

Article

Linear-Time Algorithm for Complex Uniform CDT Subproblem Based on Hidden Convexity

Zhuoyi Xu ¹, Chenyang Ma ¹ and Meijia Yang ^{2,*}¹ School of Statistics and Data Science, Capital University of Economics and Business, Beijing 100070, China; xuzhuoyi@cueb.edu.cn (Z.X.); 32023120102@cueb.edu.cn (C.M.)² School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China

* Correspondence: myang@ustb.edu.cn

Abstract

In this paper, we study the nonconvex quadratic programming over the intersection of two balls in n -dimensional complex space, which is called the uniform CDT subproblem and is significant in both optimization theory and applications. We first prove the hidden convexity of the problem by using the S-lemma. In order to construct an algorithm, we prove the hidden convexity again by reformulating it as a convex problem (C). Subsequently, we employ the eigenvalue approximation method and the generalized Nesterov's accelerated gradient descent algorithm to solve the problem (C) within an error tolerance of ϵ . Then, we prove that the proposed algorithm is linear-time with respect to the density of the matrix in the objective function. The density is related to the number of non-zero entries of the matrix. We do numerical experiments to compare the proposed linear-time algorithm with the CVX solver, with the method proposed by Burer and with the algorithm proposed by Sakaue et al. Numerical results show that the linear-time algorithm performs much better than other methods, which verify that the proposed algorithm is applicable to many large-scale problems nowadays.

Keywords: complex uniform CDT subproblem; hidden convexity; generalized Nesterov's accelerated gradient method; linear-time global algorithm

MSC: 90C20; 90C25; 90C26

1. Introduction

Optimization problems over the complex field have attracted significant attention in both theory and applications. In this paper, we study the nonconvex quadratic programming over the intersection of two balls in n -dimensional complex space. It is a special class of nonconvex quadratic programming over the intersection of two general ellipsoids, which is known as the Celis–Dennis–Tapia (CDT) problem or the two trust-region subproblem (TTRS).

$$\begin{aligned} \text{UCDT} \quad & \min \quad f(w) = w^H A w + 2\mathcal{R}(b^H w) \\ & \text{s.t.} \quad w^H w - 1 \leq 0, \\ & \quad \quad w^H w + 2\mathcal{R}(c^H w) + d \leq 0, \\ & \quad \quad w \in \mathbb{C}^n, \end{aligned}$$

where $d \in \mathbb{R}$, $b, c \in \mathbb{C}^n$ are complex vectors, $A \in \mathbb{C}^{n \times n}$ is an Hermitian matrix whose eigenvalues are real numbers. Without loss of generality, we assume that the first constraint



Academic Editor: Shin-ya Matsushita

Received: 24 May 2026

Revised: 10 July 2026

Accepted: 10 July 2026

Published: 14 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)[Attribution \(CC BY\) license](#).

is the unit ball. We call this problem ‘uniform’ as the quadratic terms in both constraints share the same identity matrix I . Throughout this paper, $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ stand for the real part and the imaginary part of $(\cdot)$, respectively. Let $\text{val}(\cdot)$ denote the optimal value of problem $(\cdot)$. Assume that $A \not\equiv 0$ and denote its eigenvalues by $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, in which at least the smallest one of them is negative, that is, $\lambda_1 < 0$. Without loss of generality, we study UCDT under the following assumptions.

Assumption 1. Assume that there exists $\hat{w}$ in the interior of the feasible region of UCDT, i.e., $\hat{w}^H \hat{w} - 1 < 0$ and $\hat{w}^H \hat{w} + 2\mathcal{R}(c^H \hat{w}) + d < 0$. In other words, Slater’s condition holds for UCDT.

Assumption 2. Define $\Omega_1 := \{w^H w - 1 \leq 0, w \in \mathbb{C}^n\}$ and $\Omega_2 := \{w^H w + 2\mathcal{R}(c^H w) + d \leq 0, w \in \mathbb{C}^n\}$. Assume that $\Omega_1 \not\subseteq \Omega_2$ and $\Omega_2 \not\subseteq \Omega_1$. Otherwise, UCDT degenerates to a quadratic problem with only one ball constraint, which is known as a trust region subproblem (TRS) in the complex domain.

The nonconvex quadratic programming over a ball constraint is the famous trust region subproblem (TRS). It is known that the standard semidefinite programming relaxation, i.e., the Shor relaxation, is an equivalent convex relaxation for TRS. By equivalent, it means that the optimal value of the SDP relaxation is equal to that of the original problem. The optimal solution of TRS can be extracted from the optimal solution of the SDP relaxation. Then, we can solve the equivalent convex problem instead of the original nonconvex problem. This property of the nonconvex problem was defined as ‘hidden convexity’, which has been thoroughly surveyed in [1]. Leveraging hidden convexity, researchers can devise algorithms for the equivalent convex formulation. Increasingly efficient algorithms have been developed. Among these algorithms, whether a linear-time algorithm can be designed for an optimization problem is of great importance because there are now a large number of large-scale optimization problems that need to be solved quickly while maintaining solution accuracy. A key milestone was reached in 2016 with the first linear-time algorithm proposed by [2]. This was soon followed by additional linear-time approaches in [3–5].

As is well-known, TRS is essential to the trust region method in nonlinear programming, which is often combined with sequential quadratic programming (SQP) in practice. SQP is an efficient iterative method for constrained nonlinear optimization. As noted by [6], the matrix in its quadratic subproblem approximates the Hessian of the Lagrangian and is typically indefinite in nonconvex settings. Consequently, when SQP is combined with a trust-region framework [7] for an optimization problem with a ball constraint, the subproblem with another ball constraint related to the stepsize to be solved at each iteration takes the form of UCDT precisely.

UCDT is a very special class problem of the CDT subproblem, which was first introduced in [8]. CDT minimizes a TRS subject to an additional convex quadratic constraint. The real CDT subproblem has been extensively studied. The presence of the second quadratic constraint makes it considerably more challenging than TRS, even when both constraints are convex. Notably, ref. [9] proved that CDT can, in general, exhibit a duality gap. Consequently, a significant body of work has focused on deriving easily verifiable conditions that certify when no duality gap occurs, along with algorithms that guarantee global optimality. The first polynomial-time algorithm for solving CDT was established by [10] in 2016, followed by further polynomial-time methods such as those in [11,12]. In 2023, ref. [13] studied a class of extended CDT subproblems, proposed a family of relaxations, and studied some of its key properties, such as convexity issues, exactness of the relaxation, and a sufficient condition for exactness. Ref. [14] showed that nonconvex homogeneous quadratically constrained quadratic optimization with at most two constraints admits no local-non-global minimizers, which motivated our investigation.

As an important member of CDT, UCDT has attracted considerable attention. Kelly et al. [15] have shown that a particular disjunctive semidefinite relaxation is exact. Recently, ref. [16] has constructed a new, polynomially sized semidefinite relaxation, which adds second-order cone constraints to the classical SDP relaxation, for UCDT and proved that the new relaxation is exact. In other words, UCDT also has hidden convexity in the real space.

Parallel to the study of optimization problems in the real domain, related research in the complex domain has also been advancing. Complex quadratic programming, and the complex CDT problem in particular, has been widely studied in the literature. Ref. [17] analyzed the minimization of an indefinite quadratic function with two quadratic constraints, proving that strong duality holds in the complex domain and deriving necessary and sufficient optimality conditions. Ref. [18] investigated convexity properties in quadratic matrix programming, which reduces to a complex-variable problem when the matrix variable is of size $n \times 2$. More recently, ref. [19] considered minimizing a general homogeneous quadratic function subject to four complex homogeneous quadratic constraints, proposing an easily verifiable necessary and sufficient condition to determine whether the associated semidefinite programming (SDP) relaxation is tight. When tightness is confirmed, a globally optimal solution can be extracted in polynomial time.

The UCDT problem also arises in many applications. It is of significant practical importance. For example, the core task of radar is to detect whether a target is present. In an interference environment with clutter and noise, to achieve optimal detection, the Generalized Likelihood Ratio Test (GLRT) detector is typically employed. The radar transmits a sequence of coded pulses, which is defined as the vector $c \in \mathbb{C}^n$ [20]. To maximize the target detection probability, it is necessary to maximize a quadratic form of c :

$$c^H A c,$$

where A is an indefinite Hermitian matrix constructed from parameters such as the interference covariance matrix and the target Doppler frequency. The radar transmitter has a limited power budget. Hence, in the waveform design process, a total transmit energy constraint must be imposed, which takes the mathematical form:

$$\|c\|^2 \leq 1.$$

Beyond radar detection, UCDT appears in statistical applications, such as robust maximum-likelihood estimation, Gaussian mixture modeling, and logistic regression. This is because the log-likelihood Hessian in these problems is inherently nonconvex and may contain negative eigenvalues at certain points (including local maxima or saddle points), without needing to impose an explicit negative quadratic perturbation. Motivated by the theoretical and practical demand for an efficient solver, this paper focuses on the complex optimization model UCDT and develops a linear-time algorithm for it, sharing the same crucial property of hidden convexity in the complex number field; we note that a linear-time algorithm for it remains lacking in existing work.

