MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

Using Multiparameter Eigenvalues for Solving Quadratic Programming with Quadratic Equality Constraints

Shinsaku SAKAUE

(Communicated by Akiko TAKEDA)

METR 2016-02 March 2016

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html

Using multiparameter eigenvalues for solving quadratic programming with quadratic equality constraints

Shinsaku Sakaue*

March 23, 2016

Abstract

We show that multiparameter eigenvalues can solve some optimization problems. Specifically, we develop an eigenvalue-based algorithm for quadratic programming with quadratic equality constraints (QECQP). QECQP models various well-known optimization problems such as the maximum-cut problem, and QECQP is also closely related to quadratically constrained quadratic programming (QCQP). Recently, for some special cases of QCQP with one or two constraints, algorithms based on eigenvalue with one or two parameter have been proposed, which can solve some nonconvex instances, for which ordinary optimization methods often fail.

In this paper, we generalize the aforementioned eigenvalue-based algorithms by allowing for larger number of constraints; using multiparameter eigenvalue problems, we propose an algorithm that is applicable to QECQP with an arbitrary fixed number of constraints. Unfortunately, the algorithm is not proved to find a global solution. However, we show in experiments that our algorithm works for small-scale instances and computes a global solution with high accuracy, as long as the effects of singular matrices are small enough for our algorithm to work well.

Keywords. multiparameter eigenvalue problem, quadratic programming with quadratic equality constraints, quadratically constrained quadratic programming.

AMS Classification. 15A18, 65F15, 90C20.

1 Introduction

Multiparameter eigenvalue problems have been studied in the literature [4, 13] and various studies have been conducted on applying them to many types of differential equations [5, 15, 20]. However, aside from these studies, not many applications seem to take advantage of them. The main message of this paper is that multiparameter eigenvalue problems can be used to solve some optimization problems. Specifically, in this paper we propose an algorithm based on multiparameter eigenvalues for solving the following quadratic programming with quadratic equality constraints (QECQP):

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^\top Q_0 x + 2q_0^\top x + \gamma_0 \\ \text{subject to} \quad & x^\top Q_i x + 2q_i^\top x + \gamma_i = 0 \quad (i = 1, \dots, m) \end{aligned} \tag{1}$$

*Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan (shinsaku_sakaue@mist.i.u-tokyo.ac.jp). This research is supported by Grants-in-Aid for Scientific Research (Challenging Exploratory Research, No. 26540007) from Japan Society for Promotion of Science.

where $q_i \in \mathbb{R}^n$, $\gamma_i \in \mathbb{R}$ and $Q_i \in \mathbb{R}^{n\times n}$ are symmetric for $i = 0, 1, \ldots, m$. QECQP models various optimization problems such as the maximum-cut problem (see [11]) and polynomial optimization problems (see [18]). Taking m as a variable, solving QECQP requires exponential computation time, and so approximation algorithms and relaxation methods are extensively studied for QECQP; one of the most famous methods is the approximation algorithm for the maximum cut problem proposed by Goemans and Williamson [11].

QECQP is also closely related to quadratically constrained quadratic programming (QCQP), which is a well-known optimization problem of the following form:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^\top Q_0 x + 2q_0^\top x + \gamma_0 \\ \text{subject to} \quad & x^\top Q_i x + 2q_i^\top x + \gamma_i \le 0 \quad (i = 1, \ldots, m). \end{aligned} \tag{2}$$

Actually, by introducing additional m slack variables $s_i \in \mathbb{R}$ $(i = 1, \ldots, m)$, QCQP can be expressed with $n + m$ variables as follows:

$$\begin{aligned} \underset{x,\ s}{\text{minimize}} \quad & x^\top Q_0 x + 2q_0^\top x + \gamma_0 \\ \text{subject to} \quad & x^\top Q_i x + 2q_i^\top x + \gamma_i + s_i^2 = 0 \quad (i = 1, \ldots, m), \end{aligned} \tag{3}$$

which can be rewritten as the following QECQP:

$$\begin{aligned} \underset{x,\ s}{\text{minimize}} \quad & \begin{bmatrix} x^\top & s^\top \end{bmatrix} \begin{bmatrix} Q_0 & O_{n\times m} \\ O_{m\times n} & O_{m\times m} \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} x + 2\begin{bmatrix} q_0^\top & \mathbf{0}_m^\top \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} + \gamma_0 \\ \text{subject to} \quad & \begin{bmatrix} x^\top & s^\top \end{bmatrix} \begin{bmatrix} Q_i & O_{n\times m} \\ O_{m\times n} & e_i e_i^\top \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} x + 2\begin{bmatrix} q_i^\top & \mathbf{0}_m^\top \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} + \gamma_i = 0 \ (i = 1, \ldots, m) \end{aligned} \tag{4}$$

where $O_{k\times\ell} \in \mathbb{R}^{k\times\ell}$ is the zero matrix, $\mathbf{0}_m \in \mathbb{R}^m$ is the zero vector, and $e_i \in \mathbb{R}^m$ is the ith column of the $m \times m$ identity matrix. This means that a practical global optimization algorithm for QECQP provides a solution method for QCQP. Solving QCQP also requires exponential computation time if m is regarded as a variable, and so various relaxation methods are studied for QCQP (see, e.g., [3]).

One of the well-known special cases of QCQP is the trust region subproblem (TRS), which has only one convex constraint with positive definite Q_1. Though TRS is a nonconvex optimization problem since Q_0 in the objective function is indefinite, its semidefinite programming (SDP) relaxation, which can be solved in polynomial-time, provides an optimal solution for the original TRS [22]. More generally, QCQP with one constraint can be solved in polynomial-time via its SDP relaxation [23].

If QCQP has two constraints with positive definite Q_1 and positive semidefinite Q_2, then it is called the Celis-Dennis-Tapia (CDT) problem, which was proposed by Celis, Dennis and Tapia [9] as a natural extension of TRS. Li and Yuan [17] proposed a Lagrangian-dual based algorithm for CDT that finds a global solution if the Hessian of Lagrangian is positive semidefinite at a global solution. As Yuan [25] proved, however, it is possible that the Hessian of the Lagrangian in the CDT problem has one negative eigenvalue at a global solution, which means that Li and Yuan's algorithm does not always find a global solution. Burer and Anstreicher [8] provided a tighter SDP relaxation by adding some constraints to the ordinary SDP relaxation, but the resulting problem can still have a relaxation gap. In general, the complexity of the CDT problem had been open for a long time until Bienstock's recent work [7] that proved its polynomial-time solvability.

