Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article
JMM

A fast and cheap approach for strengthening Lagrangian bound for the generalized Celis-Dennis-Tapia subproblem

Temadher Alassiry Almaadeed[†], Akram Taati[‡], Abdelouahed Hamdi[†*]

[†] *Department of Mathematics and Statistics, College of Arts and Sciences, Qatar University, P.O. Box 2713, Doha- Qatar.*
[‡] *Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran.*
Email(s): t.alassiry@qu.edu.qa, akram.taati@gmail.com, abhamdi@qu.edu.qa

Abstract. In this paper, we consider the generalized Celis-Dennis-Tapia problem which is the problem of minimizing a nonconvex quadratic function subject to two quadratic inequality constraints, one of which being convex. When there is a positive duality gap, by exploiting an equivalent form of the dual Lagrangian problem, we propose to improve the dual bound by adding one or two linear cuts to the Lagrangian relaxation. The present work is motivated by and generalizes the results in [8](L. Consolini and M. Locatelli, "Sharp and fast bounds for the Celis-Dennis-Tapia problem", SIAM Journal on Optimization, 33(2), 868-898, 2023) for the problem with two strictly convex quadratic constraints. Our main contribution is to show that one can include the feasible region in a convex set and then follow the approach in [8] to construct the linear cuts based on supporting hyperplanes of the convex set. Numerical experiments are conducted to assess the quality of the proposed bounds.

Keywords: Quadratically constrained quadratic programming, Celis-Dennis-Tapia problem, dual Lagrangian bound, supporting hyperplane.
AMS Subject Classification 2010: 90C20, 90C30, 90C46, 52A20.

1 Introduction

In this paper, we consider the following nonconvex quadratic programming problem with two quadratic constraints:

$$\begin{aligned} p^* := \min \quad & f_0(x) := x^T Q_0 x + 2q_0^T x \\ s.t. \quad & f_i(x) := x^T Q_i x + 2q_i^T x + \gamma_i \le 0, \quad i = 1,2, \end{aligned} \tag{1}$$

*Corresponding author
Received: 11 October 2025/ Revised: 06 December 2025/ Accepted: 08 December 2025
DOI: 10.22124/jmm.2025.31942.2884

where $Q_i \in \mathbb{R}^{n\times n}$ is a symmetric matrix, $q_i \in \mathbb{R}^n$ and $\gamma_i \in \mathbb{R}$ for $i = 0,1,2$. We further assume that Q_1 is positive definite.

Problem (1) covers the Celis-Dennis-Tapia (CDT) problem as a special case where Q_1 is positive definite and Q_2 is positive semidefinite. Hence, we call problem (1) the generalized CDT (GCDT) problem. The CDT problem first appeared in the trust-region method developed by Celis, Dennis, and Tapia to solve equality constrained nonlinear problems [7]. Several papers have studied the properties of the CDT problem and its solution methods under various assumptions [2, 4–6, 8–14, 16]. In addition to the CDT problem, other nonconvex situations have been explored in the literature, particularly in the works of Hamdi, which discuss modified Bregman proximal schemes and Moreau-Yosida regularization methods for the minimization of differences of two convex functions [9, 10].

Moreover, some papers have investigated strengthening the SDP-relaxation of the CDT problem when it is not exact. In [3, 6], the authors proposed strengthening the SDP-relaxation by adding some valid second-order cone constraints to the SDP-relaxation. The resulting relaxations are still not exact, but it has been shown that they are able to solve many instances that are not solved by the SDP-relaxation. In [19], the authors considered the GCDT problem and showed that the duality gap can be narrowed by adding an appropriate second-order cone constraint to the SDP-relaxation. However, no numerical evidence is reported by the authors to assess their theoretical foundation computationally. Furthermore, it is proved that the duality gap can be eliminated by solving two subproblems with second-order cone constraints when the second constraint is the product of two linear functions. However, due to the relatively large computational complexity, the relaxations mentioned above are not practical for large-scale applications. Most recently, Consolini and Locatelli [8] explored strengthening the Lagrangian dual bound of the CDT problem through the addition of one or two linear cuts to the Lagrangian dual problem. It has been shown that the new bounds are computationally efficient and require limited computing time. The bounds are based on the following fact. When strong Lagrangian duality fails for the CDT problem, there exist exactly two optimal solutions x_1^* and x_2^* of

$$\begin{aligned} \min \quad & x^T(Q_0+\lambda^* Q_2)x + 2(q_0+\lambda^* q_2)^T x + \lambda^*\gamma_2 \\ s.t. \quad & x^T x \le 1, \end{aligned} \tag{2}$$

such that ${x_i^*}^T x_i^* = 1, i = 1,2$, $f_2(x_1^*) < 0$, and $f_2(x_2^*) > 0$, where $\lambda^* > 0$ is the optimal solution of an equivalent form of the Lagrangian dual problem [8]. Define $\omega := \{x | f_2(x) \le 0\}$; the set ω is convex since $Q_2 \succ 0$ and $x_2^* \notin \omega$. Therefore, ω and x_2^* can be separated by a hyperplane. In particular, they can be separated by the supporting hyperplane of ω at the projection of x_2^* over $\mathrm{bd}(\omega)$. The authors exploited such a hyperplane to define the linear cuts. Since the linear cuts are the supporting hyperplanes of the convex set w, they are redundant for the CDT problem but not for problem (2). Therefore, their addition allows for an improvement of the bound.

In this paper, motivated by [8], we explore strengthening the Lagrangian dual bound of the GCDT problem by adding one or two linear cuts to its Lagrangian dual problem. In [8], it has been proved that in case the Lagrangian dual bound of the GCDT problem is not exact, it can be improved if we are able to find a set X such that $\omega \subset X$ and $x_2^* \notin X$. For the GCDT problem, if Q_2 is not positive semidefinite, then the set ω is nonconvex. Therefore, in the general case, w and x_2^* cannot be separated by a hyperplane. It is worth noting that, in this case, no specific method has been proposed in [8] to construct the set X. In this paper, we address this difficulty in applying the approach of [8] for the CDT problem to the GCDT problem.

