relaxations of QOPs with binary and box constraints, and a hierarchy of DNN relaxations of POPs with binary and box constraints, respectively.

4.1 Construction of a polyhedral cone $\mathbb{L}_{\rm d}$ for $\mathbb{K}^2$

Assume that a nonempty subset $\mathcal{A}$ of $\mathbb{Z}^n_+$ satisfying (11) is given throughout this section. Let $\boldsymbol{\Theta} = \mathcal{A} \times \mathcal{A}$. Then, $\mathbb{S}^{\mathcal{A}}$ can be identified with $\mathbb{R}^{\boldsymbol{\Theta}}$ where the $(\boldsymbol{\alpha}, \boldsymbol{\beta})$th element of $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{\Theta}}$ is written as $X_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ $((\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \boldsymbol{\Theta})$. We may regard $\boldsymbol{Q}^0 \in \mathbb{R}^{\boldsymbol{\Theta}}$ and $\boldsymbol{x}^{\square\mathcal{A}} \in \mathbb{R}^{\boldsymbol{\Theta}}$ in (12). To generalize the inequality $x_i \geq x_i^2$ for every $x_i \in [0, 1]$ to monomials in $\boldsymbol{x}^{\square\mathcal{A}}$, we introduce a binary relation $\succeq$ in $\boldsymbol{\Theta}$. By the definition (10) of $\boldsymbol{r}$, we know that $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}\boldsymbol{\beta}} = \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} = \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta})} \geq \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})} = \boldsymbol{x}^{\boldsymbol{\gamma}+\boldsymbol{\delta}} = (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\gamma}\boldsymbol{\delta}}$ holds if $\boldsymbol{x} \in S$ and $\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta}) \leq \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})$. Hence it may be natural to define $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \succeq (\boldsymbol{\gamma}, \boldsymbol{\delta})$ by $\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta}) \leq \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})$. Such a preorder will be investigated in Section 6. Here we impose an additional requirement for the preorder to induce a polyhedral cone $\mathbb{L}_{\rm d}$ which allows the efficient computation of the projection onto it.

We now define a binary relation $\succeq$ in $\boldsymbol{\Theta}$ such that for every pair of $(\boldsymbol{\alpha}, \boldsymbol{\beta}), (\boldsymbol{\gamma}, \boldsymbol{\delta}) \in \boldsymbol{\Theta}$, $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \succeq (\boldsymbol{\gamma}, \boldsymbol{\delta})$ if and only if there exists a positive number $c \geq 1$ such that

$$\begin{aligned} r_i(\boldsymbol{\alpha}+\boldsymbol{\beta}) \quad &= \quad r_i(\boldsymbol{\gamma}+\boldsymbol{\delta}) \quad (i \in I_{\rm bin}) \\ c(\alpha_j+\beta_j) = c r_j(\boldsymbol{\alpha}+\boldsymbol{\beta}) \quad &= \quad r_j(\boldsymbol{\gamma}+\boldsymbol{\delta}) = \gamma_j + \delta_j \quad (j \in I_{\rm box}), \end{aligned}$$

which obviously implies $\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta}) \leq \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})$. Using the definition (10) of $\boldsymbol{r} : \mathbb{Z}^n_+ \to \mathbb{Z}^n_+$, it is easy to verify that $\succeq$ is a preorder. By definition, we see that

$$\begin{aligned} (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}\boldsymbol{\beta}} = \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} = \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta})} \quad &\geq \quad \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})} = \boldsymbol{x}^{\boldsymbol{\gamma}+\boldsymbol{\delta}} = (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\gamma}\boldsymbol{\delta}} \\ & \quad\ \text{if } \boldsymbol{x} \in S \text{ and } (\boldsymbol{\alpha}, \boldsymbol{\beta}) \succeq (\boldsymbol{\gamma}, \boldsymbol{\delta}), \end{aligned} \tag{20}$$

where $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}\boldsymbol{\beta}}$ denotes the $(\boldsymbol{\alpha}, \boldsymbol{\beta})$th element of $\boldsymbol{x}^{\square\mathcal{A}} \in \mathbb{S}^{\mathcal{A}}$.

As shown in Section 3, the preorder $\succeq$ induces a strict preorder $\succ$ and an equivalence relation $\sim$ by (13). Let $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \boldsymbol{\Theta}$. Then $[\boldsymbol{\theta}] = \{(\boldsymbol{\gamma}, \boldsymbol{\delta}) \in \boldsymbol{\Theta} : \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta}) = \boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta})\}$. Each equivalence class is characterized by $\boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})$, and the family of equivalence classes is denoted by $\{E_{\boldsymbol{\sigma}}\ (\boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})\}$ where $E_{\boldsymbol{\sigma}} = \{(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \boldsymbol{\Theta} : \boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta}) = \boldsymbol{\sigma}\}$. By definition, $(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}\boldsymbol{\beta}} = (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\gamma}\boldsymbol{\delta}}$ if and only if $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \succeq (\boldsymbol{\gamma}, \boldsymbol{\delta})$ and $(\boldsymbol{\gamma}, \boldsymbol{\delta}) \succeq (\boldsymbol{\alpha}, \boldsymbol{\beta})$. Hence, by (20),

$$(\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\alpha}\boldsymbol{\beta}} = (\boldsymbol{x}^{\square\mathcal{A}})_{\boldsymbol{\gamma}\boldsymbol{\delta}} \text{ if } \boldsymbol{x} \in S \text{ and } (\boldsymbol{\alpha}, \boldsymbol{\beta}) \sim (\boldsymbol{\gamma}, \boldsymbol{\delta}). \tag{21}$$

Lemma 4.1.

(a) *For each equivalence class* $E_{\boldsymbol{\sigma}}$ ($\boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})$)*, there exists a unique maximal chain* Γ *containing* $E_{\boldsymbol{\sigma}}$*, which is represented as*

$$\Gamma \;=\; \{E_{\boldsymbol{\tau}} : \tau_i = \sigma_i \ (i \in I_{bin}),\ c\tau_j = \sigma_j \ (j \in I_{box}) \text{ for some } c > 0\ \}.$$

(b) *The family of maximal chains partitions the family of equivalence classes. If we denote the family of maximal chains as* $\Gamma_1, \ldots, \Gamma_r$ *then* $\bigcup_{p=1}^r \Gamma_p = \{E_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})\}$ *and* $\Gamma_p \bigcap \Gamma_q = \emptyset$ $(p \neq q)$.

(c) *If* $I_{box} = \emptyset$*, then every maximal chain consists of a single equivalence class* $E_{\boldsymbol{\sigma}}$ *for some* $\boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})$.

Proof. It suffices to show assertion (a) because assertions (b) and (c) follow from (a). Let $\boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})$. Define Γ as in assertion (a). Obviously, $E_{\boldsymbol{\sigma}} \in \Gamma$. Enumerate Γ such that $\Gamma = \{E_{\boldsymbol{\tau}^1}, \ldots, E_{\boldsymbol{\tau}^\ell}\}$. By the definition of Γ, we can take distinct positive numbers $c^1, \ldots, c^\ell$ such that $\tau_i^k = \sigma_i \ (i \in I_{\rm bin}),\ c^k\tau_j^k = \sigma_j \ (j \in I_{\rm box})$ $(k = 1, \ldots, \ell)$. Rearranging the order of the equivalence sets in Γ if necessary, we may assume that $0 < c^1 < \cdots < c^\ell$. Then $E_{\boldsymbol{\tau}^1} \prec \cdots \prec E_{\boldsymbol{\tau}^\ell}$, which implies that Γ is a chain. Furthermore, we can rewrite Γ as $\Gamma = \{E_{\boldsymbol{\tau}} : E_{\boldsymbol{\tau}} \prec E_{\boldsymbol{\sigma}}\} \cup \{E_{\boldsymbol{\sigma}}\} \cup \{E_{\boldsymbol{\tau}} : E_{\boldsymbol{\sigma}} \prec E_{\boldsymbol{\tau}}\}$. This ensures the maximality of Γ. $\square$

Let $\Gamma_1, \ldots, \Gamma_r$ be the family of maximal chains of equivalence classes. Define

$$\begin{aligned}
\boldsymbol{H}^0 &= \text{ the } |\mathcal{A}| \times |\mathcal{A}| \text{ matrix in } \mathbb{S}^{\mathcal{A}} \text{ with 1 at the } (\boldsymbol{0},\boldsymbol{0})\text{th element and 0 elsewhere,} \\
\mathbb{L}_{\rm d} &= \left\{ \boldsymbol{X} \in \mathbb{N}^{\mathcal{A}} : \begin{array}{ll} X_{\boldsymbol{\alpha\beta}} = X_{\boldsymbol{\gamma\delta}} & \text{if } (\boldsymbol{\alpha},\boldsymbol{\beta}) \sim (\boldsymbol{\gamma},\boldsymbol{\delta}), \\ X_{\boldsymbol{\alpha\beta}} \geq X_{\boldsymbol{\alpha\beta}} & \text{if } \Gamma_p \ni [(\boldsymbol{\alpha},\boldsymbol{\beta})] \succ [(\boldsymbol{\gamma},\boldsymbol{\delta})] \in \Gamma_p \\ & (p = 1, \ldots, r) \end{array} \right\} \\
&= \left\{ \boldsymbol{X} \in \mathbb{S}^{\mathcal{A}} : \begin{array}{ll} X_{\boldsymbol{\alpha\beta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ & \text{if } (\boldsymbol{\alpha},\boldsymbol{\beta}) \in E_{\boldsymbol{\sigma}} \in \Gamma_p \ (p = 1, \ldots, r), \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} & \text{if } \Gamma_p \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_p \ (p = 1, \ldots, r) \end{array} \right\}.
\end{aligned}$$

Here the last identity follows from assertion (b) of Lemma 4.1. If $\boldsymbol{x}$ is a feasible solution of (12) with the objective value $\langle \boldsymbol{Q}^0,\, \boldsymbol{x}^{\square\mathcal{A}} \rangle = f(\boldsymbol{x})$, then $\boldsymbol{X} = \boldsymbol{x}^{\square\mathcal{A}}$ satisfies that

$$\begin{aligned}
&\langle \boldsymbol{Q}^0,\, \boldsymbol{x}^{\square\mathcal{A}} \rangle = \langle \boldsymbol{Q}^0,\, \boldsymbol{X} \rangle,\ \boldsymbol{X} \in \mathbb{S}^{\mathcal{A}}_+,\ \boldsymbol{X} \in \mathbb{N}^{\mathcal{A}},\ \langle \boldsymbol{H}^0,\, \boldsymbol{X} \rangle = X_{\boldsymbol{00}} = 1, \\
&X_{\boldsymbol{\alpha\beta}} = X_{\boldsymbol{\gamma\delta}} \text{ if } (\boldsymbol{\alpha},\boldsymbol{\beta}) \sim (\boldsymbol{\gamma},\boldsymbol{\delta}), \\
&X_{\boldsymbol{\alpha\beta}} \geq X_{\boldsymbol{\gamma\delta}} \text{ if } \Gamma_p \ni [(\boldsymbol{\alpha},\boldsymbol{\beta})] \succ [(\boldsymbol{\gamma},\boldsymbol{\delta})] \in \Gamma_p \ (p = 1, \ldots, r).
\end{aligned}$$

Note that the last two relations are obtained by (20) and (21). This implies that $\boldsymbol{X}$ is a feasible solution of COP (2) with $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}}_+$ and $\mathbb{K}^2 = \mathbb{L}_{\rm d}$ and attains the same objective value $\langle \boldsymbol{Q}^0,\, \boldsymbol{x}^{\square\mathcal{A}} \rangle$. Therefore, COP (2) serves as a DNN relaxation of POP (1). Notice that the construction of $\mathbb{L}_{\rm d}$ depends on the choices of $\mathcal{A}$. The standard choices of $\mathcal{A}$ are discussed for QOPs in Section 4.3 and for POPs in Section 4.4.

Lemma 4.2. *Let* $\boldsymbol{X}$ *be a feasible solution of COP* (2) *with* $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}}_+$ *and* $\mathbb{K}^2 = \mathbb{L}_d$*. Then*

(a) $0 \leq X_{\boldsymbol{\alpha\beta}} \leq 1$ *for every* $((\boldsymbol{\alpha},\boldsymbol{\beta}) \in \boldsymbol{\Theta})$.

(b) $\langle \boldsymbol{I},\, \boldsymbol{X} \rangle \leq |\mathcal{A}|$*, where* $\boldsymbol{I}$ *denotes the identity matrix in* $\mathbb{S}^{\mathcal{A}}$.

Proof. As assertion (b) follows from assertion (a), we only prove (a). It follows from the definition of $\mathbb{L}_{\rm d}$ that each element of $\boldsymbol{X}$ is nonnegative. Since $\boldsymbol{X} \in \mathbb{S}^{\mathcal{A}}_+$, it suffices to prove that all the diagonal elements of $\boldsymbol{X}$ is not greater than 1. We know that $1 = \langle \boldsymbol{H}^0, \boldsymbol{X} \rangle = X_{\boldsymbol{00}}$. Let $\boldsymbol{0} \neq \boldsymbol{\alpha} \in \mathcal{A}$ be fixed. We will show $X_{\boldsymbol{\alpha\alpha}} \leq 1$. Consider the 2×2 principal submatrix $\begin{pmatrix} X_{\boldsymbol{00}} & X_{\boldsymbol{0\alpha}} \\ X_{\boldsymbol{\alpha 0}} & X_{\boldsymbol{\alpha\alpha}} \end{pmatrix} = \begin{pmatrix} 1 & X_{\boldsymbol{0\alpha}} \\ X_{\boldsymbol{\alpha 0}} & X_{\boldsymbol{\alpha\alpha}} \end{pmatrix}$ of $\boldsymbol{X} \in \mathbb{S}^{\mathcal{A}}_+$. The positive semdifiniteness of the submatrix implies that $0 \leq X_{\boldsymbol{\alpha\alpha}} - X^2_{\boldsymbol{0\alpha}}$. On the other hand, we see that $r_i(\boldsymbol{0}+\boldsymbol{\alpha}) = r_i(\boldsymbol{\alpha}+\boldsymbol{\alpha})$ $(i \in I_{\rm bin})$ and $2r_j(\boldsymbol{0}+\boldsymbol{\alpha}) = 2\alpha_j = r_j(\boldsymbol{\alpha}+\boldsymbol{\alpha})$ $(j \in I_{\rm box})$. As a result, $(\boldsymbol{0}, \boldsymbol{\alpha}) \succeq (\boldsymbol{\alpha}, \boldsymbol{\alpha})$, and $X_{\boldsymbol{0\alpha}} \geq X_{\boldsymbol{\alpha\alpha}}$. It follows that $0 \leq X_{\boldsymbol{\alpha\alpha}} - X^2_{\boldsymbol{\alpha\alpha}} = X_{\boldsymbol{\alpha\alpha}}(1 - X_{\boldsymbol{\alpha\alpha}})$, which implies $0 \leq X_{\boldsymbol{\alpha\alpha}} \leq 1$. □

By assertion (b) of Lemma 4.2, we can apply Algorithm 2.1 (Accelerated BP Algorithm) to COP (2) with $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}}_+$ and $\mathbb{K}^2 = \mathbb{L}_{\rm d}$.

