Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 20, No. 4, pp. 667–682, 2024

A NEW GLOBAL ALGORITHM FOR HOMOGENEOUS COMPLEX QUADRATIC PROGRAMMING PROBLEMS AND APPLICATIONS

Yingzhe Xu*, Jintao Xu†, Cheng Lu, Shu-Cherng Fang‡ and Zhibin Deng§

Dedicated to Prof. Masao Fukushima for celebrating his 75th birthday

Abstract: In this paper, we propose a new semidefinite relaxation-based branch-and-bound algorithm for a class of complex quadratic programming (CQP) problems with nonconvex constraints on modulus and phase difference. We derive a sufficient condition for the tightness of the semidefinite relaxation and estimate the gap between the nonconvex constraints and their convex hulls. The numerical results show that the proposed algorithm outperforms the well-known commercial solver Gurobi and an existing global algorithm based on semidefinite relaxation that utilizes the structure of phase constraints when applied to the typical applications of CQP problems, such as discrete and virtual beamforming problems.

Key words: *semidefinite relaxation, branch-and-bound algorithm, complex quadratic programming, beamformer design*

Mathematics Subject Classification: *90C20, 90C22, 90C57, 90C90*

---

1 Introduction

In this paper, we consider the following complex quadratic programming problems:

$$\begin{aligned} \min_{x} \quad & x^\dagger Q_0 x, \\ \text{s.t.} \quad & x^\dagger Q_i x \le b_i,\ i = 1,2,\ldots,m, \\ & l_i \le |x_i| \le u_i,\ i = 1,2,\ldots,n, \\ & \arg\left(x_i x_j^\dagger\right) \in \mathcal{A}_{ij},\ (i,j) \in \mathcal{E}, \end{aligned} \tag{CQP}$$

where $x \in \mathbb{C}^n$ is decision variable, $Q_0, Q_1, \ldots, Q_m \in \mathbb{C}^{n\times n}$ are Hermitian matrices, $b_1, b_2 \ldots, b_m \in \mathbb{R}$ are real scalars, l_i and u_i are the lower and upper bounds of the modulus on

*Yingzhe Xu and Cheng Lu's research was supported by the National Natural Science Foundation of China Grant No. 12171151.

†Jintao Xu's research was supported by the National Natural Science Foundation of China Grant No. 11771243.

‡Corresponding author

§Zhibin Deng's research was supported by the National Natural Science Foundation of China Grant No. T2293774, Fundamental Research Funds for the Central University No. E2ET0808X2, and a grant from MOE Social Science Laboratory of Digital Economic Forecasts and Policy Simulation at UCAS.

 DOI: https://doi.org/10.61208/pjo-2024-001

the decision variable $x_i \in \mathbb{C}$ for $i = 1, 2, \ldots, n$, respectively, set $\mathcal{E} \subseteq \{(i,j) | 1 \leq i < j \leq n\}$, and $\mathcal{A}_{ij}$ with $(i,j) \in \mathcal{E}$ can be either an interval of the form $\left[\underline{\theta}_{ij}, \overline{\theta}_{ij}\right]$ or a discrete set of the form $\left\{\theta_{ij}^1, \theta_{ij}^2 \ldots, \theta_{ij}^M\right\}$. We use $\arg(\cdot)$ to denote the phase of a complex number and $(\cdot)^\dagger$ to denote the conjugate transpose of a matrix or vector. The constraint $\arg\left(x_i x_j^\dagger\right) \in \mathcal{A}_{ij}$ restricts the phase difference between the complex variables x_i and x_j in $\mathcal{A}_{ij}$. This constraint arises naturally in the field of transmit beamformer design [1, 4] and optimal power flow [7, 8]. For example, according to the beamforming design principle, the phase difference between signals emitted from the antennas is required to be within certain angles to acquire constructive interference. Similarly, in the optimal power flow problem, the voltage phase difference between buses is restricted in a given range for stable and secure operation of the power systems. Moreover, other quadratic programming problems, such as the multiple-input multiple-output (MIMO) detection problem [11, 14], radar phase code design[16], angular synchronization problem [2] and the Max-3-Cut problem [5, 19], can be formulated as special cases of problem (CQP). Therefore, problem (CQP) is very general and has wide applications in various fields.

Problem (CQP) is NP-hard in general, since the Max-3-Cut problem as its special case is already known as an NP-hard problem [19]. Therefore, most of the algorithms for solving (CQP) or its special cases in the literature are suboptimal, including approximate algorithms [16, 22] and local algorithms [23]. Among these suboptimal algorithms, the approximate algorithms based on semidefinite relaxations have attracted great attention. One can refer to [15] for a survey on the applications of semidefinite relaxations in signal processing, and [18] for the applications in other areas.

On the other hand, some researchers have proposed global algorithms to solve problem (CQP). Here, we mention some works closely related to our paper. In [3], Chen et al. proposed a branch-and-bound algorithm to solve (CQP), where the set $\mathcal{A}_{ij}$ is a continuous interval $[\underline{\theta}_{ij}, \overline{\theta}_{ij}]$ with $0 < \overline{\theta}_{ij} - \underline{\theta}_{ij} < \pi$. In this paper, we remove this restriction, allowing the set $\mathcal{A}_{ij}$ to be a discrete set or a continuous interval with a range larger than π. The other two related works are [10] and [12], in which Lu et al. proposed a method to derive new semidefinite relaxations for two classes of complex quadratic programming problems. The main idea of their method is to derive new valid inequalities by representing variables in the polar coordinate form and exploring relations between the radius and phases of variables. However, when the phase difference constraints are presented in problem (CQP), the semidefinite relaxations in [10] and [12] are not directly applicable. Recently, Xu et al. [21] followed the method of [10] and [12] and proposed a new semidefinite relaxation for problem (CQP). We adopt a cutting-edge semidefinite relaxation in this paper. In addition, we derive sufficient conditions for the tightness of this relaxation and estimate the relaxation gap between the relaxation and the original problem. The tightness condition and the gap estimation are not done in the existing work [21].

Moreover, we propose a branch-and-bound algorithm to globally solve problem (CQP) with general nonconvex constraints on variable modulus and phase difference. Specifically, the phase difference of two decision variables is allowed to lie in a discrete or continuous set with a range larger than π. Such cases cannot be handled by other global algorithms in the literature. The main feature of the proposed branch-and-bound algorithm is that some complex variables are selected with their bounds on modules or phase differences partitioned in the branching procedure. Numerical experiments on two types of beamformer design problems are reported to verify the efficiency of the proposed algorithm.

The rest of this paper is organized as follows. In Section 2, we review the semidefinite relaxation in [21] and derive the sufficient conditions for the tightness of the relaxation. Then the gap between the feasible regions of the relaxation and the original problem is estimated. In Section 3, we propose a branch-and-bound algorithm based on the semidefinite relaxation to solve problem (CQP) globally, and discuss the relationship between the proposed branch-and-bound algorithm and the existing one in [12]. In Section 4, we apply the proposed algorithm to solve two application problems of (CQP) numerically and compare its performance with other algorithms. Conclusions are given in Section 5.