The rest of the paper is organized as follows. In Section 2, we first briefly review the approach of [8] to improve the Lagrangian dual bound of the CDT problem when it is not exact. Then, in Section 3, we generalize the idea of [8] to the GCDT problem. Numerical experiments are conducted in Section 4 to assess the quality of the proposed bounds. Finally, conclusions are drawn in Section 5.

Notations: Throughout this paper, for the symmetric matrix A, $\lambda_{\min}(A)$ denotes the smallest eigenvalue of A. Moreover, $\mathrm{bd}(S)$ denotes the boundary of the set S.

2 Review of the bound improvement of the CDT problem from [8]

In this section, we briefly review the approach of [8] to improve the dual Lagrangian bound of the CDT problem, i.e., problem (1) with positive definite Q_2. Then, in the next section, we generalize the idea to the GCDT problem, i.e., problem (1) with $Q_2 \not\succ 0$. Without loss of generality, from now on, we assume that $f_1(x) = x^T x - 1$. Moreover, we consider the following assumption on problem (1).

Assumption 1. Problem (1) satisfies the Slater condition, i.e., there exists $\hat{x}$ such that $f_1(\hat{x}) < 0$ and $f_2(\hat{x}) < 0$.

In [8], the authors have discussed a class of lower bounds on the solution to the problem (1) that can be obtained from its Lagrangian relaxation as follows. Define $\omega := \{x | f_2(x) \leq 0\}$. Let D denote the feasible region of problem (1) and $X \subset \mathbb{R}^n$ be a closed set such that $D \subseteq X$. Define

$$d_X^* := \max_{\lambda \geq 0} \quad \theta_X(\lambda) \tag{3}$$

where

$$\begin{aligned} \theta_X(\lambda) = \min \quad & x^T(Q_0 + \lambda Q_2)x + 2(q_0 + \lambda q_2)^T x + \lambda \gamma_2 \\ s.t \quad & x^T x \leq 1, \\ & x \in X. \end{aligned} \tag{4}$$

Obviously, for each $\lambda \geq 0$, problem (4) is a relaxation of (1) and most importantly, we have $d^* \leq d_X^* \leq p^*$, where d^* is the dual Lagrangian bound of problem (1). Let $S_X(\lambda)$ denote the set of all optimal solutions of problem (4) and λ_X^* be the maximizer of problem (3). In the case where the lower bound d_X^* is not exact, i.e., $d_X^* < p^*$, it has been proved in Proposition 2.12 of [8] that one can improve it if we are able to replace the set X with a new set Y which cut away all members of $S_X(\lambda_X^*)$ outside the set ω. Consider the CDT problem. When strong Lagrangian duality fails for the CDT problem, there exist exactly two optimal solutions x_1^* and x_2^* of problem (4) for $\lambda = \lambda_{\mathbb{R}^n}^*$ such that $f_2(x_1^*) < 0$ and $f_2(x_2^*) > 0$ (Proposition 3.2 of [8]). Since $Q_2 \succ 0$, the set ω is convex and $x_2^* \notin \omega$. Therefore, ω and x_2^* can be separated by a hyperplane. In particular, they can be separated by the supporting hyperplane of ω at the projection of x_2^* over $\mathrm{bd}(\omega)$, $\bar{x}$. In [8], the authors exploited such hyperplane to define the set X as

$$X = \Omega_{\bar{x}} = \{x | \nabla f_2(\bar{x})^T (x - \bar{x}) \leq 0\}. \tag{5}$$

Since $f_2(x)$ is convex and $f_2(\bar{x}) = 0$, we have $f_2(x) \geq \nabla f_2(\bar{x})(x - \bar{x})$. Therefore, for all $x \in D$, we have $D \subseteq X$. This choice of X also ensures that $x_2^* \notin X$. Then it follows from Proposition 2.12 of [8] that one can strictly improve the lower bound d^* by solving problem (3). If $d_{\Omega_{\bar{x}}}^*$ is not exact, i.e., $d_{\Omega_{\bar{x}}}^* < p^*$, the authors in [8] also have investigated improvement of the bound based on successive local adjustments of the linear cut through Algorithm 4.1 and adding a further linear cut to the set $\Omega_{\bar{x}}$, possibly followed by a local adjustment of the two linear cut. For more details, see Sections 4 and 5 of [8].

3 Bound improvement of the GCDT problem

In this section, we consider the GCDT problem where $Q_2 \nsucceq 0$ and generalize the idea of [8] to improve the Lagrangian dual bound when it is not exact.

Let x_1^*, x_2^*, λ^*, ω and D be defined as before. For the GCDT problem, except the case where Q_2 is positive semidefinite but singular, the set ω is nonconvex. Therefore, in the general case, w and x_2^* can not be separated by a hyperplane; see Figure 1. In such situation, unlike the CDT problem, no specific