4.2 Computation of the metric projection $\Pi_{\mathbb{L}_{\rm d}}$ from $\mathbb{S}^{\mathcal{A}}$ onto $\mathbb{L}_{\rm d}$

We have already shown in Lemma 4.1 that the family of maximal chains $\{\Gamma_1, \ldots, \Gamma_r\}$ partitions the family of equivalence classes $\{E_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})\}$. Therefore, we can use Lemma 3.2 and Algorithm 3.3 for the computation of the metric projection $\Pi_{\mathbb{L}_{\rm d}}$ from $\mathbb{S}^{\mathcal{A}}$ onto $\mathbb{L}_{\rm d}$. Next, we give a simple example to illustrate the computations in Algorithm 3.3 based on Lemma 3.2.

Example 4.3. Let $I_{\rm bin} = \{1\}$, $I_{\rm box} = \{2\}$, $S = \{(x_1, x_2) \in \mathbb{R}^2 : x_i \in \{0,1\}\ (i \in I_{\rm bin}),\ x_j \in [0,1]\ (j \in I_{\rm box})\}$ and $\mathcal{A} = \{(0,0), (1,0), (0,1), (1,1)\}$. In this case, we see that $\boldsymbol{r}(\mathcal{A}) = \mathcal{A}$ and

$$\boldsymbol{x}^{\square\mathcal{A}} = \begin{pmatrix} 1 & x_1 & x_2 & x_1x_2 \\ x_1 & x_1^2 & x_1x_2 & x_1^2x_2 \\ x_2 & x_1x_2 & x_2^2 & x_1x_2^2 \\ x_1x_2 & x_1^2x_2 & x_1x_2^2 & x_1^2x_2^2 \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_2 & x_1x_2 \\ x_1 & x_1 & x_1x_2 & x_1x_2 \\ x_2 & x_1x_2 & x_2^2 & x_1x_2^2 \\ x_1x_2 & x_1x_2 & x_1x_2^2 & x_1x_2^2 \end{pmatrix}$$

for every $\boldsymbol{x} \in S$. We also see that $\boldsymbol{r}(\mathcal{A}+\mathcal{A}) = \{(0,0), (1,0), (0,1), (1,1), (0,2), (1,2)\}$. As a result, we have 6 equivalence classes corresponding to the different monomials 1, x_1, x_2, x_1x_2, x_2^2 and $x_1x_2^2$ in the reduced moment matrix on the right, and the equivalence classes are $E_{(0,0)}$, $E_{(1,0)}$, $E_{(0,1)}$, $E_{(1,1)}$, $E_{(0,2)}$, $E_{(1,2)}$. For example, $E_{(1,2)}$ consists of $(\boldsymbol{\alpha}, \boldsymbol{\beta}) = ((0,1),(1,1))$, $((1,1),(0,1))$, $((1,1),(1,1))$ corresponding to 3 monomials $x_1x_2^2$ appeared in the reduced moment matrix. The maximal chains are $\Gamma_1 = \{E_{(0,0)}\}$, $\Gamma_2 = \{E_{(1,0)}\}$, $\Gamma_3 = \{E_{(0,1)},\ E_{(0,2)}\}$, $\Gamma_4 = \{E_{(1,1)},\ E_{(1,2)}\}$. Thus, $\mathbb{L}_{\rm d}$ is represented as

$$\mathbb{L}_{\rm d} = \left\{ \boldsymbol{X} \in \mathbb{S}^{\mathcal{A}} : \begin{array}{l} X_{\boldsymbol{\alpha\beta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ \text{ if } (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in E_{\boldsymbol{\sigma}} \in \Gamma_p\ (p = 1, \ldots, 4), \\ \xi_{(0,1)} \geq \xi_{(0,2)},\ \xi_{(1,1)} \geq \xi_{(1,2)} \end{array} \right\}.$$

To illustrate how we compute $\overline{\boldsymbol{X}} = \Pi_{\mathbb{L}_{\rm d}}(\boldsymbol{Z})$ for a given $\boldsymbol{Z} \in \mathbb{S}^{\mathcal{A}}$, take $\Gamma_p = \Gamma_4 = \{E_{(1,1)},\ E_{(1,2)}\}$. By Lemma 3.2 (or applying Steps 0, 1 and 2 of Algorithm 3.3 with

$p = 4$), we obtain that

$$\overline{X}_{\boldsymbol{\alpha\beta}} = \begin{cases} \mu_+(\boldsymbol{Z}, E_{(1,1)}) & \text{if } \mu(\boldsymbol{Z}, E_{(1,1)}) \geq \mu(\boldsymbol{Z}, E_{(1,2)}) \\ \mu_+(\boldsymbol{Z}, E_{(1,1)} \cup E_{(1,2)}) & \text{otherwise,} \end{cases}$$
$$\text{for all } (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in E_{(1,1)},$$
$$\overline{X}_{\boldsymbol{\alpha\beta}} = \begin{cases} \mu_+(\boldsymbol{Z}, E_{(1,2)}) & \text{if } \mu(\boldsymbol{Z}, E_{(1,1)}) \geq \mu(\boldsymbol{Z}, E_{(1,2)}) \\ \mu_+(\boldsymbol{Z}, E_{(1,1)} \cup E_{(1,2)}) & \text{otherwise,} \end{cases}$$
$$\text{for all } (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in E_{(1,2)}.$$

4.3 Choosing $\mathcal{A}$ for the standard DNN relaxation of a QOP (6) with binary and box constraints

For the standard DNN relaxation QOP (6) with binary and box constraints in Section 2.3, we choose $\mathcal{A} = \left\{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \sum_{i=1}^n \alpha_i \leq 1\right\} = \{\boldsymbol{0}, \boldsymbol{e}^1, \ldots, \boldsymbol{e}^n\}$, where $\boldsymbol{e}^i$ denotes the ith coordinate vector with 1 at the ith element and 0 elsewhere ($i = 1, \ldots, n$). Then the moment matrix $\boldsymbol{x}^{\square\mathcal{A}}$ coincides with $(1, \boldsymbol{x})(1, \boldsymbol{x})^T$. The equivalence classes are $E_{\boldsymbol{e}^j} = \{(\boldsymbol{0}, \boldsymbol{e}^j), (\boldsymbol{e}^j, \boldsymbol{0}), (\boldsymbol{e}^j, \boldsymbol{e}^j)\}$ ($j \in I_{\text{bin}}$) in addition to the trivial ones from the symmetry of the moment matrix $\boldsymbol{x}^{\square\mathcal{A}}$. They induce the identities $x_j = x_j^2$ ($j \in I_{\text{bin}}$). The maximal chains which have more than one equivalence classes are $\{E_{\boldsymbol{e}^i},\ E_{2\boldsymbol{e}^i}\}$ ($i \in I_{\text{box}}$), which induce the inequalities $x_i \geq x_i^2$ ($i \in I_{\text{box}}$). Therefore, the cone $\mathbb{L}_{\text{d}}$ used for $\mathbb{K}^2$ in (2) is represented as $\mathbb{L}_{\text{d}} = \{\boldsymbol{X} \in \mathbb{N}^{\mathcal{A}} : X_{\boldsymbol{0}\boldsymbol{e}^j} = X_{\boldsymbol{e}^j\boldsymbol{e}^j}\ (j \in I_{\text{bin}}),\ X_{\boldsymbol{0}\boldsymbol{e}^i} \geq X_{\boldsymbol{e}^i\boldsymbol{e}^i}\ (i \in I_{\text{box}})\}$, which coincides with the cone $\mathbb{L}$ defined by (9) if we use the coordinates $\{0, 1, \ldots, n\}$ instead of $\mathcal{A}$ and identify $\mathbb{S}^{\mathcal{A}}$ with $\mathbb{S}^{1+n}$.

4.4 Choosing $\mathcal{A}$ for a hierarchy of DNN relaxations of a POP with binary and box constraints

In this subsection, we briefly discuss how the idea of the hierarchy of SDP relaxations proposed by Lasserre [21] for general POPs is incorporated in our DNN relaxation of POP (1) with binary and box constraints. Let ω_0 be the smallest positive integer which is no less than $(\deg f)/2$. For each $\omega \geq \omega_0$, let $\mathcal{A}^\omega = \{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \sum_{i=1}^n \alpha_i \leq \omega\}$. We call ω *a relaxation order*.

First, we apply $\boldsymbol{r}$ to the exponents of all monomials of $f(\boldsymbol{x})$, *i.e.*, replace every monomial $\boldsymbol{x}^\gamma$ of $f_0(\boldsymbol{x})$ by $\boldsymbol{x}^{\boldsymbol{r}(\gamma)}$. Then condition (11) is clearly satisfied with $\mathcal{A} = \boldsymbol{r}(\mathcal{A}^\omega) = \{\boldsymbol{r}(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathcal{A}^\omega\}$ for every $\omega \geq \omega_0$. By increasing the relaxation order ω from ω_0 and constructing a DNN relaxation with each $\mathcal{A} = \boldsymbol{r}(\mathcal{A}^\omega)$, a hierarchy of DNN relaxations of POP (1) can be obtained. If $I_{\text{bin}} = \{1, \ldots, n\}$, *i.e.*, $S = \{0, 1\}^n$, this construction is essentially the same as Lasserre's when it is applied to POP (1) with $S = \{0, 1\}^n$, except that a stronger DNN relaxation than a SDP relaxation is employed at each level of the hierarchy; See [20]. When $I_{\text{bin}} \neq \{1, \ldots, n\}$, our DNN relaxation and Lasserre's SDP relaxation at each level of the hierarchies are different. In order not to lengthen an already long paper, we will leave the theoretical study of the convergence of the optimal values of our hierarchy of DNN relaxations to the optimal value of POP (1) as a future work. Here we aim to demonstrate the effectiveness of our hierarchy of DNN relaxations with numerical experiments.

5 Sparse DNN relaxations of POP (1)

The sparse DNN relaxation of POP (1) presented in this section is based on [27], where a sparse version of the hierarchy of SDP relaxations of general POPs with inequality constraints was proposed by exploiting certain structured sparsity in the objective polynomial and constraints. Notice that POP (1) involves only simple constraints $x_i - x_i^2 = 0$ for $x_i \in \{0,1\}$ and $x_i - x_i^2 \geq 0$ for $x_i \in [0,1]$, which are both separable in x_i $(i = 1,\ldots,n)$. As a result, we can focus on the objective polynomial $f(\boldsymbol{x})$ for sparsity exploitation. The structured sparsity of $f(\boldsymbol{x})$ is readily observed in the nonzero pattern of its Hessian matrix.

For each subset C of $\{1,\ldots,n\}$, let $\mathbb{Z}^C_+ = \{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \alpha_i = 0 \text{ if } i \notin C\}$. We first assume that the objective polynomial $f(\boldsymbol{x})$ with $\text{supp}f = \boldsymbol{r}(\text{supp}f)$ is represented as the sum of m polynomials $f_k(\boldsymbol{x})$ $(k = 1,\ldots,m)$ with $\text{supp}f_k = \boldsymbol{r}(\text{supp}f_k) \subset \mathbb{Z}^{C_k}_+$ for some $C_k \subset \{1,\ldots,n\}$. We assume that m and the size $|C_k|$ of each C_k are both small, *e.g.*, $m \leq n$ and $|C_k| = O(1)$. Under this assumption, the construction of the cone $\mathbb{L}_\text{s}$ used for $\mathbb{K}^2$ of the sparse DNN relaxation of the form (2) is described in Section 5.1. More details on choices of $C_1,\ldots,C_m$ from the Hessian matrix of the objective function $f(\boldsymbol{x})$ are presented in Section 5.2.

5.1 Construction of a polyhedral cone $\mathbb{L}_\text{s}$ for $\mathbb{K}^2$

For each k, choose $\mathcal{A}_k$ to be a finite subset of $\mathbb{Z}^{C_k}_+$ such that $\boldsymbol{0} \in \mathcal{A}_k = \boldsymbol{r}(\mathcal{A}_k)$ and $\text{supp}f_k = \boldsymbol{r}(\text{supp}f_k) \subset \mathcal{A}_k + \mathcal{A}_k$, and take a matrix $\boldsymbol{Q}^0_k \in \mathbb{S}^{\mathcal{A}_k}$ such that $f_k(\boldsymbol{x}) = \langle \boldsymbol{Q}^0_k,\ \boldsymbol{x}^{\square\mathcal{A}_k}\rangle$. Let $\boldsymbol{\Theta} = \{(k,\boldsymbol{\alpha},\boldsymbol{\beta}) : (\boldsymbol{\alpha},\boldsymbol{\beta}) \in \mathcal{A}_k \times \mathcal{A}_k,\ k = 1,\ldots,m\}$.

For each $\boldsymbol{X} = (\boldsymbol{X}_1,\ldots,\boldsymbol{X}_m) \in \mathbb{S}^{\mathcal{A}_1} \times \cdots \times \mathbb{S}^{\mathcal{A}_m}$, we denote the $(\boldsymbol{\alpha},\boldsymbol{\beta})$th element of $\boldsymbol{X}_k$ by $X_{k\boldsymbol{\alpha}\boldsymbol{\beta}}$ $((k,\boldsymbol{\alpha},\boldsymbol{\beta}) \in \boldsymbol{\Theta})$ $(k = 1,\ldots,m)$. Then we may identify $\mathbb{S}^{\mathcal{A}_1} \times \cdots \times \mathbb{S}^{\mathcal{A}_m}$ with the $|\boldsymbol{\Theta}|$-dimensional Euclidean space $\mathbb{R}^{\boldsymbol{\Theta}}$ with the inner product $\langle \boldsymbol{W},\ \boldsymbol{X}\rangle = \sum_{(k,\boldsymbol{\alpha},\boldsymbol{\beta})\in\boldsymbol{\Theta}} W_{k\boldsymbol{\alpha}\boldsymbol{\beta}} X_{k\boldsymbol{\alpha}\boldsymbol{\beta}}$. Let $\boldsymbol{Q}^0 = (\boldsymbol{Q}^0_1,\ldots,\boldsymbol{Q}^0_m) \in \mathbb{R}^{\boldsymbol{\Theta}}$. Then, POP (1) is equivalent to

$$\min\left\{\langle \boldsymbol{Q}^0,\ \boldsymbol{X}\rangle : \boldsymbol{X} \in T\right\}, \tag{22}$$

where $T = \left\{\boldsymbol{X} = (\boldsymbol{x}^{\square\mathcal{A}_1},\ldots,\boldsymbol{x}^{\square\mathcal{A}_m}) \in \mathbb{R}^{\boldsymbol{\Theta}} : \boldsymbol{x} \in S\right\}$. Notice that neither the representation of $f(\boldsymbol{x})$ in terms of $f_1(\boldsymbol{x}),\ldots,f_m(\boldsymbol{x})$ nor the representation of each $f_k(\boldsymbol{x})$ in terms of $\boldsymbol{Q}^0_k \in \mathbb{S}^{\mathcal{A}}$ is unique.