The following notations are adopted throughout the paper. For a given complex matrix $X \in \mathbb{C}^{n\times n}$, $\mathrm{Re}(X)$ and $\mathrm{Im}(X)$ denote its componentwise real and imaginary parts, respectively, and $X^\dagger$ denotes the conjugate transpose of X. For a given Hermitian matrix $A \in \mathbb{C}^{n\times n}$, $A \succeq 0$ means that A is positive semidefinite. For given two Hermitian matrices A and B, $A \succeq B$ means $A - B \succeq 0$. Moreover, $\mathrm{Trace}(A)$ denotes the trace of A, and $A \cdot B$ denotes $\mathrm{Trace}(A^\dagger B)$. For a set $\mathcal{S}$ in some vector space, we use $\mathrm{Conv}(\mathcal{S})$ to represent the convex hull of $\mathcal{S}$. Besides, with a slight abuse of notations, for a nonzero complex variable z and a set $\mathcal{A} \subset \mathbb{R}$, the notation $\arg(z) \in \mathcal{A}$ means that there exists a $k \in \mathbb{Z}$ such that $\arg(z) + 2k\pi$ is contained in $\mathcal{A}$. Besides, since $\arg(x_i x_j^\dagger)$ is not defined when $x_i x_j^\dagger = 0$, we always regard the constraint $\arg(x_i x_j^\dagger) \in \mathcal{A}_{ij}$ as being satisfied if $x_i x_j^\dagger = 0$ and $\mathcal{A}_{ij}$ is nonempty. For a real number x, $\lfloor x \rfloor$ is the greatest integer less than or equal to x, and $\lceil x \rceil$ is the least integer greater than or equal to x. $\mathrm{diag}(X)$ is defined as the vector consisting of the diagonal elements of the matrix X.

2 An Enhanced Semidefinite Relaxation

In this section, we review a semidefinite relaxation for problem (CQP) which has been proposed in [21] and give a sufficient condition for the tightness of the relaxation. Then we derive some inequalities to estimate the gap when the relaxation is not tight.

2.1 A semidefinite relaxation for (CQP) in [21]

For completeness of this paper, we briefly review how to obtain a positive semidefinite relaxation of problem (CQP). We refer the readers to [21] for more details.

By introducing an $n \times n$ complex Hermitian matrix $X := xx^\dagger$, problem (CQP) can be equivalently reformulated as

$$\begin{array}{ll} \min\limits_{X} & Q_0 \cdot X \\ \text{s.t.} & Q_i \cdot X \le b_i,\ i = 1, 2, \ldots, m, \\ & l_i^2 \le X_{ii} \le u_i^2,\ i = 1, 2, \ldots, n, \\ & \arg(X_{ij}) \in \mathcal{A}_{ij},\ (i,j) \in \mathcal{E}, \\ & \mathrm{rank}(X) = 1. \end{array} \tag{CQP2}$$

where $X \in \mathbb{C}^{n\times n}$ is decision variable. After dropping the phase difference constraints $\arg(X_{ij}) \in \mathcal{A}_{ij}$ for $(i,j) \in \mathcal{E}$ and relaxing the nonconvex constraint $\mathrm{rank}(X) = 1$ to semidefinite constraint $X \succeq 0$, we have the following classic semidefinite relaxation of (CQP):

$$\begin{array}{ll} \min\limits_{X} & Q_0 \cdot X \\ \text{s.t.} & Q_i \cdot X \le b_i,\ i = 1, 2, \ldots, m, \\ & l_i^2 \le X_{ii} \le u_i^2,\ i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array} \tag{CSDP}$$

It is notable that the bound provided by the relaxation problem (CSDP) may not be tight since the constraints $\arg(X_{ij}) \in \mathcal{A}_{ij},\ (i,j) \in \mathcal{E}$ have been dropped directly and $\mathcal{A}_{ij}$ is not even involved in (CSDP). Therefore, it is likely to improve the tightness of relaxation (CSDP) by deriving new valid inequalities from the phase difference constraints. For this purpose, we first reformulate problem (CQP) by introducing a new matrix variable R as follows:

$$\begin{array}{ll} \min\limits_{X,R} & Q_0 \cdot X \\ \text{s.t.} & Q_i \cdot X \le b_i,\ i = 1,\ldots,m, \\ & l_i^2 \le X_{ii} = R_{ii} \le u_i^2,\ i = 1,\ldots,n, \\ & R_{ij}^2 = R_{ii}R_{jj}, (i,j) \in \mathcal{E}, \\ & |X_{ij}| = R_{ij},\ \arg(X_{ij}) \in \mathcal{A}_{ij}, (i,j) \in \mathcal{E}, \\ & X \succeq 0. \end{array} \tag{CQP3}$$

where $X \in \mathbb{C}^{n\times n}$ and $R \in \mathbb{R}^{n\times n}$ are decision variables. The equivalence between (CQP) and (CQP3) above is obvious by noting that $X_{ij} = x_i x_j$ and $R_{ij} = |x_i||x_j|$. By relaxing the nonconvex constraints $R_{ij}^2 = R_{ii}R_{jj}$ and $|X_{ij}| = R_{ij}$ in problem (CQP3), we have the following semidefinite relaxation of problem (CQP):

$$\begin{array}{ll} \min\limits_{X,R} & Q_0 \cdot X \\ \text{s.t.} & Q_i \cdot X \le b_i,\ i = 1,\ldots,m, \\ & l_i^2 \le R_{ii} = X_{ii} \le u_i^2,\ i = 1,\ldots,n, \\ & (R_{ii}, R_{jj}, R_{ij}) \in \text{Conv}(\mathcal{H}_{ij}),\ (i,j) \in \mathcal{E}, \\ & X_{ij} \in \text{Conv}(\mathcal{G}_{ij}(R_{ij})),\ (i,j) \in \mathcal{E}, \\ & X \succeq 0. \end{array} \tag{ECSDP1}$$

where the sets $\mathcal{H}_{ij}$ and $\mathcal{G}_{ij}(R_{ij})$ are defined as

$$\mathcal{H}_{ij} := \left\{ (R_{ii}, R_{jj}, R_{ij}) \mid l_i^2 \le R_{ii} \le u_i^2, l_j^2 \le R_{jj} \le u_j^2, R_{ij}^2 = R_{ii}R_{jj} \right\}, \tag{2.1}$$

and

$$\mathcal{G}_{ij}(R_{ij}) := \{ X_{ij} \mid |X_{ij}| = R_{ij},\ \arg(X_{ij}) \in \mathcal{A}_{ij} \}, \tag{2.2}$$

respectively. Usually, it may be hard to describe the convex hull of a given set. Fortunately, [3] and [21] show that the convex hulls of $\mathcal{H}_{ij}$ and $\mathcal{G}_{ij}(R_{ij})$ in problem (CQP3) have a closed-form characterization. We cite the related results here with some modifications in the notation.

Theorem 2.1 (Corollary 5 in [3])**.** *The following two linear inequalities are valid for $\mathcal{H}_{ij}$ with a given index $(i,j) \in \mathcal{E}$:*

$$(l_i + u_i)(l_j + u_j) R_{ij} \ge \left(l_j^2 + l_j u_j\right) R_{ii} + \left(l_i^2 + l_i u_i\right) R_{jj} + l_i l_j u_i u_j - l_i^2 l_j^2, \tag{2.3}$$

and

$$(l_i + u_i)(l_j + u_j) R_{ij} \ge \left(u_j^2 + l_j u_j\right) R_{ii} + \left(u_i^2 + l_i u_i\right) R_{jj} + l_i l_j u_i u_j - u_i^2 u_j^2. \tag{2.4}$$

Moreover, the convex hull of $\mathcal{H}_{ij}$ can be represented as follows:

$$\text{Conv}(\mathcal{H}_{ij}) = \left\{ (R_{ii}, R_{jj}, R_{ij}) \;\middle|\; \begin{array}{l} l_i^2 \le R_{ii} \le u_i^2,\ l_j^2 \le R_{jj} \le u_j^2 \\ R_{ij}^2 \le R_{ii}R_{jj} \\ (R_{ii}, R_{jj}, R_{ij}) \textit{ satisfies inequalities } (2.3) \textit{ and } (2.4) \end{array} \right\}. \tag{2.5}$$

As for the set $\mathcal{G}_{ij}(R_{ij})$, its convex hull depends on the structure of $\mathcal{A}_{ij}$. The next theorem describes the convex hull of $\mathcal{G}_{ij}(R_{ij})$.

