Solving Two-Trust-Region Subproblems using Semidefinite Optimization with Eigenvector Branching

Kurt M. Anstreicher

February 11, 2022

Abstract Semidefinite programming (SDP) problems typically utilize a constraint of the form $X \succeq xx^T$ to obtain a convex relaxation of the condition $X = xx^T$, where $x \in \mathbb{R}^n$. In this paper we consider a new hyperplane branching method for SDP based on using an eigenvector of $X - xx^T$. This branching technique is related to previous work of Saxeena, Bonami and Lee [19] who used such an eigenvector to derive a disjunctive cut. We obtain excellent computational results applying the new branching technique to difficult instances of the two-trust-region subproblem.

Keywords Semidefinite programming · Semidefinite optimization · Conic optimization · Nonconvex quadratic programming · Trust region subproblem

Mathematics Subject Classification (2010) 90C20 · 90C22 · 90C26

1 Introduction

We consider a semidefinite programming (SDP) [21] problem of the form

$$\begin{aligned} \text{SDP} : \min\ & C \bullet Y \\ \text{s.t.}\ & A_i \bullet Y = b_i, \quad i \in \mathcal{E} \\ & A_i \bullet Y \le b_i, \quad i \in \mathcal{I} \\ & Y \succeq 0. \end{aligned}$$

In SDP the matrix Y has the form

$$Y = \begin{pmatrix} Y_{00} & x^T \\ x & X \end{pmatrix},$$

Kurt M. Anstreicher
Department of Business Analytics
University of Iowa
Iowa City, IA, 52242 USA
kurt-anstreicher@uiowa.edu

where $x \in \mathbb{R}^n$. The constraint $Y \succeq 0$, with $Y_{00} = 1$, is then equivalent to $X \succeq xx^T$, which is a convex relaxation of the rank-one condition $X = xx^T$. We will consider the rows and columns of Y to be indexed $0, 1, \ldots, n$, and assume that the equality constraint for $i = 0 \in \mathcal{E}$ is the constraint $Y_{00} = 1$. Since the constraints for all other $i > 0$, $i \in \mathcal{E} \cup \mathcal{I}$ are written in terms of Y, if desired we may assume without loss of generality that $b_i = 0$ for all $i > 0$. This assumption is convenient at some points in the sequel.

Problems of the form SDP have wide applications in approximating difficult Mixed-Integer Nonlinear Programming (MINLP) problems with binary variables and/or nonconvexities in the objective or constraints [5]. The condition that a variable $x_i \in \{0, 1\}$ can be incorporated in SDP via the equality constraint $X_{ii} = x_i$. If the solution of SDP results in such a variable having a nonbinary value then one may *branch* on the binary condition and produce two child problems, one with $x_i = 0$ and the other with $x_i = 1$. If the child problems contain additional binary variables with nonbinary solution values then this process can be continued to produce a *branching tree* which will eventually enforce all binary conditions. Branching is generally viewed as the "method of last resort" since the branching tree is potentially of exponential size in the nunber of binary variables, but in many cases there is no alternative if an exact solution is required.

We are particularly interested in the application of SDP to problems with continuous variables and a nonconvex quadratic objective or constraints. For example, an indefinite quadratic objective $f(x) = x^T Q x + c^T x$ can be modeled in SDP using the matrix

$$C = \begin{pmatrix} 0 & \frac{1}{2}c^T \\ \frac{1}{2}c & Q \end{pmatrix}.$$

A solution with $X = xx^T$ then has $C \bullet X = f(x)$. If the solution of SDP is *not* rank-one, then *spatial branching* can be applied to produce child problems that provide tighter relaxations of the original nonconvex problems. Spatial branching is usually performed by splitting the range of a continuous variable x_i to produce two child problems with constraints $x_i \le \theta$ and $x_i \ge \theta$, respectively, where θ is often taken to be the the value of x_i in the current solution. In order for spatial branching to produce an improvement in the child problems, it is essential that the reduced range for x_i be used to improve the relaxation in each child problem [6]. For example, when SDP contains Reformulation-Linearization Technique (RLT) constraints [20] on (x, X) generated from linear inequality constraints on x, then the reduced range for x_i in the child problems can be used to tighten these constraints. Spatial branching is even less efficient than branching on binary variables, but nevertheless has been successfully applied in global optimization solvers such as BARON [18] and Couenne [13].