The rest of the paper is structured as follows. In Section 2, we prove the hidden convexity of UCDT using the S-lemma. We also give a tight convex relaxation with strong duality, reconfirming this property. In Section 3, we propose a linear-time global algorithm that employs the Lanczos method and a generalized Nesterov accelerated gradient technique. In Section 4, we do numerical experiments to compare the proposed linear-time algorithm with the CVX solver, with the method proposed by Burer [16] and with the algorithm proposed by Sakaue et al. [12]. Conclusions are made in Section 5.

2. Hidden Convexity of UCDD

In this section, we first prove the hidden convexity of UCDD by applying the S-lemma over the complex domain, following the framework of [17], which builds on the foundational result of [21]. This proof establishes the theoretical strong duality of the problem. Alternatively, we provide a second proof by reformulating UCDD as an equivalent convex program, an approach inspired by [22]. This equivalent convex reformulation, presented in Section 2.2, serves as a computationally tractable foundation for our linear-time algorithm.

2.1. Proof Method Based on S-Lemma

Before restating the S-lemma, we first translate the property of a quadratic form being positive semidefinite into an equivalent matrix condition.

Lemma 1. Let $A \in \mathbb{C}^{n \times n}$ be an Hermitian matrix, $w, b \in \mathbb{C}^n$, $c \in \mathbb{R}$. We can adjust a quadratic inequality to a semidefinite matrix:

$$w^H A w + 2\mathcal{R}(b^H w) + c \geq 0 \quad \Leftrightarrow \quad \begin{bmatrix} A & b \\ b^H & c \end{bmatrix} \succeq 0.$$

Proof. By introducing $w = z/\tau \in \mathbb{C}^n$ with $z \in \mathbb{C}^n$ and $0 \neq \tau \in \mathbb{C}$, we have

$$\begin{aligned} w^H A w + 2\mathcal{R}(b^H w) + c \geq 0, \forall w &\Leftrightarrow \left(\frac{z}{\tau}\right)^H A \left(\frac{z}{\tau}\right) + 2\mathcal{R}\left(b^H \frac{z}{\tau}\right) + c \geq 0, \forall z, \tau \neq 0, \\ &\Leftrightarrow z^H A z + 2\mathcal{R}(\bar{\tau} b^H z) + c|\tau|^2 \geq 0, \forall z, \tau \neq 0, \\ &\Leftrightarrow z^H A z + 2\mathcal{R}(\bar{\tau} b^H z) + c|\tau|^2 \geq 0, \forall z, \tau, \\ &\Leftrightarrow \begin{pmatrix} \bar{z} & \bar{\tau} \end{pmatrix} \begin{pmatrix} A & b \\ b^H & c \end{pmatrix} \begin{pmatrix} z \\ \tau \end{pmatrix} \succeq 0. \end{aligned}$$

□

Then, we recall the S-lemma proposed by [21] over the complex domain, which reads as:

Theorem 1. (S-lemma) Let $f_k(w) = w^H A_k w + 2\mathcal{R}(b_k^H w) + c_k$, where $A_k \in \mathbb{C}^{n \times n}$ are Hermitian matrices, $w, b_k \in \mathbb{C}^n$ and $c_k \in \mathbb{R}$, ($k = 0, 1, 2$). If there exists $w_0 \in \mathbb{C}^n$ such that $f_1(w_0) < 0$, $f_2(w_0) < 0$ hold, then the following two statements are equivalent:

- (i) The system $f_0(w_0) < 0$, $f_1(w_0) \leq 0$, $f_2(w_0) \leq 0$ is unsolvable.
- (ii) There exists $\mu_1 \geq 0$, $\mu_2 \geq 0$ such that

$$\forall w \in \mathbb{C}^n, \quad f_0(w) + \mu_1 f_1(w) + \mu_2 f_2(w) \geq 0$$

holds.

To apply the S-lemma, the optimal value of the problem UCDD can be formulated by

$$\begin{aligned} \text{val}(\text{UCDD}) = \sup \left\{ t \mid \left\{ w^H A w + 2\mathcal{R}(b^H w) - t < 0, \right. \right. \\ \left. \left. w^H w - 1 \leq 0, \quad w^H w + 2\mathcal{R}(c^H w) + d \leq 0, w \in \mathbb{C}^n \right\} = \emptyset \right\}. \quad (1) \end{aligned}$$

Let

$$f_0(w) = w^H A w + 2\mathcal{R}(b^H w) - t, \quad f_1(w) = w^H w - 1, \quad f_2(w) = w^H w + 2\mathcal{R}(c^H w) + d.$$

With Assumption 1, there exists $\hat{w}$ such that

$$f_1(\hat{w}) < 0, \quad f_2(\hat{w}) < 0.$$

Therefore, the conditions for using the S-lemma have been met, and we have

$$\begin{aligned} \text{val}(\text{UCDT}) &= \sup \left\{ t \mid \forall w \in \mathbb{C}^n, \quad f_0(w) + \mu_1 f_1(w) + \mu_2 f_2(w) \geq 0, \quad \mu_1, \mu_2 \geq 0 \right\} \\ &= \sup \left\{ t \mid \left[w^H A w + 2\mathcal{R}(b^H w) - t \right] + \mu_1 \left[w^H w - 1 \right] \right. \\ &\quad \left. + \mu_2 \left[w^H w + 2\mathcal{R}(c^H w) + d \right], \quad \forall w \in \mathbb{C}^n, \mu_1, \mu_2 \geq 0 \right\} \\ &= \sup \left\{ t \mid \begin{bmatrix} A + \mu_1 I + \mu_2 I & b + \mu_2 c \\ (b + \mu_2 c)^H & -t - \mu_1 + \mu_2 d \end{bmatrix} \succeq 0, \quad \mu_1, \mu_2 \geq 0 \right\}. \end{aligned} \quad (2)$$

The last equation is deduced from Lemma 1 and the problem on the right side of the equation is convex as the objective function is linear and the feasible set is defined by linear matrix inequality, which is convex, together with nonnegativity constraints on the variables, which are also convex.

From the process above, we reveal the hidden convexity of UCDT in optimization theory by reformulating it to an equivalent semidefinite programming (2), which is convex and can be solved in polynomial time.

2.2. Proof Method Based on an Equivalent Convex Quadratic Programming

Following the ideas in [5,22], we provide an alternative proof of the hidden convexity of UCDT via a tight convex quadratic programming relaxation. This equivalent convex reformulation not only reconfirms the hidden convexity property but also lays the groundwork for the linear-time algorithm presented in the following section.

Introduce a new real variable t and use the smallest eigenvalue λ_1 to relax the objective function; we have the following convex quadratic programming:

$$\begin{aligned} \text{(C)} \quad \min \quad & g(w, t) = w^H (A - \lambda_1 I) w + 2\mathcal{R}(b^H w) + \lambda_1 t \\ \text{s.t.} \quad & w^H w \leq t, \\ & t - 1 \leq 0, \\ & t + 2\mathcal{R}(c^H w) + d \leq 0, \\ & w \in \mathbb{C}^n, t \in \mathbb{R}. \end{aligned} \quad (3)$$

Theorem 2. UCDT is equivalent to the convex quadratic programming relaxation (C) in the sense that $\text{val}(C) = \text{val}(\text{UCDT})$. Moreover, let $v_1 = x + iy$, with $x, y \in \mathbb{R}^n$ and $(x, y) \neq (0, 0)$ be an eigenvector of matrix A corresponding to the smallest eigenvalue λ_1 . Then, for any optimal solution of (C) which is denoted by (w^*, t^*) ,

$$w(s) = w^* + s\tilde{v}$$

can globally solve UCDT, where

$$\begin{aligned} s &= - \frac{(w^{*H} \tilde{v} + \tilde{v}^H w^*) \pm \sqrt{(w^{*H} \tilde{v} + \tilde{v}^H w^*)^2 - 4\tilde{v}^H \tilde{v} (w^{*H} w^* - t^*)}}{2\tilde{v}^H \tilde{v}}, \\ \tilde{v} &= \begin{cases} (x + iy) + \frac{\mathcal{R}(c)^T x + \mathcal{I}(c)^T y}{\mathcal{R}(c)^T y - \mathcal{I}(c)^T x} (ix - y), & \mathcal{R}(c)^T y - \mathcal{I}(c)^T x \neq 0, \\ y - ix, & \mathcal{R}(c)^T y - \mathcal{I}(c)^T x = 0. \end{cases} \end{aligned}$$

Proof. Firstly, we prove that $val(C) \leq val(UCDT)$, which is obvious as (C) is a relaxation of UCDT.