Theorem 2.2 (Propositions 1 and 2 in [21]). *For a given index $(i,j) \in \mathcal{E}$ and R_{ij}, when $\mathcal{A}_{ij} = [\underline{\theta}_{ij}, \overline{\theta}_{ij}]$ and $\overline{\theta}_{ij} - \underline{\theta}_{ij} < 2\pi$, the convex hull of $\mathcal{G}_{ij}(R_{ij})$ is*

$$\mathrm{Conv}\left(\mathcal{G}_{ij}\left(R_{ij}\right)\right) = \left\{ X_{ij} \;\middle|\; \begin{array}{l} a_{ij}\,\mathrm{Re}\left(X_{ij}\right) + b_{ij}\,\mathrm{Im}\left(X_{ij}\right) \geq c_{ij} R_{ij} \\ |X_{ij}| \leq R_{ij} \end{array} \right\}, \tag{2.6}$$

where

$$a_{ij} = \cos\left(\frac{\underline{\theta}_{ij} + \overline{\theta}_{ij}}{2}\right),\ b_{ij} = \sin\left(\frac{\underline{\theta}_{ij} + \overline{\theta}_{ij}}{2}\right) \text{ and } c_{ij} = \cos\left(\frac{\overline{\theta}_{ij} - \underline{\theta}_{ij}}{2}\right).$$

When $\mathcal{A}_{ij} = \{\theta_{ij}^1, \theta_{ij}^2, \ldots, \theta_{ij}^M\}$ where $0 \leq \theta_{ij}^1 < \theta_{ij}^2 < \cdots < \theta_{ij}^M < 2\pi$, the convex hull of $\mathcal{G}_{ij}(R_{ij})$ is

$$\mathrm{Conv}\left(\mathcal{G}_{ij}\left(R_{ij}\right)\right) = \left\{ X_{ij} \;\middle|\; \begin{array}{l} a_{ij}^t\,\mathrm{Re}\left(X_{ij}\right) + b_{ij}^t\,\mathrm{Im}\left(X_{ij}\right) \leq c_{ij}^t R_{ij} \\ t = 1, 2, \ldots, M \end{array} \right\}, \tag{2.7}$$

where $\theta_{ij}^{M+1} := \theta_{ij}^1 + 2\pi$ and

$$a_{ij}^t = \cos\left(\frac{\theta_{ij}^t + \theta_{ij}^{t+1}}{2}\right),\ b_{ij}^t = \sin\left(\frac{\theta_{ij}^t + \theta_{ij}^{t+1}}{2}\right),\ c_{ij}^t = \cos\left(\frac{\theta_{ij}^{t+1} - \theta_{ij}^t}{2}\right)$$

for $t = 1, 2, \ldots, M$.

In addition, the constraint $R \succeq 0$ implies the constraints $R_{ij}^2 \leq R_{ii}R_{jj}$ for all $(i,j) \in \mathcal{E}$, hence $R_{ij}^2 \leq R_{ii}R_{jj}$ becomes redundant if $R \succeq 0$ is added into (ECSDP1). In this way, we have another semidefinite relaxation of (CQP) as follows:

$$\begin{array}{ll} \min\limits_{X,R} & Q_0 \cdot X \\ \text{s.t.} & Q_i \cdot X \leq b_i, i = 1, \ldots, m, \\ & l_i^2 \leq R_{ii} = X_{ii} \leq u_i^2, i = 1, \ldots, n, \\ & X_{ij} \in \mathrm{Conv}\left(\mathcal{G}_{ij}\left(R_{ij}\right)\right), (i,j) \in \mathcal{E}, \\ & \left(R_{ii}, R_{jj}, R_{ij}\right) \text{ satisfies (2.3) and (2.4)}, (i,j) \in \mathcal{E}, \\ & X \succeq 0,\ R \succeq 0. \end{array} \tag{ECSDP2}$$

We point out that (ECSDP2) could be tighter than (ECSDP1). Section 4 of [21] provides an example of (CQP) to support this claim and explores the relationship between (ECSDP1) and other relaxations in the literature.

2.2 Tightness and relaxation gap

Since (ECSDP1) and (ECSDP2) are two relaxations of problem (CQP), a natural concern is the tightness of these relaxations. We address this issue from two perspectives. We first establish the conditions under which the proposed relaxation (ECSDP1) is tight. Subsequently, we quantify the disparity between the sets $\mathcal{H}_{ij}$ and $\mathcal{G}_{ij}(R_{ij})$ and their corresponding convex hulls $\mathrm{Conv}(\mathcal{H}_{ij})$ and $\mathrm{Conv}(\mathcal{G}_{ij}(R_{ij}))$, respectively.

The following theorem gives the condition for the tightness of the relaxation (ECSDP1) whose proof line is similar to the one in [9] and [13].

Theorem 2.3. *Assume that the undirected graph $\mathcal{U} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and the edge set $\mathcal{E} \subseteq \{(i,j) | 1 \leq i \leq j\}$ is connected, and (X, R) is an optimal solution of* (ECSDP1)*. If X satisfies $X_{ii} > 0$ for $i = 1, 2, \ldots, n$ and $X_{ii}X_{jj} = |X_{ij}|^2$ for $(i,j) \in \mathcal{E}$, then the rank of matrix X is one, and there is no gap between* (ECSDP1) *and* (CQP).

Proof. Since X is a positive definite Hermite matrix, it has a decomposition $X = V^\dagger V$ with $V = [v_1, v_2, \ldots, v_n]$ and $v_i \in \mathbb{C}^n$. Since $X_{ii} > 0$, v_i and v_j are all nonzero vectors. Then the equation $X_{ii}X_{jj} = |X_{ij}|^2$ can be expressed equivalently as $|v_i|^2|v_j|^2 = |{v_i}^\dagger v_j|^2$. Based on the Cauchy-Schwartz inequality, the equation $|v_i|^2|v_j|^2 = |{v_i}^\dagger v_j|^2$ holds if and only if there exists a $\lambda_{ij} \in \mathbb{C}$, such that $v_i = \lambda_{ij}v_j$ for any $(i,j) \in \mathcal{E}$. For any two different columns v_p and v_q in V, under the assumption that $\mathcal{U}$ is a connected graph, there is a path connecting p and q. Without loss of generality, such a path can be denoted by a set $\{s_1, s_2, \ldots, s_\ell\}$ where $s_1 = p$, $s_\ell = q$ and $(s_i, s_{i+1}) \in \mathcal{E}$ for $i = 1, 2, \ldots, \ell - 1$. It follows that $v_{s_i} = \lambda_{s_i s_{i+1}} v_{s_{i+1}}$ for some nonzero $\lambda_{s_i s_{i+1}}$, $i = 1, 2, \ldots, \ell - 1$. Hence, $v_p = \prod_{i=1}^{\ell-1} \lambda_{s_i s_{i+1}} v_q$, i.e., v_p and v_q is linearly dependent for any $p, q \in \{1, 2, ..., n\}$. It follows that the matrix $V = [v_1, v_2, \ldots, v_n]$ is rank-one, and so is X. That is, X has a rank-one decomposition as $X = xx^\dagger$ for some x, which is an optimal solution of (CQP). The tightness of (ECSDP1) then holds straightforward. □

Remark 2.4. Since relaxation (ECSDP2) is at least as tight as (ECSDP1), the condition in Theorem 2.3 is also sufficient for the tightness of (ECSDP2).