Spatial branching based on a single continuous variable may result in child problems that have improved bounds, but such branching has little or nothing to do with the semidefiniteness condition $Y \succeq 0$. It would be desireable to have a branching mechanism that is more closely related to this condition, especially when $Y \succeq 0$ is being enforced through the use of a conic solver. This is exactly the topic that we consider in this paper. In the next section we describe a new branching technique for SDP that we term *eigenvector branching*. Our terminology is based on the use of the eigenvector corresponding to the maximal eigenvalue of $X - xx^T$ as the basis for spatial branching. The use of this maximal eigenvector to strengthen SDP was first suggested in [19], but the approach taken there is not based on branching but rather the construction of a disjunctive cut that can be added to SDP. In the next section we describe eigenvector branching and how different kinds of constraints can be generated that strengthen the resulting child problems. We then apply the technique to a set of difficult instances of the two-trust-region subproblem (TTRS), a well-studied nonconvex quadratic problem. We consider several versions of the algorithm based on differences in branching and the constraints that are added to the child problems. We obtain excellent computational results, particularly when the branching inequalities are used to generate SOC-RLT constraints [10] that are added to the child problems.

In Section 3 we consider an alternative use of the maximal eigenvector of $X - xx^T$, suggested in [19], to generate a disjunctive cut that can be used to strengthen SDP. We extend the methodology in [19], which is based on polyhedral relaxations of SDP, to apply to the conic problem where the condition $Y \succeq 0$ is enforced and SOC-RLT constraints are also present. We then apply an algorithm incorporating these disjunctive cuts to the same set of TTRS instances used in Section 2. Our computational results using disjunctive cuts compare favorably to previous methods based on other classes of cuts but are not as good as the results using eigenvector branching. Examining the instances where the method based on disjunctive cuts fails shows that failure is always due to not generating a sufficiently accurate feasible solution rather than failure to generate an accurate lower bound.

Notation. All matrices are square and symmetric. The matrix inner product is denoted $X \bullet Y = \mathrm{tr}(XY)$, $X \succeq Y$ denotes that $X - Y$ is positive semidefinite, and $X \succ Y$ denotes that $X - Y$ is positive definite. We use e to denote a vector of arbitrary dimension with each component equal to one.

2 Eigenvector branching

We are interested in the situation where a feasible solution to SDP has $\bar{Y} \succeq 0$, but $\bar{X} - \bar{x}\bar{x}^T \neq 0$. In this case there is an $a \in \mathbb{R}^n$ with $a^T(\bar{X} - \bar{x}\bar{x}^T)a > 0$, or equivalently $a^T\bar{X}a > (a^T\bar{x})^2$. The constraint $a^TXa \leq (a^Tx)^2$ certainly holds for all rank-one $X = xx^T$, but this is not a convex constraint in the variables (x, X). However by making a secant approximation of the function $(a^Tx)^2$ we can obtain a valid disjunction, one part of which must hold for any $X = xx^T$ for which Y is feasible in SDP.

Proposition 2.1 ([19]) *Suppose that $\bar{Y}$ is feasible in SDP, with $a^T\bar{x} = \theta$ and $a^T\bar{X}a > \theta^2$. Assume that $\mu^- \leq a^Tx \leq \mu^+$ for all x feasible in SDP. Then if*

Y is feasible in SDP with $X = xx^T$, one of the following must hold:

$$a^T x \geq \theta, \quad a^T Xa \leq (\theta + \mu^+)a^T x - \theta\mu^+, \tag{1}$$
$$a^T x \leq \theta, \quad a^T Xa \leq (\theta + \mu^-)a^T x - \theta\mu^-. \tag{2}$$

Note that $\mu^- \leq a^T x \leq \theta$ is equivalent to $-\theta \leq -a^T x \leq -\mu^-$. It is then obvious that (2) in Proposition 2.1 is exactly (1) with a replaced by $-a$, θ replaced by $-\theta$ and μ^+ replaced by $-\mu^-$. When $\bar{Y} \succeq 0$ but $\bar{X} - \bar{x}\bar{x}^T \neq 0$, a good candidate for the vector a, as suggested in [19], is the eigenvector corresponding to the maximum eigenvalue of $\bar{X} - \bar{x}\bar{x}^T$. In [19] a problem of the form SDP is further relaxed by using linear constraints to approximately enforce the condition $Y \succeq 0$. The eigenvector for a negative eigenvalue of $\bar{X} - \bar{x}\bar{x}^T$ can then be used to generate a cut implied by $Y \succeq 0$, and the eigenvector for a positive eigenvalue can be used along with the disjunction in Proposition 2.1 to generate a valid constraint, or cut, that can be added to the problem. We will consider these disjunctive cuts in the next section.