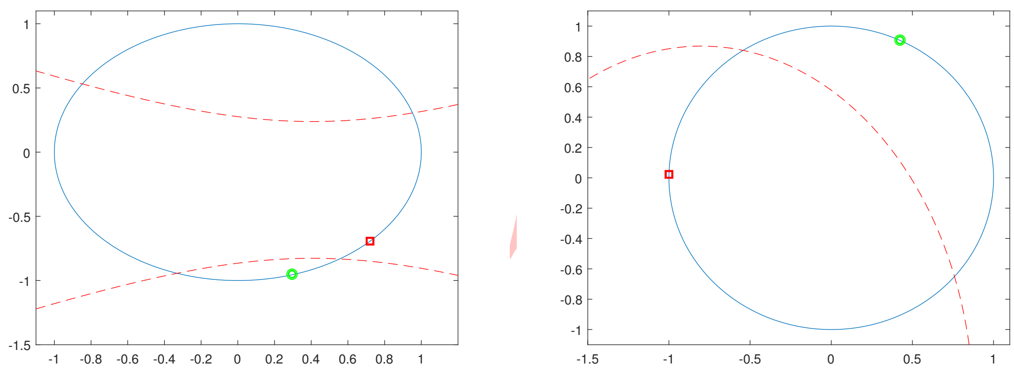

Figure 1: The solid line ('−'): $f_1(x) = 0$, the dashed line ('- -'): $f_2(x) = 0$, □: x_2^* the optimal solution of problem (4) for $\lambda = \lambda^*_{\mathbb{R}^n}$ outside the set ω, 'o': x_1^* the optimal solution of problem (4) for $\lambda = \lambda^*$ in the interior of ω

method has been proposed in [8] to construct the set X. This is the main difficulty in applying the approach of [8] for the CDT problem to the GCDT problem. To address this difficulty, we first introduce a convex set denoted by ω' such that $D \subseteq \omega'$ and $x_2^* \notin \omega'$. Then we exploit the separating hyperplane for ω' and x_2^*, in particular, the one that is the supporting hyperplane of ω' at projection of x_2^* over $\mathrm{bd}(\omega')$ to define the set X as in [8]. Assuming that $\lambda_{\min}(Q_2) < 0$, define $\hat{f}_2(x) := f_2(x) - (\lambda_{\min}(Q_2) - \delta)(x^T x - 1)$ where δ is a small positive constant and $\omega' := \{x | \hat{f}_2(x) \leq 0\}$. The following result plays an important role in the rest of the paper.

Lemma 1. *Let $\lambda_{min}(Q_2) < 0$, D, ω and ω' be defined as before. Then*

1. *ω' is a convex set.*
2. *$D \subseteq \omega'$.*
3. *Let $x^T x = 1$. Then $x \notin \omega$ if and only if $x \notin \omega'$.*
4. *Let $x^T x = 1$. Then $f_2(x) = 0$ if and only if $\hat{f}_2(x) = 0$.*
5. *Let $x^T x < 1$ and $\hat{f}_2(x) = 0$. Then $f_2(x) > 0$.*

Proof. 1. Since the Hessian of $\hat{f}_2(x)$ is positive definite, ω' is a convex set.

2. Let $x \in D$. Then $x^T x \leq 1$ and $f_2(x) \leq 0$. Therefore, $f_2(x) - (\lambda_{\min}(Q_2) - \delta)(x^T x - 1) \leq 0$, and thus $x \in \omega'$.

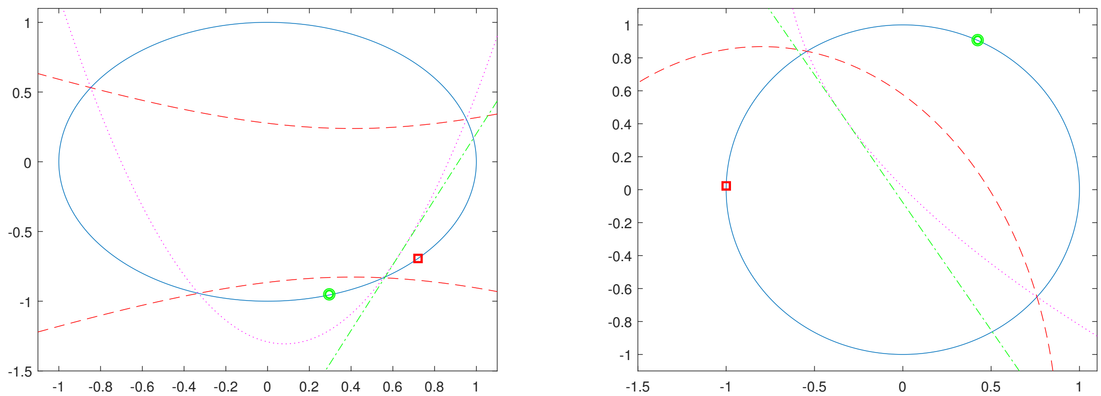

Figure 2: The solid line ('–'): $f_1(x) = 0$, the dashed line ('- -'): $f_2(x) = 0$, the dotted line (':'): $\hat{f}_2(x) = 0$, the dashed-dot line ('-.'): the hyperplane in $\bar{\Omega}_{\bar{x}}$, □: x_2^* the optimal solution of problem (4) for $\lambda = \lambda^*_{\mathbb{R}^n}$ outside the set ω, 'o': x_1^* the optimal solution of problem (4) for $\lambda = \lambda^*_{\mathbb{R}^n}$ in interior of ω

3. $x \notin \omega$ if and only if $f_2(x) > 0$. Furthermore, since $x^T x - 1 = 0$, $f_2(x) > 0$ if and only if $\hat{f}_2(x) > 0$ which completes the proof.
4. The proof is similar to part (3).
5. We have $f_2(x) = \hat{f}_2(x) + (\lambda_{\min}(Q_2) - \delta)(x^T x - 1)$. Since $\hat{f}_2(x) = 0$ and $x^T x < 1$, we have $f_2(x) > 0$. □

Since ${x_2^*}^T x_2^* = 1$ and $x_2^* \notin \omega$, it follows from Lemma 1 that $x_2^* \notin \omega'$. Let $\bar{x}$ be the projection of x_2^* over $\mathrm{bd}(\omega')$. Then we exploit the supporting hyperplane of ω' at $\bar{x}$ to define the set X as

$$X = \bar{\Omega}_{\bar{x}} = \{x | \nabla \hat{f}_2(\bar{x})^T (x - \bar{x}) \leq 0\}.$$

See Figure 2 where we show the set ω' and the supporting hyperplane in $\bar{\Omega}_{\bar{x}}$ corresponding to the examples in Figure 1.