In general, there may be a non-zero gap between (CQP) and the relaxation (ECSDP1). In this case, Theorem 2.3 implies that there is at least one index $(i,j) \in \mathcal{E}$ such that $R_{ij} < \sqrt{R_{ii}R_{jj}}$ or $|X_{ij}| < R_{ij}$ for the optimal solution (X, R) of (ECSDP1). That is, $(R_{ii}, R_{jj}, R_{ij}) \in \mathrm{Conv}(\mathcal{H}_{ij}) \setminus \mathcal{H}_{ij}$ or $X_{ij} \in \mathrm{Conv}(\mathcal{G}_{ij}(R_{ij}) \setminus \mathcal{G}_{ij}(R_{ij})$. Next, we derive some inequalities to bound the difference between R_{ij} and $\sqrt{R_{ii}R_{jj}}$, and the difference between $|X_{ij}|$ and R_{ij} in the following two theorems, respectively.

Theorem 2.5. *For a given index $(i,j) \in \mathcal{E}$, R_{ij} and $X_{ij} \in Conv(\mathcal{G}_{ij}(R_{ij}))$, when $\mathcal{A}_{ij} = \left[\underline{\theta}_{ij}, \overline{\theta}_{ij}\right]$ and $\overline{\theta}_{ij} - \underline{\theta}_{ij} < 2\pi$, the following inequality holds*

$$R_{ij} \geq |X_{ij}| \geq R_{ij} \cos\left(\frac{\overline{\theta}_{ij} - \underline{\theta}_{ij}}{2}\right). \tag{2.8}$$

When $\mathcal{A}_{ij} = \left\{\theta_{ij}^1, \theta_{ij}^2, \ldots, \theta_{ij}^M\right\}$ where $0 \leq \theta_{ij}^1 < \theta_{ij}^2 < \cdots < \theta_{ij}^M < 2\pi$, the following inequality holds

$$R_{ij} \geq |X_{ij}| \geq R_{ij} \cos\left(\frac{\theta_{ij}^M - \theta_{ij}^1}{2}\right) \tag{2.9}$$

Proof. For the case $\mathcal{A}_{ij} = \left[\underline{\theta}_{ij}, \overline{\theta}_{ij}\right]$ with $\overline{\theta}_{ij} - \underline{\theta}_{ij} < 2\pi$, by the definition of $\mathrm{Conv}(\mathcal{G}_{ij}(R_{ij}))$ in (2.5), we have $|X_{ij}| \leq R_{ij}$. Thus we only need to show that

$$|X_{ij}| \geq R_{ij} \cos\left(\frac{\overline{\theta}_{ij} - \underline{\theta}_{ij}}{2}\right).$$

Since $X_{ij} \in \mathrm{Conv}(\mathcal{G}_{ij}(R_{ij}))$, we have

$$a_{ij} \operatorname{Re}(X_{ij}) + b_{ij} \operatorname{Im}(X_{ij}) \geq c_{ij} R_{ij},$$

where a_{ij}, b_{ij} and c_{ij} are defined in Theorem 2.2. Using the Cauchy-Schwartz inequality, it follows that

$$a_{ij} \operatorname{Re}(X_{ij}) + b_{ij} \operatorname{Im}(X_{ij}) \leq \sqrt{\left[a_{ij}^2 + b_{ij}^2\right]\left[\operatorname{Re}^2(X_{ij}) + \operatorname{Im}^2(X_{ij})\right]} = |X_{ij}|.$$

The previous two inequalities show that $|X_{ij}| \geq c_{ij} R_{ij}$. Using the definition of c_{ij}, we obtain the desired inequality

$$R_{ij} \geq |X_{ij}| \geq R_{ij} \cos\left(\frac{\overline{\theta}_{ij} - \underline{\theta}_{ij}}{2}\right).$$

When $\mathcal{A}_{ij} = \left\{\theta_{ij}^1, \theta_{ij}^2, \ldots, \theta_{ij}^M\right\}$, we have the following inequality hold when $t = M$ from $\mathrm{Conv}(\mathcal{G}_{ij}(R_{ij}))$ defined in (2.7):

$$\cos\left(\frac{\theta_{ij}^1 + \theta_{ij}^M}{2}\right) \mathrm{Re}\left(X_{ij}\right) + \sin\left(\frac{\theta_{ij}^1 + \theta_{ij}^M}{2}\right) \mathrm{Im}\left(X_{ij}\right) \geq \cos\left(\frac{\theta_{ij}^M - \theta_{ij}^1}{2}\right) R_{ij}.$$

Again, using the Cauchy-Schwartz inequality, we have $|X_{ij}| \geq R_{ij} \cos\left(\frac{\theta_{ij}^M - \theta_{ij}^1}{2}\right)$. On the other hand, $\mathrm{Conv}(\mathcal{G}_{ij}(R_{ij}))$ defined in (2.7) is the polyhedral set generated by the set of extreme points $\{A_1, A_2, \ldots, A_M\}$, which all lie on a circle with radius R_{ij}. So we have $|X_{ij}| \leq R_{ij}$. This completes the proof. □

Theorem 2.6. *If $(R_{ii}, R_{jj}, R_{ij}) \in \mathrm{Conv}(\mathcal{H}_{ij})$, then the following inequality holds*

$$\sqrt{R_{ii} R_{jj}} - R_{ij} \leq \frac{1}{2} u_i \left(u_j - l_j\right) + \frac{1}{2} u_j \left(u_i - l_i\right). \tag{2.10}$$

Proof. Following from the inequalities (2.3) and (2.4), we have the following two inequalities:

$$\sqrt{R_{ii} R_{jj}} - R_{ij} \leq \sqrt{R_{ii} R_{jj}} - \frac{\left(l_j^2 + l_j u_j\right) R_{ii} + \left(l_i^2 + l_i u_i\right) R_{jj} + l_i l_j u_i u_j - l_i^2 l_j^2}{\left(l_i + u_i\right)\left(l_j + u_j\right)},$$
$$\sqrt{R_{ii} R_{jj}} - R_{ij} \leq \sqrt{R_{ii} R_{jj}} - \frac{\left(u_j^2 + l_j u_j\right) R_{ii} + \left(u_i^2 + l_i u_i\right) R_{jj} + l_i l_j u_i u_j - u_i^2 u_j^2}{\left(l_i + u_i\right)\left(l_j + u_j\right)}.$$

Adding the above two inequalities and dividing both sides by 2, we have

$$\sqrt{R_{ii} R_{jj}} - R_{ij} \leq \frac{1}{2}\left[\frac{\left(u_i u_j - l_i l_j\right)^2 - \left(\left(l_j + u_j\right)\sqrt{R_{ii}} - \left(l_i + u_i\right)\sqrt{R_{jj}}\right)^2}{\left(l_i + u_i\right)\left(l_j + u_j\right)}\right].$$

Dropping the two terms involving $\sqrt{R_{ii}}$ and $\sqrt{R_{jj}}$ in the right-hand side of the above inequality, we have

$$\sqrt{R_{ii} R_{jj}} - R_{ij} \leq \frac{1}{2}\left[\frac{\left(u_i u_j - l_i l_j\right)^2}{\left(l_i + u_i\right)\left(l_j + u_j\right)}\right].$$

Since $u_i > l_i \geq 0$ and $u_j > l_j \geq 0$, it follows that

$$\frac{u_i u_j - l_i l_j}{\left(l_i + u_i\right)\left(l_j + u_j\right)} \leq \frac{u_i u_j - l_i l_j}{u_i u_j} \leq 1. \tag{2.11}$$

Now we have

$$\begin{aligned}\sqrt{R_{ii}R_{jj}} - R_{ij} &\le \frac{(u_iu_j - l_il_j)^2}{2(l_i+u_i)(l_j+u_j)} \\ &\le \frac{1}{2}(u_iu_j - l_il_j) \\ &= \frac{1}{2}(l_i(u_j-l_j) + l_j(u_i-l_i) + (u_j-l_j)(u_i-l_i)) \\ &\le \frac{1}{2}(l_i(u_j-l_j) + l_j(u_i-l_i) + 2(u_j-l_j)(u_i-l_i)) \\ &= \frac{1}{2}u_i(u_j-l_j) + \frac{1}{2}u_j(u_i-l_i)\end{aligned}$$

where the second inequality follows from (2.11) and the last inequality follows from $u_i > l_i \ge 0$ and $u_j > l_j \ge 0$. □