Let $\tilde{w}$ be the optimal solution of UCDT. Then, $\tilde{w}$ satisfies that

$$\tilde{w}^H \tilde{w} - 1 \leq 0, \quad \tilde{w}^H \tilde{w} + 2\mathcal{R}(c^H \tilde{w}) + d \leq 0.$$

Let $\tilde{t} = \tilde{w}^H \tilde{w}$, so that $(\tilde{w}, \tilde{t})$ is a feasible solution of (C), then

$$\begin{aligned} val(C) &\leq g(\tilde{w}, \tilde{t}) \\ &= \tilde{w}^H (A - \lambda_1 I) \tilde{w} + 2\mathcal{R}(b^H \tilde{w}) + \lambda_1 \tilde{t} \\ &= \tilde{w}^H A \tilde{w} + 2\mathcal{R}(b^H \tilde{w}) \\ &= val(UCDT). \end{aligned}$$

Secondly, we prove that the lower bound $val(C)$ is attainable for $val(UCDT)$.

Let (w^*, t^*) be the optimal solution of UCDT. Then, (w^*, t^*) satisfies that

$$w^{*H} w^* \leq t^*, \quad t^* - 1 \leq 0, \quad t^* + 2\mathcal{R}(c^H w^*) + d \leq 0.$$

When $w^{*H} w^* = t^*$, w^* satisfies the constraints of UCDT, so that w^* is a feasible solution of UCDT, then it holds that

$$\begin{aligned} val(UCDT) &\leq f(w^*) \\ &= w^{*H} A w^* + 2\mathcal{R}(b^H w^*) \\ &= w^{*H} A w^* + 2\mathcal{R}(b^H w^*) + \lambda_1 (t - w^{*H} w^*) \\ &= w^{*H} (A - \lambda_1 I) w^* + 2\mathcal{R}(b^H w^*) + \lambda_1 t^* \\ &= val(C). \end{aligned}$$

We continue our proof under the assumption $w^{*H} w^* < t^*$. With the assumption of A and λ_1 , the matrix $A - \lambda_1 I$ has a zero eigenvalue, which shares the same corresponding eigenvectors of λ_1 for matrix A .

Let v_1 denote an eigenvector. It satisfies that

$$(A - \lambda_1 I)v_1 = 0,$$

where $v_1 = x + iy$, $x, y \in \mathbb{R}^n$, $x^T x + y^T y \neq 0$. Then, $v_2 = iv_1 = ix - y$ is also an eigenvector of the zero eigenvalue for the matrix $A - \lambda_1 I$. We can prove that eigenvectors v_1 and v_2 are linearly independent when $x \neq 0$ and $y \neq 0$.

Suppose that v_1 and v_2 are linear dependent, and then there must exist $k_1, k_2 \in \mathbb{R}$ not both zero such that

$$\begin{aligned} k_1 v_1 + k_2 v_2 &= k_1(x + iy) + k_2(ix - y) \\ &= (k_1 x - k_2 y) + i(k_1 y + k_2 x) \\ &= 0. \end{aligned}$$

Therefore, $k_1 x - k_2 y = 0$ and $k_1 y + k_2 x = 0$ hold at the same time. As $x \neq 0$ and $y \neq 0$, the two equalities hold if and only if $k_1 = k_2 = 0$, which causes a contradiction with the assumption. Thus v_1 and v_2 are linearly independent.

Since v_1 and v_2 are linear independent when $x \neq 0$ and $y \neq 0$, then we construct a linear combination of v_1 and v_2 as:

$$\tilde{v} = v_1 + rv_2, \quad r \in \mathbb{R}.$$

By appropriate choice of r , we can construct a $\tilde{v}$ such that

$$\mathcal{R}(c^H \tilde{v}) = 0.$$

As $c^H \tilde{v} = (\mathcal{R}(c) - i\mathcal{I}(c))^T((x - ry) + i(y + rx))$, then

$$\begin{aligned} \mathcal{R}(c^H \tilde{v}) &= \mathcal{R}(c)^T(x - ry) + \mathcal{I}(c)^T(y + rx) \\ &= [\mathcal{R}(c)^T x + \mathcal{I}(c)^T y] - r[\mathcal{R}(c)^T y - \mathcal{I}(c)^T x]. \end{aligned}$$

If $\mathcal{R}(c)^T y - \mathcal{I}(c)^T x \neq 0$, let $r = \frac{\mathcal{R}(c)^T x + \mathcal{I}(c)^T y}{\mathcal{R}(c)^T y - \mathcal{I}(c)^T x}$, then $\mathcal{R}(c^H \tilde{v}) = 0$.

If $\mathcal{R}(c)^T y - \mathcal{I}(c)^T x = 0$, we adjust the linear combination of v_1 and v_2 and redefine $\tilde{v} = -v_2 = y - ix$, then $\mathcal{R}(c^H \tilde{v}) = 0$.

Furthermore, define

$$w(s) = w^* + s\tilde{v}, \quad s \in \mathbb{R}.$$

The constraint

$$\begin{aligned} t^* + 2\mathcal{R}(c^H w(s)) + d &= t^* + 2\mathcal{R}(c^H(w^* + s\tilde{v})) + d \\ &= t^* + 2\mathcal{R}(c^H w^*) + 2s\mathcal{R}(c^H \tilde{v}) + d \\ &= t^* + 2\mathcal{R}(c^H w^*) + d \leq 0 \end{aligned}$$

still holds. Then $(w(s), t^*)$ is a feasible solution of (C) for any s such that

$$w(s)^H w(s) \leq t^*.$$

Substitute $(w(s), t^*)$ into the object function of (C), we have

$$\begin{aligned} g(w(s), t^*) &= w(s)^H(A - \lambda_1 I)w(s) + 2\mathcal{R}(b^H w(s)) + \lambda_1 t^* \\ &= (w^* + s\tilde{v})^H(A - \lambda_1 I)(w^* + s\tilde{v}) + 2\mathcal{R}(b^H(w^* + s\tilde{v})) + \lambda_1 t^* \\ &= w^{*H}(A - \lambda_1 I)w^* + 2\mathcal{R}(b^H w^*) + 2s\mathcal{R}(b^H \tilde{v}) + \lambda_1 t^*. \end{aligned}$$

Given that $w(0)^H w(0) < t$ and the function $w(s)^H w(s)$ is continuous in $s \in \mathbb{R}$, it follows from the continuity that there exists a neighborhood $U(0)$ of 0 satisfying $\forall s \in U(0)$, $w(s)^H w(s) \leq t$.

Since (w^*, t^*) is the optimal solution of (C), we can conclude that $\mathcal{R}(b^H \tilde{v}) = 0$ and $(w(s), t^*)$ remains an optimal solution of (C). Otherwise, we can always find an s that is of opposite sign to $\mathcal{R}(b^H \tilde{v})$, such that $s\mathcal{R}(b^H \tilde{v}) < 0$. This implies $g(w(s), t^*) < g(w^*, t^*)$, yielding a contradiction. As $(w(s), t^*)$ satisfies that

$$\begin{aligned} w(s)^H w(s) &\leq t^* \leq 1, \\ w(s)^H w(s) + 2\mathcal{R}(c^H w(s)) + d &\leq t^* + 2\mathcal{R}(c^H w(s)) + d \leq 0, \end{aligned}$$

$w(s)$ is a feasible solution of UCDT.

Substituting $w(s)$ into the object function of UCDT, we have

$$\begin{aligned}
f(w(s)) &= w(s)^H A w(s) + 2\mathcal{R}(b^H w(s)) \\
&= (w^* + s\tilde{v})^H (A - \lambda_1 I) (w^* + s\tilde{v}) + 2\mathcal{R}(b^H (w^* + s\tilde{v})) + \lambda_1 w(s)^H w(s) \\
&= w^{*H} (A - \lambda_1 I) w^* + 2\mathcal{R}(b^H w^*) + \lambda_1 w(s)^H w(s) \\
&= w^{*H} (A - \lambda_1 I) w^* + 2\mathcal{R}(b^H w^*) + \lambda_1 t^* + \lambda_1 [w(s)^H w(s) - t^*] \\
&= \text{val}(C) + \lambda_1 [w(s)^H w(s) - t^*] \\
&\geq \text{val}(C).
\end{aligned}$$

The last inequality holds since $\lambda_1 < 0$ and $w(s)^H w(s) \leq t^*$. Furthermore, it holds as an equality when $w(s)^H w(s) = t^*$. We can deduce that this equation has a root in terms of s that

$$s = -\frac{(w^{*H} \tilde{v} + \tilde{v}^H w^*) \pm \sqrt{(w^{*H} \tilde{v} + \tilde{v}^H w^*)^2 - 4\tilde{v}^H \tilde{v} (w^{*H} w^* - t^*)}}{2\tilde{v}^H \tilde{v}}.$$

Under this condition, the lower bound $\text{val}(C)$ is attainable for $\text{val}(\text{UCDT})$.