The sparse DNN relaxations of POP (1) is derived from (22) in the same way as the dense DNN relaxations of (1) from (12) in Section 4. Using $\boldsymbol{r} : \mathbb{Z}^n_+ \to \mathbb{Z}^n_+$ defined by (10), we introduce a preorder $\succeq$ in $\boldsymbol{\Theta}$: for every pair of $(k,\boldsymbol{\alpha},\boldsymbol{\beta}) \in \boldsymbol{\Theta}$ and $(\ell,\boldsymbol{\gamma},\boldsymbol{\delta}) \in \boldsymbol{\Theta}$, $(k,\boldsymbol{\alpha},\boldsymbol{\beta}) \succeq (\ell,\boldsymbol{\gamma},\boldsymbol{\delta})$ if and only if there exists a positive number $c \geq 1$ such that $r_i(\boldsymbol{\alpha}+\boldsymbol{\beta}) = r_i(\boldsymbol{\gamma}+\boldsymbol{\delta})$ $(i \in I_\text{bin})$ and $c(\alpha_j + \beta_j) = \gamma_j + \delta_j$ $(j \in I_\text{box})$. By definition, we see that

$$\begin{aligned}(\boldsymbol{x}^{\square\mathcal{A}_k})_{\boldsymbol{\alpha}\boldsymbol{\beta}} \quad &= \quad \boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} = \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta})} \geq \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})} = \boldsymbol{x}^{\boldsymbol{\gamma}+\boldsymbol{\delta}} = (\boldsymbol{x}^{\square\mathcal{A}_\ell})_{\boldsymbol{\gamma}\boldsymbol{\delta}} \\ &\quad\ \text{if } \boldsymbol{x} \in S \ \text{ and } \boldsymbol{\Theta} \ni (k,\boldsymbol{\alpha},\boldsymbol{\beta}) \succeq (\ell,\boldsymbol{\gamma},\boldsymbol{\delta}) \in \boldsymbol{\Theta}.\end{aligned} \tag{23}$$

The preorder $\succeq$ induces a strictly preorder $\succ$ and an equivalence relation $\sim$ by (13). Let $(\ell,\boldsymbol{\alpha},\boldsymbol{\beta}) \in \boldsymbol{\Theta}$. Then $[(\ell,\boldsymbol{\alpha},\boldsymbol{\beta})] = \{(k,\boldsymbol{\gamma},\boldsymbol{\delta}) \in \boldsymbol{\Theta} : \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta}) = \boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta})\}$ (the equivalence class containing $(\ell, \boldsymbol{\alpha}, \boldsymbol{\beta})$). As a result, each equivalence class is characterized by $\boldsymbol{\sigma} \in \bigcup_{k=1}^{m} \boldsymbol{r}(\mathcal{A}_k + \mathcal{A}_k)$, and the family of equivalence classes is denoted by $\{E_{\boldsymbol{\sigma}} \ (\boldsymbol{\sigma} \in \bigcup_{k=1}^{m} \boldsymbol{r}(\mathcal{A}_k + \mathcal{A}_k))\}$ where $E_{\boldsymbol{\sigma}} = \{(k, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in \boldsymbol{\Theta} : \boldsymbol{r}(\boldsymbol{\alpha} + \boldsymbol{\beta}) = \boldsymbol{\sigma}\}$. By definition and (23),

$$(\boldsymbol{x}^{\square \mathcal{A}_k})_{\boldsymbol{\alpha}\boldsymbol{\beta}} = (\boldsymbol{x}^{\square \mathcal{A}_\ell})_{\boldsymbol{\gamma}\boldsymbol{\delta}} \text{ if } \boldsymbol{x} \in S \text{ and } (k, \boldsymbol{\alpha}, \boldsymbol{\beta}) \sim (\ell, \boldsymbol{\gamma}, \boldsymbol{\delta}). \tag{24}$$

We can extend Lemma 4.1 established for the dense DNN relaxation to the sparse DNN relaxation. In fact, the extended lemma ensures that the family of maximal chains of equivalence classes, denoted by $\{\Gamma_1, \ldots, \Gamma_r\}$, partitions the family of equivalence classes $\{E_{\boldsymbol{\sigma}} \ (\boldsymbol{\sigma} \in \bigcup_{k=1}^{m} \ \boldsymbol{r}(\mathcal{A}_k + \mathcal{A}_k))\}$. Now define

$$\begin{aligned}
\boldsymbol{H}_1^0 &= \text{the } |\mathcal{A}_1| \times |\mathcal{A}_1| \text{ matrix in } \mathbb{S}^{\mathcal{A}_1} \text{ with 1 at the } (\boldsymbol{0}, \boldsymbol{0})\text{th element} \\
&\quad\ \text{and 0 elsewhere,} \\
\boldsymbol{H}^0 &= (\boldsymbol{H}_1^0, \boldsymbol{O}, \ldots, \boldsymbol{O}) \in \mathbb{R}^{\boldsymbol{\Theta}} = \mathbb{S}^{\mathcal{A}_1} \times \cdots \times \mathbb{S}^{\mathcal{A}_m}, \\
\mathbb{L}_{\rm s} &= \left\{ \boldsymbol{X} \in \mathbb{R}^{\boldsymbol{\Theta}} : \begin{array}{ll} X_{k\boldsymbol{\alpha}\boldsymbol{\beta}} = \xi_{\boldsymbol{\sigma}} \in \mathbb{R}_+ & \text{if } (k, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in E_{\boldsymbol{\sigma}} \in \Gamma_p \\ \xi_{\boldsymbol{\sigma}} \geq \xi_{\boldsymbol{\tau}} & \text{if } \Gamma_p \ni E_{\boldsymbol{\sigma}} \succ E_{\boldsymbol{\tau}} \in \Gamma_p \\ & (p = 1, \ldots, r) \end{array} \right\}.
\end{aligned}$$

If $\boldsymbol{x}$ is a feasible solution, then $\boldsymbol{X} = (\boldsymbol{x}^{\square \mathcal{A}_1}, \ldots, \boldsymbol{x}^{\square \mathcal{A}_m}) \in \mathbb{R}^{\boldsymbol{\Theta}}$ satisfies $\boldsymbol{X} \in \mathbb{S}_+^{\mathcal{A}_1} \times \cdots \times \mathbb{S}_+^{\mathcal{A}_m}$, $\langle \boldsymbol{H}^0, \ \boldsymbol{X} \rangle = 1$, $\boldsymbol{X} \in \mathbb{L}_{\rm s}$ and $f(\boldsymbol{x}) = \langle \boldsymbol{Q}^0, \ \boldsymbol{X} \rangle$. Thus, we can see that it is a feasible solution of COP (2) with $\mathbb{K}^1 = \mathbb{S}_+^{\mathcal{A}_1} \times \cdots \times \mathbb{S}_+^{\mathcal{A}_m}$ and $\mathbb{K}^2 = \mathbb{L}_{\rm s}$, and that it also attains the same objective value $f(\boldsymbol{x})$. This implies that COP (2) serves as a sparse DNN relaxation of POP (1).

Lemma 4.2 can be extended to the sparse DNN relaxation of POP (1), so that Algorithm 2.1 can be used for solving COP (2) with $\mathbb{K}^1 = \mathbb{S}_+^{\mathcal{A}_1} \times \cdots \times \mathbb{S}_+^{\mathcal{A}_m}$ and $\mathbb{K}^2 = \mathbb{L}_{\rm s}$. Lemma 3.2 and Algorithm 3.3 can also be applied to the metric projection $\Pi_{\mathbb{L}_{\rm s}}$ from $\mathbb{R}^{\boldsymbol{\Theta}}$ onto $\mathbb{L}_{\rm s}$.

5.2 The chordal graph sparsity of the Hessian matrix of the objective polynomial of POP (1)

In this section, we describe how the objective polynomial $f(\boldsymbol{x})$ of POP (1) can be represented as the sum of m polynomials $f_k(\boldsymbol{x})$ $(k = 1, \ldots, m)$ with $\mathrm{supp} f_k \subset \mathbb{Z}^{C_k}$ for some $C_k \subset \{1, \ldots, n\}$. We assume that $f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathrm{supp} f} c(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}}$ with $\mathrm{supp} f = \boldsymbol{r}(\mathrm{supp} f) \subset \mathbb{Z}_+^n$.

Let $\boldsymbol{H} f(\boldsymbol{x})$ denotes the $n \times n$ Hessian matrix of $f(\boldsymbol{x})$. Each element of $\boldsymbol{H} f(\boldsymbol{x})$ is a polynomial. Throughout this section, we assume that many elements of $\boldsymbol{H} f(\boldsymbol{x})$ are identically zero. To find a structured sparsity of $\boldsymbol{H} f(\boldsymbol{x})$ which can be exploited, we introduce an undirected graph $G(N, \mathcal{E})$ with a node set $N = \{1, \ldots, n\}$ and an edge set $\mathcal{E} \subset N \times N$ consisting of all (i, j)s with $i \neq j$ such that the (i, j)th element of $\boldsymbol{H} f(\boldsymbol{x})$ is not identically zero. We identify each edge $(i, j) \in \mathcal{E}$ with (j, i). Let $G(N, \overline{\mathcal{E}})$ be a chordal extension of $G(N, \mathcal{E})$, a chordal graph such that $\mathcal{E} \subset \overline{\mathcal{E}}$. Here a graph is called chordal when every cycle consisting of more than 3 edges has a chord. Let $C_1, \ldots, C_m \subset N$ denote the maximal cliques of $G(N, \overline{\mathcal{E}})$. See [6,8] for chordal graphs and

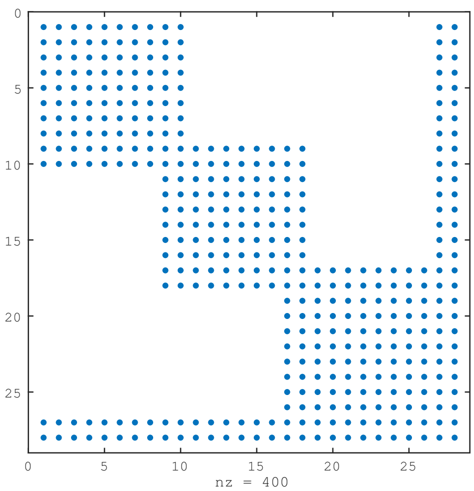

Figure 1: The arrow-type nonzero pattern of the Hessian matrix $\boldsymbol{H}f(\boldsymbol{x})$ of $f(\boldsymbol{x})$. $m = 3,\ a = 2,\ b = 10,\ c = 2$.

their fundamental properties. An example is given next to illustrate the construction of a family of maximal cliques corresponding to an arrow-type nonzero pattern as shown in Figure 1.

Example 5.1. (An arrow-type nonzero pattern) For nonnegative integers m, a, b and c with $m \geq 1$, $b \geq 1$ and $a \leq b/2$, let $n = b + (b-a)(m-1) + c,\ N = \{1, \ldots, n\}$,

$$\begin{aligned} B &= \{1, \ldots, b\},\ F = \{1, \ldots, c\} + \{n - c\}, \\ B_k &= B + \{(b-a)(k-1)\}\ (k = 1, \ldots, m),\ C_k = B_k \cup F\ (k = 1, \ldots, m), \end{aligned}$$

where $+$ stands for the Minkowski sum of two sets. Every C_k consists of two disjoint subsets B_k and F of N. We see that $|C_k| = b + c$, $|B_k \cap B_{k+1}| = a\ (k = 1, \ldots, m-1)$ and $B_j \cap B_k = \emptyset$ if $|j - k| \geq 2$. We may regard $C_1, \ldots, C_m \subset N$ as the maximal cliques which characterize the nonzero pattern of the Hessian matrix $\boldsymbol{H}f(\boldsymbol{x})$ of a sparse polynomial $f(\boldsymbol{x})$. Figure 1 shows the nonzero pattern of the Hessian matrix $\boldsymbol{H}f(\boldsymbol{x})$ of such a polynomial with $m = 3$, $b = 10$, $a = 2$ and $c = 2$. We observe that each B_k induces a $b \times b$ nonzero block along the diagonal and F leads to c nonzero rows and columns at the bottom and the right end of $\boldsymbol{H}f(\boldsymbol{x})$. Furthermore, the graph $G(N, \mathcal{E})$ with the node set $N = \{1, \ldots, n\}$ and the edge set $\mathcal{E} = \{(i, j) \in N \times N : (i, j) \in C_k \times C_k \text{ and } i < j \text{ for some } k\}$ forms a chordal graph having the maximal cliques $C_1, \ldots, C_m$. We see that this example with the described $\boldsymbol{H}f(\boldsymbol{x})$ of a sparse polynomial $f(\boldsymbol{x})$ illustrates the previous discussion.

To represent $f(\boldsymbol{x})$ as the sum of m polynomials $f_k(\boldsymbol{x})$ $(k = 1, \ldots, m)$ with $\mathrm{supp} f_k = \boldsymbol{r}(\mathrm{supp} f_k) \subset \mathbb{Z}^{C_k}$ in an iterative way, we initially set $f_k(\boldsymbol{x}) \equiv 0$ $(k = 1, \ldots, m)$ and $\mathcal{F} = \mathrm{supp} f = \boldsymbol{r}(\mathrm{supp} f)$. Choose $\boldsymbol{\alpha} \in \mathcal{F}$. If $C = \{i \in N : \alpha_i \geq 1\} = \{i\}$ for some $i \in N$ or equivalently $\boldsymbol{\alpha} = c\boldsymbol{e}^i$ for some nonzero $c \in \mathbb{Z}_+$, then C is contained in some maximal clique C_k since $\bigcup_{k=1}^m C_k = N$. Thus, $\boldsymbol{\alpha} \in \mathbb{Z}_+^{C_k}$. Otherwise C consists of more than one element, say, $1, \ldots, \ell$ or $\boldsymbol{\alpha} = \sum_{i=1}^{\ell} c_i \boldsymbol{e}^i$ for some nonzero $c_i \in \mathbb{Z}_+$. In this case, for every

$(i,j) \in C \times C$, the (i,j)th element of the Hessian matrix $\boldsymbol{H}f(\boldsymbol{x})$ of $f(\boldsymbol{x})$ is not zero, so that C forms a clique of $G(N,\mathcal{E})$. Hence, C is contained in some maximal clique C_k and $\boldsymbol{\alpha} \in \mathbb{Z}^{C_k}_+$. In both cases, update $f_k(\boldsymbol{x}) \leftarrow f_k(\boldsymbol{x}) + c(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$ and $\mathcal{F} \leftarrow \mathcal{F}\backslash\{\boldsymbol{\alpha}\}$. Repeating this procedure until $\mathcal{F}$ becomes empty, we obtain the desired m polynomials to represent $f(\boldsymbol{x})$.