In this section our intent is to use Proposition 2.1 as the basis for spatial branching in the x variables. We use the term *eigenvector branching* to refer to spatial branching based on the conditions $a^T x \geq \theta$, $a^T x \leq \theta$ as in (1) and (2), where $\bar{Y} \succeq 0$, a is the eigenvector for the maximal eigenvalue of $\bar{X} - \bar{x}\bar{x}^T \neq 0$ and $\theta = a^T \bar{x}$. Then $a^T \bar{X} a > (a^T \bar{x})^2 = \theta^2$, and note that

$$(\theta + \mu^+)a^T \bar{x} - \theta\mu^+ = (\theta + \mu^-)a^T \bar{x} - \theta\mu^- = \theta a^T \bar{x} = \theta^2.$$

Therefore the current point $(\bar{x}, \bar{X})$ is infeasible for both (1) and (2). The upper and lower bounds μ^+ and μ^- in Proposition 2.1 can be obtained in several different ways. One possibility, suggested in [19], is to maximize and minimize $a^T x$ for x feasible in SDP. In some situations it may also be possible to obtain values for μ^+ and μ^- without performing these explicit optimizations. For example, if the constraints of SDP imply that $\|x\| \leq M$, then one may immediately take $\mu^+ = M\|a\|$, $\mu^- = -M\|a\|$.

We will apply eigenvector branching as described above to SDP relaxations of the two-trust-region subproblem (TTRS). The TTRS, also referred to as the Celis-Dennis-Tapia (CDT) problem [11], arises as a direction-finding subproblem in certain trust-region based methods for nonlinear optimization [12]. The TTRS has the form

$$\begin{aligned} \text{TTRS} : \min\ & x^T Qx + c^T x \\ \text{s.t.}\ & x^T x \leq 1, \quad x^T Hx + 2h^T x \leq 1, \end{aligned}$$

where Q is indefinite and $H \succ 0$. The constraints of TTRS can equivalently be written in second-order cone (SOC) form, for example as $\|x\| \leq 1$, $\|H^{1/2}x\|^2 + (h^T x)^2 \leq (1 - h^T x)^2$.

There is a large literature that considers problems of the form TTRS. Optimality conditions for TTRS are given in [8] and [16], and papers such as [1] and [4] give conditions under which the problem can be efficiently solved. A convergent path-following method for TTRS is described in [23]. This method is not provably polynomial-time, but a polynomial-time algorithm for TTRS based on methods for polynomial equations [3] is described in [7].

The basic SDP (Shor) relaxation for TTRS is

$$\begin{aligned} \mathrm{TTRS_{SDP}} : \min\ & Q \bullet X + c^T x \\ \text{s.t.}\ & I \bullet X \le 1, \quad H \bullet X + 2h^T x \le 1, \\ & X \succeq xx^T. \end{aligned}$$

Then $\mathrm{TTRS_{SDP}}$ can easily be written as a problem of the form SDP. It is well known that the relaxation $\mathrm{TTRS_{SDP}}$ can have a nonzero optimality gap, unlike the simpler trust-region subproblem TRS (TTRS without the second ellipsoid constraint), for which the Shor relaxation is tight [17]. Approximation results for the Shor relaxation applied to problems with an indefinite quadratic objective and two or more ellipsoidal constraints are given in [14] and [15].

For our computational tests here we will use a set of TTRS instances that were first considered in [10]. These problems were generated in such a way that they are likely to have a gap for $\mathrm{TTRS_{SDP}}$. If such a gap exists, the approach taken in [10] is to add up to 25 SOC-RLT constraints based on the two ellipsoid constraints of TTRS. At termination, an instance is considered to be solved if the relative gap satisfies

$$\frac{v(\bar{x}) - z(\bar{x}, \bar{X})}{|v(\bar{x})|} < 10^{-4}, \tag{3}$$

where $v(\bar{x}) = \bar{x}^T Q \bar{x} + c^T \bar{x}$, $z(\bar{x}, \bar{X}) = Q \bullet \bar{X} + c^T \bar{x}$ and we are using the fact that if $(\bar{x}, \bar{X})$ is feasible in $\mathrm{TTRS_{SDP}}$ then $\bar{x}$ is feasible in TTRS. When applied to 1000 instances each of size $n = 5$, $n = 10$ and $n = 20$, the resulting numbers of unsolved instances are then 41, 70 and 104, respectively.