The proof is complete. $\square$

The computational complexity of calculating the eigenvalues of a matrix is known as $O(n^3)$. In order to achieve the linear-time computational efficiency, we need the following estimation on the largest eigenvalue of A , denoted by $\lambda_{\max}(A)$.

Theorem 3 ([2,23]). *Given a symmetric matrix $A \in \mathbb{C}^{n \times n}$ with $\|A\| \leq \rho$, where $\|\cdot\|$ denotes the spectral norm of matrix $\cdot$, parameter $\epsilon, \delta > 0$ and let $N(\geq n)$ be an upper bound of the number of non-zero entries in A , the Lanczos method runs in time $O(\frac{N\sqrt{\rho}}{\epsilon} \log \frac{n}{\delta})$ and returns a unit vector $w \in \mathbb{C}^n$ for which*

$$w^H A w \geq \lambda_{\max}(A) - \epsilon$$

with probability at least $1 - \delta$.

As $\|\rho I - A\| \leq 2\rho$ and the largest eigenvalue of matrix $\rho I - A$ is $\rho - \lambda_1$. We can use Theorem 3 to estimate λ_1 and obtain the following result.

Corollary 1. *Given a symmetric matrix $A \in \mathbb{C}^{n \times n}$ with $\|A\| \leq \rho$, parameter $\epsilon, \delta > 0$ and let $N(\geq n)$ be an upper bound of the number of non-zero entries in A , the Lanczos method runs in time $O(\frac{N\sqrt{\rho}}{\epsilon} \log \frac{n}{\delta})$ and returns a unit vector $u \in \mathbb{C}^n$ for which*

$$u^H A u \leq \lambda_1 + \epsilon$$

with probability at least $1 - \delta$.

Define $\tilde{\lambda}_1 = u^H A u - \epsilon$. Then, we have

$$A - \tilde{\lambda}_1 I \succeq A - \lambda_1 I \succeq 0.$$

Therefore, we solve the following approximate convex programming of (3) instead of solving problem (3) directly.

$$\begin{aligned}
 (\text{AC}) \quad & \min \quad \tilde{g}(w, t) = w^H(A - \tilde{\lambda}_1 I)w + 2\mathcal{R}(b^H w) + \tilde{\lambda}_1 t \\
 & \text{s.t.} \quad w^H w \leq t, \\
 & \quad t - 1 \leq 0, \\
 & \quad t + 2\mathcal{R}(c^H w) + d \leq 0, \\
 & \quad w \in \mathbb{C}^n, t \in \mathbb{R}.
 \end{aligned} \tag{4}$$

As the two problems (3) and (4) share the same feasible region, the difference between their objective functions is given by $\tilde{\lambda}_1(t - w^H w)$, where $0 \leq t - w^H w \leq 1$. The Lanczos method guarantees with high probability that $|\tilde{\lambda}_1 - \lambda_1| \leq \epsilon$, where λ_1 is the smallest eigenvalue of A . Consequently, for any feasible solution, the difference in objective values is bounded by ϵ , and hence the difference between the optimal values $|\text{val}(\text{AC}) - \text{val}(\text{C})|$ is also bounded by ϵ .

3. Nesterov's Accelerated Gradient Method and Complexity

In this section, we first transform the complex problem (AC) into an equivalent real problem (AC^r). Then, we solve the problem (AC^r) using the Nesterov accelerated gradient method and analyze its worst-case complexity.

By decomposing the complex vector into its real and imaginary components, we convert the n -dimensional complex vector w into a $2n$ -dimensional real vector z . Therefore, (AC) is transformed into an equivalent real-variable problem (AC^r) of dimension $(2n + 1)$. The problem (AC^r) is formulated as follows.

$$\begin{aligned}
 (\text{AC}^r) \quad & \min \quad \hat{g}(z, t) = z^T(Q - \tilde{\lambda}_1 I)z + 2q^T z + \tilde{\lambda}_1 t \\
 & \text{s.t.} \quad z^T z - t \leq 0, \\
 & \quad h^T z + d + t \leq 0, \\
 & \quad t \leq 1, \\
 & \quad z \in \mathbb{R}^{2n}, t \in \mathbb{R},
 \end{aligned}$$

where

$$\begin{aligned}
 Q &= \begin{pmatrix} \mathcal{R}(A) & -\mathcal{I}(A) \\ \mathcal{I}(A) & \mathcal{R}(A) \end{pmatrix}, \quad \|Q\| = \|A\|, \\
 q &= (\mathcal{R}(b), \mathcal{I}(b)), \\
 h &= 2(\mathcal{R}(c), \mathcal{I}(c)).
 \end{aligned}$$

Let $\Omega := \{z^T z - t \leq 0, h^T z + d + t \leq 0, t \leq 1, z \in \mathbb{R}^{2n}, t \in \mathbb{R}\}$.

Lemma 2. Problem (AC) and problem (AC^r) are equivalent by means of

$$\begin{aligned}
 \text{val}(\text{AC}) &= \tilde{g}(w^*, t^*), \\
 \text{val}(\text{AC}^r) &= \hat{g}(z^*, t^*) = \hat{g}(\mathcal{R}(w^*), \mathcal{I}(w^*), t^*), \\
 \text{val}(\text{AC}) &= \text{val}(\text{AC}^r),
 \end{aligned}$$

where (w^*, t^*) and (z^*, t^*) are the optimal solutions of (AC) and (AC^r) respectively.

Proof. It can be easily verified that we can construct a feasible solution of (AC^r) from the optimal solution (w^*, t^*) of (AC), and vice versa. $\square$

Thus

$$\begin{aligned}\nabla_z \hat{g}(z, t) &= 2(Q - \tilde{\lambda}_1 I)z + 2q, \\ \nabla_t \hat{g}(z, t) &= \tilde{\lambda}_1, \\ \nabla \hat{g}(z, t) &= \begin{pmatrix} 2(Q - \tilde{\lambda}_1 I)z + 2q \\ \tilde{\lambda}_1 \end{pmatrix},\end{aligned}$$

and we can conclude that $\hat{g}(z, t)$ is Lipschitz continuous, since $\forall (z_1, t_1), (z_2, t_2) \in \Omega$,

$$\begin{aligned}\|\nabla \hat{g}(z_1, t_1) - \nabla \hat{g}(z_2, t_2)\|_2 &= \left\| \begin{pmatrix} 2(Q - \tilde{\lambda}_1 I)(z_1 - z_2) \\ 0 \end{pmatrix} \right\|_2 \\ &= 2\|(Q - \tilde{\lambda}_1 I)(z_1 - z_2)\|_2 \\ &\leq 2(\|Q\| + |\tilde{\lambda}_1|)\|z_1 - z_2\|_2 \\ &\leq 2(\|Q\| + |\tilde{\lambda}_1|)\|(z_1, t_1) - (z_2, t_2)\|_2,\end{aligned}$$

where $\|Q\| + |\tilde{\lambda}_1| \leq 2\|A\| \leq 2\rho$.

Next, we utilize the following algorithm proposed in [24], which serves as a generalized form of Nesterov's accelerated gradient method, to solve (AC^r) .

Theorem 4 ([24,25]). *Let $\{(z_k, t_k)\}$ be the sequence generated by Algorithm 1, (z^*, t^*) be the global minimizer of (AC^r) . Then, it holds that*

$$\forall k, \quad \hat{g}(z_k, t_k) - \hat{g}(z^*, t^*) \leq O\left(\frac{\rho}{k^2}\right),$$

and Algorithm 1 returns an ϵ -approximation solution of (AC^r) in $O(\sqrt{\frac{\rho}{\epsilon}})$ iterations.