Bienstock in fact proved that general QCQP with an arbitrary fixed number of quadratic constraints can be solved in polynomial time, employing a polynomial-time feasibility algorithm based on Barvinok's construction [6]. His algorithm finds an ϵ-feasible solution

with ϵ-accuracy, or a solution that satisfies relaxed constraints: $x^\top Q_i x + 2q_i^\top x + \gamma_i \leq \epsilon$, and in exact arithmetic the objective value corresponding to the solution is within ϵ from the optimal value. His algorithm is proved to run in polynomial-time with respect to the number of bits in the data and $\log \epsilon^{-1}$. Unfortunately, however, the polynomial-time feasibility algorithm looks difficult to implement, and so Bienstock's polynomial-time algorithm does not appear to be very practical.

On the other hand, recently, practical algorithms via multiparameter eigenvalues have been studied for some special cases of QCQP and QECQP. In an algorithm proposed by Iwata, Nakatsukasa and Takeda [14] for computing the signed distance between overlapping two ellipsoids, a solution method via two-parameter eigenvalues are developed for a special case of QECQP with two constraints. Their algorithm is extended for a generalized version of CDT problem in [21]. For TRS, Adachi et al. [1] proposed an algorithm employing generalized eigenvalue computation, showing that their algorithm outperforms exiting ones in accuracy and efficiency, particularly for large-sparse instances.

In this paper we derive an algorithm with multiparameter eigenvalue computation for QECQP, which is a generalization of algorithms in [1, 14, 21] in that we use an m-parameter eigenvalue problem instead of one-parameter or two-parameter as in previous studies.

Our approach is based on the one developed by Iwata, Nakatsukasa and Takeda [14], in which a special case of QECQP is solved via two-parameter eigenvalues. More specifically, we consider finding all Karush-Kuhn-Tucker (KKT) points by computing the Lagrange multipliers satisfying the KKT conditions of QECQP via multiparameter eigenvalues. The KKT conditions of QECQP are expressed as rational equations of Lagrange multipliers. We convert these rational equations into polynomial equations by constructing certain polynomial multivariate matrix pencils whose zeros of determinants are the zeros of the rational equations. This reduces the problem to a multiparameter linear eigenvalue problem, which we solve via a single-parameter linear generalized eigenvalue problem of larger size, for which reliable algorithms are available. Throughout this paper we employ the QZ algorithm [19] for solving generalized eigenvalue problems since it worked better than other algorithms in numerical experiments (considering numerical stability, it is desirable to apply GUPTRI [10] to singular generalized eigenvalue problems that we see later, however we observed in our experiments that it did not work well for some reason).

Unfortunately, unlike the algorithms in [1, 14, 21], our algorithm is not guaranteed to find a global solution for QECQP since it is difficult to deal with some singular matrix pencils that appear in our algorithm. We provide an efficient solution method for a multiparameter eigenvalue problem derived from QECQP, and thus improve the performance and stability of our algorithm. We show in the numerical experiments that our algorithm works for small-scale instances and computes global solutions with high accuracy as long as the effects of singular matrices are so small that our algorithm woks well.

This paper is organized as follows. In Section 2, we derive the KKT conditions of QECQP and express them as a multiparameter linear eigenvalue problem with certain polynomial matrix pencils, whose solutions include the Lagrange multipliers. Section 3 discusses the solution method of the multiparameter eigenvalue problem. Section 4 provides the outline and complexity analysis of our algorithm. In Section 5, we present numerical experiments to demonstrate the performance of our algorithm.

1.1 Notation

Throughout this paper, $\mathbb{R}^k$ denotes the k-dimensional real vector space and $\mathbb{R}^{k\times\ell}$ denotes the space of $k \times \ell$ real matrices. We denote the zero vector in $\mathbb{R}^k$ by $\mathbf{0}_k$, or just by 0

when the dimension is clear. Similarly, we denote the zero matrix in $\mathbb{R}^{k\times\ell}$ by $O_{k\times\ell}$, or just by O when the size is clear. The unit matrix of size k is denoted by I_k. If a matrix $X \in \mathbb{R}^{k\times\ell}$ satisfies $\operatorname{rank} X = r$, then $X^\perp \in \mathbb{R}^{k\times(\ell-r)}$ denotes a matrix whose columns form an orthogonal basis of the null space of X. For a pair of symmetric matrices X and Y, we write $X \succ Y$ if $X - Y$ is positive definite and $X \succeq Y$ if $X - Y$ is positive semidefinite. We use $X \bullet Y$ to express the Frobenius inner product of X and Y. For two matrices $X \in \mathbb{R}^{k\times\ell}$ and $Y \in \mathbb{R}^{m\times n}$, we define the Kronecker product as follows:

$$X \otimes Y := \begin{bmatrix} X_{11}Y & X_{12}Y & \cdots & X_{1l}Y \\ X_{21}Y & X_{22}Y & \cdots & X_{2l}Y \\ \vdots & \vdots & \ddots & \vdots \\ X_{k1}Y & X_{k2}Y & \cdots & X_{kl}Y \end{bmatrix}.$$

Note that the Kronecker product satisfies $(X_1 \otimes Y_1)(X_2 \otimes Y_2) = X_1X_2 \otimes Y_1Y_2$.

2 Finding the KKT points for QECQP

2.1 Assumptions and the constraint qualification

Since we cannot easily check the feasibility of a given QECQP instance, we assume that QECQP instances we deal with have at least one global solution. Furthermore, we assume some constraint qualification so that a global solution can be found among all KKT points. In summary, throughout this paper, we impose the following assumptions on QECQP:

Assumption 2.1.

i) There exists at least one global solution for QECQP.

ii) The linear independence constraint qualification (LICQ) holds.

Assumption ii) implies that the gradients of all constraint $\nabla g_i(x) = 2Q_ix + 2q_i$ $(i = 1,\dots,m)$ are linearly independent for every feasible point x, which guarantees that the KKT conditions are necessary for local optimality. There are various constraint qualifications (CQ) such as LICQ and Mangasarian-Fromovitz CQ that make the KKT conditions necessary for local optimality. The LICQ is shown to be the weakest CQ which ensures the existence and uniqueness of Lagrange multipliers (see [24]).

2.2 The KKT conditions for QECQP

The assumptions i) and ii) imply that a global solution of QECQP can be obtained by computing all KKT points. In this section, we show the KKT conditions of QECQP and discuss how to obtain all KKT points.

Let $\lambda_i \in \mathbb{R}$ $(i = 1,\dots,m)$ be Lagrange multipliers and $\lambda = (\lambda_1,\dots,\lambda_m)$ be a m-dimensional vector whose ith element is λ_i. If $x \in \mathbb{R}^n$ is a local solution of QECQP, then there exists a vector $\lambda = (\lambda_1,\dots,\lambda_m)$ satisfying the following KKT conditions:

$$H(\lambda)x = y(\lambda), \tag{5}$$

$$x^\top Q_i x + 2q_i^\top x + \gamma_i = 0 \quad (i = 1,\dots,m), \tag{6}$$

where

$$H(\lambda) := Q_0 + \lambda_1 Q_1 + \lambda_2 Q_2 + \cdots + \lambda_m Q_m \tag{7}$$

and

$$y(\lambda) := -(q_0 + \lambda_1 q_1 + \lambda_2 q_2 + \cdots + \lambda_m q_m). \tag{8}$$

Note that, once λ is obtained, the matrix $H(\lambda)$ and the vector $y(\lambda)$ are fixed. We also remark that the matrix $H(\lambda)$ is the Hessian of the Lagrangian.