Theorems 2.5 and 2.6 show that the bounds on $|X_{ij}| - R_{ij}$ and $\sqrt{R_{ii}R_{jj}} - R_{ij}$ are controlled by $\overline{\theta}_{ij} - \underline{\theta}_{ij}$ and $u_i - l_i$, $u_j - l_j$, respectively. When $\overline{\theta}_{ij} - \underline{\theta}_{ij}$ approaches zero, $|X_{ij}|$ will be close to R_{ij}. Similarly, When $u_i - l_i$ and $u_j - l_j$ become close to zero, $R_{ii}R_{jj}$ will be close to R_{ij}. That is, for problem (CQP), if the bounds on the modulus of variables and the phase difference are narrow, the relaxation (ECSDP1) may provide a good approximation. Furthermore, the relaxation (ECSDP2) could be tighter than (ECSDP1), so it may provide an even better lower bound.

3 A New Branch-and-Bound Algorithm for (CQP)

This section presents a new branch-and-bound algorithm for solving problem (CQP) based on semidefinite relaxation (ECSDP2) (since it could provide a tighter lower bound than (ECSDP1)). Then we discuss the relationship between the proposed algorithm and the existing one in [12].

3.1 The proposed algorithm

The proposed branch-and-bound algorithm is an enumeration procedure that partitions the feasible region into smaller subregions and recursively generates subproblems over the partitioned subregions. The algorithm estimates a lower bound of problem (CQP) by solving the semidefinite relaxation (ECSDP2) in each enumeration node. Meanwhile, an upper bound is calculated using a local or heuristic algorithm. The procedure ends when the difference between the upper and lower bounds is less than a given error tolerance ϵ. Then an ϵ-optimal solution is found.

For ease of description, we adopt the following notations. For each enumeration node, we associate it with a set of parameters $\{x, (X,R), L, U, \mathcal{A}, \mathcal{R}\}$, where x is a feasible solution obtained by a local or heuristic algorithm, (X,R) is the optimal solution of the relaxation in the enumeration node, L and U are the lower and upper bounds of the node, respectively, $\mathcal{A}$ collects the upper and lower bounds on the phase differences, and $\mathcal{R}$ is the set of upper and lower bounds on the modulus of variables. Let $ECSDP(\mathcal{A}, \mathcal{R})$ denote the semidefinite relaxation for this instance with parameters $\mathcal{A}_{ij} \in \mathcal{A}$ for $(i,j) \in \mathcal{E}$, and $[l_i, u_i] \in \mathcal{R}$, $i = 1, 2, ..., n$.

The main procedures of the proposed algorithm are described as below.

Lower Bound: For a generic node indexed by k, the optimal value L^k of each relaxation problem with parameter sets $\mathcal{A}^k$ and $\mathcal{R}^k$ serves as a lower bound for the subproblem. Then,

L^*, the minimal value of the lower bounds of all incumbent nodes, is a valid lower bound of problem (CQP).

Upper Bound: If the relaxation problem (ECSDP2) of a node is infeasible, then this node is fathomed. If the convex relaxation (ECSDP2) at a node in the search tree is exact, which means that the optimal solution of (ECSDP2) satisfies $|X_{ij}| = R_{ij}$ and $R_{ij}^2 = R_{ii}R_{jj}$ for all $(i,j) \in \mathcal{E}$, then the current node is also fathomed and its optimal value is a valid upper bound of (CQP). Otherwise, we will try to recover a feasible solution of problem (CQP) from the optimal solution of the relaxation problem. In detail, let (X, R) be the optimal relaxation solution, and let v be the eigenvector corresponding to the largest eigenvalue of the solution X. Then, we construct a solution $\overline{x}$ by setting

$$|\overline{x}| = \text{diag}(X) \text{ and } \arg(\overline{x}) = \arg(v).$$

Note that $\overline{x}$ may not satisfy the nonconvex quadratic constraints, thus we use it as an initial point for a local solver, such as IPOPT [20], to solve the subproblem to find a feasible solution x of problem (CQP). In the case where x is found, its objective value is an upper bound of the subproblem. We use U^* and x^* to denote the best upper bound and the best feasible solution throughout the enumeration process.

Node Selection Rule: Once the upper bound U^* of problem (CQP) is updated, any node with the lower bound no less than U^* is fathomed. Let $\mathcal{P}$ denote the set of all incumbent nodes that are not yet fathomed. In the enumeration procedure, we always select an active node with the smallest lower bound from $\mathcal{P}$. If there is a tie in the selection of the node, we break it arbitrarily.

Branching Rule: Our goal is to continuously obtain tighter lower bounds via effective branching. Theorems 2.5 and 2.6 indicate that the tightness of the proposed semidefinite relaxation may be affected by the values of $\overline{\theta}_{ij} - \underline{\theta}_{ij}$ and $u_i - l_i$, $u_j - l_j$. Therefore, our branching strategy aims to partition the modular and phase spaces such that the ranges of modulus and phase difference become narrow.

Let $\{\mathcal{A}^k, \mathcal{R}^k\}$ be the parameter of a selected node k in the current enumeration, and (X^k, R^k) is the optimal solution of the relaxation $ECSDP(\mathcal{A}^k, \mathcal{R}^k)$. We calculate the following two groups of values:

$$S_1^k = \max_{(i,j)\in\mathcal{E}} \left\{R_{ij}^k - |X_{ij}^k|\right\}, \quad \left(i_1^k, j_1^k\right) = \arg\max_{(i,j)\in\mathcal{E}} \left\{R_{ij}^k - |X_{ij}^k|\right\},$$

and

$$S_2^k = \max_{(i,j)\in\mathcal{E}} \left\{\sqrt{R_{ii}^k R_{jj}^k} - R_{ij}^k\right\}, \quad \left(i_2^k, j_2^k\right) = \arg\max_{(i,j)\in\mathcal{E}} \left\{\sqrt{R_{ii}^k R_{jj}^k} - R_{ij}^k\right\}.$$

If multiple indices achieve the maximum, we select the smallest one. The quantity $\max\{S_1^k, S_2^k\}$ serves as a proxy for the relaxation gap. If $S_2^k \le S_1^k$, then motivated by the bound in Theorem 2.6, we select $\mathcal{A}_{i_1^k j_1^k} \in \mathcal{A}^k$ to branch. When $\mathcal{A}_{i_1^k j_1^k}$ is an interval $[\underline{\theta}_{i_1^k j_1^k}, \overline{\theta}_{i_1^k j_1^k}]$, it will be partitioned into $[\underline{\theta}_{i_1^k j_1^k}, (\underline{\theta}_{i_1^k j_1^k} + \overline{\theta}_{i_1^k j_1^k})/2]$ and $[(\underline{\theta}_{i_1^k j_1^k} + \overline{\theta}_{i_1^k j_1^k})/2, \overline{\theta}_{i_1^k j_1^k}]$. When $\mathcal{A}_{i_1^k j_1^k}$ is a discrete set $\left\{\theta^1_{i_1^k j_1^k}, \theta^2_{i_1^k j_1^k}, \ldots, \theta^M_{i_1^k j_1^k}\right\}$, it will be branched into $\left\{\theta^1_{i_1^k j_1^k}, \theta^2_{i_1^k j_1^k}, \ldots, \theta^{\lfloor M/2 \rfloor}_{i_1^k j_1^k}\right\}$ and $\left\{\theta^{\lceil M/2 \rceil}_{i_1^k j_1^k}, \theta^{\lceil M/2 \rceil + 1}_{i_1^k j_1^k}, \ldots, \theta^M_{i_1^k j_1^k}\right\}$. Otherwise, if $S_2^k \ge S_1^k$, we choose the longer of the intervals $[l_{i_2^k}, u_{i_2^k}]$ and $[l_{j_2^k}, u_{j_2^k}]$, and partition it into two equal length subintervals.