5.3 Choosing $\mathcal{A}_k$ $(k = 1,\ldots,m)$ for sparse DNN relaxations of QOPs and POPs with binary and box constraints

Suppose that $\deg f = 2$ in POP (1) to deal with a QOP with binary and box constraints. If we represent the quadratic objective function $f(\boldsymbol{x}) = \langle \boldsymbol{Q}^0,\ \boldsymbol{x}\boldsymbol{x}^T\rangle + \boldsymbol{a}\boldsymbol{x}^T$, then the Hessian matrix $\boldsymbol{H}f(\boldsymbol{x})$ coincides with $\boldsymbol{Q}^0$. Therefore, the nonzero pattern of $\boldsymbol{Q}^0$ determines the graph $G(N,\mathcal{E})$, which induces a chordal extension of $G(N,\overline{\mathcal{E}})$ and its maximal cliques $C_1,\ldots,C_m$. We choose $\{\boldsymbol{0},\ \boldsymbol{e}^i\ (i \in C_k)\}$ for $\mathcal{A}_k$ $(k = 1,\ldots,m)$ to construct the cone $\mathbb{L}_{\mathrm{s}}$ and the sparse DNN relaxation of the form (2).

We now extend the hierarchy of dense DNN relaxations of POP (1) with binary and box constraints presented in Section 4.4, to the sparse case. Let ω_0 be the smallest positive integer which is not less than $(\deg f)/2$. First, construct nonempty subsets C_k of N and polynomial functions $f_k(\boldsymbol{x})$ with $\mathrm{supp} f_k = \boldsymbol{r}(\mathrm{supp} f_k \subset \mathbb{Z}^{C_k})$ $(k = 1,\ldots,m)$ from the Hessian matrix $\boldsymbol{H}f(\boldsymbol{x})$ of $f(\boldsymbol{x})$ as in Section 5.2. For a positive integer $\omega \geq \omega_0$, let $\mathcal{A}^\omega_k = \left\{\boldsymbol{\alpha} \in \mathbb{Z}^{C_k}_+ : \sum_{i\in N}\alpha_i \leq \omega\right\}$ and $\mathcal{A}_k = \boldsymbol{r}(\mathcal{A}^\omega_k)$ $(k = 1,\ldots,m)$. Then, $\mathrm{supp} f_k \subset \mathcal{A}_k + \mathcal{A}_k$ holds $(k = 1,\ldots,m)$. Thus, we can find a matrix $\boldsymbol{Q}^0_k \in \mathbb{S}^{\mathcal{A}_k}$ such that $f_k(\boldsymbol{x}) = \langle \boldsymbol{Q}^0_k,\ \boldsymbol{x}^{\mathcal{A}_k}\rangle$ $(k = 1,\ldots,m)$ to lift POP (1) to the problem (22) in $\mathbb{R}^{\boldsymbol{\Theta}}$ with $\boldsymbol{\Theta} = \mathbb{S}^{\mathcal{A}_1} \times \cdots \mathbb{S}^{\mathcal{A}_k}$. Note that $\mathcal{A}_k = \boldsymbol{r}(\mathcal{A}^\omega_k)$ $(k = 1,\ldots,m)$ with I_{bin} and I_{box} determine the preorder $\succeq$, the equivalence relation $\sim$, the family of maximal chains of equivalence classes, $\mathbb{L}_{\mathrm{s}}$, and eventually the sparse DNN relaxation of the form (2) with the relaxation order ω.

6 A full preorder $\succeq_{\mathrm{f}}$ in $\mathcal{A} \times \mathcal{A}$ and a Lagrangian-DNN relaxation

Let $\mathcal{A}$ be a nonempty subset of $\mathbb{Z}^n_+$ satisfying (11), and let $\succeq$ denote the preorder introduced in Section 4. A binary relation $\succeq_{\mathrm{f}}$ in $\boldsymbol{\Theta} := \mathcal{A} \times \mathcal{A}$ defined by $(\boldsymbol{\alpha},\boldsymbol{\beta})\succeq_{\mathrm{f}}(\boldsymbol{\gamma},\boldsymbol{\delta}) \Leftrightarrow \boldsymbol{r}(\boldsymbol{\alpha}+\boldsymbol{\beta}) \leq \boldsymbol{r}(\boldsymbol{\gamma}+\boldsymbol{\delta})$ also serves a preorder in $\boldsymbol{\Theta}$ satisfying (20). We call $\succeq_{\mathrm{f}}$ the *full preorder* in $\mathcal{A} \times \mathcal{A}$.

Lemma 6.1.

(a) The preorders $\succeq$ and $\succeq_f$ induce a common equivalence relation $\sim$.

(b) For every pair of $(\boldsymbol{\alpha},\boldsymbol{\beta})$ and $(\boldsymbol{\gamma},\boldsymbol{\delta})$ in $\boldsymbol{\Theta}$, $(\boldsymbol{\alpha},\boldsymbol{\beta})\succeq_f(\boldsymbol{\gamma},\boldsymbol{\delta})$ if and only if $\boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \geq \boldsymbol{x}^{\boldsymbol{\gamma}+\boldsymbol{\delta}}$ for every $\boldsymbol{x} \in S$.

Proof. Assertion (a) follows directly from the definitions of $\succeq$ and $\succeq_{\mathrm{f}}$. For assertion (b), it suffices to show that $\boldsymbol{r}(\boldsymbol{\sigma}) \leq \boldsymbol{r}(\boldsymbol{\tau})$ if and only if $\boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\sigma})} \geq \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\tau})}$ for every $\boldsymbol{x} \in S$. The "only if" part is obvious by definition. To prove the "if part", we assume that $\boldsymbol{r}(\boldsymbol{\sigma}) \not\leq \boldsymbol{r}(\boldsymbol{\tau})$ holds, and we will show that $\boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\sigma})} < \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\tau})}$ for some $\boldsymbol{x} \in S$. By the assumption, there exists an i such that $r_i(\boldsymbol{\sigma}) > r_i(\boldsymbol{\tau})$. Fix $x_j = 1$ for every $j \neq i$. Then $x_j^{r_i(\boldsymbol{\sigma})} = x_j^{r_i(\boldsymbol{\tau})} = 1$ for every $j \neq i$. If $i \in I_{\rm bin}$, then $r_i(\boldsymbol{\sigma}) > r_i(\boldsymbol{\tau})$ implies that $r_i(\boldsymbol{\sigma}) = 1 > r_i(\boldsymbol{\tau}) = 0$. In this case, taking $x_i = 0$, we obtain that $\boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\sigma})} = x_i^{r_i(\boldsymbol{\sigma})} = 0 < 1 = x_i^{r_i(\boldsymbol{\tau})} = \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\tau})}$. Otherwise, $i \in I_{\rm box}$ and $r_i(\boldsymbol{\sigma}) = \sigma_i > r_i(\boldsymbol{\tau}) = \tau_i$. In this case, taking $x_i = 0.5$, we obtain that $\boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\sigma})} = x_i^{\boldsymbol{\sigma}_i} < x_i^{\boldsymbol{\tau}_i} = \boldsymbol{x}^{\boldsymbol{r}(\boldsymbol{\tau})}$. □

Now, let us consider to use the preorder $\succeq_{\rm f}$ instead of $\succeq$ to derive a DNN relaxation of POP (1), which will be called as the *full DNN relaxation*. By assertion (b) of Lemma 6.1, the preorder $\succeq_{\rm f}$ may be regarded as the strongest preorder to generate inequalities between two monomials $\boldsymbol{x}^{\boldsymbol{\alpha}+\boldsymbol{\beta}}$ and $\boldsymbol{x}^{\boldsymbol{\gamma}+\boldsymbol{\delta}}$ for every $\boldsymbol{x} \in S$. In particular, the preorder $\succeq_{\rm f}$ is stronger than $\succeq$ in the sense that if $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \succeq (\boldsymbol{\gamma}, \boldsymbol{\delta})$ then $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \succeq_{\rm f} (\boldsymbol{\gamma}, \boldsymbol{\delta})$. This ensures that every chain with respect to $\succeq$ is a chain with respect to $\succeq_{\rm f}$. Therefore, the resulting cone, which is denoted by $\mathbb{L}_{\rm f}$, is expected to be smaller than the one induced by $\succeq$, and the resulting DNN relaxation of POP (1) is expected to be stronger than the DNN relaxation given in Section 4. However, the family of maximal chains with respect to $\succeq_{\rm f}$ are not necessarily disjoint with each other. This makes the computation of the metric projection from $\mathbb{S}^{\mathcal{A}}$ onto $\mathbb{L}_{\rm f}$ highly challenging and expensive. To apply Lemma 3.2 and Algorithm 3.3, we need to choose a family of chains (with respect to $\succeq_{\rm f}$) which partitions the family of equivalence classes $\{E_{\boldsymbol{\sigma}} \ (\boldsymbol{\sigma} \in \boldsymbol{r}(\mathcal{A}+\mathcal{A})\}$. The family of maximal chains with respect to $\succeq$ may be regarded as such a family.

In this section, we focus on QOP (6) with binary and box constraints to describe its full DNN relaxation using $\succeq_{\rm f}$, and show its close relations with the Lagrangian-DNN relaxation proposed in [2] for a class of QOPs with linear, binary and complementarity constraints.

6.1 A full DNN relaxation of a QOP with binary and box constraints

First we observe that the preorder $\succeq_{\rm f}$ induces the identities $x_i = x_i^2$ $(i \in I_{\rm bin})$ and inequalities $x_i \geq x_i^2$ $(i \in I_{\rm box})$, $x_i \geq x_i x_j$ $(1 \leq i < j \leq n)$ which hold for every $\boldsymbol{x} \in S$. Hence, the finest family of chains of equivalence classes mentioned in Remark 3.1 leads to the *full DNN relaxation*

$$\psi_1 = \min \left\{ \left\langle \boldsymbol{Q}^0, \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \right\rangle : \ X_{00} = 1, \begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n}_+ \cap \mathbb{L}_{\rm f} \right\}, \tag{25}$$

where

$$\mathbb{L}_{\rm f} = \left\{ \begin{pmatrix} V_{00} & \boldsymbol{v}^T \\ \boldsymbol{v} & \boldsymbol{V} \end{pmatrix} \in \mathbb{N}^{1+n} : \begin{array}{l} v_i = V_{ii} \ (i \in I_{\rm bin}), \\ v_i \geq V_{ii} \ (i \in I_{\rm box}), \\ v_i \geq V_{ij} \ (1 \leq i < j \leq n) \end{array} \right\}.$$

6.2 Strengthening the full DNN relaxation (25) with slack variables

By introducing slack variables $u_i \geq 0$ $(i = 1, \ldots, n)$ into the QOP (6), we can rewrite the problem as:

$$\min\{\langle \boldsymbol{Q}^0,\ (1, \boldsymbol{x})(1, \boldsymbol{x})^T\rangle : \boldsymbol{x} \in S,\ \boldsymbol{x} + \boldsymbol{u} = \boldsymbol{e},\ \boldsymbol{u} \in S\},$$

where $\boldsymbol{e}$ denotes the n-dimensional row vector of all ones. Obviously this QOP is equivalent to (6). By applying the discussion in Section 6.1 twice in the $\boldsymbol{x}$-space and $\boldsymbol{u}$-space, we have a DNN relaxation of this problem as

$$\psi_2 = \min\left\{ \left\langle \boldsymbol{Q}^0, \begin{pmatrix} X_{00} & \boldsymbol{x} \\ \boldsymbol{x}^T & \boldsymbol{X} \end{pmatrix} \right\rangle : \begin{array}{l} X_{00} = U_{00} = 1,\ \boldsymbol{x} + \boldsymbol{u} = \boldsymbol{e}, \\ \begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n}_+ \cap \mathbb{L}_{\mathrm{f}}, \\ \begin{pmatrix} U_{00} & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix} \in \mathbb{S}^{1+n}_+ \cap \mathbb{L}_{\mathrm{f}} \end{array} \right\}. \tag{26}$$

We call (26) the *twin DNN relaxation* of QOP (6). It is easy to see that $\psi_1 \leq \psi_2$.

In the next subsection, another DNN relaxation for the QOP (6) will be derived with cones $\mathbb{K}^1 = \mathbb{S}^{1+2n}_+$ and $\mathbb{K}^2 = \mathbb{L}_{\mathrm{d}} \in \mathbb{S}^{1+2n}$, which is at least as strong as (26) and can be approximately solved by the accelerated BP algorithm (Algorithm 2.1) combined with Algorithm 3.3 for the computation the metric computation of $\mathbb{S}^{1+2n}$ onto $\mathbb{L}_{\mathrm{d}}$.