Algorithm 1 Generalized Nesterov's Accelerated Gradient Method

1. Input: A, b, c, d
2. Choose $\theta_0 = \theta_{-1} \in (0, 1)$, $(z_0, t_0) = (z_{-1}, t_{-1}) \in \Omega$. Let $k := 0$.
3. Let

$$\begin{aligned}y_k &= z_k + \theta_k(\theta_{k-1}^{-1} - 1)(z_k - z_{k-1}), \\ s_k &= t_k + \theta_k(\theta_{k-1}^{-1} - 1)(t_k - t_{k-1})\end{aligned}$$

and

$$\begin{aligned}\nabla_z \hat{g}_k &= \nabla_z \hat{g}(y_k, s_k), \\ \nabla_t \hat{g}_k &= \nabla_t \hat{g}(y_k, s_k).\end{aligned}$$

Update

$$(z_{k+1}, t_{k+1}) = \arg \min_{(z, t) \in \Omega} [z^T \nabla_z \hat{g}_k + \rho \|z - y_k\|^2] + [t \cdot \nabla_t \hat{g}_k + \rho (t - s_k)^2]. \quad (5)$$

4. Choose $\theta_{k+1} \in (0, 1]$ satisfying

$$\frac{1 - \theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2}.$$

5. Output: z_k, t_k

If the stopping criterion does not hold, update $k := k + 1$ and go to step (2).

In the algorithm above, we need to solve (5), which is a proximal quadratic subproblem constrained on Ω :

$$\begin{aligned} G(z, t) &= z^T \nabla_z g_k + \rho \|z - y_k\|^2 + t \cdot \nabla_t g_k + \rho(t - s_k)^2 \\ &= z^T (2(Q - \tilde{\lambda}_1 I)y_k + 2q) + \rho z^T z - 2\rho y_k^T z + \rho y_k^T y_k + \tilde{\lambda}_1 t + \rho t^2 - 2\rho s_k t + \rho s_k^2. \end{aligned}$$

To simplify the expression, define

$$\begin{aligned} p &= 2[(Q - \tilde{\lambda}_1 I)y_k + q - \rho y_k], \\ r &= \rho y_k^T y_k, \\ \phi &= \tilde{\lambda}_1 - 2\rho s_k, \\ \psi &= \rho s_k^2, \end{aligned}$$

then we have

$$G(z, t) = \rho z^T z + p^T z + r + \rho t^2 + \phi t + \psi.$$

Therefore, the task of solving (5) is replaced by solving the following equivalent problem:

$$(SA) \quad \min_{(z,t) \in \Omega} G(z, t) = \rho z^T z + p^T z + \rho t^2 + \phi t$$

Since this problem is convex and quadratic, we can find the optimal $(\hat{z}, \hat{t})$ by solving the KKT conditions of problem (SA).

The KKT system for problem (SA) consists of three parts as follows:

1. Stationarity

$$2(\rho + \mu_1)z + p + \mu_2 h = 0, \quad 2\rho t + \phi - \mu_1 + \mu_2 + \mu_3 = 0. \quad (6)$$

2. Complementary slackness

$$\mu_1(z^T z - t) = 0, \quad \mu_2(h^T z + d + t) = 0, \quad \mu_3(t - 1) = 0. \quad (7)$$

3. Primal feasibility

$$z^T z - t \leq 0, \quad h^T z + d + t \leq 0, \quad t - 1 \leq 0, \quad (8)$$

where $\mu_1, \mu_2, \mu_3 \geq 0$ are the Lagrangian multipliers corresponding to the primal constraints, respectively.

We solve the KKT system from the two equations in (6). It can be deduced that

$$z(\mu_1, \mu_2) = -\frac{p + \mu_2 h}{2(\rho + \mu_1)}, \quad t(\mu_1, \mu_2, \mu_3) = \frac{\mu_1 - \mu_2 - \mu_3 - \phi}{2\rho}.$$

Since $\rho > 0$ and $\mu_1, \mu_2, \mu_3 \geq 0$, we divide the system into different cases.

Case 1. Suppose $\mu_1 = 0, \mu_2 = 0$. Then we have

$$z(0, 0) = -\frac{p}{2\rho}, \quad t(0, 0, \mu_3) = \frac{-\mu_3 - \phi}{2\rho}.$$

(I) If $\mu_3 = 0$, then $t(0, 0, \mu_3) = t(0, 0, 0) = -\frac{\phi}{2\rho}$.

(II) If $\mu_3 > 0$, then $t(0, 0, \mu_3) = 1$, so that

$$\mu_3 = -2\rho - \varphi > 0.$$

In either case, if $z = z(0, 0)$ and $t = t(0, 0, \mu_3)$ satisfy the three inequalities in (8), $(z(0, 0), t(0, 0, \mu_3)) = (\hat{z}, \hat{t})$ is optimal.

Case 2. Suppose $\mu_1 = 0, \mu_2 > 0$. Then, we have

$$z(0, \mu_2) = -\frac{p + \mu_2 h}{2\rho}, \quad t(0, \mu_2, \mu_3) = \frac{-\mu_2 - \mu_3 - \varphi}{2\rho},$$

and

$$h^T z(0, \mu_2) + d + t(0, \mu_2, \mu_3) = 0.$$

(I) If $\mu_3 = 0$, then

$$\mu_2 = \frac{2\rho d - \varphi - h^T p}{h^T h + 1}.$$

(II) If $\mu_3 > 0$, then $t(0, \mu_2, \mu_3) = 1$, so that

$$\mu_3 = -\mu_2 - 2\rho - \varphi, \quad \mu_2 = \frac{2\rho(d + 1) - h^T p}{h^T h}.$$

In either case, if $z = z(0, \mu_2)$ and $t = t(0, \mu_2, \mu_3)$ satisfy the three inequalities in (8), $(z(0, \mu_2), t(0, \mu_2, \mu_3)) = (\hat{z}, \hat{t})$ is optimal.

Case 3. Suppose $\mu_1 > 0, \mu_2 = 0$. Then, we have

$$z(\mu_1, 0) = -\frac{p}{2(\rho + \mu_1)}, \quad t(\mu_1, 0, \mu_3) = \frac{\mu_1 - \mu_3 - \varphi}{2\rho},$$

and

$$z(\mu_1, 0)^T z(\mu_1, 0) - t(\mu_1, 0, \mu_3) = 0.$$

(I) If $\mu_3 = 0$, then we have

$$2(\mu_1 - \varphi)(\rho + \mu_1)^2 - \rho p^T p = 0,$$

which is a unary cubic equation. μ_1 can be solved by the known formula from this equation.

(II) If $\mu_3 > 0$, then $t(\mu_1, 0, \mu_3) = 1$, so that

$$\mu_3 = \mu_1 - \varphi - 2\rho, \quad \mu_1 = -\rho + \frac{\sqrt{p^T p}}{2}.$$

In either case, if $z = z(\mu_1, 0)$ and $t = t(\mu_1, 0, \mu_3)$ satisfy the three inequalities in (8), $(z(\mu_1, 0), t(\mu_1, 0, \mu_3)) = (\hat{z}, \hat{t})$ is optimal.

Case 4. Suppose $\mu_1 > 0, \mu_2 > 0$. Then, we have

$$z(\mu_1, \mu_2) = -\frac{p + \mu_2 h}{2(\rho + \mu_1)}, \quad t(\mu_1, \mu_2, \mu_3) = \frac{\mu_1 - \mu_2 - \mu_3 - \varphi}{2\rho},$$

and

$$h^T z(\mu_1, \mu_2) + d + t(\mu_1, \mu_2, \mu_3) = 0, \quad z(\mu_1, \mu_2)^T z(\mu_1, \mu_2) - t(\mu_1, \mu_2, \mu_3) = 0.$$

(I) If $\mu_3 = 0$, then we have

$$\begin{cases} \rho(p + \mu_2 h)^T(p + \mu_2 h) - 2(\rho + \mu_1)^2(\mu_1 - \mu_2 - \varphi) = 0, \\ (\mu_1 - \mu_2 - \varphi)(\rho + \mu_1) + 2d\rho(\rho + \mu_1) - \rho h^T(p + \mu_2 h) = 0. \end{cases}$$

We can obtain μ_1 and μ_2 by solving the above binary cubic equations.

(II) If $\mu_3 > 0$, then $t(\mu_1, \mu_2, \mu_3) = 1$, so that

$$\mu_3 = \mu_1 - \mu_2 - \varphi - 2\rho,$$

$$\begin{cases} (p + \mu_2 h)^T(p + \mu_2 h) - 4(\rho + \mu_1)^2 = 0, \\ 2(d + 1)(\rho + \mu_1) - h^T(p + \mu_2 h) = 0. \end{cases}$$

We can obtain μ_1 and μ_2 by solving the above binary quadratic equations.

In either case, if $z = z(\mu_1, \mu_2)$ and $t = t(\mu_1, \mu_2, \mu_3)$ satisfy the three inequalities in (8), $(z(\mu_1, \mu_2), t(\mu_1, \mu_2, \mu_3)) = (\hat{z}, \hat{t})$ is optimal.

Remark 1. In Case 4, we need to solve a system of binary quadratic equations. The common methods include the iterative method and the resultant method. The resultant method's calculation process can be segmented into the following steps.