2.3 Formulation as a pair of bivariate matrix equations

Since solving the equations (5)–(8) directly is difficult, we formulate a pair of matrix equations that provide appropriate Lagrange multipliers as in [1, 14, 21]; we define matrices $M_i(\lambda)$ for $i = 1, \ldots, m$ as follows:

$$M_i(\lambda) = \begin{bmatrix} Q_i & -H(\lambda) & q_i \\ -H(\lambda) & O & y(\lambda) \\ q_i^\top & y(\lambda)^\top & \gamma_i \end{bmatrix}. \tag{9}$$

Lemma 1. *For every x that satisfies the KKT conditions* (5) *and* (6) *with multipliers $\lambda_1, \ldots, \lambda_m$, we have* $\det M_i(\lambda) = 0$ $(i = 1, \ldots, m)$.

Proof. By the equation (5), $y(\lambda)$ must belong to $\operatorname{Im} H(\lambda)$. Therefore, if $H(\lambda)$ is singular, we have $\operatorname{rank} \begin{bmatrix} -H(\lambda) & y(\lambda) \end{bmatrix} < n$, which implies that $M_i(\lambda)$ is singular. Hence we have $\det M_i(\lambda) = 0$.

Now suppose that $H(\lambda)$ is nonsingular. For the computation of $\det M_i(\lambda)$, we use the Schur complement of $M_i(\lambda)$ with respect to

$$A_i := \begin{bmatrix} Q_i & -H(\lambda) \\ -H(\lambda) & O \end{bmatrix}.$$

Since

$$A_i^{-1} = \begin{bmatrix} O & -H(\lambda)^{-1} \\ -H(\lambda)^{-1} & -H(\lambda)^{-1} Q_i H(\lambda)^{-1} \end{bmatrix},$$

we have

$$\det M_i(\lambda) = (-1)^n \det H(\lambda)^2 \times \left(\gamma_i + 2 q_i^\top H(\lambda)^{-1} y(\lambda) + y(\lambda)^\top H(\lambda)^{-1} Q_i H(\lambda)^{-1} y(\lambda) \right).$$

Thus, using (5) for the above equation, we obtain

$$\det M_i(\lambda) = (-1)^n \det H(\lambda)^2 \left(x^\top Q_i x + 2 q_i^\top x + \gamma_i \right). \tag{10}$$

It then follows from (6) that $\det M_i(\lambda) = 0$. □

Lemma 1 suggests computing all possible pairs of Lagrange multipliers $\lambda_1, \ldots, \lambda_m$ for the KKT points by solving the determinantal equations

$$\det M_i(\lambda) = 0 \quad (i = 1, \ldots, m). \tag{11}$$

We will discuss how to solve (11) in Section 3.

For each Lagrange multiplier vector $\lambda = (\lambda_1, \ldots, \lambda_m)$ thus obtained, one can compute x by solving the linear equation (5) if $H(\lambda)$ is nonsingular. However, unlike the case with two constraints (i.e., $m = 2$) shown in [21], if the obtained $H(\lambda)$ is singular, it is difficult to compute x satisfying the KKT conditions. This fact is one of the reasons why we cannot theoretically prove that our algorithm finds a global solution for QECQP. One possible remedy for such cases is to perturb the original problem so that $H(\lambda)$ is nonsingular at a global solution.

3 Solving the determinantal equations

In this section, we consider solving the determinantal equations (11). We here rewrite $M_i(\lambda)$ defined by (9) as the following matrix polynomial form:

$$M_i(\lambda) = C_i + \lambda_1 D_1 + \lambda_2 D_2 + \cdots + \lambda_m D_m \tag{12}$$

where

$$C_i := \begin{bmatrix} Q_i & -Q_0 & q_i \\ -Q_0 & O & -q_0 \\ q_i^\top & -q_0^\top & 2\gamma_i \end{bmatrix},\ D_i := \begin{bmatrix} O & -Q_i & \mathbf{0}_n \\ -Q_i & O & -q_i \\ \mathbf{0}_n^\top & -q_i^\top & 0 \end{bmatrix} \quad (i = 1, \ldots, m). \tag{13}$$

By (12) we see that (11) is a multiparameter eigenvalue problem expressed as

$$\begin{aligned} \det M_1(\lambda) &= \det(C_1 + \lambda_1 D_1 + \lambda_2 D_2 + \cdots + \lambda_m D_m) = 0, \\ \det M_2(\lambda) &= \det(C_2 + \lambda_1 D_1 + \lambda_2 D_2 + \cdots + \lambda_m D_m) = 0, \\ &\vdots \\ \det M_m(\lambda) &= \det(C_m + \lambda_1 D_1 + \lambda_2 D_2 + \cdots + \lambda_m D_m) = 0. \end{aligned} \tag{14}$$

To solve this, we reduce the multiparameter eigenvalue problem (14) to univariate linear generalized eigenvalue problems based on [4, 20].

3.1 Reduction to univariate linear eigenvlaue problems

In this section, we see the solution method for the following general form of the multiparameter eigenvalue problem:

$$\begin{aligned} A_{1,0}v_1 &= \lambda_1 A_{1,1}v_1 + \cdots + \lambda_m A_{1,m}v_1, \\ A_{2,0}v_2 &= \lambda_1 A_{2,1}v_2 + \cdots + \lambda_m A_{2,m}v_2, \\ &\vdots \\ A_{m,0}v_m &= \lambda_1 A_{m,1}v_m + \cdots + \lambda_m A_{m,m}v_m \end{aligned} \tag{15}$$

where $A_{i,j} \in \mathbb{R}^{n\times n}$ for $i, j = 1, \ldots, m$. $\lambda_i \in \mathbb{R}$ and $v_i \in \mathbb{R}^n$ for $i = 1, \ldots, m$ are eigenvalues and eigenvectors respectively. As in [4, 20], we consider reducing the problem (15) to single-parameter linear generalized eigenvalue problems of larger size.