The results of [10] are improved on in [22]. The methodology of [22] is based on a detailed study of TTRS for $n = 2$. This approach results in an exact cutting-plane algorithm for $n = 2$ that can be extended heuristically to $n > 2$. When applied to the test problems from [10], the algorithm of [22] solves some of the instances that are unsolved using only SOC-RLT cuts. Due to differences in the solver and parameter settings, the number of instances that are unsolved using SOC-RLT cuts for dimensions 5, 10 and 20 are taken to be 38, 71 and 106, respectively.

The test instances that we will use here are the 38, 70 and 104 problems of size 5, 10 and 20, repectively, that were considered to be unsolved in both [10] and [22]. These same problems were also considered in the computational results of [2]. The approach taken in [2] is to start with the Shor relaxation and then add up to 25 constraints that are obtained from the Kronecker product of the SOC representations of the two ellipsoid constraints in TTRS. It is proved in [2] that these Kronecker product constraints imply the SOC-RLT constraints used in [10], so problems that are solved using the SOC-RLT constraints would always be solved using the Kronecker product constraints. The approach used in [2] also solves some of the instances that are not solved using only SOC-RLT cuts. Table 1, reproduced from [2], shows that the overall results using

Table 1 Comparison of previous results on TTRS test instances

		Number of instances solved in:			
n	Instances	A. [2] only	Y.B. [22] only	Both	Neither
5	38	8	8	12	10
10	70	34	7	14	15
20	104	35	14	24	31
	212	77	29	50	56

the Kronecker product constraints on these instances are better than those from [22], but neither method dominates the other.

When applying eigenvector branching, the problem for a node at depth $k \geq 0$ in the branching tree includes k branching inequalities. These inequalities are used to derive additional constraints that are added to the Shor relaxation. The resulting problem is solved to obtain a lower bound $z(\bar{x}, \bar{X})$ and feasible objective value $v(\bar{x}) = \bar{x}^T Q\bar{x} + c^T\bar{x}$. If $v(\bar{x})$ is less than the best known value (BKV) then the BKV is updated, and the problem is fathomed if either it is infeasible, $z(\bar{x}, \bar{X}) \geq$ BKV or the relative gap condition (3) is satisfied. If none of these conditions holds then two child problems are created by using branching inequalities $a^T x \geq \theta$, $a^T x \leq \theta$, where a with $\|a\| = 1$ is the eigenvector corresponding to the maximal eigenvalue of $\bar{X} - \bar{x}\bar{x}^T$ and $\theta = a^T\bar{x}$.

We will first consider using linear branching inequalities to generate SOC-RLT constraints [10] that can be added to the Shor relaxation. With $h = 0$, which is the case for the test problems here, a branching inequality $a^T x \geq \theta$ implies the two SOC-RLT constraints

$$\|Xa - \theta x\| \leq a^T x - \theta, \qquad \|H^{1/2}Xa - \theta H^{1/2}x\| \leq a^T x - \theta, \tag{4}$$

and for each such inequality we add the two constraints from (4) to the Shor relaxation. Note that $\|a\| = 1$ and $\|x\| \leq 1$ together imply that $a^T x \leq 1$, so we could take $\mu^+ = 1$ in Proposition 2.1. Then the SOC-RLT constraint $\|Xa - \theta x\| \leq a^T x - \theta$ implies that $a^T(Xa - \theta x) \leq a^T x - \theta$, which can be written as $a^T Xa \leq (1 + \theta)a^T x - \theta$, and this is exactly the linear constraint in (1) with $\mu^+ = 1$.