Step 1: Reformulate the binary quadratic equations into the following form.

$$\begin{cases} \mu_2^2 + a_{11}(\mu_1)\mu_2 + a_{12}(\mu_1) = 0, \\ \mu_2^2 + a_{21}(\mu_1)\mu_2 + a_{22}(\mu_1) = 0, \end{cases}$$

where $a_{11}(\mu_1), a_{12}(\mu_1), a_{21}(\mu_1), a_{22}(\mu_1)$ are polynomials of μ_1 .

Step 2: Solve the resultant, which is an equation deduced from the coefficient determinant, a polynomial of μ_1 . If there exist roots of μ_1 , it indicates that the whole system has solutions. Otherwise, it has no solution.

Step 3: Substitute each value of μ_1 back into the original equations to solve for the corresponding value of μ_2 .

After analyzing all possible cases, we can deduce that (5) can be modeled in $O(m)$ time, where $m \geq n$ is an upper bound of the number of non-zero entries in Q . Then, (5) can be solved in $O(n)$ time. With Theorem 4, we have the following conclusion.

Lemma 3. The time complexity of approximately solving (AC^r) under ϵ is

$$O\left(m\sqrt{\frac{\rho}{\epsilon}}\right).$$

Combining Corollary 1, Theorem 2, and Lemma 3, we can deduce the following main result.

Theorem 5. Given the parameters $\epsilon > 0$, $\delta > 0$, with probability at least $1 - \delta$ over the randomization of an approximate eigenvector oracle, we can find an ϵ -approximate solution of UCDT, $\tilde{w} \in \mathbb{C}^n$, satisfying the ball constraints

$$\tilde{w}^H \tilde{w} - 1 \leq 0, \quad \tilde{w}^H \tilde{w} + 2\mathcal{R}(c^H \tilde{w}) + d \leq 0,$$

and the objective value

$$f(\tilde{w}) \leq \text{val}(\text{UCDT}) + \epsilon$$

in total time

$$O\left(m\sqrt{\frac{\rho}{\epsilon}}\log\frac{n}{\delta}\right).$$

From this Theorem, we can have a linear-time algorithm for UCDT. By “linear”, it refers to the number of nonzero elements in the matrix being linear, so this algorithm is suitable for sparse matrices.

Remark 2. It is known that (CDT_C) has hidden convexity and admits a polynomial-time solution. However, no linear-time algorithm currently exists. Notably, Algorithm 1 can be extended to approximate (CDT_C) by bounding it. Specifically, by replacing the ellipsoidal constraint with a ball constraint defined via the maximum or minimum eigenvalue, our algorithm yields a lower bound (via relaxation) or an upper bound (via restriction).

Remark 3. The two proofs and the algorithm presented in Sections 2 and 3 exploit the specific structure of the complex domain. The corresponding real problem cannot be directly addressed by the same simple convex relaxation and solution recovery method. There exist other tight relaxations for the real case, such as the BETA-SDP relaxation proposed by Burer [16] mentioned in Section 1, but that is not the relaxation considered in Section 2 of this paper. Solving that convex relaxation requires lifting to the matrix level and introducing additional second-order cone constraints. To the best of our knowledge, the lifted problem does not admit a linear-time algorithm.

4. Numerical Experiment

In this section, we test the performance of Algorithm 1. All numerical tests are implemented in Python 3.13.5 and run on a laptop equipped with an Intel (R) Core (TM) i7-13620H processor (with a base frequency of 2.40 GHz) and 16 GB RAM. All instances are generated randomly. We initialize ρ as the square root of the largest eigenvalue of the matrix $Q^T Q$.

To rigorously evaluate the performance of the proposed algorithm, we designed a generator capable of precisely controlling the problem dimension n , the matrix sparsity d , and the condition number κ . For sparse construction, $N_{nz} = \lfloor d \cdot n^2/2 \rfloor$ positions are uniformly sampled and filled with complex Gaussian random variables $\mathcal{CN}(0, 1)$ in the lower triangular region. A Hermitian matrix M is then obtained via symmetrization, and the diagonal entries are filled with real Gaussian variables. To precisely control the condition number κ , we employ an affine spectral transformation. The extreme eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ of M are extracted using the Lanczos method, and the spectrum is linearly mapped to the interval $[-1/\kappa, 1]$ via $A = aM + bI$, ensuring that A attains the exact target condition number. For constraint parameterization, the linear terms $b, c \in \mathbb{C}^n$ are sampled from the complex standard normal distribution after scaling by $1/\sqrt{2}$, while the scalar constant d is chosen from $[-2, -0.1]$ to guarantee strict feasibility of the constraint sphere.

First, we compare the computational efficiency of different methods under different ϵ values and matrix dimensions. Among them, t_{CVX} , t_{TPE} , and t_{Alg1} denote solving the problem directly by the CVX solver, the algorithm proposed by [12], and Algorithm 1 proposed by us, respectively.

We can read from the results in Table 1 that Algorithm 1 runs faster than the other two methods when the dimension is large.

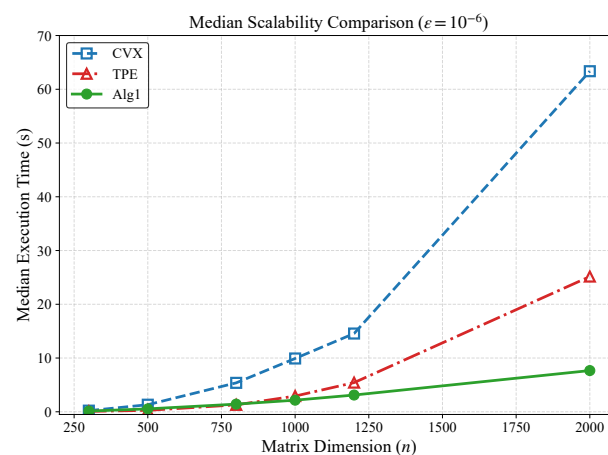
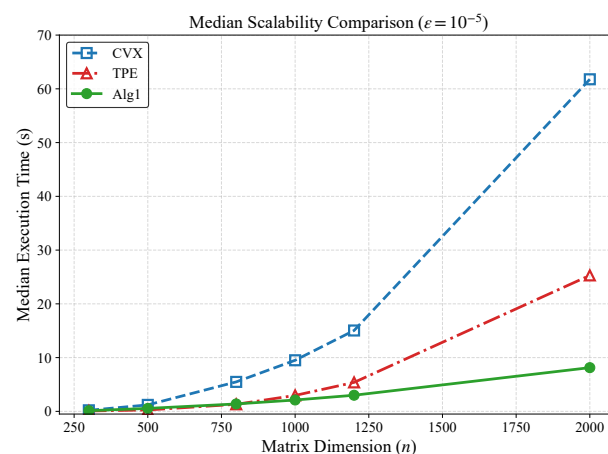
Additionally, we specifically visualized the difference in computational efficiency between the CVX solver and our Algorithm 1 under specific ϵ . The comparison of the running time among Algorithm 1, the CVX solver and the algorithm proposed by Sakaue et al. [12] with $\epsilon = 10^{-6}$, $\epsilon = 10^{-5}$, $\epsilon = 10^{-4}$ is displayed in Figure 1, Figure 2 and Figure 3, respectively.

Table 1. Numerical results of median execution time (s) for different optimality tolerances ϵ . The best performances for each dimension are highlighted in bold.

$\epsilon \backslash n$		300	500	800	1000	1200	2000
10^{-4}	t_{CVX}	0.195	1.192	5.087	9.458	14.520	61.437
	t_{TPE}	0.072	0.277	1.307	3.002	5.259	25.471
	t_{Alg1}	0.110	0.535	1.380	2.101	2.993	8.339
10^{-5}	t_{CVX}	0.196	1.204	5.480	9.505	15.043	61.778
	t_{TPE}	0.073	0.261	1.311	2.963	5.386	25.291
	t_{Alg1}	0.122	0.551	1.379	2.112	2.999	8.132
10^{-6}	t_{CVX}	0.212	1.316	5.381	9.916	14.557	63.361
	t_{TPE}	0.076	0.266	1.287	2.920	5.424	25.155
	t_{Alg1}	0.129	0.541	1.423	2.158	3.114	7.668

The computational efficiency of the algorithm is also assessed through comparison with the BETA SDP [16]. The corresponding execution time curves are illustrated in Figures 4–6.

From the figure, it is observed that the presence of the second-order cone constraints in the BETA leads to a significantly high-order polynomial growth as n increases. Therefore, we only do tests when dimension n is relatively small. In contrast, Algorithm 1 demonstrates the advantage of linear-time complexity.