6.3 Representing binary constraints with complementarity constraints

Since each binary variable x_i is characterized by $x_i + u_i = 1$, $x_i \geq 0$, $u_i \geq 0$ and $x_i u_i = 0$ with a slack box variable u_i, QOP (6) is equivalent to

$$\min\left\{ \langle \boldsymbol{Q}^0,\ (1, \boldsymbol{x})(1, \boldsymbol{x})^T\rangle : \begin{array}{l} x_0 = 1,\ \boldsymbol{x} \in [0,1]^n,\ \boldsymbol{u} \in [0,1]^n, \\ \left((-1,\ \boldsymbol{e}_j,\ \boldsymbol{e}_j)(1, \boldsymbol{x}, \boldsymbol{u})^T\right)^2 = 0\ (j = 1, \ldots, n), \\ x_i u_i = 0\ (i \in I_{\mathrm{bin}}) \end{array} \right\} \tag{27}$$

which becomes a box constrained QOP with equality constraints. Here $\boldsymbol{e}^j$ denotes the jth coordinate row vector in $\mathbb{R}^n$. Note that the equality constraints $\left((-1,\ \boldsymbol{e}_j,\ \boldsymbol{e}_j)(1, \boldsymbol{x}, \boldsymbol{u})^T\right)^2 = 0$ $(j = 1, \ldots, n)$ are equivalent to a linear equality $\boldsymbol{x} + \boldsymbol{u} = \boldsymbol{e}$. Define

$$\begin{aligned} \boldsymbol{Q}^1_j &= (-1,\ \boldsymbol{e}_j,\ \boldsymbol{e}_j)^T(-1,\ \boldsymbol{e}_j,\ \boldsymbol{e}_j) \in \mathbb{S}^{1+2n}_+\ (j = 1, \ldots, n), \\ \boldsymbol{Q}^2_i &= \begin{pmatrix} 0 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{O} & \boldsymbol{E}_{ii} \\ \boldsymbol{0}^T & \boldsymbol{E}_{ii} & \boldsymbol{O} \end{pmatrix} \in \mathbb{N}^{1+2n}\ (i \in I_{\mathrm{bin}}), \\ \boldsymbol{E}_{ii} &= \text{the } n \times n \text{ matrix with 1 at the } (i,i)\text{th element and 0 elsewhere } (i \in I_{\mathrm{bin}}). \end{aligned}$$

Then we can rewrite problem (27) as

$$\min\left\{ \langle \boldsymbol{Q}^0,\ (1, \boldsymbol{x})(1, \boldsymbol{x})^T\rangle : \begin{array}{l} x_0 = 1, \boldsymbol{x} \in [0,1]^n,\ \boldsymbol{u} \in [0,1]^n, \\ \langle \boldsymbol{Q}^1_j,\ (1, \boldsymbol{x}, \boldsymbol{u})(1, \boldsymbol{x}, \boldsymbol{u})^T\rangle = 0\ (j = 1, \ldots, n), \\ \langle \boldsymbol{Q}^2_i,\ (1, \boldsymbol{x}, \boldsymbol{u})(1, \boldsymbol{x}, \boldsymbol{u})^T\rangle = 0\ (i \in I_{\mathrm{bin}}). \end{array} \right\}$$

Replacing $(1,\boldsymbol{x},\boldsymbol{u})(1,\boldsymbol{x},\boldsymbol{u})^T$ by $\boldsymbol{Z} = \begin{pmatrix} X_{00} & \boldsymbol{x} & \boldsymbol{u} \\ \boldsymbol{x}^T & \boldsymbol{X} & \boldsymbol{W} \\ \boldsymbol{u}^T & \boldsymbol{W}^T & \boldsymbol{U} \end{pmatrix}$ and taking into account of $\boldsymbol{x} \in [0,1]^n,\ \boldsymbol{u} \in [0,1]^n$, we obtain a DNN relaxation of QOP (27):

$$\psi_3 \ = \ \left\{ \left\langle \boldsymbol{Q}^0, \begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \right\rangle : \begin{array}{l} \boldsymbol{Z} \in \mathbb{S}^{1+2n}_+ \cap \mathbb{L}_{\rm d},\ \langle \boldsymbol{H}^0,\ \boldsymbol{Z}\rangle = 1, \\ \langle \boldsymbol{Q}^1_j,\ \boldsymbol{Z}\rangle = 0\ (j=1,\ldots,n), \\ \langle \boldsymbol{Q}^2_i,\ \boldsymbol{Z}\rangle = 0\ (i \in I_{\rm bin}) \end{array} \right\}, \tag{28}$$

where $\boldsymbol{H}^0 =$ the $(1+2n)\times(1+2n)$ matrix with all elements equal to 0 except 1 at the $(0,0)$th element, and

$$\mathbb{L}_{\rm d} \ = \ \left\{ \boldsymbol{Z} \in \mathbb{N}^{1+2n} : x_i \geq X_{ii},\ u_i \geq U_{ii}\ (i=1,\ldots,n) \right\}.$$

Since $\mathbb{L}_{\rm d}$ is constructed using the preorder $\succeq$ in the $(\boldsymbol{x},\boldsymbol{u})$ space, $\mathbb{R}^{2n}$ (see Section 4.3), Algorithm 3.3 can be applied for computation of the metric projection $\Pi_{\mathbb{L}_{\rm d}}$ from $\mathbb{S}^{1+2n}$ onto $\mathbb{L}_{\rm d}$.

The next proposition shows that the DNN relaxation (28) is at least as strong as the twin DNN relaxation (26).

Proposition 6.2. *The DNN relaxation (28) is at least as strong as the twin DNN relaxation (26) in the sense that $\psi_3 \geq \psi_2$.*

Proof. To see this, suppose that $\boldsymbol{Z}$ is a feasible solution of (28). Obviously,

$$X_{00} = 1,\ \begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \in \mathbb{S}^{1+n}_+ \cap \mathbb{N}^{1+n},\ \begin{pmatrix} X_{00} & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix} \in \mathbb{S}^{1+n}_+ \cap \mathbb{N}^{1+n}.$$

Since $\boldsymbol{Z} \in \mathbb{S}^{1+2n}_+$, we see from the identity $\langle \boldsymbol{Q}^1_j,\ \boldsymbol{Z}\rangle = 0\ (j=1,\ldots,n)$ and the definition of $\boldsymbol{Q}^1_j$ that $\boldsymbol{Z}(-1,\ \boldsymbol{e}_j,\ \boldsymbol{e}_j)^T = \boldsymbol{0}\ (j=1,\ldots,n)$, or equivalently,

$$\begin{array}{l} X_{00} - x_j - u_j = 0\ (j=1,\ldots,n), \\ x_i - X_{ij} - W_{ij} = 0\ (1 \leq j \leq n),\ u_i - U_{ij} - W_{ji} = 0\ (1 \leq j \leq n). \end{array}$$

We also know from $\langle \boldsymbol{Q}^2_i,\ \boldsymbol{Z}\rangle = 0\ (i \in I_{\rm bin})$ and $\boldsymbol{W} \geq \boldsymbol{O}$ that $W_{ii} = 0\ (i \in I_{\rm bin})$, $W_{ii} \geq 0\ (i \in I_{\rm box})$ and $W_{ij} \geq 0\ (1 \leq j \leq n)$. Thus, the previous relations imply

$$\begin{array}{l} \boldsymbol{x} + \boldsymbol{u} = \boldsymbol{e}, \\ x_i = X_{ii}\ (i \in I_{\rm bin}),\ x_i \geq X_{ii}\ (i \in I_{\rm box}),\ x_i \geq X_{ij}\ (1 \leq i < j \leq n), \\ u_i = U_{ii}\ (i \in I_{\rm bin}),\ u_i \geq U_{ii}\ (i \in I_{\rm box}),\ u_i \geq U_{ij}\ (1 \leq i < j \leq n), \\ x_i - W_{ij} \geq 0\ (1 \leq i \leq j \leq n),\ u_i - W_{ji} \geq 0\ (1 \leq i \leq j \leq n). \end{array}$$

As a result, both $\begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix}$ and $\begin{pmatrix} X_{00} & \boldsymbol{u}^T \\ \boldsymbol{u} & \boldsymbol{U} \end{pmatrix}$ lie in $\mathbb{L}_{\rm f}$, and they provide a feasible solution of (26). Consequently, $\psi_3 \geq \psi_2$ follows. The inequalities in the last line above are irrelevant for the conclusion, but it shows that the DNN relaxation (28) also incorporates the inequalities induced from $x_i - x_i u_j \geq 0\ (1 \leq i \leq j \leq n)$ and $u_i - x_j u_i \geq 0\ (1 \leq i \leq j \leq n)$. □

6.4 A Lagrangian-DNN relaxation

Since $\langle \boldsymbol{Q}_j^1, \boldsymbol{Z}\rangle \geq 0$ for every $\boldsymbol{Z} \in \mathbb{S}_+^{1+n}$ $(j = 1, \ldots, n)$ and $\langle \boldsymbol{Q}_i^2, \boldsymbol{Z}\rangle \geq 0$ for every $\boldsymbol{Z} \in \mathbb{N}_+^{1+n}$ $(i \in I_{\text{bin}})$, we can rewrite (28) as

$$\psi_3 = \min \left\{ \left\langle \boldsymbol{Q}^0, \begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \right\rangle : \begin{array}{l} \boldsymbol{Z} \in \mathbb{S}_+^{1+2n} \cap \mathbb{L}_{\text{d}}, \ \langle \boldsymbol{H}^0, \boldsymbol{Z}\rangle = 1, \\ \langle \boldsymbol{H}^1, \boldsymbol{Z}\rangle = 0, \end{array} \right\}, \tag{29}$$

where $\boldsymbol{H}^1 = \sum_{j=1}^n \boldsymbol{Q}_j^1 + \sum_{i \in I_{\text{bin}}} \boldsymbol{Q}_i^2$. We notice that (29) is not in the form of the COP (2) yet. By further applying the Lagrangian relaxation to (29) with a given parameter $\lambda > 0$, we obtain the Lagrangian-DNN relaxation of (27)

$$\psi_4(\lambda) = \min \left\{ \left\langle \boldsymbol{Q}^0, \begin{pmatrix} X_{00} & \boldsymbol{x}^T \\ \boldsymbol{x} & \boldsymbol{X} \end{pmatrix} \right\rangle + \lambda \langle \boldsymbol{H}^1, \boldsymbol{Z}\rangle : \begin{array}{l} \boldsymbol{Z} \in \mathbb{S}_+^{1+2n} \cap \mathbb{L}_{\text{d}}, \\ \langle \boldsymbol{H}^0, \boldsymbol{Z}\rangle = 1, \end{array} \right\}, \tag{30}$$

which is in the same form as COP (2). Since the term $\langle \boldsymbol{H}^1, \boldsymbol{Z}\rangle$ is nonnegative for every $\boldsymbol{Z} \in \mathbb{S}_+^{1+2n} \cap \mathbb{L}_{\text{d}}$, it serves as a penalty term such that if $\boldsymbol{Z} \in \mathbb{S}_+^{1+2n} \cap \mathbb{L}_{\text{d}}$ is not feasible for (29), then the term diverges to ∞ as $\lambda \uparrow \infty$. Using this fact, it is easy to prove that the optimal value $\psi_4(\lambda)$ of (30) converges to the optimal value ψ_3 of (29) as $\lambda \uparrow \infty$.

Applying the Lagrangian-DNN relaxation (30) to QOP (27) with box constraints serves as a relaxation of QOP (6) with binary and box constraints. As a result, we can say that the Lagrangian-DNN relaxation of (27) provides another way to strengthen the standard relaxation of QOP (6) mentioned in Sections 2.3 and 4.3, in addition to applying the hierarchy of DNN relaxations to (6). This Lagrangian-DNN relaxation was originally proposed in [2] for the CPP reformulation of a class of QOPs with linear, binary and complementarity constraints; see also [4, 16].

In Section 7.3, we compare the standard DNN relaxation (given in Section 2.3 with $\mathbb{K}^1 = \mathbb{S}_+^{1+n}$ and $\mathbb{K}^2 = \mathbb{L}$ defined in (9)) applied to the QOP (6) and the Lagraingian-DNN relaxation applied to (27) through numerical results, which shows that the Lagrangian-DNN relaxation is more effective in obtaining a tight lower bound for the optimal value of QOP (6), and that the standard DNN relaxation requires less computational time.

7 Numerical results

We tested Algorithm 2.1 (the accelerated BP algorithm) on binary and box POPs, the maximum complete satisfiability problem [14] and binary QOPs. The purpose of the numerical experiments is to demonstrate the performance of Algorithm 2.1 in comparison to some existing methods, SeDuMi [25], SDPT3 [26] and SDPNAL+ [30]. SeDuMi and SDPT3 are popular Matlab software packages based on the primal-dual interior-point method for solving general medium scale SDPs. SDPNAL+ is a Matlab implementation of the majorized semismooth Newton-CG augmented Lagrangian method developed for large scale SDPs with bounded variables. We applied SparsePOP [28] to the POP (1) to generate an SDP relaxation problem and solved it by SeDuMi, SDPT3 and SDPNAL+ to compute a lower bound of the optimal value of POP (1). SparsePOP is a MATLAB implementation of a sparse version [27] of the hierarchy of SDP relaxations [21] for POPs.

We first explain how polynomial objective functions for binary and box POPs were generated. Any polynomial objective function $f(\boldsymbol{x})$ of degree d can be written as $f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{F}} c(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}}$. For $\mathcal{F}$, we consider two cases: the fully dense case and the sparse case with the Hessian matrix of f having an arrow-type nonzero sparsity pattern described in Example 5.1. For the fully dense case, we take $\mathcal{F} = \{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \sum_{i=1}^n \alpha_i \leq d\}$. For the arrow-type sparse case, the following $\mathcal{F}$ is considered:

$$\mathcal{F}_k = \{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \sum_{i=1}^n \alpha_i \leq d,\ \alpha_i = 0 \text{ if } i \notin C_k\}(k = 1,\dots,m),\ \mathcal{F} = \bigcup_{i=1}^m \mathcal{F}_k,$$

where C_k is determined by nonnegative integers m, a, b and c with $m \geq 1$, $b \geq 1$ and $a \leq b/2$ as in Example 5.1. We fixed $a = c = 2$, $b = 10$ for degree 3 and 4 polynomial objective functions, and $a = c = 2$, $b = 8$ for degree 5 and 6 polynomial objective functions in the numerical experiments on sparse binary and box constrained POPs. See also Figure 1 where $a = c = 2$, $b = 10$ and $m = 3$ are used. The coefficients $c(\boldsymbol{\alpha})$ ($\boldsymbol{\alpha} \in \mathcal{F}$) were chosen randomly from the interval $(-1, 1)$. All POP test instances used in this paper can be downloaded from [17].

For the hierarchy of SDP relaxations of POP (1), we need to explicitly represent the constraints of POP (1) in terms of equalities and inequalities. For the binary case $S = \{0, 1\}^n$, we rewrite it as

$$\min f(\boldsymbol{x}) \text{ subject to } x_i(1 - x_i) = 0\ (1 \leq i \leq n). \tag{31}$$

The box POP can be expressed as

$$\min f(\boldsymbol{x}) \text{ subject to } x_i \geq 0,\ 1 - x_i \geq 0\ (1 \leq i \leq n). \tag{32}$$

Alternatively, it can also be forumlated as

$$\min f(\boldsymbol{x}) \text{ subject to } x_i(1 - x_i) \geq 0\ (1 \leq i \leq n). \tag{33}$$

We applied SparsePOP to POPs (31), (32) and (33) with the relaxation order $\omega = \lceil \deg f/2 \rceil$ and

$$\begin{aligned}
\mathcal{A} &= \left\{\boldsymbol{r}(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{Z}^n_+, \sum_{i=1}^n \alpha_i \leq \omega\right\} \text{ if } f \text{ is dense,} \\
\mathcal{A} &= \bigcup_{k=1}^m \left\{\boldsymbol{r}(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{Z}^n_+, \sum_{i\in C_k} \alpha_i \leq \omega, \alpha_i = 0\ (i \notin C_k)\right\} \text{ if } f \text{ is arrow-type} \textit{sparse}
\end{aligned}$$

to generate the lowest hierarchy of SDP relaxation problems [21] (see also [27] for the hierarchy of sparse SDP relaxations). Note that when the degree of f is even, the formulation (32) is not effective in obtaining tight lower bounds for the box POP. This can be easily verified from the construction of the hierarchy of SDP relaxation problems.