Before looking at the 212 test problems from Table 1, consider the following instance with $n = 2$, $h = 0$ from [10]:

$$H = \frac{1}{2}\begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} -4 & 1 \\ 1 & -2 \end{pmatrix} \quad c = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{5}$$

The true solution value for this problem is -4, and the value for the Shor relaxation is -4.25. Applying the SOC-RLT cuts from [10] improves the lower bound to approximately -4.036. The Kronecker product constraints used in [2] produce a small further improvement but do not obtain the true solution value. The method of [22], which is exact for $n = 2$, obtains the value $z(\bar{x}, \bar{X}) = -4$ for the strengthened Shor relaxation, but does not obtain a feasible $\bar{x}$ with

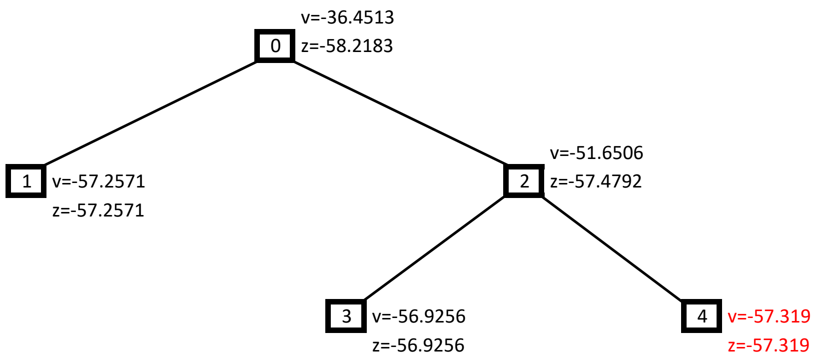

Fig. 1 Branching tree for instance_10_607

$v(\bar{x}) = -4$ [9]. This is not surprising since this problem has two optimal solutions, as shown in [10, Fig.1]. In the presence of multiple optimal solutions, no method based on cutting planes alone can eliminate the convex hull of these optimal solutions, and an interior-point solver is very unlikely to generate an extreme point of the problem's feasible set. When the objective is concave, as in this example, it is then to be expected that $v(\bar{x}) > z(\bar{x}, \bar{X})$ will always hold.

The solution of the Shor relaxation for (5) has $z(\bar{x}, \bar{X}) = -4.25$ at a point $(\bar{x}.\bar{X})$ with $v(\bar{x}) = -.75$. Applying eigenvector branching results in two child problems, both having solutions that are numerically rank-one with objective values and lower bounds equal to -4 (the gap between the lower of the two objective values and the lower of the two bounds is 3.63E-9). Thus the problem is exactly solved with a total of 3 nodes in the branching tree.

The performance of eigenvector branching on the TTRS instance (5) is extremely good, but the qualitative behavior on the test instances from Table 1 is typically not much worse. Consider for example instance_10_607, which is one of the problems that was not solved in either [2] or [22]. The branching tree for this problem is shown in Figure 1. The relative gap at the root node is about 38%. The problem for node 1 has a solution that is numerically rank-one and is therefore fathomed. Node 2 on level 1 produces 2 child nodes, both of which have solutions that are numerically rank-one. The problem solution, from node 4, has a relative gap of about 3E-8 obtained using a total of 5 nodes with a maximum depth of 2.

In Figures 2, 3 and 4 we show the number of nodes, maximum depth and final relative gap when solving the 212 instances from Table 1 using several versions of eigenvector branching. Computations were performed on a 64-bit PC with an Intel i7-6700 CPU running at 3.40 GHz with 16G of RAM, using the Matlab-based SeDuMi solver. Results obtained by branching on the eigenvector corresponding to the maximum eigenvalue of $\bar{X} - \bar{x}\bar{x}^T$, using the