**Figure 1.** $\epsilon = 10^{-6}$.**Figure 2.** $\epsilon = 10^{-5}$.

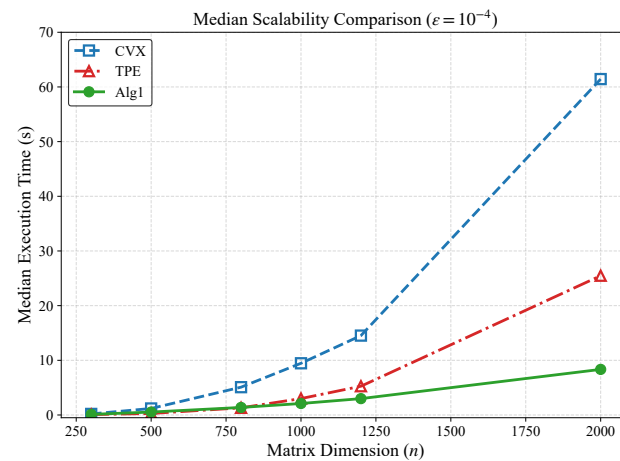


Figure 3. $\epsilon = 10^{-4}$.

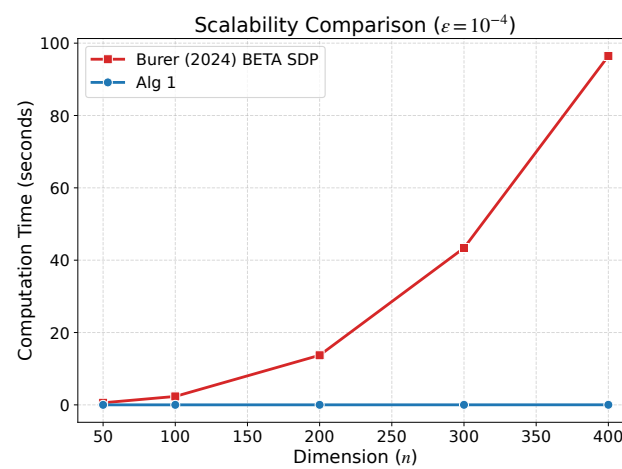


Figure 4. Result for $\epsilon = 10^{-4}$.

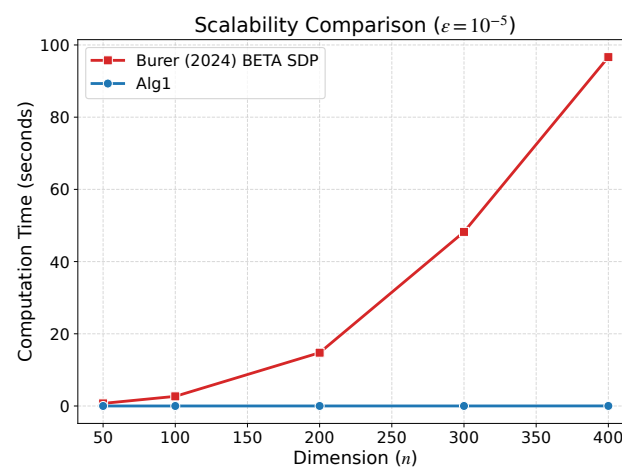


Figure 5. Result for $\epsilon = 10^{-5}$.

Next, we employ boxplots to illustrate the algorithm's resilience to ill-conditioning at $\kappa = 10^3$. In the vast majority of cases, the optimality gaps consistently fall within the order of 10^{-7} to 10^{-9} , significantly outperforming the prescribed tolerance of 10^{-5} . The results are illustrated in Figure 7.

Then, we investigated the impact of the eigenvalue approximation tolerance $\epsilon \in \{10^{-4}, 10^{-5}, 10^{-6}\}$ on the solution time. Experiment results in Figure 8 show that the algorithm is empirically highly robust with respect to this parameter.

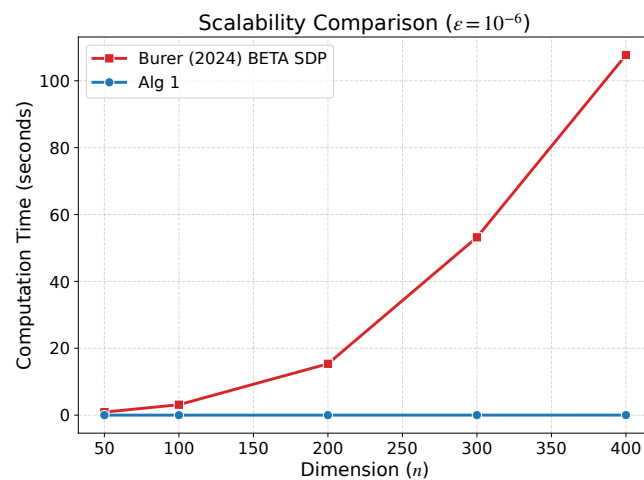


Figure 6. Result for $\epsilon = 10^{-6}$.

To investigate the impact of spectral aliasing on the algorithm, we constructed three extreme spectral distribution test cases ($n = 1000, \kappa = 10$): Uniform, with eigenvalues evenly spaced; Top-Clustered, with eigenvalues clustered near λ_{max} ; and Bottom-Clustered, with eigenvalues clustered near λ_{min} , which is the worst-case for the algorithm.

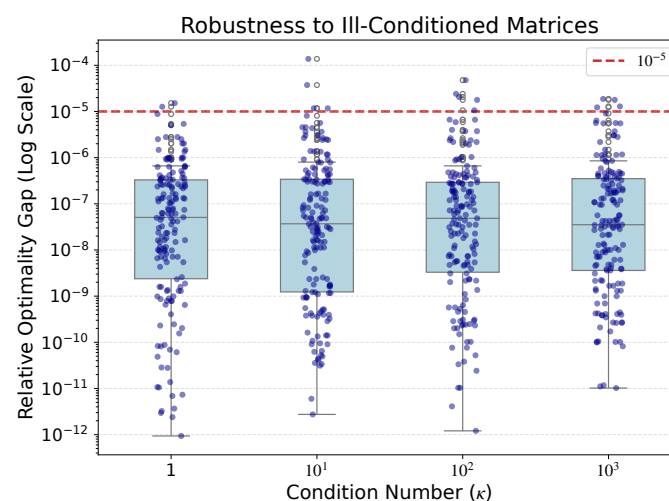


Figure 7. Robustness to ill-conditioned Matrices under different condition number.

Table 2 shows that although clustered spectra increase the time cost in the Lanczos methods stage, the main Nesterov acceleration stage exhibits stable to spectral distributions, converging within approximately six iterations. This finding confirms that the algorithm maintains exceptionally high numerical stability even when handling ill-conditioned problems with complex geometric structures.

Table 2. Robustness analysis across different eigenvalue spectral distributions.

Spectral Distribution	Lanczos (s)	Nesterov (s)	Total (s)	Iterations
Uniform	0.350	0.033	0.383	6.0
Bottom-Clustered	1.491	0.032	1.523	5.4
Top-Clustered	1.571	0.033	1.604	6.0

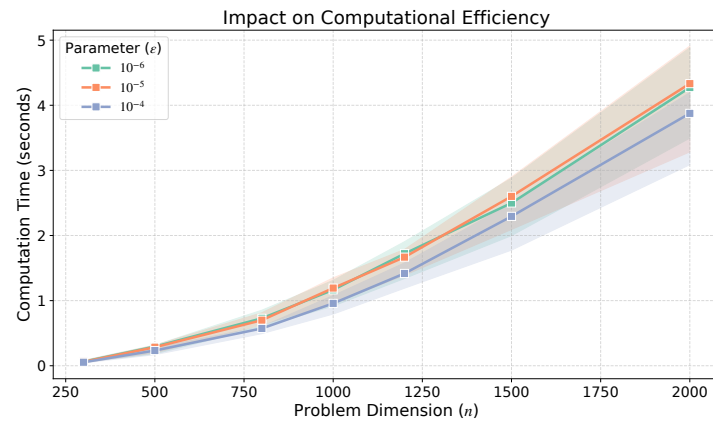


Figure 8. The impact of the eigenvalue approximation tolerance.

Define density d of matrix Q as $d = \frac{m}{n^2}$, where m is the number of nonzero entries in Q and n is the dimension. We also compare the computational time of our Algorithm 1 with $\epsilon = 10^{-6}$ under different density d of matrix Q and show the results in Figures 9 and 10. From the two pictures, we can observe intuitively that for the same dimension, the smaller the density, the faster the computational efficiency. We can also observe that the computational time increases linearly with the dimension.