Based on the above description, the pseudo code of the proposed algorithm is given in Algorithm 1.

Algorithm 1 A branch-and-bound algorithm for problem (CQP).

Require: A feasible instance of (CQP) with parameters $\mathcal{A}$ and $\mathcal{R}$, a limit on iteration number K and a given error tolerance ϵ.

Set $k = 0$, initialize $\mathcal{A}^k = \mathcal{A}$ and $\mathcal{R}^k = \mathcal{R}$. Set the node list $\mathcal{P} = \emptyset$.
Solve root node relaxation $ECSDP(\mathcal{A}^0, \mathcal{R}^0)$ for its optimal solution (X^0, R^0) and optimal value L^0.
Compute a feasible point x^0 using a local solver or heuristic method (if possible) and get an upper bound U^0.
Set the initial optimal upper bound $L^* = L^0$ and optimal lower bound $U^* = U^0$.
Add $\{x^0, (X^0, R^0), L^0, U^0, \mathcal{A}^0, \mathcal{R}^0\}$ into the node list $\mathcal{P}$.
while $(U^* - L^*)/U^* > \epsilon$ and $k < K$ **do**
    Set $k = k + 1$
    Use **Node Selection Rule**, choose a node from $\mathcal{P}$, denoted as $\{x^k, (X^k, R^k), L^k, U^k, \mathcal{A}^k, \mathcal{R}^k\}$
    Delete the chosen node from $\mathcal{P}$.
    Use the **Branching Rule** to either branch $\mathcal{A}^k$ into $\mathcal{A}^k_-$ and $\mathcal{A}^k_+$ or branch $\mathcal{R}^k$ into $\mathcal{R}^k_+$ and $\mathcal{R}^k_-$.
    for $s \in \{+, -\}$ **do**
        Solve relaxation $ECSDP(\mathcal{A}^k_s, \mathcal{R}^k_s)$ to get an optimal solution (X^k_s, R^k_s) and a lower bound L^k_s.
        Obtain an feasible solution x^k_s and get an upper bound U^k_s from x^k_s by using the method in the **Upper Bound** procedure.
        if $U^k_s \leq U^*$ **then**
            Update $U^* = U^k_s$ and $x^* = x^k_s$.
        end if
        if $L^k_s \leq U^*$ **then**
            Insert the node $\{x^k_s, (X^k_s, R^k_s), L^k_s, U^k_s, \mathcal{A}^k_s, \mathcal{R}^k_s\}$ into $\mathcal{P}$.
        end if
    end for
    Delete all the nodes in $\mathcal{P}$ whose lower bound is larger than U^*.
    Update lower bound L^* according to **Lower Bound** procedure.
end while
Return x^*, L^* and U^*.

3.2 Comparison with the algorithm in [12]

In this subsection, we analyze the relationship between the proposed branch-and-bound algorithm and the one in [12].

Lu et al. have studied the following nonhomogeneous quadratic programming problem in [12]:

$$
\begin{aligned}
\min\ & \frac{1}{2} x^\dagger Q_0 x + \mathrm{Re}(c^\dagger x) \\
& l_i \leq |x_i| \leq u_i,\ i = 1, 2, \ldots, n, \\
& \arg(x_i) \in \mathcal{A}_i \subseteq [0, 2\pi],\ i = 1, 2, \ldots, n.
\end{aligned}
\tag{3.1}
$$

Note that problem (3.1) is a subclass of problem (CQP). In fact, by appending x_{n+1} to the vector $x \in \mathbb{C}^n$ and setting $x_{n+1} = 1$, the constraint $\arg(x_i) \in \mathcal{A}_i \subseteq [0, 2\pi]$ can be rewritten as $\arg(x_i x_{n+1}^\dagger) \in \mathcal{A}_i \subseteq [0, 2\pi]$, $i = 1, 2, \ldots, n$. In this way, problem (3.1) is a special case of (CQP) by setting $Q_i = 0$, $b_i = 0$, $i = 1, 2, ..., m$, and $\mathcal{E} = \{(1, n+1), (2, n+1), ..., (n, n+1)\}$.

In [12], Lu et al. propose a branch-and-bound algorithm, called ECSDR-BB, for problem (3.1) based on the following semidefinite relaxation:

$$\begin{aligned} \min\ & \frac{1}{2} Q_0 \cdot X + \mathrm{Re}(c^\dagger x) \\ & l_i \le r_i \le u_i,\ i = 1, 2, \ldots, n, \\ & x_i \in \mathrm{Conv}(\mathcal{G}_i(r_i)),\ i = 1, 2, \ldots, n, \\ & X_{ii} \ge r_i^2,\ X_{ii} - (l_i + u_i) r_i + l_i u_i \le 0,\ i = 1, 2, \ldots, n, \\ & X \succeq x x^\dagger, \end{aligned} \tag{ECSDP3}$$

where $\mathcal{G}_i(r_i) := \{x_i \mid r_i = |x_i|, \arg(x_i) \in \mathcal{A}_i\}$ for $r_i > 0$ and $\mathcal{G}_i(0) = \{0\}$.

We now compare the relaxations adopted in the proposed algorithm and ECSDR-BB. Theorem 7 in [21] shows that (ECSDP1) is equivalent to (ECSDP3). Indeed, it is straightforward to check that, if (X, x, r) is a feasible solution of (ECSDP3), the following solution (Y, R) such that

$$Y = \begin{bmatrix} X & x \\ x^\dagger & 1 \end{bmatrix} \text{ and } R = \begin{bmatrix} r r^\dagger & r \\ r^\dagger & 1 \end{bmatrix} \tag{3.2}$$

is a feasible solution of (ECSDP1), and vice versa. Hence, ECSDR-BB uses (ECSDP1) as the relaxation for the lower bound, while the proposed algorithm in this paper uses (ECSDP2). As we have mentioned in Section 2.1, (ECSDP2) could be tighter than (ECSDP1), the lower bounds returned by the proposed algorithm is at least as tight as the one by ECSDR-BB.

We then analyze the relationship between the branching rules of the two algorithms. In ECSDR-BB, after obtaining an optimal solution $(\bar{X}, \bar{x}, \bar{r})$ of (ECSDP3), a feasible solution $\hat{\mathbf{x}}$ of (3.1) is generated as follows:

$$\hat{\mathbf{x}} = \mathrm{Scale}(\bar{\mathbf{x}}, \bar{\mathbf{r}}) := \left[\bar{r}_1 e^{\mathrm{i}\arg(\bar{x}_1)}, \bar{r}_2 e^{\mathrm{i}\arg(\bar{x}_2)}, \ldots, \bar{r}_n e^{\mathrm{i}\arg(\bar{x}_n)}\right]^\top.$$

Let

$$\begin{aligned} i_1^* &= \arg\max_i \{|\hat{x}_i - \bar{x}_i|\}, \quad S_1^* = \max_i \{|\hat{x}_i - \bar{x}_i|\}, \\ i_2^* &= \arg\max_i \left\{\bar{X}_{ii} - (\bar{r}_i)^2\right\}, \quad S_2^* = \max_i \left\{\bar{X}_{ii} - (\bar{r}_i)^2\right\}. \end{aligned} \tag{3.3}$$