Since $x_2^* \notin \bar{\Omega}_{\bar{x}}$, it follows from Proposition 2.12 of [8] that we can strictly improve the lower bound d^* by solving problem (3) with $X = \bar{\Omega}_{\bar{x}}$. We apply Algorithm 2.1 of [8] to solve problem (3). To do this, we initialize Algorithm 2.1 with $X = \bar{\Omega}_{\bar{x}}$ and $\lambda_{\text{init}} = \lambda^*_{\mathbb{R}^n}$. Moreover, we apply Algorithm 1 of [18] to solve problem (4) that is an eTRS with one linear inequality constraint. Let x^* be an optimal solution of problem (4) with $X = \bar{\Omega}_{\bar{x}}$ and $\lambda = \lambda^*_{\bar{\Omega}_{\bar{x}}}$. Then either $\nabla \hat{f}_2(\bar{x})^T (x^* - \bar{x}) < 0$ or $\nabla \hat{f}_2(\bar{x})^T (x^* - \bar{x}) = 0$. In the latter case, x^* is an optimal solution of problem (4) with the equality linear constraint:

$$\begin{aligned} \min \quad & x^T (Q_0 + \lambda Q_2) x + 2(q_0 + \lambda q_2)^T x + \lambda \gamma_2 \\ s.t \quad & x^T x \leq 1, \\ & \nabla \hat{f}_2(\bar{x})^T (x - \bar{x}) = 0. \end{aligned} \tag{6}$$

Let $S^{eq}_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$ denote the set of all optimal solutions of problem (6). Since $0 \in \partial \theta_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$, it is easy to verify that (i) $S_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}}) = S^{eq}_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$ with $\bar{x} \in S^{eq}_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$ and $\bar{x}^T \bar{x} = 1$, or (ii) $S_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}}) = S^{eq}_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}}) \cup \{x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}\}$

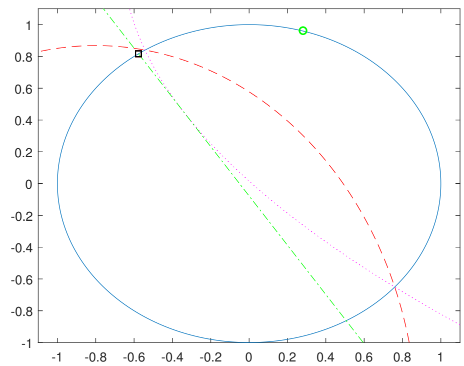

Figure 3: The dashed-dot line ('-.'): the supporting hyperplane of ω' at $\bar{x}$, $\square$: the optimal solution of (4) outside ω, 'o': the optimal solution of (4) in interior of ω

with $f_2(x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}) \leq 0$ or (iii) $S_{\bar{\Omega}_{\bar{x}}}(\lambda) = \{x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}\}$ with $f_2(x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}) = 0$ where $x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}$ is the local non-global minimizer of problem (4) without the linear constraint [18]. Note that in case (i), we have $\bar{x}^T\bar{x} = 1$ and it is the optimal solution of problem (1). If $\bar{x}^T\bar{x} < 1$, then by Lemma 1 and the fact that $\hat{f}_2(\bar{x}) = 0$, we have $f_2(\bar{x}) > 0$ and consequently in this case $0 \notin \partial\theta_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$ that is a contradiction. We know that the lower bound $d^*_{\bar{\Omega}_{\bar{x}}}$ is exact, i.e., $d^*_{\bar{\Omega}_{\bar{x}}} = p^*$, if and only if there exists $x^* \in S_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$ such that $f_2(x^*) = 0$. Therefore, when $d^*_{\bar{\Omega}_{\bar{x}}} < p^*$, it holds that $S_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}}) = S^{eq}_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}}) \cup \{x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}\}$, $f_2(x^{\ell}_{\lambda^*_{\bar{\Omega}_{\bar{x}}}}) < 0$ and $f_2(x^*) > 0$ for all $x^* \in S^{eq}_{\bar{\Omega}_{\bar{x}}}(\lambda^*_{\bar{\Omega}_{\bar{x}}})$. In the other words, there exists one optimal solution of (4) in interior of ω and at least one optimal solution outside ω. We illustrate all this in Figure 3. When $d^*_{\bar{\Omega}_{\bar{x}}}$ is not exact, the same approaches as in Sections 4 and 5 of [8] where set H is replaced by ω' and $f_2(x)$ is replaced by $\hat{f}_2(x)$ can be applied to possibly further improve the lower bound $d^*_{\bar{\Omega}_{\bar{x}}}$.

4 Numerical experiments

In this section, we present the numerical results of the proposed bounds:

- `LbDual`, the Lagrangian dual bound returned by Algorithm 2.1 of [8] with $X = \mathbb{R}^n$, i.e., d^*,
- `LbOneCut`, the bound obtained by adding one linear cut, i.e., $d^*_{\bar{\Omega}_{\bar{x}}}$,
- `LbOneAdj`, the bound obtained by local adjustments of the linear cut returned by Algorithm 4.1 of [8],
- `LbTwoCut`, the bound obtained by adding two linear cuts.
- `LbTwoAdj`, the bound obtained by local adjustments of the two linear cuts,

on numerous randomly generated instances of problem (1) to demonstrate the effectiveness and applicability of the proposed bounds in practical scenarios.