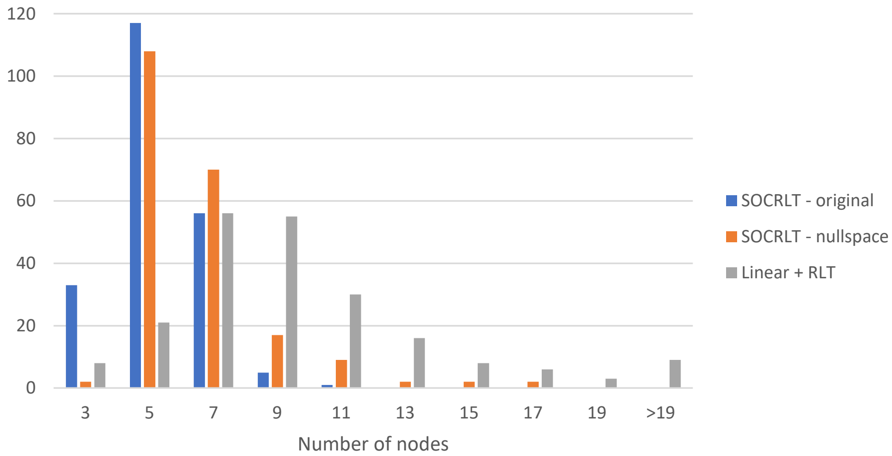

Fig. 2 Number of nodes using eigenvector branching

SOC-RLT constraints (4), are labeled SOCRLT - original. For this version, 206 of the 212 problems require 7 or fewer nodes and the maximum number of nodes is 11. The maximum depth is 4, and 204 of the problems are solved with a depth of 2 or less. The branching trees were traversed breadth-first, and none of the solution times exceeded 2.5 seconds. The relative gaps shown in Figure 4 are especially notable because the algorithm is implemented with the fathoming criterion from (3). Despite the fact that nodes are fathomed based on a relative gap of 1E-4, 184 of the 212 problems have solutions with relative gaps of 1E-6 or less. The reason for this is that many nodes in the branching trees for these problems have solutions that are numerically rank-one, as was the case for all of the terminal nodes shown in Figure 1.

Figures 2, 3 and 4 also show results for two other versions of the algorithm. The series labeled SOCRLT - nullspace give results when branching is based on $a = Z\bar{a}$, where $\bar{a}$ is the eigenvector for the maximum eigenvalue of $Z^T\bar{X}Z$ and the columns of Z are an orthonormal basis for the nullspace of $\bar{x}^T$. By first projecting onto the nullspace of $\bar{x}^T$ we obtain a symmetric disjunction with $\theta = a^T\bar{x} = 0$. It should be noted that the conditions $Y \succeq 0$ and $Z^TXZ = 0$ imply that $X = \lambda xx^T$ for $\lambda \geq 1$ but do not ensure that $\lambda = 1$. The case of a solution with $\lambda > 1$ is unlikely to occur in applications, but can be excluded in the presence of any equality constraint $g^Tx = h$ by adding the "squared" constraint $g^TXg = h^2$. As shown in the figures, the performance for the nullspace version of eigenvector branching is still excellent, but not as good as for the original version. For both the original and nullspace versions we also considered adding the RLT constraints that can be generated using the branching inequalities at each node, but adding these constraints had virtually no effect.