In preparation we define the operator $|\cdot|_\otimes$ as in [20]; the operator $|\cdot|_\otimes$ has m^2 matrices $A_{i,j} \in \mathbb{R}^{n\times n}$ $(i, j = 1, \ldots, m)$ as inputs and returns a matrix of size $n^m \times n^m$ as follows:

$$\begin{vmatrix} A_{1,1} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,m} \end{vmatrix}_\otimes := \sum_{\sigma\in S_m} \operatorname{sgn}(\sigma) A_{1,\sigma(1)} \otimes \cdots \otimes A_{m,\sigma(m)} \tag{16}$$

where S_m denotes the set of all permutations of $\{1, \ldots, m\}$. This operation is analogous to the determinant expansion; the elements are given by matrices $A_{i,j}$ and the product of two elements are defined by the Kronecker product. However, since the Kronecker product is not commutative, we must be careful about the order in which we calculate the Kronecker products. Note that, in each term of the right-hand side of (16), the Kronecker products of

matrices are calculated in row order. Using this operator, we define the following matrices that are sometimes called $m \times m$ operator determinants (see [20]):

$$
\begin{aligned}
\Delta_0 &:= \begin{vmatrix} A_{1,1} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,m} \end{vmatrix}_{\otimes}, \\
\Delta_i &:= \begin{vmatrix} A_{1,1} & \cdots & A_{1,i-1} & A_{1,0} & A_{1,i+1} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,i-1} & A_{m,0} & A_{m,i+1} & \cdots & A_{m,m} \end{vmatrix}_{\otimes},
\end{aligned} \tag{17}
$$

where $i = 1, \ldots, m$. The following lemma gives a connection between the multiparameter eigenvalue problem (15) and univariate linear eigenvalue problems with operator determinants (see [4, 20] for details).

Lemma 2. *Assume λ_i and v_i $(i = 1, \ldots, m)$ satisfy (15) and define $w := v_1 \otimes v_2 \otimes \cdots \otimes v_m$. Then the following equations hold:*

$$
\begin{aligned}
\Delta_1 w &= \lambda_1 \Delta_0 w, \\
\Delta_2 w &= \lambda_2 \Delta_0 w, \\
&\vdots \\
\Delta_m w &= \lambda_m \Delta_0 w.
\end{aligned}
$$

Proof. Considering the definition (16), one can easily check the following equality holds:

$$
\begin{vmatrix} A_{1,1} & \cdots & A_{1,i} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,i} & \cdots & A_{m,m} \end{vmatrix}_{\otimes} = \begin{vmatrix} A_{1,1} & \cdots & A_{1,i} + \alpha A_{1,j} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,i} + \alpha A_{m,j} & \cdots & A_{m,m} \end{vmatrix}_{\otimes}
$$

where $\forall \alpha \in \mathbb{R}$ and $i, j \in \{1, \ldots, m\}$. Using this, for $i = 1, \ldots, m$, we get

$$
\begin{aligned}
&\Delta_i - \lambda_i \Delta_0 \\
&= \begin{vmatrix} A_{1,1} & \cdots & A_{1,0} - \lambda_i A_{1,i} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,0} - \lambda_i A_{m,i} & \cdots & A_{m,m} \end{vmatrix}_{\otimes} \\
&= \begin{vmatrix} A_{1,1} & \cdots & A_{1,0} - \lambda_1 A_{1,1} - \cdots - \lambda_m A_{1,m} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,0} - \lambda_1 A_{m,1} - \cdots - \lambda_m A_{m,m} & \cdots & A_{m,m} \end{vmatrix}_{\otimes}.
\end{aligned}
$$

Therefore, we have

$$
\Delta_i w - \lambda_i \Delta_0 w = \begin{vmatrix} A_{1,1} & \cdots & A_{1,0} - \lambda_1 A_{1,1} - \cdots - \lambda_m A_{1,m} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,0} - \lambda_1 A_{m,1} - \cdots - \lambda_m A_{m,m} & \cdots & A_{m,m} \end{vmatrix}_{\otimes} (v_1 \otimes v_2 \otimes \cdots \otimes v_m).
$$

Considering the determinant expansion of the right-hand side as in (16), we see each term includes

$$
(A_{i,0} - \lambda_1 A_{i,1} - \cdots - \lambda_m A_{i,m}) v_i = 0.
$$

Thus $\Delta_i w - \lambda_i \Delta_0 w = 0$ holds for $i = 1, \ldots, m$. □

Lemma 1 suggests solving the generalized eigenvalue problem $\det(\Delta_i - \lambda_i \Delta_0) = 0$ to obtain λ_i that satisfies (15).

Here, assuming an ideal case, we observe that, once one of the m eigenvalue problems $\Delta_i w = \lambda_i \Delta_0 w$ $(i = 1, \ldots, m)$ is solved, the remaining $m - 1$ eigenvalue problems can be solved easily. For definiteness, we assume all eigenpairs (λ_1, w) satisfying $\Delta_1 w = \lambda_1 \Delta_0 w$ are obtained, and we show how to compute $\lambda_2, \ldots, \lambda_m$. Assume that the eigenvalue problems $\Delta_i w = \lambda_i \Delta_0 w$ $(i = 1, \ldots, m)$ are regular, i.e., there exists $\lambda_i \in \mathbb{R}$ such that $\det(\Delta_i - \lambda_i \Delta_0) \neq 0$. Note that this regularity assumption ensures $\Delta_0 w \neq 0$; this is because $\Delta_0 w = 0$ means $\Delta_i w = 0$ from $\Delta_i w = \lambda_i \Delta_0 w$, and thus w is a common null vector of Δ_0, Δ_i, which contradicts the regularity assumption. In this case, since $\Delta_i w = \lambda_i \Delta_0 w$ holds for $i = 1, \ldots, m$, we can compute the remaining eigenvalues $\lambda_2, \ldots, \lambda_m$ as follows:

$$\lambda_i = w^\top \Delta_i w / w^\top \Delta_0 w \quad (i = 2, \ldots, m). \tag{18}$$

However, in our problem, the matrices Δ_i $(i = 0, \ldots, m)$ derived from (14) are singular and thus the generalized eigenvalue problems $\Delta_i w = \lambda_i \Delta_0 w$ $(i = 1, \ldots, m)$ can be singular, i.e., infinitely many $\lambda_i \in \mathbb{R}$ satisfy $\Delta_i w = \lambda_i \Delta_0 w$ for some w, which makes it difficult to compute appropriate eigenvalues $\lambda_1, \ldots, \lambda_m$ for solving (14). In the following section, we discuss how to deal with the singularity that arises in our problem.