If $S_1^* \le S_2^*$, ECSDR-BB selects $\mathcal{A}_{i_1^*}$ to branch; otherwise, ECSDR-BB selects the interval $[l_{i_2^*},\ u_{i_2^*}]$. That is, the branching rule of ECSDR-BB only selects the index $i \in \{1, 2, \ldots, n\}$. Since problem (3.1) can be formulated as (CQP), the branching rule of ECSDR-BB is equivalent to the one of the proposed algorithm, in which $\mathcal{E}$ is set to $\{(1, n+1), (2, n+1), \ldots, (n, n+1)\}$. On the other hand, we can add the redundant constraints $\arg(x_i x_j^\dagger) \in \mathcal{A}_{ij} = [0, 2\pi]$ for (i, j) with $1 \le i < j \le n$ into (CQP), that is, $\mathcal{E}$ is extended by adding (i, j)'s with $1 \le i < j \le n$. In this case, the candidate for branching in the proposed algorithm is the entire set of elements (i, j) with $1 \le i < j \le n + 1$, which strictly includes the original set $\mathcal{E} = \{(1, n+1), (2, n+1), \ldots, (n, n+1)\}$. Therefore, the branching rule in the proposed algorithm measures the proxy of the relaxation gap over a wider set, making branch decisions that could lead to smaller relaxation gaps in the child nodes.

In summary, the proposed algorithm is different from ECSDR-BB in the following two aspects.

(1) The proposed algorithm uses the tighter relaxation (ECSDP2) than (ECSDP1) which in turn is equivalent to the relaxation (ECSDP3) used in ECSDR-BB.

(2) The branching rule of the proposed algorithm allows us to select candidates in the set $\{(i,j) \mid 1 \le i < j \le n+1\}$, while the branching rule of ECSDR-BB only considers the set $\{(1,n+1),(2,n+1),\dots,(n,n+1)\}$.

4 Numerical Experiment

In this section, we evaluate the performance of the proposed branch-and-bound algorithm. Two subclasses of problem (CQP) are selected to test the proposed algorithm: discrete beamforming problem and virtual beamforming problem. For both subclasses, we compare the proposed algorithm with the existing ones in the literature. Specifically, for the discrete beamforming design problem, we draw a comparison with the one in [4], while for the virtual beamforming design problem, our benchmark is the algorithm proposed in [12].

All experiments are carried out on a personal computer with Intel Core(TM) i5-7300HQ CPU (3.5 GHz) and 16 GB RAM. We use Mosek (Version 10.0) [17] to solve the semidefinite relaxations, and Gurobi [6] for mixed integer linear problems. Algorithms are implemented in Matlab R2019a.

4.1 Discrete beamforming problem

We refer to [4] for an introduction and a model of the discrete beamforming problem as below. Transmit beamforming design has found widespread applications in different fields such as communications, radar, sonar, etc. In the field of communications, there is a strong interest in the use of a transmit beamformer at the base station in order to improve the quality of service (QoS) and maximize the signal-to-noise ratio (SNR) for the users. Practical hardware in radar and communication systems is composed of discrete phase and amplitude shifters, hence the discrete transmit beamforming problem can be formulated as follows:

$$\begin{array}{ll} \max\limits_{t,x} & t \\ \text{s.t.} & x^\dagger Q_k x \ge t\gamma_k\sigma_k^2,\ k=1,2,\dots,N, \\ & |x_i|^2 \le P_{\max},\ i=1,2,\dots,M, \\ & x^\dagger x \le P_{\text{tot}}, \\ & \arg(x_i) \in \{0,\xi,2\xi,\dots,(2^n-1)\,\xi\},\ \xi = 2\pi/2^n, \\ & |x_i| \in \{\Delta,2\Delta,\dots,2^m\Delta\},\ \Delta=\sqrt{P_{\max}}/2^m, \end{array} \tag{DBP}$$

where $t\in\mathbb{R}$ and $x\in\mathbb{C}^n$ are decision variables, M is the number of transmit antennas, N is the number of receivers, $Q_k = h_k h_k^\dagger$ with $h_k\in\mathbb{C}^M$ being the complex channel vector for the k-th receiver, $P_{\max}$ is the maximal per-antenna power, P_{tot} is the total transmit power, γ_k is the power proportion for the k-th target, m and n are the number of bits to represent the discrete phase and amplitude, respectively, and ξ and Δ are the discrete step size for phase and amplitude, respectively.

In [4], problem (DBP) is reformulated as a mixed integer problem (MIP), and solved by the off-the-shelf solver Gurobi. Meanwhile, (DBP) can also be solved by the proposed Algorithm 1 in its original formulation. To compare the performance of the two algorithms, we randomly generated several groups of instances as follows: We set $M\in\{4,5\}$, $N\in\{4,8,12\}$, $n\in\{3,4\}$, $m\in\{3,4\}$, $\gamma_k=1$, $\sigma_k^2=1$ for $k=1,2,\dots,N$, $P_{\max}=20$ and $P_{\text{tot}}=225$. For each given group of parameter setting (M,N,n,m), the real and imaginary parts of the complex vector $h_k\in\mathbb{C}^M$ are randomly sampled from the standard M-dimensional Gaussian distribution. Ten instances are generated for each group. For all instances, the tolerance for relative optimality is set to 0.01%, and the running time is limited to 600 seconds.

Table 1: Numerical results on discrete beamforming problem.

Group	Gurobi		Algorithm 1	
	Solved	Time (Sec.)	Solved	Time (Sec.)
$M=4, N=4, n=3, m=3$	10/10	1.35	10/10	0.71
$M=4, N=8, n=3, m=3$	10/10	3.12	10/10	1.57
$M=4, N=12, n=3, m=3$	10/10	3.31	10/10	4.34
$M=4, N=4, n=4, m=4$	10/10	11.70	10/10	0.51
$M=4, N=8, n=4, m=4$	10/10	55.51	10/10	2.88
$M=4, N=12, n=4, m=4$	9/10	120.82	10/10	21.38
$M=5, N=4, n=4, m=4$	8/10	143.95	10/10	1.53
$M=5, N=8, n=4, m=4$	3/10	177.68	10/10	11.51
$M=5, N=12, n=4, m=4$	2/10	274.86	10/10	35.53

The computational results for problem (DBP) are shown in Table 1. The column "Solved" lists the number of instances that can be solved within 600s, and the column "Time" lists the average computational time over instances that can be solved within the time limit of 600 seconds. It is evident that, on average, the proposed algorithm is much faster than Gurobi in solving problem (DBP). As the instance size rises, Gurobi fails to solve more cases and requires more time to solve larger instances, while the proposed algorithm successfully solves all instances within one minute on average regardless of problem size. The main reason is probably that the semidefinite relaxation applied in the proposed algorithm is much tighter than the linear relaxation used in Gurobi, and the mixed integer reformulation introduces a lot of binary variables as the problem size increases. In addition, the special branching rule in the proposed algorithm, which chooses bounds on the modulus or phase of the complex variables for branching, may be also effective in reducing the gap.

4.2 Virtual beamforming problem

In this subsection, we investigate the performance of the proposed algorithm in the case of continuous phase difference constraints. In [12], Lu et al. proposed a branch-and-bound algorithm called ECSDR-BB for the virtual beamforming problem, which can formulated as follows:

$$\begin{aligned} \min_{y}\ & \frac{1}{2} y^\dagger \tilde{Q}_0 y \\ & l_i \le |y_i| \le u_i,\ i = 1, 2, \ldots, n, \\ & \arg(y_i y_j^\dagger) \in \mathcal{A}_{ij},\ (i,j) \in \mathcal{E} \\ & y_{n+1} = 1, \end{aligned} \tag{VBP}$$

where $\mathcal{E} = \{(1, n+1), (2, n+1), \ldots, (n, n+1)\}$ and $\tilde{Q}_0 = \begin{bmatrix} Q_0 & c \\ c^\dagger & 0 \end{bmatrix} \in \mathbb{C}^{(n+1)\times(n+1)}$.