Each dense test problem solved by Algorithm 2.1 is of the form (2) with $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}}_+$ and $\mathbb{K}^2 = \mathbb{L}_{\mathrm{d}} \subset \mathbb{S}^{\mathcal{A}}$ for some $\mathcal{A} \subset \mathbb{Z}^n_+$ (see Section 4), while each sparse test problem is of the form (2) with $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}_1}_+ \times \cdots \times \mathbb{S}^{\mathcal{A}_m}_+$ and $\mathbb{K}^2 = \mathbb{L}_s \subset \mathbb{S}^{\mathcal{A}_1} \times \cdots \times \mathbb{S}^{\mathcal{A}_m}$ for some $\mathcal{A}_1, \ldots, \mathcal{A}_m \subset \mathbb{Z}^n_+$ (see Section 5).

All the numerical experiments were run in Matlab on a Mac Pro with Intel Xeon E5 8-core CPU (3.0 GHZ) and 64 GB memory. We use the following notation in Tables on numerical results.

SeDuMi: SeDuMi applied the SDP relaxation problem of POP (1) generated by SparsePOP with the relaxation order $\omega = \lceil \mathrm{deg} f/2 \rceil$.

SDPT3: SDPT3 applied the SDP relaxation problem of POP (1) generated by SparsePOP with the relaxation order $\omega = \lceil \mathrm{deg} f/2 \rceil$.

SDPNAL+: SDPNAL+ applied the SDP relaxation problem of POP (1) generated by SparsePOP with the relaxation order $\omega = \lceil \mathrm{deg} f/2 \rceil$. Termination code "termcode" is included in Tables 2 through 7: termcode 0 means "successfully solved", -2 "partially solved", *i.e.*, "primal feasibility is good but dual feasibility is not good enough", and 1 "not solved". Note that SDPNAL+ is a solver that is expected to work well on nondegenerate SDPs but it may not be robust for many degenerate problems which are often encountered in the SDP relaxations of POPs.

Alg 2.1-D: Algorithm 2.1 applied to the dense DNN relaxation (2) (with $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}}_+$ and $\mathbb{K}^2 = \mathbb{L}_d$ defined in Sections 4.1 and 4.4) of POP (1) (or the dense DNN relaxation (see Section 4.3) of QOP (6)).

Alg 2.1-S: Algorithm 2.1 applied to the sparse DNN relaxation (2) (with $\mathbb{K}^1 = \mathbb{S}^{\mathcal{A}_1}_+ \times \cdots \times \mathbb{S}^{\mathcal{A}_m}$ and $\mathbb{K}^2 = \mathbb{L}_s$ defined in Sections 5.1 and 5.3) of POP (1).

Alg 2.1-LD: Algorithm 2.1 applied to the Lagrangian-Dual DNN relaxation (30) of QOP (27) (see Section 6.4).

All tables include the smallest feasible objective values computed by applying the MATLAB toolbox `fmincon` to the POPs with the initial points obtained from SparsePOP combined with SDPNAL+.

Table 1 shows the numerical results on small and medium-sized binary and box POPs. We observe that

(a) SeDuMi and SDPT3 based on the primal-dual interior-point method can handle only small to medium scale SDP relaxation problems because they are too time-consuming and memory-intensive to solve large scale problems.

Thus, for the subsequent experiments, only the numerical results obtained by only two solvers, SDPNAL+ and Algorithm 2.1, are shown.

Next, we summarize the important points commonly observed from the numerical results comparing SDPNAL+ with Algorithm 2.1 in Tables 2, 3, 4 and 5 as follows.

(b) When SDPNAL+ converges, the lower bounds are tight. (Tables 2, 3, 4 and 5).

(c) For large scale sparse POPs, Algorithm 2.1 is much faster than SDPNAL+ (Tables 4 and 5).

Table 1: A brief comparison of Algorithm 2.1 with SeDuMi, SDPT3 and SDPNAL+ for small and medium-sized randomly-generated dense binary and box constrained POPs. For each problem pb$d$_n (or pd_n), d denotes the degree and n the number of variables. ‡ : not converged in 100 iterations.

Problem	Smallest feasible obj.val. computed	Lower bound(seconds) SeDuMi	SDPT3	SDPNAL+	Alg 2.1-D
Binary constrained					
pb3_10	-1.0746e1(8.3e-1)	-1.0746e1(9.0e-1)	-1.0746e1(1.1e0)	-1.0746e1(2.7e0)	-1.0746e1(8.1e0)
pb3_20	-3.2664e1(6.0e-1)	-3.2664e1(1.3e3)	-4.0348e1(8.2e2)‡	-3.2664e1(3.7e1)	-3.2664e1(9.3e1)
pb3_40	-1.3431e2(1.9e0)	Out of memory	Out of memory	-1.3431e2(1.6e3)	-1.3431e2(3.7e3)
pb4_20	-6.9048e1(2.2e0)	-6.9048e1(1.3e3)	-8.8985e1(8.5e2)‡	-6.9048e1(4.7e1)	-6.9049e1(6.3e1)
Box constrained					
p3_10	-1.6500e1(1.2e-1)	-1.6500e1(4.0e0)	-1.6500e1(8.0e-1)	-1.6500e1(1.7e0)	-1.6530e1(8.8e0)
p3_20	-4.0158e1(3.7e-1)	-4.0158e1(4.9e3)	-4.0158e1(7.3e0)	-4.0158e1(7.1e0)	-4.0232e1(1.3e2)
p3_40	-1.7888e2(2.2e0)	Out of memory	Out of memory	-1.7888e2(7.9e1)	-1.8092e2(4.1e3)
p4_20	-9.0463e1(2.5e0)	-9.0463e1(5.4e3)	-9.0463e1(1.1e3)	-9.0463e1(2.2e1)	-9.0878e1(9.1e1)

(d) The lower bounds obtained by Algorithm 2.1 are slightly worse than those obtained by SDPNAL+ for binary POPs (Tables 2 and 4).

(e) The lower bounds obtained by Algorithm 2.1 are worse than those obtained by SDPNAL+ for box POPs (Tables 3 and 5). This is mainly because the relaxation problems solved by Algorithm 2.1 and SDPNAL+ are not equivalent.

(f) The lower bounds computed by SDPNAL+ are invalid in many instances if the iteration is terminated due to exceeding the maximum execution time before satisfying the other stopping criteria (Tables 2, 3, 4 and 5). It requires very strict stopping criteria to generate a valid lower bound.

(g) Algorithm 2.1 consistently generates a valid lower bound at each iteration, and even if the iteration is terminated due to consuming the maximum execution time without satisfying the other stopping criteria, the lower bound computed is always valid (Tables 2, 3, 4 and 5).

7.1 Dense binary and box constrained POPs with randomly generated coefficients

Tables 2 and 3 show numerical results on dense binary and box constrained POPs with randomly generated coefficients, respectively. For the largest problems shown in Table 2, pb3_60, pb4_60, pb5_25 and pb6_25, two results obtained using the maximum execution time 20,000 and 100,000 seconds are shown. In Table 3, the maximum execution time to solve p6_20 was increased from 20,000 to 100,000 seconds for SDPNAL+ and two results are shown in the last row.

In Tables 2 and 3, we notice that Observation (b), (d), (e) and (g) hold. It should be emphasized that the lower bounds generated by Algorithm 2.1 is always valid, while those by SDPNAL+ are not, in particular, when the iteration is terminated after spending the prescribed maximum execution time (Observation (g)). For the instances pb6_25 of Table 2 and p6_20 of Table 3, increasing the maximum execution time allowed for SDPNAL+ did not provide better valid lower bounds. We observe some differences between SDPNAL+ applied to (32) and SDPNAL+ applied to (33) in the box constrained case. The former is more efficient than the latter for POPs of odd degree 3 and 5, and the converse is true for POPs of even degree. However, Observation (c) is not so clear in Tables 2 and 3

Table 2: Randomly-generated **dense binary** POPs. For each problem pb$d$_n, d denotes the degree and n the number of variables. † : not converged in 20,000 seconds. ‡ : not converged in 100,000 seconds. ♯ : an invalid lower bound greater than the smallest feasible objective value is computed.

Problem pb$d$_n	Smallest feasible obj. val. computed	Lower bound(seconds,iterations,termcode) SDPNAL+	Alg 2.1-D
pb3_40	-1.3431168e2(1.9e0)	-1.3431168e2(1.6e3,1553,0)	-1.3431426e2(3.7e3,20)
pb3_50	-2.4309905e2(3.7e0)	-2.4309905e2(4.2e3,943)	-2.4555535e2(5.4e3,19)
pb3_60	-3.0170687e2(5.6e1)	-3.0028013e2(2.0e4,2212,1)†♯ -2.7422592e2(1.0e5,19855,1)‡♯	-3.4077258e2(2.0e4, 7)† -3.0177697e2(8.1e4,20)
pb4_40	-3.7797114e2(2.3e1)	-3.7797114e2(8.7e2,947,0)	-3.7798193e2(2.7e3,19)
pb4_50	-7.8894627e2(7.2e1)	-7.8894627e2(1.4e4,7019,0)	-7.8939585e2(1.9e4,18)
pb4_60	-1.0991199e3(1.0e3)	-1.0723163e3(2.0e4,2758,1)†♯ -1.1114847e3(1.0e5,16783,1)‡	-1.1415732e3(2.0e4, 9)† -1.1180038e3(8.8e4,20)
pb5_15	-3.4951734e1(2.6e0)	-3.4951734e1(4.3e2,228,0)	-3.4953479e1(7.7e2,18)
pb5_20	-1.3404547e2(7.7e0)	-1.3404547e2(6.2e3,945,0)	-1.4007048e2(1.1e4,19)
pb5_25	-2.5880453e2(3.1e2)	-1.3326898e2(2.0e4)†♯ -2.3500146e2(1.0e5,5281)‡♯	-3.1239181e3(2.0e4,2)† -3.8009333e2(1.0e5,7)‡
pb6_15	-5.9278171e1(6.0e0)	-5.9278171e1(9.5e2,346,0)	-5.9279006e1(9.9e2,19)
pb6_20	-2.2733885e2(3.8e1)	-2.2733885e2(1.0e4,2170,0)	-2.2813238e2(1.3e4,18)
pb6_25	-5.2927540e2(2.8e3)	-8.3542993e1(2.1e4,105,1)†♯ -3.3574264e2(1.0e5,3633,1)‡♯	-4.0292443e3(2.0e4,2)† -6.1935818e2(1.0e5,10)‡

Table 3: Same as Table 2 but for randomly generated **dense box constrained** POPs.

		Lower bound(seconds,iterations,termcode)		
Problem	Smallest feasible	SDPNAL+		Alg 2.1-D
pd_n_i	obj. val. computed	Applied to (32)	Applied to (33)	
p3_40	-1.7887545e2(2.2e0)	-1.7887545e2(7.9e1,331,0)	-1.7887545e2(1.7e3,955,0)	-1.8092428e2(4.1e3,19)
p3_50	-2.4947782e2(3.5e1)	-2.4947539e2(1.3e3,1216,1)♯	-2.4947782e2(1.1e04,4605,0)	-2.5453114e2(2.0e4,19)
p3_60	-3.0639131e2(9.7e1)	-3.1404614e2(3.3e3,4552,-2)	-3.0638937e2(2.0e4,4852,1)†♯	-3.2737213e2(2.2e4,10)†
p4_40	-4.6789000e2(3.9e1)	-1.7126932e3(7.8e3,10073,0)	-4.6789000e2(8.4e2,339,0)	-4.8304755e2(4.2e3,19)
p4_50	-9.5277366e2(8.3e1)	Out of memory	-9.5277366e2(1.4e4,5192,0)	-1.0256302e3(2.1e4,12)†
p4_60	-1.3377196e3(6.2e1)	Out of memory	-1.3377196e3(1.8e4,4817,0)	-1.4979588e3(2.0e4,10)†
p5_15	-7.3225057e1(2.7e0)	-7.3225057e1(1.2e2,224,0)	-7.3225052e1(7.2e2,1558,0)♯	-7.8113948e1(1.4e3,18)
p5_20	-1.4864741e2(9.5e0)	-1.4864741e2(6.3e2,227,0)	-1.4765645e2(2.0e4,11362,1)†♯	-1.5522496e2(1.3e4,20)
p5_25	-3.5593726e2(3.1e1)	-3.5593726e2(1.4e3,222,0)	-2.3370923e2(2.0e4,326,1)†♯	-6.1178277e2(2.0e4, 6)†
p6_10	-1.9308541e1(1.0e0)	-5.9155001e1(6.1e1,337,1)	-1.9308541e1(3.6e1,110,0)	-2.0388298e1(1.9e2,20)
p6_15	-1.2763216e2(9.5e0)	-3.0861920e2(1.8e3,5206,-2)	-1.2763216e2(1.1e3,956,0)	-1.3005318e2(1.4e3,20)
p6_20	-3.5573291e2(9.6e2)	-9.9222209e2(5.2e3,1863,-2)	-3.6483341e2(2.0e4,9195,1)† -4.0419402e2(1.0e5,84394,1)‡	-3.8853628e2(2.0e4,21)

7.2 Sparse binary and box POPs

To clearly see Observation (c) and the others, Tables 4 and 5 display numerical results on five instances for each type of binary and box POP instances. We first mention that for each degree d =3, 4, 5 and 6 in Table 4, SDPNAL+ spent more than 5,000 seconds for most POP instances with smaller number of blocks (20,000 seconds for most POP instances with larger number of blocks), while Algorithm 2.1 less than 2,000 seconds for the former instances (5,000 seconds for the latter instances). From Table 5, we observe that Algorithm 2.1 spent less than 4,000 seconds for the POPs of degree 3 and 4, and less than 10,000 seconds for the POPs of degree 5 and 6. On the other hand, SDPNAL+ applied to (32) for the POP instances of degree 5 and 6, and SDPNAL+ applied to (33) for the POP instances of all degrees spent more 20,000 seconds. The execution time of SDPNAL+ applied to (32) for the POP instances of degree 3 and 4 is not much longer than the one of Algorithm 2.1. However, the lower bound obtained by SDPNAL+ is worse than the one by Algorithm 2.1. The other observations (b), (d), (e), (f) and (g) can also be confirmed in Tables 4 and 5.