3.2 Removing the common null space of the matrices Δ_i

The multiparameter eigenvalue problem (14) can be expressed as follows with eigenvectors v_i $(i = 1, \ldots, m)$:

$$\begin{aligned} -C_1 v_1 &= \lambda_1 D_1 v_1 + \cdots + \lambda_m D_m v_1, \\ -C_2 v_2 &= \lambda_1 D_1 v_2 + \cdots + \lambda_m D_m v_2, \\ &\vdots \\ -C_m v_m &= \lambda_1 D_1 v_m + \cdots + \lambda_m D_m v_m. \end{aligned} \tag{19}$$

For this problem, we construct the following $m \times m$ operator determinants Δ_i $(i = 0, \ldots, m)$ according to the definition (17):

$$\begin{aligned} \Delta_0 &:= \begin{vmatrix} D_1 & \cdots & D_m \\ \vdots & \ddots & \vdots \\ D_1 & \cdots & D_m \end{vmatrix}_{\otimes}, \\ \Delta_i &:= \begin{vmatrix} D_1 & \cdots & D_{i-1} & -C_1 & D_{i+1} & \cdots & D_m \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ D_1 & \cdots & D_{i-1} & -C_m & D_{i+1} & \cdots & D_m \end{vmatrix}_{\otimes} \quad (i = 1, \ldots, m). \end{aligned} \tag{20}$$

By Lemma 2, for the eigenvalues $\lambda_1, \ldots, \lambda_m$ and eigenvectors $v_1, \ldots, v_m$ satisfying (19), we have

$$\Delta_i w = \lambda_i \Delta_0 w \quad (i = 1, \ldots, m) \tag{21}$$

where $w := v_1 \otimes v_2 \otimes \cdots \otimes v_m$. Therefore, if we obtain all pair of solutions $\lambda_1, \ldots, \lambda_m$ for the generalized eigenvalue problems (21), we can find the solutions for (19) among them.

Unfortunately, however, the matrix pencils $\Delta_i - \lambda_i \Delta_0$ $(i = 1, \ldots, m)$ derived from (20) are singular, which means infinitely many λ_i satisfy $\det(\Delta_i - \lambda_i \Delta_0) = 0$. To make matters

worse, the null space of $\Delta_i - \lambda_i \Delta_0$ cannot be described explicitly; actually some null vectors can depend on the value of λ_i, and thus it is hard to remove the null space completely. Therefore, in what follows, we consider removing some parts of the common null space of $\Delta_i - \lambda_i \Delta_0$ $(i = 1, \ldots, m)$ whose basis can be computed relatively easily. More specifically, we describe a projection procedure that removes the common null space of Δ_i $(i = 0, \ldots, m)$, which is always included in the common null space of $\Delta_i - \lambda_i \Delta_0$ $(i = 1, \ldots, m)$. By this procedure the numerical stability of the eigensolver is expected to improve and the size of matrices Δ_i $(i = 0, \ldots, m)$ can be reduced.

We first compute $\Delta_0^{\perp}$, an orthogonal basis of the null space of Δ_0, via the singular value decomposition (SVD) as follows. Let $\Delta_0 = U \Sigma V$ be the SVD of Δ_0. Then $\Delta_0^{\perp}$ is given by the columns of V that correspond to the zero singular values, which we denote by V_0 in what follows.

We then consider computing an orthogonal basis of the common null space of $\Delta_0, \ldots, \Delta_m$ sequentially. Let V_i be an orthogonal basis of the common null of $\Delta_0, \ldots, \Delta_i$. We now consider how to compute V_{i+1} given V_i. Since V_i forms the orthogonal basis of the common null space of $\Delta_0, \ldots, \Delta_i$, there exists some matrix X such that $V_{i+1} = V_i X$, i.e., each column of V_{i+1} can be described as a linear combination of the columns of V_i. Now our purpose is finding the matrix X such that $V_{i+1} = V_i X$, which can be obtained by finding the matrix X such that $\Delta_{i+1} V_i X = 0$. We here compute the matrix X as $(\Delta_{i+1} V_i)^{\perp}$ via SVD as shown above replacing Δ_0 by $\Delta_{i+1} V_i$. Since the columns of X obtained via SVD are orthogonal, the columns of $V_{i+1} = V_i X$ form an orthogonal basis of the common null space of $\Delta_0, \ldots, \Delta_{i+1}$.

Considering the above, the common null space of Δ_i $(i = 0, \ldots, m)$ can be obtained as V_m by Algorithm 3.1.

Alternatively the common null space V_m can be obtained by computing the SVD

$$\begin{bmatrix} \Delta_0 \\ \vdots \\ \Delta_m \end{bmatrix} = U \Sigma V \tag{22}$$

and extracting the columns of V that correspond to the zero singular values. However, the matrix in left-hand side of (22) tends to be too large in practice. Therefore we employ the above sequential method for computing V_m to save memory space.

Algorithm 3.1 Finding the common null space of $\Delta_0, \ldots, \Delta_m$.

1: Compute SVD of Δ_0 to get V_0.
2: **for** $i = 1$ to m **do**
3: Compute $X = (\Delta_i V_{i-1})^{\perp}$ using SVD and let $V_i = V_{i-1} X$.
4: **end for**
5: Return V_m.

Finally, we get the projected Δ_i $(i = 0, \ldots, m)$ as

$$\widetilde{\Delta}_i := (V_m{}^{\perp})^{\top} \Delta_i V_m{}^{\perp}. \tag{23}$$

Note that, as a consequence of the above projection procedure, we may project out some parts of null space that are necessary for finding the Lagrange multipliers. This means the solution method for QECQP presented here may fail to find a global solution, and this is one of the reasons why our algorithm is not theoretically guaranteed to find a global solution.

3.3 Solving the generalized eigenvalue problems with projected matrices

We discuss how to compute the eigenvalues satisfying the following eigenvalue problems:

$$\widetilde{\Delta}_i \widetilde{w}_i = \lambda_i \widetilde{\Delta}_0 \widetilde{w}_i \quad (i = 1, \ldots, m) \tag{24}$$

where $\widetilde{\Delta}_i$ is as defined in (23) and $\widetilde{w}_i$ $(i = 1, \ldots, m)$ are eigenvectors.

First, as shown for the eigenvalue problems (21), we observe that the eigenvalue problems (24) have common eigenvectors, i.e., there exists eigenvectors of (24) whose entries are independent of $i = 1, \ldots, m$. Let v_i $(i = 1, \ldots, m)$ be eigenvectors satisfying (19) and $w := v_1 \otimes v_2 \otimes \cdots \otimes v_m$. Note that w satisfies $\Delta_i w = \lambda_i \Delta_0 w$ for $i = 1, \ldots, m$. Furthermore, this vector w can be expressed with some vectors α, β as

$$w = {V_m}^{\perp} \alpha + V_m \beta$$

where V_m is the output of Algorithm 3.1, i.e., the orthogonal basis of the common null space of Δ_i $(i = 0, \ldots, m)$. Therefore we have

$$\begin{aligned} \Delta_i w = \lambda_i \Delta_0 w &\Leftrightarrow \Delta_i({V_m}^{\perp}\alpha + V_m\beta) = \lambda_i \Delta_0({V_m}^{\perp}\alpha + V_m\beta) \\ &\Leftrightarrow \Delta_i {V_m}^{\perp}\alpha = \lambda_i \Delta_0 {V_m}^{\perp}\alpha \\ &\Rightarrow \widetilde{\Delta}_i \alpha = \lambda_i \widetilde{\Delta}_0 \alpha \end{aligned}$$

for $i = 1, \ldots, m$. This means the generalized eigenvalue problems (24) have the common eigenvector α for $i = 1, \ldots, m$. Consequently the problems (24) can be written as follows:

$$\widetilde{\Delta}_i \widetilde{w} = \lambda_i \widetilde{\Delta}_0 \widetilde{w} \quad (i = 1, \ldots, m) \tag{25}$$

where $\widetilde{w}$ is the common eigenvector of reduced size.