As discussed in Section 3, when strong Lagrangian duality does not hold for the GCDT problem, we can improve the bound `LbDual` by computing the bound `LbOneCut`. Furthermore, when the bound

LbOneCut is not exact, we may be able to improve it by computing the bound LbOneAdj or LbTwoCut. Finally, if the bound LbTwoCut is not exact, we may be able to improve it by computing the bound LbTwoAdj. Hence, to report the numerical results, we follow the following steps:

- **Step 1.** Compute the bound LbDual. If the instance is solved by the bound LbDual, then stop, otherwise go to Step 2.

- **Step 2.** Compute the bound LbOneCut. If the instance is solved by the bound LbOneCut, then stop, otherwise go to Step 3.

- **Step 3**. Compute the bounds LbOneAdj and LbTwoCut. If the instance is not solved by the bound LbTwoCut, then go to Step 4.

- **Step 4.** Compute the bound LbTwoAdj.

Let LB denote the proposed bound. As in [6, 8], we say that an instance is solved by the bound if the following relative gap (RG) is less than 10^{-4}:

$$\mathrm{RG} = \frac{UB - LB}{|UB|} < 10^{-4},$$

where UB is an upper bound. After the computation of the bound LB, i.e., d_X^*, for $\lambda = \lambda_X^*$, there exists an optimal solution of problem (4), x_X^*, that is also feasible for problem (1), i.e., $f_2(x_X^*) \leq 0$. We set UB equal to the lowest objective value of problem (1) at such feasible solutions obtained after the computation of the bound LB and all the previously computed bounds by which the instance was not solved.

To compute the bounds LbOneCut and LbOneAdj, we apply Algorithm 1 of [18] to solve problem (4) and to compute the bounds LbTwoCut and LbTwoAdj, we apply the 2-eTRS Algorithm in [15] to solve problem (4). To assess the quality of the computed bounds, we compare them with the optimal objective value returned by Algorithm 6.1 from [17]. The comparison with Algorithm 6.1 of [17] is conducted for dimensions up to 25 since it requires a longer time to solve larger dimensions; for instance, its run time for a GCDT problem with $n = 50$ was about 13344 seconds. To measure the accuracy of the computed bounds, we have computed the relative objective function difference as follows:

$$\text{Accuracy} = \frac{p^* - LB}{|p^*|}, \tag{7}$$

where p^* is the optimal objective value returned by Algorithm 6.1 of [17] and LB is the computed bound. All computations are performed in MATLAB R2023a on a 1.80 GHz laptop with 16 GB of RAM.

4.1 First class of test problems

In this subsection, we test the proposed bounds on instances of problem (1) for which strong Lagrangian duality does not hold [1]. To this end, we set $Q_1 = I$, $q_1 = 0$, and $\gamma_1 = -1$. Then, we construct a positive semidefinite matrix H with $\text{Rank}(H) = n-1$ as $H = V^T DV$, where V is an orthogonal matrix generated via `V=gallery('orthog',n,2)` and D is a diagonal matrix whose first diagonal entry is zero and the other diagonal entries are generated uniformly in $[0.5,2]$. Then we set $Q_0 = H - \mu I - \lambda Q_2$, where Q_2 is a diagonal matrix whose diagonal entries are generated uniformly in $[-2,1]$, μ and λ are chosen uniformly from $[1,2]$ and $[0.5,1]$, respectively. Next, we set $q_0 = -Hx_0 - \lambda q_2$, where x_0 is computed via `x0=randn(n,1)/norm(x0)` and q_2 is a vector of uniformly distributed random numbers in the interval $(0,1)$. Finally, we set $\gamma_2 = \frac{c_1+c_2}{2}$ if $|c_1 - c_2| > 0.1$, where $c_1 = -(x_1^T Q_2 x_1 + 2q_2^T x_1)$, $c_2 = -(x_2^T Q_2 x_2 + 2q_2^T x_2)$, and x_1 and x_2 are the solutions of the system of linear equations $Hx = -(q_0 + \lambda q_2)$ satisfying $x_i^T x_i = 1$ and $i = 1,2$. Otherwise, we repeat the procedure until $|c_1 - c_2| > 0.1$ holds.

For each dimension, we generated 50 instances, and the corresponding numerical results are reported in Tables 1, 2, and 3. In Tables 1 and 2, we compare the proposed bounds with Algorithm 6.1 of [17] for dimensions up to 25, as it takes longer to solve larger dimensions. Recall that an instance is considered to be solved by the bound when RG $< 10^{-4}$ and the instance was not solved by all the previously computed bounds. Therefore, we say an instance is unsolved if it was solved by none of the proposed bounds. Such instances are those that are solved by neither of the bounds `LbOneAdj` (in Step 3) and `LbTwoAdj` (in Step 4). In Table 1, we compare Algorithm 6.1 of [17] for the solved instances out of the 50 instances, and in Table 2, the comparison is done for unsolved instances out of the same 50 instances. We report the number of solved instances by each bound, denoted by "Nsolved"; the number of unsolved instances out of the 50 instances, denoted by "Nunsolved"; the average computing time of each bound; and the average run time of Algorithm 6.1 of [17] in seconds, denoted by "Time_LB"and "Time_Alg_6.1", respectively; and the average accuracy of each bound over the involved instances, denoted by "Accuracy". Moreover, we report the number of unsolved instances for which Accuracy $< 10^4$ in column "Accuracy $< 10^{-4}$".