To compare the performance of the proposed algorithm with ECSDR-BB, we generated instances using the following settings: $\mathcal{A}_i = [0, 2\pi]$, $l_i = 1$, $u_i = 2$ for all $i = 1, 2, \ldots, n$. The elements of $Q_0 \in \mathbb{C}^{n\times n}$ and $c \in \mathbb{C}^n$ are randomly sampled from the standard Gaussian distribution. We set the size of the instances $n = 10, 15, 20, 25$. For each size n, we randomly generated 10 instances and solved them using two algorithms. The tolerance for relative optimality is set to 0.001% and the running time is limited to 600 seconds.

The performance of the two algorithms is reported in Table 2. The column "Enum" lists the average number of enumerations over the instances that can be solved within a time limit. Other column names have the same meaning as in Table 1.

Table 2: Numerical results on virtual beamforming problem.

n	ECSDR-BB in [12]			Algorithm 1		
	Solved	Enum	Time (Sec.)	Solved	Enum	Time (Sec.)
10	10/10	23.5	1.02	10/10	15.8	0.90
15	10/10	232.7	14.11	10/10	154.6	11.78
20	10/10	1050.4	99.10	10/10	701.0	78.87
25	7/10	871.9	107.35	9/10	918.1	121.23

From the results in Table 2, we can observe that the proposed algorithm is faster than ECSDR-BB in terms of solution time. In detail, when $n \leq 20$, the proposed algorithm is about one order faster than ECSDR-BB. In addition, the proposed algorithm can solve more instances than ECSDR-BB when $n = 25$. The results indicate that the proposed algorithm is more efficient and robust than ECSDR-BB for the virtual beamforming problem. The superior performance of the proposed algorithm over ECSDR-BB in [12] is not surprising. According to the discussion in Section 3.2, the proposed algorithm adopts tighter semidefinite relaxation (ECSDP2) than the one used in ECSDR-BB for the virtual beamforming problem, and the branching rule in the proposed algorithm selects in a wider range of candidates for possible better improvement of lower bounds during the progress.

5 Conclusions

In this paper, we propose a branch-and-bound algorithm for a general class of complex quadratic programming problems containing nonconvex quadratic constraints and various modulus and phase difference constraints. The adaptive branching strategy selects the modulus or phase difference to branch for the purpose of efficiency and flexibility. Numerical results on the discrete and virtual beamforming problems show that the proposed algorithm is superior to the commercial solver Gurobi and other algorithms in the literature.

Acknowledgments

The authors are very grateful and would like to thank the anonymous referees and the editor for their suggestions and helpful comments, which significantly improved the quality of this paper.

References

[1] A. Arora, C.G. Tsinos, M.R.B. Shankar, S. Chatzinotas and B. Ottersten, Efficient algorithms for constant-modulus analog beamforming, *IEEE Trans. Signal Process.* 70 (2022) 756–771.

[2] A.S. Bandeira, N. Boumal and A. Singer, Tightness of the maximum likelihood semidefinite relaxation for angular synchronization, *Math. Program.* 163 (2016) 145–167.

[3] C. Chen, A. Atamtürk and S.S. Oren, A spatial branch-and-cut method for nonconvex QCQP with bounded complex variables, *Math. Program.* 165 (2016) 549–577.

[4] Ö.T. Demir and T.E. Tuncer, Optimum discrete transmit beamformer design, *Digital Signal Processing* 36 (2015) 57–68.

[5] M.X. Goemans and D.P. Williamson, Approximation algorithms for Max-3-Cut and other problems via complex semidefinite programming, *J. Comput. System Sci.* 68 (2004) 442–470.

[6] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2023.

[7] S.H. Low, Convex relaxation of optimal power flow–Part I: Formulations and equivalence, *IEEE Trans. Control Netw. Syst.* 1 (2014) 15–27.

[8] S.H. Low, Convex relaxation of optimal power flow–Part II: Exactness, *IEEE Trans. Control Netw. Syst.* 1 (2014) 177–189.

[9] C. Lu, Z. Deng, W. Xing and S.C. Fang, A new global algorithm for Max-Cut problem with chordal sparsity, *J. Optim. Theory Appl.* 197 (2023) 608–638.

[10] C. Lu, Z. Deng, W.-Q. Zhang and S.-C. Fang, Argument division based branch-and-bound algorithm for unit-modulus constrained complex quadratic programming, *J. Global Optim.* 70 (2018) 171–187.

[11] C. Lu, Y.-F. Liu, W.-Q.Zhang and S. Zhang, Tightness of a new and enhanced semidefinite relaxation for MIMO detection, *SIAM J. Optim.* 29 (2019) 719–742.

[12] C. Lu, Y.-F. Liu and J. Zhou, An enhanced SDR based global algorithm for nonconvex complex quadratic programs with signal processing applications, *IEEE Open Journal of Signal Processing* 1 (2020) 120–134.

[13] C. Lu, J. Ma, Z. Deng, W. Xing, A graphic structure based branch-and-bound algorithm for complex quadratic optimization and applications to magnitude least-square problem, *J. Global Optim.*, online first (2023).

[14] X. Luo and Z. Luo, Joint source and relay optimization for a MIMO communication system, *Pac. J. Optim.* 4 (2008) 53–62.

[15] Z. Luo, W.-K. Ma, S.A. Man-Cho, Y. Ye and S. Zhang, Semidefinite relaxation of quadratic optimization problems, *IEEE Signal Processing Magazine* 27 (2010) 20–34.

[16] A. Maio, Y. Huang, M. Piezzo, S. Zhang and A. Farina, Design of optimized radar codes with a peak to average power ratio constraint, *IEEE Transactions on Signal Processing* 59 (2011) 2683–2697.

[17] MOSEK ApS, *The MOSEK Optimization Toolbox for MATLAB Manual, Version 10.0*, 2022.

[18] R. Nishimura, S. Hayashi and M. Fukushima, SDP reformulation for robust optimization problems based on nonconvex QP duality, *Comput. Optim. Appl.* 55 (2013) 21–47.

[19] I. Waldspurger, A. d'Aspremont and S. Mallat, Phase recovery, MaxCut and complex semidefinite programming, *Math. Program.* 149 (2013) 47–81.

[20] A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.* 106 (2005) 25–57.

[21] Y. Xu, C. Lu, Z. Deng and Y.-F. Liu, New semidefinite relaxations for a class of complex quadratic programming problems, *J. Global Optim.* 87 (2023) 255–275.

[22] S. Zhang and Y. Huang, Complex quadratic optimization and semidefinite programming, *SIAM J. Optim.* 16 (2006) 871–890.

[23] S. Zhang and Y. Xia, Two fast Complex-Valued algorithms for solving complex quadratic programming problems, *IEEE Transactions on Cybernetics* 46 (2016) 2837–2847.

Manuscript received 11 September 2023
revised 11 December 2023
accepted for publication 25 December 2023

Yingzhe Xu
School of Economics and Management
North China Electric Power University, Beijing 102206, China
E-mail address: xuyingzhe1@163.com

Jintao Xu
Department of Applied Mathematics
The Hong Kong Polytechnic University, Hong Kong, China
E-mail address: jintao.xu@polyu.edu.hk

Cheng Lu
School of Economics and Management
North China Electric Power University, Beijing 102206, China
E-mail address: lucheng1983@163.com

Shu-Cherng Fang
Department of Industrial and System Engineering
North Carolina State University, Raleigh 27695-7906, USA
E-mail address: fang@ncsu.edu

Zhibin Deng
School of Economics and Management
University of Chinese Academy of Sciences, Beijing 100190, China
E-mail address: zhibindeng@ucas.edu.cn