We now consider solving (25). We first consider computing λ_1 satisfying $\widetilde{\Delta}_1 \widetilde{w} = \lambda_1 \widetilde{\Delta}_0 \widetilde{w}$ and then we see how to compute the remaining eigenvalues $\lambda_2, \ldots, \lambda_m$.

Although $\widetilde{\Delta}_i$ $(i = 0, \ldots, m)$ have no common null space due to the projection procedure in Section 3.2, for the particular eigenvalue problem $\widetilde{\Delta}_1 \widetilde{w} = \lambda_1 \widetilde{\Delta}_0 \widetilde{w}$, there may still exist common null space of $\widetilde{\Delta}_0, \widetilde{\Delta}_1$. Therefore, before computing λ_1, we do the following projection procedure to obtain matrices $\widehat{\Delta}_0, \widehat{\Delta}_1$ that have no common null space; we apply Algorithm 3.1 to $\widetilde{\Delta}_0, \widetilde{\Delta}_1$ to obtain their common null space $\widetilde{V_1}$, and then we project out the null space as follows:

$$\widehat{\Delta}_0 = (\widetilde{V_1}^{\perp})^{\top} \widetilde{\Delta}_0 \widetilde{V_1}^{\perp}, \ \widehat{\Delta}_1 = (\widetilde{V_1}^{\perp})^{\top} \widetilde{\Delta}_1 \widetilde{V_1}^{\perp}.$$

Now we compute the eigenvalue λ_1 by solving the generalized eigenvalue problem

$$\widehat{\Delta}_1 \widehat{w}_1 = \lambda_1 \widehat{\Delta}_0 \widehat{w}_1 \tag{26}$$

where $\widehat{w}_1$ is the eigenvector. A standard algorithm for solving the generalized algorithm is the QZ algorithm, which we used in our experiments.

We then compute the remaining eigenvalues $\lambda_2, \ldots, \lambda_m$ that correspond to the value of λ_1. More precisely, for each value of λ_1 obtained by solving (26), we show how to compute $\lambda_2, \ldots, \lambda_m$ that satisfy (25) with a common eigenvector $\widetilde{w}$.

We first consider expressing the vector $\widetilde{w}$ in (25) with some vectors r, s as follows:

$$\widetilde{w} = \widetilde{V_1}^{\perp} r + \widetilde{V_1} s \tag{27}$$

where $\widetilde{V_1}$ is an orthogonal basis of the common null space of $\widetilde{\Delta}_0, \widetilde{\Delta}_1$. Note that the vector r can be obtained as the eigenvector $\widehat{w}_1$ in (26), i.e., $r = \widehat{w}_1$, since

$$\begin{aligned}\widetilde{\Delta}_1\widetilde{w} = \lambda_1\widetilde{\Delta}_0\widetilde{w} &\Leftrightarrow \widetilde{\Delta}_1(\widetilde{V_1}^{\perp}r + \widetilde{V_1}s) = \lambda_1\widetilde{\Delta}_0(\widetilde{V_1}^{\perp}r + \widetilde{V_1}s)\\ &\Leftrightarrow \widetilde{\Delta}_1\widetilde{V_1}^{\perp}r = \lambda_1\widetilde{\Delta}_0\widetilde{V_1}^{\perp}r\\ &\Rightarrow \widehat{\Delta}_1 r = \lambda_1\widehat{\Delta}_0 r.\end{aligned}$$

Now our purpose is to find λ_i with some vector s such that λ_i and $\widetilde{w} = \widetilde{V_1}^{\perp}\widehat{w}_1 + \widetilde{V_1}s$ satisfy (25) for $i = 2, \ldots, m$. Plugging $\widetilde{w} = \widetilde{V_1}^{\perp}\widehat{w}_1 + \widetilde{V_1}s$ into (25), we have

$$\begin{aligned}&\widetilde{\Delta}_i(\widetilde{V_1}^{\perp}\widehat{w}_1 + \widetilde{V_1}s) = \lambda_i\widetilde{\Delta}_0(\widetilde{V_1}^{\perp}\widehat{w}_1 + \widetilde{V_1}s)\\ &\Leftrightarrow \widetilde{\Delta}_i\widetilde{V_1}^{\perp}\widehat{w}_1 + \widetilde{\Delta}_i\widetilde{V_1}s = \lambda_i\widetilde{\Delta}_0\widetilde{V_1}^{\perp}\widehat{w}_1\\ &\Leftrightarrow \begin{bmatrix}\widetilde{\Delta}_0\widetilde{V_1}^{\perp}\widehat{w}_1 & -\widetilde{\Delta}_i\widetilde{V_1}\end{bmatrix}\begin{bmatrix}\lambda_i\\ s\end{bmatrix} = \widetilde{\Delta}_i\widetilde{V_1}^{\perp}\widehat{w}_1.\end{aligned}$$

Therefore, by solving the linear equations

$$\begin{bmatrix}\widetilde{\Delta}_0\widetilde{V_1}^{\perp}\widehat{w}_1 & -\widetilde{\Delta}_i\widetilde{V_1}\end{bmatrix}\begin{bmatrix}\lambda_i\\ s\end{bmatrix} = \Delta_i\widetilde{V_1}^{\perp}\widehat{w}_1 \quad (i = 2, \ldots, m) \tag{28}$$

for $\begin{bmatrix}\lambda_i & s\end{bmatrix}^{\top}$, we get λ_i $(i = 2, \ldots, m)$ as the first elements of the solutions.

We remark that our algorithm may fail to find appropriate λ_1 if the generalized eigenvalue problem (26) is singular. Moreover, if the matrix in the right-hand side of (28) is singular, it is hard to compute appropriate λ_i $(i = 2, \ldots, m)$. These issues make it difficult to guarantee that our algorithm finds a global solution. In practice these problems appear to arise often, although our algorithm still can find a global solution in most cases as shown in Section 5.

4 Summary and analysis of the algorithm

In this section, we summarize the entire algorithm for solving QECQP and give complexity analysis.

4.1 Outline of the algorithm

We here show the pseudocode for the whole algorithm for solving QECQP.

Algorithm 4.1 Outline of algorithm for solving QECQP.