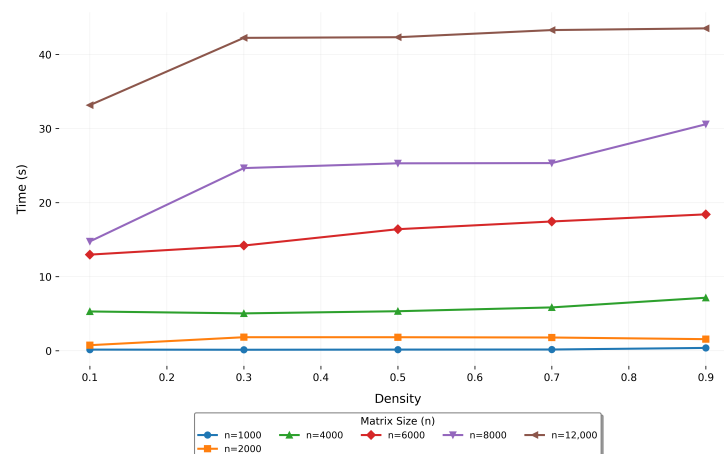


Figure 9. The running time of Algorithm 1 for different density d with $\epsilon = 10^{-6}$.

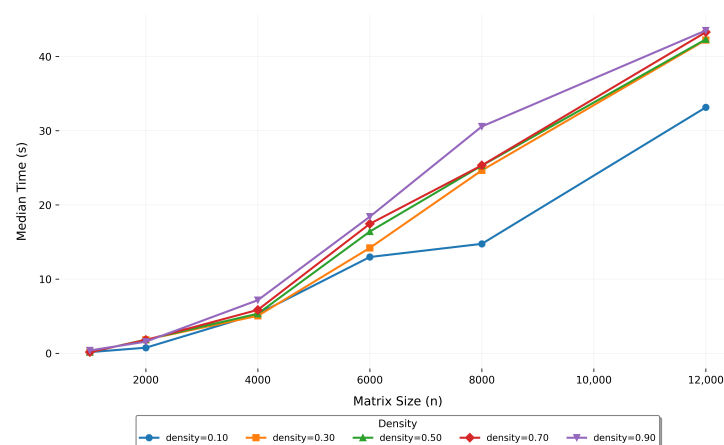


Figure 10. The running time of Algorithm 1 for different matrix size n with $\epsilon = 10^{-6}$.

5. Conclusions

In this paper, we propose a linear-time ϵ -approximation scheme for the n -dimensional complex problem UCDT. We first establish the hidden convexity of UCDT via the S-lemma and an equivalent convex reformulation (C). Subsequently, by employing eigenvalue approximation and the complex-to-real transformation (C), we derive an efficiently solvable real approximate problem (AC^r). Finally, we solve (AC^r) using a generalized Nesterov accelerated gradient descent algorithm, achieving an overall linear-time complexity. Numerical experiments confirm that the proposed algorithm significantly outperforms the standard CVX solver in terms of computational efficiency, while demonstrating strong robustness and insensitivity to the condition number.

However, the proposed algorithm relies on Assumptions 1 and 2. When Slater's condition fails, or one constraint is redundant, the equivalence to (C) may break down, and the algorithm may require a fallback to the SDP formulation. Moreover, the linear-time guarantee is asymptotic and may be affected by the conditioning of Q in practice, as the Lipschitz constant ρ depends on A .

Our future work includes extending the algorithm to the general complex CDT subproblem and applying the proposed algorithm within an SQP framework for nonconvex complex nonlinear programming. In addition, we will also focus on the hidden convexity of more nonconvex optimization problems and designing fast global and approximate algorithms for the real UCDT subproblem and the CDT subproblem.

Author Contributions: Conceptualization, Z.X. and M.Y.; methodology, Z.X. and M.Y.; software, C.M.; validation, Z.X., C.M., and M.Y.; formal analysis, M.Y.; investigation, Z.X. and M.Y.; resources, Z.X. and M.Y.; data curation, C.M.; writing—original draft preparation, Z.X. and C.M.; writing—review and editing, Z.X. and C.M.; visualization, C.M.; supervision, M.Y.; project administration, M.Y.; funding acquisition, Z.X. and M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grants 12401410, 12101041, by the Interdisciplinary Research Project for Young Teachers of USTB (Fundamental Research Funds for the Central Universities) under grant FRF-IDRY-24-026, by the Beijing Natural Science Foundation under grant 4252006.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used Deepseek-V4 for the purposes of text polishing and grammar checking. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Xia, Y. A survey of hidden convex optimization. *J. Oper. Res. Soc. China* **2020**, *8*, 1–28. [[CrossRef](#)]
2. Hazan, E.; Koren, T. A linear-time algorithm for trust region problems. *Math. Program.* **2016**, *158*, 363–381.
3. Beck, A.; Vaisbourd, Y. Globally solving the trust region subproblem using simple first-order methods. *SIAM J. Optim.* **2018**, *28*, 1951–1967. [[CrossRef](#)]
4. Ho-Nguyen, N.; Kılınç-Karzan, F. A second-order cone based approach for solving the trust region subproblem and its variants. *SIAM J. Optim.* **2017**, *27*, 1485–1512. [[CrossRef](#)]

5. Wang, J.; Xia, Y. A linear-time algorithm for the trust region subproblem based on hidden convexity. *Optim. Lett.* **2017**, *11*, 1639–1646.
6. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer: Berlin/Heidelberg, Germany, 2006.
7. Conn, A.; Gould, N.; Toint, P. *Trust-Region Methods*; SIAM: Philadelphia, PA, USA, 2000.
8. Celis, M.; Dennis, J.; Tapia, R. A Trust Region Strategy for Nonlinear Equality Constrained Optimization. In *Numerical Optimization*; Boggs, R.T., Byrd, R.H., Schnabel, R.B., Eds.; SIAM: Philadelphia, PA, USA, 1984; pp. 71–82.
9. Peng, J.; Yuan, Y. Optimality conditions for the minimization of a quadratic with two quadratic constraints. *SIAM J. Optim.* **1997**, *7*, 579–594. [[CrossRef](#)]
10. Bienstock, D. A note on polynomial solvability of the CDT problem. *SIAM J. Optim.* **2016**, *26*, 488–498. [[CrossRef](#)]
11. Consolini, L.; Locatelli, M. On the complexity of quadratic programming with two quadratic constraints. *Math. Program.* **2017**, *164*, 91–128.
12. Sakaue, S.; Nakatsukasa, Y.; Takeda, A.; Iwata, S. Solving generalized CDT problems via two-parameter eigenvalues. *SIAM J. Optim.* **2016**, *26*, 1669–1694. [[CrossRef](#)]
13. Armijo, N.; Gómez, W.; Concha, J. On the exactness of a simple relaxation for the extended Celis-Dennis-Tapia subproblem. *Optimization* **2023**, *72*, 3137–3162.
14. Song, M.; Liu, H.; Wang, J.; Xia, Y. On local minimizers of nonconvex homogeneous quadratically constrained quadratic optimization with at most two constraints. *SIAM J. Optim.* **2023**, *33*, 267–293. [[CrossRef](#)]
15. Kelly, S.; Ouyang, Y.; Yang, B. Semidefinite representable reformulations for two variants of the trust-region subproblem. *Oper. Res. Lett.* **2023**, *51*, 695–701. [[CrossRef](#)]
16. Burer, S. A slightly lifted convex relaxation for nonconvex quadratic programming with ball constraints. *Math. Program.* **2025**, *211*, 157–179.
17. Beck, A.; Eldar, Y. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM J. Optim.* **2006**, *17*, 844–860. [[CrossRef](#)]
18. Beck, A. Convexity properties associated with nonconvex quadratic matrix functions and applications to quadratic programming. *J. Optim. Theory Appl.* **2009**, *142*, 1–29. [[CrossRef](#)]
19. Ai, W.; Wei, L.; Yuan, J. On the tightness of an SDP relaxation for homogeneous QCQP with three real or four complex homogeneous constraints. *Math. Program.* **2025**, *211*, 5–48.
20. Palomar, D.; Eldar, Y. *Convex Optimization in Signal Processing and Communications*; Cambridge University Press: Cambridge, UK, 2020.
21. Fradkov, A.; Yakubovich, V. The S-procedure and duality relations in nonconvex problems of quadratic programming. *Vestn. Leningr. Univ. Leningr. Russ.* **1979**, *6*, 101–109. English translation. *Vestn. Leningr. Univ. Seriya Mat.* **1973**, *1*, 81–87.
22. Flippo, E.; Jansen, B. Duality and sensitivity in nonconvex quadratic optimization over an ellipsoid. *Eur. J. Oper. Res.* **1996**, *94*, 167–178. [[CrossRef](#)]
23. Kuczynski, J.; Wozniakowski, H. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.* **1992**, *13*, 1094–1122. [[CrossRef](#)]
24. Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence. *Sov. Math. Dokl.* **1983**, *27*, 372–376.
25. Tseng, P. On accelerated proximal gradient methods for convex-concave optimization. *SIAM J. Optim.* **2008**, *submit.* [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.