As we see in Table 1, 48 instances out of the 50 instances with $n = 10$, 45 instances out of the 50 instances with $n = 20$ and 46 instances out of the 50 instances with $n = 25$ were solved by at least one of the proposed bounds. Moreover, the column "Accuracy" show that the proposed bounds give solutions that are almost as accurate as Algorithm 6.1 of [17]. We also see that the proposed bounds are significantly faster than Algorithm 6.1 of [17] as expected. The time difference become much more significant as dimension increases. This is due to the fact that the proposed bounds are defined based on TRS (`LbDual`), eTRS with one linear cut (`LbOneCut` and `LbOneAdj`) or eTRS with two linear cuts (`LbTwoCut` and `LbTwoAdj`) and that the TRS and eTRSs require at most $O(n^3)$ time (see [15, 18]) while Algorithm 6.1 of [17] requires $O(n^6)$. In Table 2, in column "Accuracy $< 10^{-4}$", we observe that for all the unsolved instances out of the 50 instances, the accuracy of the computed bounds `LbOneAdj` and `LbTwoAdj` are less than 10^{-4}, implying that the bounds give good approximations of the optimal objective value of problem (1) in much too less time than Algorithm 6.1 of [17]. Finally, it is worth mentioning that the value of RG depends on the upper bound UB and it does affect the number of solved instances. Since the optimal objective value returned by Algorithm 6.1 of [17] is the best available upper bound (UB), we see that for all the unsolved instances with RG $\geq 10^{-4}$, we have Accuracy $< 10^{-4}$. For larger dimensions, we just report the results of the proposed bounds since Algorithm 6.1 of [17] can not handle such dimensions (we got out of memory error for $n = 100$ running Algorithm 6.1 of [17]). In Table 3, we report the average relative gap over the solved instances denoted by "ARG(solved)", the average relative gap over unsolved instances denoted by "ARG(unsolved)", the maximum relative gap denoted by "ARG(Max)"and the average computing time over the solved and unsolved instances in seconds denoted by "Time". As we see in Table 3, all the instances were solved by at least one of the proposed bounds.

Table 1: Comparison with Algorithm 6.1 of [17] for the first class of test problems: solved instances

$n = 10$			
Bound	Nsolved	Accuracy	Time_LB(Time_Alg_6.1)
`LbDual`	0	–	–
`LbOneCut`	0	–	–
`LbOneAdj`	46	5.8522e-08	0.03(0.26)
`LbTwoCut`	46	5.4293e-06	0.06(0.26)
`LbTwoAdj`	2	9.2145e-07	0.07(0.23)
$n = 20$			
`LbDual`	0	–	–
`LbOneCut`	0	–	–
`LbOneAdj`	42	1.8474e-07	0.09(28.67)
`LbTwoCut`	41	5.4293e-06	0.04(28.67)
`LbTwoAdj`	4	2.6563e-11	0.08(28.85)
$n = 25$			
`LbDual`	0	–	–
`LbOneCut`	0	–	–
`LbOneAdj`	44	1.6402e-09	0.02(135.42)
`LbTwoCut`	44	5.7309e-06	0.06(135.43)
`LbTwoAdj`	2	6.3685e-11	0.09(135.43)

Table 2: Comparison with Algorithm 6.1 of [17] for the first class of test problems: unsolved instances

n	Nunsolved	Bound	Accuracy	Accuracy $< 10^{-4}$	Time_LB(Time_Alg_6.1)
10	2	`LbOneAdj`	3.7254e-06	2	0.02(0.24)
		`LbTwoAdj`	1.5398e-04	1	0.05(0.24)
20	5	`LbOneAdj`	5.5247e-06	5	0.01(30.61)
		`LbTwoAdj`	5.5287e-06	5	0.07(28.85)
25	4	`LbOneAdj`	8.8533e-07	4	0.11(143.21)
		`LbTwoAdj`	9.1485e-07	4	0.01(140.62)

4.2 Second class of test problems

In [17], the authors based on the idea of [6], generated instances of problem (1) with several candidates for a global optimal solution, that makes the instances challenging. Here, we generate such instances in the same way as follows [17]:

1. Set $Q_1 = I$, $q_1 = 0$ and $\gamma_1 = -n^2$, where n is the dimension of the problem.

Table 3: Computational results of the first class of test problems for larger dimensions

	Nsolved	ARG(solved)	ARG(unsolved)	ARG(Max)	Time(s)
$n = 100$					
LbDual	0	–	3.4166e-01	1.6651e+00	0.0395
LbOneCut	0	–	3.0113e-01	6.3287e-01	0.0527
LbOneAdj	48	4.6234e-09	3.3009e-01	3.3018e-01	0.0789
LbTwoCut	48	1.5519e-05	3.3011e-01	3.3019e-01	0.5636
LbTwoAdj	2	1.8634e-13	–	–	1.3466
$n = 500$					
LbDual	0	–	9.2819e-01	9.9881e+00	0.7842
LbOneCut	0	–	7.4313e-01	1.0563e+01	1.0263
LbOneAdj	47	6.4761e-09	3.0012e+00	8.0329e+00	1.4952
LbTwoCut	49	2.1059e-05	8.0330e+00	8.0320e+00	13.435
LbTwoAdj	1	1.1716e-10	–	–	26.374
$n = 800$					
LbDual	0	–	4.8873e-01	2.2240e+00	1.6244
LbOneCut	0	–	4.0748e-01	3.5908e+00	2.1377
LbOneAdj	49	8.4286ee-10	1.0700e-01	1.0700e-01	3.1288
LbTwoCut	50	1.1853e-05	–	–	27.121
LbTwoAdj	–	–	–	–	–

2. Let Q_0 be a diagonal matrix with diagonal entries uniformly distributed in $[-1,1]$ and generate q_0 with uniform entries in $[-\frac{1}{2},\frac{1}{2}]$.

3. Solve the trust-region subproblem of minimizing $f_0(x)$ subject to $f_1(x) \leq 0$ and let x^* be its optimal solution. Construct an orthogonal matrix V such that $\frac{x^*}{\|x^*\|} = V^T e_1$, where e_1 is the unit vector. Then the TRS instance with $f_0(x) = x^T V Q_0 V^T x + 2q_0^T V^T x$ and $f_1(x) = x^T x - n^2$ has optimal solution ne_1. Update $(Q_0, q_0) \longleftarrow (V Q_0 V^T, V q_0)$.