In the two columns under SDPNAL+ in Table 5, we compare the results obtained by SDPNAL+ applied to (32) and SDPNAL+ applied to (33). For the test instance of degree 3 and even degree 4, the former performed more efficiently than the latter in the sense that it converged for more instances in less execution time. However, the quality of lower bounds are worse than those obtained by Algorithm 2.1 in many instances. For the test instance of degree 5 and 6, both did not converge in the maximum execution time 20,000 seconds.

If we compare the column SDPNAL+ applied to (33) in Table 5 with the column SDPNAL+ in Table 4, we notice that the performance of SDPNAL+ for the box POP instances are worse than that for the binary POP instances. In particular, SDPNAL+ spent a large number of iterations even for the instance of smaller degree and dimension. This suggests that the sparse box POP instances are so ill-conditioned that SDPNAL+, which is designed under the nondegeneracy assumption, could not avoid numerical difficulty. On the other hand, the performance of Algorithm 2.1 is stable and robust against ill-conditioning of the instances.

7.3 The maximum complete satisfiability problem [14]

Let x_i denote a boolean variable of value 1 or 0 for true or false ($i = 1, \ldots, n$). For each $k = 1, \ldots, m$, we let I_k and J_k be disjoint subsets of $\{1, \ldots, n\}$, and define a conjunctive clause $(\wedge_{i \in I_k} x_i) \wedge (\wedge_{j \in J_k}(1 - x_j))$. The maximum complete satisfiability problem [14] is to find an $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0, 1\}^n$ that maximizes the total weighted sum of the satisfied clauses. This problem is different from the well-known maximum satisfiability problem (see *e.g*, [15]) and it is NP-hard in general. Let $\boldsymbol{w} = (w_1, \ldots, w_m)$ denote a weight vector. Then the problem is formulated as a binary POP with the objective polynomial $f(\boldsymbol{x}) = \sum_{k=1}^{m} w_k \left(\Pi_{i \in I_k} x_i\right) \left(\Pi_{j \in J_k}(1 - x_j)\right)$. For the numerical experiment, each problem is constructed as follows: Let $10 \leq m = n \leq 120$, $d = 4$, and choose w_k randomly from $\{1, \ldots, n\}$ ($k = 1, \ldots, m$). Choose d elements for H_k ($k = 1, \ldots, m$) with the first element k and the other $d - 1$ elements randomly from $\{1, \ldots, n\}$; eliminate any

Table 4: **Sparse binary** POPs with an arrow-type Hessian matrix. See Example 5.1. For each problem pb$d$_n_i, d denotes the degree, n the number of variables and i the instance number. † : not converged in 20,000 seconds. ♯ : an invalid lower bound greater than the smallest feasible objective value computed.

Problem	No. of	Smallest feasible	Lower bound(seconds,iterations,termcode)	
pb$d$_d_n_i	blocks	obj.val. computed	SDPNAL+	Alg 2.1-S
pb3_1604_1	200	-2.1997436e3	-2.1997436e3(8.3e3,947,0)	-2.2002569e3(1.1e3,17)
pb3_1604_2		-2.2461680e3	-2.2461680e3(5.9e3,340,0)	-2.2471557e3(8.4e2,16)
pb3_1604_3		-2.1021807e3	-2.1021807e3(6.3e3,943,0)	-2.1027581e3(1.4e3,17)
pb3_1604_4		-2.1562787e3	-2.1562787e3(7.4e3,1559,0)	-2.1569333e3(1.3e3,17)
pb3_1604_5		-2.3105033e3	-2.3105033e3(8.1e3,1552,0)	-2.3111817e3(1.4e3,18)
pb3_3204_1	400	-4.3839531e3	-4.3839196e3(2.1e4,223,1)†♯	-4.3852815e3(1.8e3,17)
pb3_3204_2		-4.5327543e3	-4.5323085e3(2.1e4,228,1)†♯	-4.5340863e3(2.1e3,17)
pb3_3204_3		-4.2702357e3	-4.2698532e3(2.1e4,326,1)†♯	-4.2714012e3(2.8e3,17)
pb3_3204_4		-4.3029253e3	-4.3021819e3(2.1e4,329,1)†♯	-4.3043842e3(1.7e3,17)
pb3_3204_5		-4.4931418e3	-4.4930846e3(2.1e4,223,1)†♯	-4.4948996e3(1.7e3,16)
pb4_1604_1	200	-3.4331422e3	-3.4331422e3(8.8e3,1556,0)	-3.4344221e3(1.0e3,18)
pb4_1604_2		-3.4992151e3	-3.4992151e3(1.0e4,2165,0)	-3.5006061e3(9.6e2,16)
pb4_1604_3		-3.4189951e3	-3.4189952e3(6.6e3,334,0)	-3.4197441e3(1.0e3,17)
pb4_1604_4		-3.4765899e3	-3.4765899e3(8.6e3,2166,0)	-3.4777729e3(1.3e3,17)
pb4_1604_5		-3.5926287e3	-3.5926287e3(8.6e3,1553,0)	-3.5939874e3(1.1e3,17)
pb4_3204_1	400	-7.0085677e3	-7.0024449e3(2.1e4,218,1)†♯	-7.0115762e3(2.4e3,18)
pb4_3204_2		-7.0968703e3	-7.0913638e3(2.1e4,225,1)†♯	-7.0999936e3(2.2e3,16)
pb4_3204_3		-6.9608273e3	-6.9608798e3(2.1e4,219,1)†	-6.9708816e3(3.1e3,18)
pb4_3204_4		-7.0953870e3	-7.0945816e3(2.0e4,941,1)†♯	-7.0970411e3(2.6e3,17)
pb4_3204_5		-7.0384205e3	-7.0384205e3(2.0e4,1558,0)	-7.0413237e3(2.3e3,17)
pb5_304_1	50	-6.0642468e2	-6.0642468e2(4.6e3,680,0)	-6.0653651e2(1.6e3,18)
pb5_304_2		-7.7386178e2	-7.7386178e2(7.9e3,1120,0)	-7.7401619e2(1.5e3,17)
pb5_304_3		-6.4975793e2	-6.4975794e2(4.8e3,675,0)	-6.4992372e2(1.6e3,19)
pb5_304_4		-6.6325702e2	-6.6325702e2(6.3e3,1004,0)	-6.6334678e2(1.6e3,17)
pb5_304_5		-7.0608611e2	-7.0608611e2(6.8e3,1119,0)	-7.0617419e2(1.6e3,18)
pb5_604_1	100	-1.2753461e3	-1.2753461e3(1.7e4,896,-2)	-1.2756316e3(3.7e3,19)
pb5_604_2		-1.3614391e3	-1.3614429e3(2.0e4,931,1)†	-1.3617865e3(3.9e3,19)
pb5_604_3		-1.2208977e3	-1.2208977e3(1.9e4,1234,0)	-1.2211126e3(3.2e3,18)
pb5_604_4		-1.3550429e3	-1.3549722e3(2.0e4,904,1)†♯	-1.3551998e3(3.6e3,18)
pb5_604_5		-1.3764567e3	-1.3762575e3(2.0e4,1013,1)†♯	-1.3766243e3(3.6e3,18)
pb6_304_1	50	-6.7242425e2	-6.7242425e2(5.8e3,1012,0)	-6.7252514e2(1.5e3,19)
pb6_304_2		-8.0649282e2	-8.0649282e2(6.5e3,1117,0)	-8.0657419e2(1.8e3,18)
pb6_304_3		-6.7021267e2	-6.7021267e2(5.7e3,900,0)	-6.7029384e2(1.5e3,17)
pb6_304_4		-7.6331606e2	-7.6331608e2(6.5e3,1229,0)	-7.6342791e2(1.7e3,18)
pb6_304_5		-8.8961209e2	-8.8961209e2(5.7e3,1007,0)	-8.8970295e2(1.6e3,18)
pb6_604_1	100	-1.3512216e3	-1.3511445e3(2.0e4,1121,1)†♯	-1.3515195e3(2.9e3,16)
pb6_604_2		-1.5560040e3	-1.5560040e3(1.6e4,676,0)	-1.5562514e3(3.4e3,17)
pb6_604_3		-1.2982412e3	-1.2982412e3(2.0e4,1121,0)	-1.2984504e3(3.3e3,17)
pb6_604_4		-1.4998704e3	-1.4993158e3(2.0e4,1355,1)†♯	-1.5004078e3(3.3e3,17)
pb6_604_5		-1.5609811e3	-1.5609811e3(1.9e4,1345,0)	-1.5612215e3(3.0e3,18)

Table 5: Same as Table 4 but for **sparse box constrained** POPs with an arrow-type Hessian matrix. See Example 5.1. For each problem pd_n_i, d denotes the degree, n the number of variables and i the instance number. † : not converged in 20,000 seconds. ♯ : an invalid lower bound greater than the smallest feasible objective value computed.

Problem pd_n_i	No. of blocks	n	Smallest feasible obj. val. computed	Lower bound(seconds,iterations,termcode) SDPNAL+ applied to (32)	SDPNAL+ applied to (33)	Alg 2.1-S
p3_404_1	50	404	-6.4835915e2	-6.6646634e2(2.1e3)	-6.4855392e2(2.0e4,57689,-2)†	-6.5627769e2(1.7e3,17)
p3_404_2			-7.7490100e2	-7.8571762e2(2.2e3)	-7.7490058e2(1.2e4,16774,-2)♯	-7.8108367e2(1.2e3,15)
p3_404_3			-6.4887431e2	-6.7209707e2(4.0e3)	-6.4892030e2(2.0e4,55508,-2)‡	-6.5976179e2(1.7e3,16)
p3_404_4			-6.6387872e2	-6.8146402e2(2.4e3)	-6.6387871e2(1.0e4,25753,-2)♯	-6.7207477e2(1.7e3,18)
p3_404_5			-7.2540206e2	-7.4642800e2(3.6e3)	-7.2542336e2(1.3e4,22511,-2)	-7.3555104e2(1.4e3,18)
p3_804_1	100	804	-1.2539954e3	-1.2983094e3(4.7e3)	-1.2542217e3(2.0e4,14307,-2)†	-1.2735872e3(2.6e3,18)
p3_804_2			-1.4885950e3	-1.5114329e3(6.4e3)	-1.4886093e3(2.0e4,11961,-2)†	-1.5005260e3(2.8e3,19)
p3_804_3			-1.3262266e3	-1.3693319e3(5.3e3)	-1.3275511e3(2.0e4,22322,-2)†	-1.3451431e3(2.7e3,16)
p3_804_4			-1.2905418e3	-1.3447272e3(6.7e3)	-1.2905801e3(2.0e4,10685,1)†	-1.3150838e3(3.0e3,18)
p3_804_5			-1.4290667e3	-1.4636710e3(1.4e4)	-1.4294010e3(2.0e4,21671,-2)†	-1.4500834e3(2.1e3,16)
p4_404_1	50	404	-1.2304645e3	-2.6498582e3(2.0e3)	-1.2308196e3(2.0e4,41584,-2)†	-1.2792926e3(1.6e3,17)
p4_404_2			-1.4196942e3	-2.7553991e3(1.6e3)	-1.4197537e3(1.9e4,32794,-2)	-1.4509952e3(1.7e3,18)
p4_404_3			-1.2505430e3	-2.6355390e3(2.0e3)	-1.2505407e3(2.0e4,54956,-2)†♯	-1.2800518e3(2.0e3,19)
p4_404_4			-1.3102123e3	-2.6864821e3(1.7e3)	-1.3100763e3(2.0e4,45999,-2)†♯	-1.3552020e3(1.6e3,18)
p4_404_5			-1.4373974e3	-2.7187567e3(1.8e3)	-1.4371755e3(2.0e4,46798,-2)‡♯	-1.4700240e3(1.5e3,19)
p4_804_1	100	804	-2.7166498e3	-5.3880387e3(3.4e3)	-2.7171469e3(2.0e4,10684,-2)†	-2.7892288e3(2.9e3,18)
p4_804_2			-2.6593645e3	-5.4099418e3(5.1e3)	-2.6597150e3(2.0e4,10077,1)†	-2.7461688e3(3.2e3,19)
p4_804_3			-2.7019525e3	-5.3400505e3(4.5e3)	-2.7029392e3(2.0e4,20676,-2)†	-2.7798426e3(2.8e3,18)
p4_804_4			-2.6001496e3	-5.3218009e3(4.0e3)	-2.6023429e3(2.0e4,11297,1)†	-2.6796387e3(2.5e3,18)
p4_804_5			-2.7832477e3	-5.4164729e3(5.0e3)	-2.7834017e3(2.0e4,13091,1)†	-2.8498279e3(2.8e3,18)
p5_124_1	20	124	-5.2684417e2	-5.2682336e2(2.0e4)†♯	-5.2613953e2(2.0e4,3451,1)†♯	-5.3974903e2(3.9e3,19)
p5_124_2			-6.8299731e2	-6.8306215e2(2.0e4)†	-6.8232896e2(2.0e4,4504,-2)†♯	-6.8957408e2(2.9e3,18)
p5_124_3			-5.3833723e2	-5.3900337e2(2.0e4)†	-5.3799620e2(2.0e4,3581,-2)†♯	-5.5374117e2(4.1e3,19)
p5_124_4			-6.3297885e2	-6.3806634e2(2.0e4)†	-6.3826134e2(2.0e4,5260,1)†	-6.4999465e2(3.7e3,19)
p5_124_5			-7.2165102e2	-7.2175872e2(2.0e4)†	-7.2169047e2(2.0e4,2536,1)†	-7.3071052e2(4.3e3,18)
p5_244_1	40	244	-9.5766733e2	-9.9585794e2(2.0e4)†	-8.4394200e2(2.0e4,329,1)†♯	-1.0285497e3(1.0e4,19)
p5_244_2			-1.3996523e3	-1.4099641e3(2.0e4)†	-1.0549446e3(2.0e4,221,1)†♯	-1.4356384e3(5.8e3,17)
p5_244_3			-1.1018976e3	-1.1336361e3(2.0e4)†	-1.1364095e3(2.0e4,217,1)†♯	-1.1620551e3(7.4e3,19)
p5_244_4			-1.1990841e3	-1.2124008e3(2.0e4)†	-9.9241399e2(2.0e4,223,1)†♯	-1.2358589e3(8.8e3,19)
p5_244_5			-1.2606745e3	-1.2704666e3(2.0e4)†	-1.0316164e3(2.0e4,220,1)†♯	-1.2904775e3(5.1e3,16)
p6_124_1	20	124	-1.0290734e3	-1.6541582e3(2.0e4)†	-1.0293307e3(2.0e4,2110,1)†	-1.0521775e3(3.0e3,18)
p6_124_2			-1.0331931e3	-1.6711682e3(2.0e4)†	-1.0310701e3(2.0e4,2489,1)†♯	-1.0607584e3(3.6e3,18)
p6_124_3			-9.4085465e2	-1.6799981e3(2.0e4)†	-9.3726051e2(2.0e4,2991,1)†♯	-9.7581187e2(3.8e3,16)
p6_124_4			-9.4832099e2	-1.6796790e3(2.0e4)†	-9.5668416e2(2.0e4,4247,-2)†	-9.8586919e2(3.2e3,18)
p6_124_5			-1.0728526e3	-1.6467684e3(2.0e4)†	-1.0737019e3(2.0e4,4509,-2)†	-1.0899702e3(3.1e3,18)
p6_244_1	40	244	-2.0650153e3	-3.3717560e3(2.0e4)†	-1.8389846e3(2.0e4,216,1)†♯	-2.1598254e3(5.6e3,15)
p6_244_2			-2.1412371e3	-3.4110162e3(2.0e4)†	-1.5542580e3(2.1e4,331,1)†♯	-2.2345968e3(1.0e4,17)
p6_244_3			-1.9789327e3	-3.4384583e3(2.0e4)†	-1.5826219e3(2.0e4,376,1)†♯	-2.1274461e3(6.9e3,19)
p6_244_4			-2.0624482e3	-3.4270695e3(2.0e4)†	-1.5578560e3(2.0e4,553,1)†♯	-2.1398578e3(9.3e3,19)
p6_244_5			-1.9863986e3	-3.3487017e3(2.0e4)†	-1.5494532e3(2.0e4,667,1)†♯	-2.0484788e3(6.2e3,18)

duplicated element from H_k. Finally, distribute the elements of H_k randomly to I_k and J_k ($k = 1, \ldots, m$) so that $H_k = I_k \cup J_k$ and $I_k \cap J_k = \emptyset$.