1: Construct the operator determinants Δ_i $(i = 0, \ldots, m)$ according to (20).
2: Remove the common null space of Δ_i $(i = 0, \ldots, m)$ to get $\widetilde{\Delta}_i$ $(i = 0, \ldots, m)$.
3: Remove the common null space $\widetilde{\Delta}_0, \widetilde{\Delta}_1$ to get $\widehat{\Delta}_0, \widehat{\Delta}_1$.
4: Solve $\widehat{\Delta}_1\widehat{w}_1 = \lambda_1\widehat{\Delta}_0\widehat{w}_1$ to get eigenpairs $(\lambda_1, \widehat{w}_1)$ using QZ.
5: For each eigenpairs $(\lambda_1, \widehat{w}_1)$, solve (28) to get $\lambda_2, \ldots, \lambda_m$.
6: For every $\lambda = (\lambda_1, \ldots, \lambda_m)$ thus obtained, solve (5) to get x.
7: For every x thus obtained, check the feasibility (6) and rule out infeasible x.
8: For every feasible x thus obtained, compute the objective function values. The vector x corresponding to the smallest value is a global solution.

4.2 Complexity analysis

We here examine the computational costs of some major parts in Algorithm 4.1.

First, in Step 2, the projection procedure for Δ_i $(i = 0, \ldots, m)$ requires $O((m+1)(2n+1)^{3m})$ computational cost. This is because the complexity of SVD for a $M \times N$ matrix is $O(MN^2 + N^3)$ [12, §8.6] and the size of matrices Δ_i $(i = 0, \ldots, m)$ is bounded by $(2n+1)^m \times (2n+1)^m$.

In Step 4 the algorithm requires eigenpairs of the linear generalized eigenvalue problem $\widehat{\Delta}_1 \widehat{w}_1 = \lambda_1 \widehat{\Delta}_0 \widehat{w}_1$, whose size is bounded by $(2n+1)^m$. Since the standard QZ algorithm for computing the eigenpairs of an $N \times N$ linear generalized eigenvalue problem requires $O(N^3)$ floating point operations [12, §7.7.7], the computational cost for solving $\widehat{\Delta}_1 \widehat{w}_1 = \lambda_1 \widehat{\Delta}_0 \widehat{w}_1$ is $O((2n+1)^{3m})$ flops.

Furthermore, assuming that the generalized eigenvalue problem $\widehat{\Delta}_1 \widehat{w}_1 = \lambda_1 \widehat{\Delta}_0 \widehat{w}_1$ is regular, the number of eigenpairs $(\lambda_1, \widehat{w}_1)$ obtained in Step 4 is at most $(2n+1)^m$. Since the computational cost for solving (28) for $\lambda_2, \ldots, \lambda_m$ is bounded by $O((m-1)(2n+2)^{3m})$, the total complexity of Step 5 is $O((m-1)(2n+2)^{4m})$, which is the dominant cost in our algorithm.

Note that, if m is fixed, the complexity of our algorithm is polynomial time, which is consistent with the result shown by Bienstock [7].

5 Numerical experiments

In this section, we present numerical experiments to show the performance of our algorithm for finding a global solution of QECQP. Since the performance of similar algorithms for quadratic programming with two constraints has already studied in [14, 21], we implemented our algorithm for QECQP with three constraints. To compare with our algorithm, we also consider solving QECQP (1) via the following ordinary SDP relaxation:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & Q_0 \bullet X + 2q_0^\top x + \gamma_0 \\ \text{subject to} & Q_i \bullet X + 2q_i^\top x + \gamma_i = 0 \quad (i = 1, \ldots, m), \\ & X - xx^\top \succeq O. \end{array} \tag{29}$$

We compare these two methods regarding the runtime, the number of solved instances and the accuracy of computed solutions. All experiments were conducted in MATLAB R2014b on a Core i7 machine with 16GB RAM. We solved SDP (29) by SeDuMi 1.3, fixing its desired accuracy (pars.eps) to 0 so that it computes solutions as accurately as possible.

In this experiment we consider QECQP instances given by the following form so that we can easily check whether a computed solutions is optimal or not:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) = (x - x^*)^\top Q_0 (x - x^*) \\ \text{subject to} & g_i(x) = (x - x^*)^\top Q_i (x - x^*) = 0 \quad (i = 1, 2, 3) \end{array} \tag{30}$$

where $x^* \in \mathbb{R}^n$ is a given vector. We generated x^* and Q_i $(i = 1, 2, 3)$ randomly. Q_0 is randomly generated to satisfy $Q_0 \succ 0$. Apparently $x = x^*$ is a feasible solution and $f(x^*) = 0$ is the smallest objective value, which means x^* is a global solution for (30).

For $n = 4, 5$ we thus generated 100 instances, and solved them by our algorithm and by the SDP relaxation method. We measured the runtime of our algorithm and the SDP relaxation method. Table 1 shows the runtime of both algorithms by the average and standard deviation over 100 instances for each dimension.

We then counted the number of instances solved by our algorithm and by the SDP relaxation method respectively. We regard a problem is solved if the computed solution x satisfies $\|x - x^*\|_2/\|x^*\|_2 < 10^{-3}$. Table 2 indicates the number instances solved by our algorithm and by the SDP relaxation for each dimension.

Finally, we compared the accuracy of solutions computed by both algorithms. Table 3 indicates the average and standard deviation of the relative error $\|x - x^*\|_2/\|x^*\|_2$ over all solved instances for $n = 4, 5$. Note that the instances our algorithm failed to solve were ruled out on calculating its accuracy.

We observe that the SDP relaxation method outperforms our algorithm regarding the runtime and the number of solved instances. However, for the instances where our algorithm works well, it computes global solutions much more accurately than the SDP relaxation method.

Table 1: Runtime (s) of our algorithm and the SDP relaxation method for $n = 4, 5$. The results are shown by average ($\pm$ standard deviation) over 100 instances.

	Our algo.	SDP
$n = 4$	11.4 ($\pm$1.23)	0.106 ($\pm 3.41 \times 10^{-2}$)
$n = 5$	158 ($\pm$14.5)	0.109 ($\pm 2.89 \times 10^{-2}$)

Table 2: Number of instances solved by our algorithm and SDP for $n = 4, 5$.

	Our algo.	SDP
$n = 4$	98	100
$n = 5$	97	100

Table 3: Rerative error $\|x - x^*\|_2/\|x^*\|_2$ of solutions x computed by our algorithm and SDP for $n = 4, 5$. The results are shown by average ($\pm$ standard deviation) over solved instances.

	Our algo.	SDP
$n = 4$	$1.19 \times 10^{-15} (\pm 4.29 \times 10^{-15})$	$3.53 \times 10^{-6} (\pm 6.15 \times 10^{-6})$
$n = 5$	$3.63 \times 10^{-15} (\pm 1.39 \times 10^{-14})$	$2.27 \times 10^{-6} (\pm 2.82 \times 10^{-6})$

Remarks on the duality gap For the SDP relaxation method, we confirmed that the duality gap is bounded by $[2 \times 10^{-14}, 3 \times 10^{-6}]$ in all instances tested above, implying that all generated instances are relatively easy for the SDP relaxation method to solve. Probably, by the positive definiteness of Q_0, the Hessian of Lagrangian tends to be positive semidefinite at a global solution; in the CDT problem, it is known to be a necessary and sufficient condition for a given instance to have no duality gap [2], and it seems that similar situation is happening in the case of QECQP.