4. Consider Q_2 as a diagonal matrix where the first diagonal entry is fixed to be 2 and the other diagonal entries are generated uniformly in $[-1,1]$. Set $q_2 = 0$ and $\gamma_2 = -n^2$. This form of Q_2 makes ne_1, the optimal solution of TRS in step 3, infeasible for problem (1).

Similar to the first class of test problems, for each dimension, we generated 50 instances and the corresponding numerical results are reported in Tables 4, 5 and 6. We found out that, for the second class of test problems, Algorithm 6.1 of [17] failed to find the global minimizer of problem (1) for some instances (7 instances with $n = 20$ and 18 instances with $n = 25$). This is justified by the fact that the obtained upper bound UB after computing the bounds was smaller than the optimal objective value returned by Algorithm 6.1 of [17]. In the other words, we found a feasible solution of problem (1) with smaller objective value than the output of Algorithm 6.1 of [17] for those instances, see Figure 4. Hence,

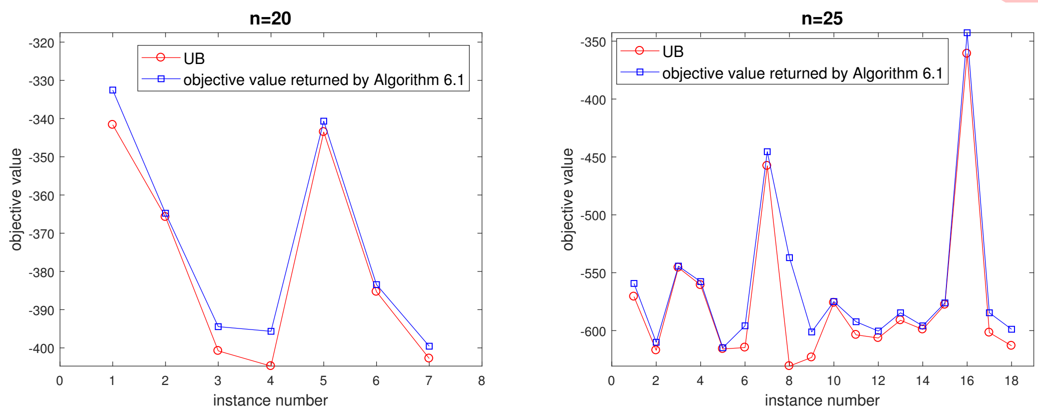

Figure 4: Some instances of the second class of test problems for which the upper bound UB is less than the optimal objective value returned by Algorithm 6.1 of [17]

in Tables 4 and 5, we measure the accuracy as

$$\frac{\min\{p^*, UB\} - LB}{|\min\{p^*, UB\}|},$$

where p^* is the objective value returned by Algorithm 6.1 of [17]. We see again in Tables 4 and 5 that for 96% of the generated instances with $n = 10$, 94 % of the instances with $n = 20$ and 80% of the instances with $n = 25$ (including solved and unsolved instances), at least one of the proposed bounds give an approximation of the optimal objective value of problem (1) with Accuracy $< 10^{-4}$. Moreover, the proposed bounds are always faster than Algorithm 6.1 of [17]. In all tables, we observe that the bound `LbOneAdj` takes more time than the bound `LbOneCut` because of extra time taken for adjusting the linear cut. Moreover, the bound `LbTwoCut` takes more time than the bound `LbOneCut` as it is based on solving the eTRS with two linear cuts rather than the eTRS with one linear cut. Finally, the bound `LbTwoAdj` takes more time than the bound `LbTwoCut` because of extra time taken for adjusting the two linear cuts. The results for larger dimensions are reported in Table 6.

5 Conclusions

This paper focuses on addressing the generalized CDT (GCDT) problem. To tackle this problem, we utilize an equivalent form of the Lagrangian dual problem and generalize the idea of [8] to enhance the dual Lagrangian bound. This enhancement involves incorporating one or two linear cuts into the equivalent form of the Lagrangian dual problem. These linear cuts are derived from the supporting hyper- planes of a convex set that is implied by the two constraints present in the GCDT problem. Our computational experiments conducted on numerous randomly generated instances demonstrate that the introduced bounds offer significant advantages in terms of affordability and efficiency. This is mainly due to their ability to leverage efficient algorithms that have been previously developed for the TRS, eTRS with one linear cut, and eTRS with two linear cuts. These findings highlight the potential of our approach to efficiently solve the GCDT problem, providing a valuable contribution to the field.

Table 4: Comparison with Algorithm 6.1 of [17] for the second class of test problems: solved instances

Bound	Nsolved	Accuracy	Time_LB(Time_Alg_6.1)
$n = 10$			
LbDual	23	9.8202e-10	0.02(0.45)
LbOneCut	10	2.5737e-07	0.08(0.45)
LbOneAdj	0	–	–
LbTwoCut	2	8.4606e-08	0.17(0.52)
LbTwoAdj	2	5.8815e-12	0.37(0.52)
$n = 20$			
LbDual	17	2.8789e-07	0.19(33.93)
LbOneCut	8	1.1183e-06	0.43(33.80)
LbOneAdj	2	1.4130e-08	1.62(34.23)
LbTwoCut	3	3.8025e-08	1.38(34.46)
LbTwoAdj	0	–	–
$n = 25$			
LbDual	24	2.3995e-06	0.20(170.32)
LbOneCut	5	2.4887e-07	0.85(207.3)
LbOneAdj	2	1.9631e-10	2.33(190.18)
LbTwoCut	2	1.9678e-10	2.41(190.18)
LbTwoAdj	–	–	–

Table 5: Comparison with Algorihm 6.1 for the second class of test problems: unsolved instances

n	Nunsolved	Bound	Accuracy	Accuracy $< 10^{-4}$	Time_LB(Time_Alg_6.1)
10	13	LbOneAdj	6.79454-04	9	0.30(0.46)
		LbTwoAdj	1.6120e-03	11	0.34(0.42)
20	22	LbOneAdj	2.1127e-03	15	2.62(33.99)
		LbTwoAdj	2.7124e-04	19	2.13(33.94)
25	19	LbOneAdj	6.9492e-03	7	3.64(165.15)
		LbTwoAdj	5.6511e-03	9	3.40(165.24)

One of the future research direction would be to investigate the ways of strengthening the Lagrangian dual bound of problem (1) with two nonconvex constraints.