We see in Table 6 that SDPNAL+ attains very tight lower bounds in less execution time than Algorithm 2.1 in the instances with dimension 60 and 80 (Observation (b)). But the validity of the obtained lower bounds in the largest instances with dimension 100 is not guaranteed when it terminated before convergence (Observation (f)). For each of mcs4_100 instances, three lower bounds obtained by Alg 2.1-S are displayed. The first two lower bounds were obtained by setting the maximum number of iterations to 3 and 6, respectively. The lower bounds show that Alg 2.1-S generates valid and meaningful (or nontrivial) lower bounds even if it was terminated at iteration 3 and 6. Observations (d) and (g) also remain valid. The structured sparsity of the test problems does not exist to the extent to be exploited in comparison to the sparse binary POP instances presented in Table 4 since each conjunctive clause is generated randomly. Thus SDPNAL+ and Alg 2.1-S can not solve instances with dimension larger than 100 in 20,000 seconds, and we are not be able to confirm whether Alg 2.1-S possibly competes with SDPNAL+ for larger scale instances (Observation (c)).

Table 6: Results for **maximum complete satisfiability** problems. For each problem mcs$d$_n_i, d denotes the degree, n the number of variables, and i the instance number. † : not converged in 20,000 seconds. ♯ : an invalid lower bound greater than the smallest feasible objective value computed.

Problem	Smallest feasible	Lower bound(seconds,iterations,termcode)	
mcs$d$_n	obj. val. computed	SDPNAL+	Alg 2.1-S
mcs4_60_1	-8.1100000e2(2.1e-1)	-8.1100000e2(9.4e2,957,0)	-8.1179365e2(3.1e3,18)
mcs4_60_2	-8.1800000e2(3.9e1)	-8.1799999e2(4.8e2,,231,0)	-8.1988907e2(2.2e3,17)
mcs4_60_3	-8.3100000e2(3.4e-1)	-8.3100000e2(3.5e2,337,0)	-8.3161053e2(2.4e3,17)
mcs4_60_4	-9.0100000e2(4.7e1)	-9.0100000e2(9.9e2,2074,0)	-9.0525489e2(2.4e3,17)
mcs4_60_5	-8.3100000e2(1.8e-1)	-8.3100000e2(4.6e2,238,0)	-8.3162079e2(2.6e3,18)
mcs4_80_1	-1.2790000e3(8.6e-1)	-1.2789997e3(5.5e3,3218,0)	-1.2839913e3(1.0e4,17)
mcs4_80_2	-1.5290000e3(2.2e0)	-1.5290008e3(4.4e3,1299,0)	-1.5349946e3(1.7e4,17)
mcs4_80_3	-1.1990000e3(5.3e-1)	-1.1990000e3(6.6e3,1476,0)	-1.2062165e3(2.1e4,16)†
mcs4_80_4	-1.4610000e3(8.9e-1)	-1.4610000e3(9.0e3,2167,0)	-1.4774670e3(2.1e4,17)†
mcs4_80_5	-1.5110000e3(6.2e1)	-1.5109987e3(5.9e3,1872,0)	-1.5201273e3(2.0e4,17)†
mcs4_100_1			-3.6120060e3(2.9e3,3)
			-2.7090128e3(8.8e3,6)
	-2.5020000e3(4.3e0)	-2.5020091e3(2.0e4,2827,-2)†	-2.5326438e3(2.1e4,12)†
mcs4_100_2			-3.221253803(3.6e3,3)
			-2.4159464e3(1.2e4,6)
	-2.3000000e3(1.6e0)	-2.3007635e3(2.0e4,1751,-2)†	-2.3579718e3(2.2e4,9)†
mcs4_100_3			-3.3927573e3(4.5e3,3)
			-2.5445680e3(1.5e4,6)
	-2.4200000e3(3.8e0)	-2.4160633e3(2.0e4,807,-2)†♯	-2.4488138e3(2.1e4,8)†
mcs4_100_4			-3.1077532e3(3.6e3,3)
			-2.7192847e3(1.7e4,6)
	-2.3800000e3(3.2e0)	-2.3793849e3(2.0e4,1463,-2)†♯	-2.4279324e3(2.1e4,8)†
mcs4_100_5			-2.7567547e3(3.3e3,3)
			-2.4121636e3(1.4e3,6)
	-2.2110000e3(3.9e0)	-2.2116967e3(2.0e4,1548,1)†	-2.2742079e3(2.1e4,9)†

7.4 Binary constrained QOPs from BIQMAC [29]

From the numerical results on binary constrained QOPs in Table 7, we see that Alg 2.1-LD on (30) provides better bounds than Alg 2.1-D, SDPNAL+ and SDPNAL+ on (30), taking shorter execution time than SDPNAL+ on (30). If we compare the results of bpq100-1 through bpq100-5 to Table 1 of [16], we notice that the execution time is comparable. The lower bounds are slightly worse than the ones in [16] but the validity of their bounds is not guaranteed there.

Table 7: Binary QOP instances with $n = 100$ and $n = 500$ from BIQMAC [29]. We took $\lambda = 1000$ in the Lagrangian-Dual DNN Relaxation (30).

	Lower bounds (seconds,termcode)			
Instance(Opt.val)	Alg 2.1-D	SDPNAL+	Alg 2.1-LD on (30)	SDPNAL+ on (30)
bqp100-1(-7.970e3)	-8.3808e3 (1.3e1)	-8.3804e3(7.0e0,0)	-8.0862e3(2.8e1)	-8.1306e3(1.3e2,1)
bqp100-2(-1.1036e4)	-1.1490e4(1.8e1)	-1.1489e4(1.2e1,0)	-1.1067e4(2.9e1)	-1.1212e4(1.1e2,1)
bqp100-3(-1.2723e4)	-1.3154e4(1.6e1)	-1.3153e4(1.2e1,0)	-1.2743e4(2.6e1)	-1.2874e4(1.6e2,1)
bqp100-4(-1.0368e4)	-1.0732e4(1.7e1)	-1.0732e4(9.2e0,0)	-1.0394e4(2.4e1)	-1.0466e4(1.3e2,1)
bqp100-5(-9.083e3)	-9.4873e3(1.5e1)	-9.4870e3(8.1e0,0)	-9.1111e3(2.9e1)	-9.1550e3(1.6e2,1)
bqp500-1(-1.16586e5)	-1.2597e3(7.2e2)	-1.2596e5(1.4e2,0)	-1.2315e5(1.7e3)	-1.2846e5(5.7e3,1)
bqp500-2(-1.28223e5)	-1.3602e5(8.8e2)	-1.3601e5(2.1e2,0)	-1.3330e5(2.2e3)	-1.3813e5(8.2e3,1)
bqp500-3(-1.30812e5)	-1.3847e5(6.8e2)	-1.3845e5(2.0e2,0)	-1.3547e5(2.2e3)	-1.3984e5(7.4e3,1)
bqp500-4(-1.30097e5)	-1.3934e5(7.1e2)	-1.3933e5(1.6e2,0)	-1.3612e5(2.2e3)	-1.4102e5(7.9e3,1)
bqp500-5(-1.25487e5)	-1.3411e5(8.0e2)	-1.3409e5(2.8e2,0)	-1.3100e5(2.1e3)	-1.3575e5(7.1e3,1)

8 Concluding remarks

For POPs with binary and box constraints, we have provided a theoretical framework under which many DNN relaxations can be obtained and uniformly formulated as a simple COP. Moreover, the computation of the metric projection onto the polyhedral cone in the COP can be carried out efficiently and accurately. The framework has also been used to prove why the (accelerated) BP algorithm was successful in obtaining tight bounds when applied to the COP from the QOPs in [4, 16]. As the most important step of the BP algorithm is in computing the metric projection, the framework presented in this paper expands the applicability of the BP algorithm to POPs from QOPs. In fact, a wide range of general POPs can be solved with the proposed method as described in the following.

We see that any variable y_i bounded by an interval $[\ell_i, u_i]$ with $-\infty < \ell_i < u_i < \infty$ can be scaled to $x_i = (y_i - \ell_i)/(u_i - \ell_i) \in [0, 1]$. Thus, it is possible to assume that all the lower and upper bounds for variables are 0 and 1 in a POP with bounded variables. Moreover, if an additional inequality constraint $h(\boldsymbol{x}) \geq 0$ is included in the POP (1), it can be written as an equality constraint $h(\boldsymbol{x}) - ay = 0$ with a slack variable $y \geq 0$, where a is a positive number. Since $h(\boldsymbol{x})$ is bounded in S, $y \in [0, 1]$ can be added by taking a sufficiently large $a > 0$.

Now, suppose that equality constraints $g_i(\boldsymbol{x}) = 0$ $(i = 1, \ldots, m)$ are added to (1):

$$\text{minimize } f_0(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in S,\ g_i(\boldsymbol{x}) = 0\ (i = 1, \ldots, m). \tag{34}$$

Assume that the resulting problem (34) is feasible. Obviously, each equality constraint $g_i(\boldsymbol{x}) = 0$ can be rewritten as $g_i(\boldsymbol{x})^2 = 0$. One can consider using a Lagrangian relaxation method to solve (34) as follows. For a sufficiently large number parameter $\lambda > 0$, consider the Lagrangian relaxation of (34):

$$\text{minimize } f_0(\boldsymbol{x}) + \lambda \textstyle\sum_{i=1}^m g_i(\boldsymbol{x})^2 \text{ subject to } \boldsymbol{x} \in S, \tag{35}$$

where the term $\lambda \sum_{i=1}^m g_i(\boldsymbol{x})^2$ added to the objective function $f_0(\boldsymbol{x})$ of (34) serves as a penalty in the sense that if $\boldsymbol{x} \in S$ is not a feasible solution of (34), then the objective value $f_0(\boldsymbol{x}) + \lambda \sum_{i=1}^m g_i(\boldsymbol{x})^2$ of (35) diverges to $+\infty$ as $\lambda \to \infty$. Using the compactness of the feasible region S of (35), it is easy to prove that the optimal value of (35) converges to the optimal value of (34) as $\lambda \to \infty$. Consequently, a lower bound of the optimal value of POP (34) can be computed by applying Algorithm 2.1 to the DNN relaxation of POP (35) with binary and box constraints for a sufficiently large $\lambda > 0$.

Acknowledgement. The authors would like to thank Professor Akiko Takeda for suggesting the maximum complete satisfiability problem as a test example, and Mr. Naoki Ito for improving the MATLAB codes on Algorithm 2.1.

References

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

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

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

[4] N. Arima, S. Kim, M. Kojima, and K.C. Toh. A robust Lagrangian-DNN method for a class of quadratic optimization problems. *Comput. Optim. Appl.*, 66(3):453–479, 2017.

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

[6] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In Liu J.W.H. George A., Gilbert J. R., editor, *Graph Theory and Sparse Matrix Computation*. Springer-Verlag, New York, 1993.

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

[8] M.C. Columbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[9] H. Dong and K. Anstreicher. Separating doubly nonnegative and completely positive matrices. *Math. Program.*, 137(131-153), 2013.

[10] T. Fujie and M. Kojima. Semidefinite programming relaxation for nonconvex quadratic programs. *Journal of Global Optimization*, 10:367–368, 1997.

[11] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion. I: General framework. *SIAM J. Optim.*, 11:647–674, 2000.

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

[13] B. Ghaddar, J. C. Vera, and M. F. Anjos. A dynamic inequality generation scheme for polynomial programming. *Math. Program.*, 156(1):21–57, 2016.

[14] S. He, Z. Li, and S. Zhang. Approximation algorithms for discrete polynomial optimization. *J. of Oper. Res. Soc. China*, 1(1):3–36, 2013.

[15] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9:256–278, 1974.

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

[17] S. Kim, M. Kojima, and K. C. Toh. Binary and box polynomial optimization test problems. http://math.ewha.ac.kr/˜skim/TestProb/TestProb.html, September 2017.

[18] M. Kobayashi, S. Kim, and M. Kojima. Correlative sparsity in primal-dual interior-point methods for lp, sdp, and socp. *Appl. Math. Optim.*, 58:69–88, 2008.

[19] M. Kojima, S. Shindoh, and S. Hara. Interior-point methods for the monotone linear complementarity problem in symmetric matrices. *SIAM J. Optim.*, 7:86–125, 1997.

[20] J. B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *Integer Programming and Combinatorial Optimization*, pages 293–303. Springer, 2001.

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

[22] J. J. Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cones mutuellement polaires. *C. R. Acad. Sci.*, 255:238–240, 1962.

[23] Y. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM J. Optim.*, 8:324–363, 1998.

[24] N. Z. Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25:1–11, 1987.

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

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

[27] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.*, 17:218–242, 2006.

[28] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto. Algorithm 883: Sparsepop: A sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Trans. Math. Softw.*, 35(15), 2008.

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

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

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