As one can see from the derivation of our algorithm, the presence of duality gap does not affect the performance of our algorithm, while the singularity of matrices is problematic as mentioned later. Therefore, our algorithm can be effective for QECQP instances with

nonsingular matrices and having duality gap, which is hard to solve for the SDP relaxation method because of the duality gap.

Remarks on the instances with singular matirces We also observed via experiments that our algorithm often fail to solve QECQP instances (30) with low-rank matrices Q_i, e.g., diagonal matrices with only one or two nonzero entries. This is probably because low-rank matrices Q_i spawn singular matrices in many parts of our algorithm (e.g., the generalized eigenvalue problem (26) and the linear equation (28)), and thus causes numerical difficulties in our algorithm.

6 Conclusion and discussion

In this paper we have developed an algorithm for QECQP. Our algorithm solves QECQP as follows: find all Lagrange multipliers by solving a system of multivariate determinantal equations, compute the KKT points corresponding to the multipliers and then obtain a global solution with the smallest objective value among the KKT points. The key step of our algorithm is to convert the KKT conditions into the multiparameter eigenvalue problem, which is reduced to the m linear generalized eigenvalue problems of larger size.

We also proposed a projection method that removes the common null space of the operator determinants to improve the performance and stability of the eigenvalue computation. We also showed that, once one of the generalized eigenvalue problems is solved, the remaining eigenvalues can be obtained without solving the large-size generalized eigenvalue problems. Although our algorithm is not theoretically guaranteed to find a global solution of QECQP, the numerical experiments show that, for QECQP instances where our algorithm works well, it computes a global solution with high accuracy.

A possible future work is to prove that our algorithm finds a global solution for QECQP by modifying some procedures of our algorithm or imposing some assumptions on QECQP instances. One promising approach is to generalize Theorem 1.1 of [16], which shows the connection between two-parameter eigenvalue problems and Bézoutian (a counterpart of the operator determinant for the two-parameter case), so that we can clarify the relationship between the multiparameter eigenvalue problem and the operator determinant. It is also important to study how the presence of singular matrices and the projection process affect the solvability of the original QECQP instance. Additionally, considering the numerical stability, it is desirable if we can apply GUPTRI [10] for solving singular generalized eigenvalue problems that appear in our algorithm. Although our straightforward implementation of replacing QZ with GUPTRI resulted in failure to find a solution for some reason, it is possible that GUPTRI would work better than QZ with some appropriate modification.

Acknowledgements

I would like to express my sincere gratitude to Yuji Nakatsuaksa for his insightful suggestions and invaluable support. I would also like to thank Stefan Johansson for providing the source code of GUPTRI. My thanks also goes to Bor Plestenjak for comments on multiparameter eigenvalue problems.

References

[1] S. Adachi, S. Iwata, Y. Nakatsukasa, and A. Takeda. Solving the trust region subproblem by a generalized eigenvalue problem. *METR 2015-14, University of Tokyo, April 2015.*

[2] W. Ai and S. Zhang. Strong duality for the CDT subproblem: a necessary and sufficient condition. *SIAM J. Optim.*, 19(4):1735–1756, 2009.

[3] K. Anstreicher. On convex relaxations for quadratically constrained quadratic programming. *Math. Program.*, 136(2):233–251, 2012.

[4] F. V. Atkinson. *Multiparameter Eigenvalue Problems.* Academic Press, New York-London, 1972. Volume I: Matrices and compact operators, Mathematics in Science and Engineering, Vol. 82.

[5] F. V. Atkinson and A. B. Mingarelli. *Multiparameter Eigenvalue Problems: Sturm-Liouville Theory.* CRC Press, Boca Raton, FL, 2011.

[6] A. I. Barvinok. Feasibility testing for systems of real quadratic equations. *Discrete Comput. Geom.*, 10:1–13, 1993.

[7] D. Bienstock. A note on polynomial solvability of the CDT problem. *ArXiv 1406.6429*, June 2014.

[8] S. Burer and K. Anstreicher. Second-order-cone constraints for extended trust-region subproblems. *SIAM J. Optim.*, 23(1):432–451, 2013.

[9] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization*, pages 71–82. SIAM Philadelphia, 1985.

[10] J. Demmel and B. Kågström. The generalized schur decomposition of an arbitrary pencil $A - \lambda B$: robust software with error bounds and applications. part I: theory and algorithms. *ACM Trans. Math. Soft.*, 19(2):160–174, 1993.

[11] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[12] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 4th edition, 2012.

[13] M. E. Hochstenbach and B. Plestenjak. Backward error, condition numbers, and pseudospectra for the multiparameter eigenvalue problem. *Linear Algebra Appl.*, 375:63–81, 2003.

[14] S. Iwata, Y. Nakatsukasa, and A. Takeda. Computing the signed distance between overlapping ellipsoids. *SIAM J. Optim.*, 25(4):2359–2384, 2015.

[15] E. Jarlebring and M. E. Hochstenbach. Polynomial two-parameter eigenvalue problems and matrix pencil methods for stability of delay-differential equations. *Linear Algebra Appl.*, 431(3):369–380, 2009.

[16] L. Lerer and M. Tismenetsky. The Bézoutian and the eigenvalue-separation problem for matrix polynomials. *Integral Equations and Operator Theory*, 5(1):386–445, 1982.

[17] G. Li and Y. Yuan. Compute a Celis-Dennis-Tapia step. *J. Comput. Math.*, 23:463–478, 2005.

[18] M. Mevissen and M. Kojima. SDP relaxations for quadratic optimization problems derived from polynomial optimization problems. *Asia-Pac. J. Oper. Res.*, 27(01):15–38, 2010.

[19] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM J. Numer. Anal.*, 10(2):241–256, 1973.

[20] B. Plestenjak, C. I. Gheorghiu, and M. E. Hochstenbach. Spectral collocation for multiparameter eigenvalue problems arising from separable boundary value problems. *J. Comput. Phys.*, 298:585–601, 2015.

[21] S. Sakaue, Y. Nakatsukasa, A. Takeda, and S. Iwata. A polynomial-time algorithm for nonconvex quadratic optimization with two quadratic constraints. *METR 2015-03, University of Tokyo, January 2015.*

[22] R. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optim.*, 5(2):286–313, 1995.

[23] J. F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Math. Oper. Res.*, 28(2):246–267, 2003.

[24] G. Wachsmuth. On LICQ and the uniqueness of Lagrange multipliers. *Oper. Res. Lett.*, 41(1):78–80, 2013.

[25] Y. Yuan. On a subproblem of trust region algorithms for constrained optimization. *Math. Program.*, 47:53–63, 1990.