Table 6: Computational results of the second class of test problems for larger dimensions

	Nsolved	ARG(solved)	ARG(unsolved)	ARG(MAX)	Time(s)
$n = 100$					
`LbDual`	24	2.1515e-06	7.7101e-03	3.9224e-02	0.2096
`LbOneCut`	9	8.2720e-06	7.9646e-03	3.8760e-02	1.0815
`LbOneAdj`	0	–	7.5612e-03	3.8727e-02	5.9561
`LbTwoCut`	1	1.1556e-07	7.5795e-03	3.8728e-02	2.6933
`LbTwoAdj`	0	–	7.4059e-03	3.8727e-02	5.0415
$n = 500$					
`LbDual`	34	1.4526e-05	1.8857e-03	6.2712e-03	1.0604
`LbOneCut`	8	7.7310e-06	2.5798e-03	6.1471e-03	5.2168
`LbOneAdj`	0	–	2.7157e-03	6.0976e-03	38.880
`LbTwoCut`	0	–	2.3638e-03	5.9516e-03	16.614
`LbTwoAdj`	1	2.4599e-05	2.6616e-03	5.9516e-03	21.443
$n = 800$					
`LbDual`	35	1.6545e-05	1.3831e-03	5.3188e-03	3.4859
`LbOneCut`	7	2.0546e-05	1.7890e-03	5.1269e-03	15.737
`LbOneAdj`	1	2.5804e-05	1.9392e-03	5.1085e-03	86.410
`LbTwoCut`	0	–	1.6751e-03	5.1085e-03	72.832
`LbTwoAdj`	0	–	1.6750e-03	5.1085e-03	112.00

6 Acknowledgement

The authors would like to express their thanks to Qatar University for supporting their project under Grant NCBP-QUCP-CAS-2020-1.

References

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

[2] T.A. Almaadeed, A. Taati, M. Salahi, A. Hamdi, *The generalized trust-region subproblem with additional linear inequality constraints: two convex quadratic relaxations and strong duality*, Symmetry **12(9)** (2020) 1369.

[3] K.M. Anstreicher, *Kronecker product constraints with an application to the two-trust-region subproblem*, SIAM J. Optim. **27(2)** (2017) 368–378.

[4] D. Bienstock, *A note on polynomial solvability of the CDT problem*, SIAM J. Optim. **26(1)** (2016) 488–498.

[5] I.M. Bomze, M.L. Overton, *Narrowing the difficulty gap for the Celis-Dennis-Tapia problem*, Math. Program. **151(2)** (2015) 459–476.

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

[7] M.R. Celis, J.E. Dennis, R. A. Tapia, *A trust region algorithm for nonlinear equality constrained optimization*, in Numerical Optimization, R.T. Boggs, R.H. Byrd, and R.B. Schnabel, eds., SIAM, Philadelphia, (1984) 71–82.

[8] L. Consolini, M. Locatelli, *Sharp and fast bounds for the Celis-Dennis-Tapia problem*, SIAM J. Optim. **33(2)** (2023) 868–898.

[9] A. Hamdi, *A modified Bregman proximal scheme to minimize the difference of two convex functions*, Appl. Math. E-Notes **6(2)** (2006) 132–140.

[10] A. Hamdi, *A Moreau-Yosida regularization of a difference of two convex functions*, Appl. Math. E-Notes **5(2)** (2005) 164–170.

[11] A. Hamdi, A. Taati, T.A. Al-Maadeed, *Quadratic problems with two quadratic constraints: convex quadratic relaxation and strong Lagrangian duality*, RAIRO Oper. Res. **55**, (2021) 2905–2922.

[12] A. Hamdi, M. Salahi, S. Ansary Karbasy, T. A. Al-Maadeed, *Quadratic optimization with a ball and a reverse ball constrains*, Commun. Comb. Optim. **1**, (2025) 1–13.

[13] S. Ansary Karbasy, M. Salahi, *Quadratic optimization with two ball constraints*, Numer. Algebra Control Optim. **10(2)** (2020) 165–175.

[14] S. Ansary Karbasy, M. Salahi, *A hybrid algorithm for the two-trust-region subproblem*, Comput. Appl. Math. **38(3)** (2019) 1–19.

[15] S. Ansary Karbasy, M. Salahi, *An efficient algorithm for the extended trust-region subproblem with two linear constraints*, Bull. Iran. Math. Soc. **48(2)** (2022) 715–737.

[16] G. Li, Y. Yuan, *Compute a Celis-Dennis-Tapia step*, J. Comput. Math. **23(5)** (2005) 463–478.

[17] S. Sakaue, Y. Nakatsukasa, A. Takeda, S. Iwata, *Solving generalized CDT problems via two-parameters eigenvalues*, SIAM J. Optim. **26(3)** (2016) 1669–1694.

[18] M. Salahi, A. Taati, H. Wolkowicz, *Local nonglobal minima for solving large-scale extended trust-region subproblems*, Comput. Optim. Appl. **66(2)** (2017) 223–244.

[19] J. Yuan, M. Wang, W. Ai, T. Shuai, *New results on narrowing the duality gap of the extended Celis-Dennis-Tapia problem*, SIAM J. Optim. **27(2)** (2017) 890–909.