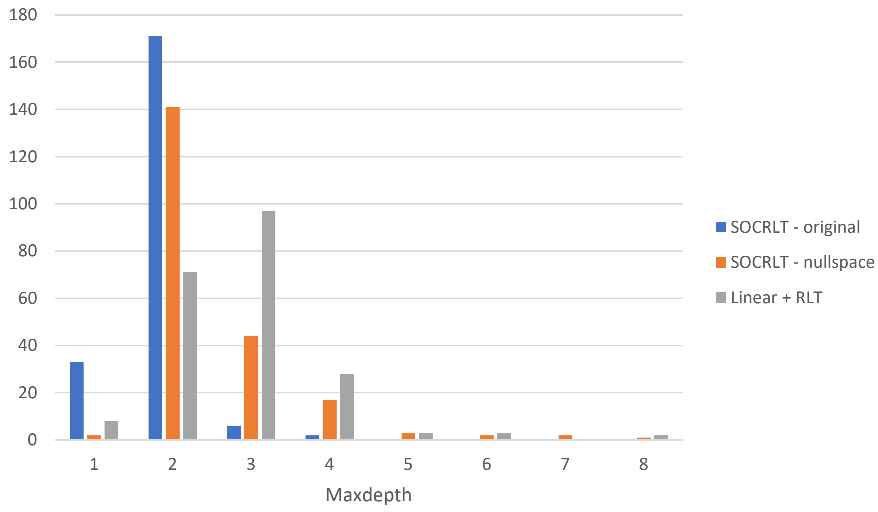

Fig. 3 Maximum depth using eigenvector branching

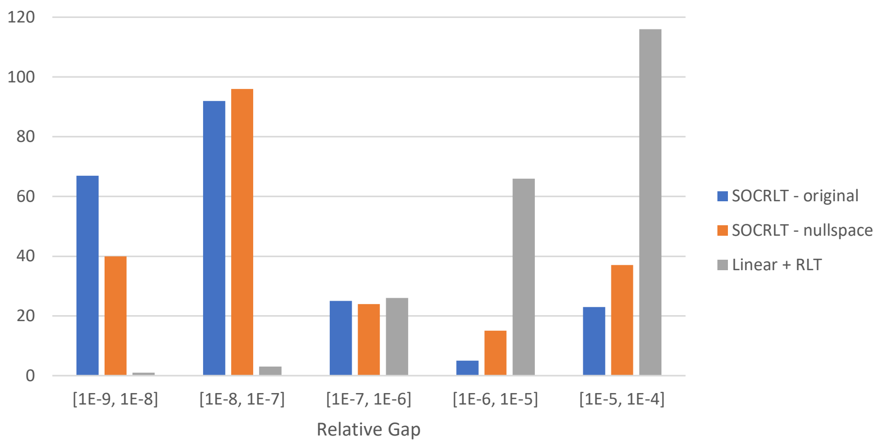

Fig. 4 Relative gap using eigenvector branching

In addition to the algorithm based on the SOC-RLT constraints in (4) we considered an implementation that used the linear constraints in (1) and (2). For the branching inequalities (1) and (2), with $\|a\| = 1$, we first considered a version using the values $\mu^+ = \min(1, \|H^{-1/2}a\|)$, $\mu^- = -\mu^+$, corresponding to maximizing and minimizing $a^T x$ over each of the two individual ellipsoids in TTRS. This version of the algorithm successfully solves all 212 instances, but the number of nodes required is sometimes much higher than for the versions using SOC-RLT constraints; 77 of the problems required 21 or more nodes, and the maximum number of nodes was over 300.

We considered two modifications to improve the performance of the algorithm using linear constraints. The first was to add the RLT constraints that can be obtained from the branching inequalities at each node. The results using this version of the algorithm are shown in Figures 2, 3 and 4 as the series Linear + RLT. The addition of the RLT constraints has a significant effect on performance when using the linear constraints in (1) and (2). It is evident from Figures 2 and 3 that the number of nodes and maxdepth distributions for this version are still good, but notably worse than for the versions using SOC-RLT constraints. The relative gaps shown in Figure 4 are substantially worse than for the SOC-RLT versions; 116 of the 212 problems have final relative gaps in the range 1E-5 to 1E-4, compared to only 23 problems for SOCRLT - original and 37 problems for SOCRLT - nullspace. In addition to the version where RLT constraints are added, we also considered a version of the algorithm where the upper and lower bounds μ^+ and μ^- are obtained by maximizing or minimizing $a^T x$ over the ellipsoid constraints and any other branching inequalities that are in force at the node where the branching is being applied. The performance for this version is very similar to the version where RLT constraints are added. Using both the RLT constraints and the optimized bounds produces a small marginal improvement over either method used individually.

3 Disjunctive cuts

As described in the previous section, the original use of Proposition 2.1 in [19] was not as the basis for branching but rather to generate a disjunctive constraint, or cut, that could be used to improve a polyhedral outer approximation of SDP. In this section we consider applying such disjunctive cuts to TTRS_{SDP}, but with two differences compared to the treatment in [19]. First, we assume that the semidefiniteness condition $Y \succeq 0$ is being explicitly maintained, and second we consider the possibility that each half of the disjunction in Proposition 2.1 can be used to generate conic but non-polyhedral constraints. For example, in the previous section a branching constraint $a^T x \geq \theta$ for TTRS_{SDP} was used to generate two SOC-RLT constraints (4), and the performance of the algorithm was markedly better using these SOC-RLT constraints than when the original linear constraints in (1) and (2) were used.

Consider TTRS_{SDP} written as a problem of the form SDP, with one equality constraint $Y_{00} = 1$ and two homogenous inequality constraints $A_i \bullet Y \leq 0$, $i = 1, 2$. Assume that $\bar{Y} \succeq 0$, but $\bar{X} - \bar{x}\bar{x}^T \neq 0$. Although $\bar{Y}$ is evidently not rank-one, it could be that $\bar{Y}$ is in the convex hull of rank-one solutions of TTRS_{SDP}. Applying Proposition 2.1, but using SOC-RLT constraints as in (4), there must then be Y^+ and Y^- so that

$$Y^+ + Y^- = \bar{Y}, \tag{6}$$

where Y^+ satisfies the constraints

$$\begin{aligned} A_i \bullet Y^+ &\le 0, \quad i=1,\dots,m,\\ \|X^+a - \theta x^+\| &\le a^T x^+ - \theta Y^+_{00},\\ \|H^{1/2}X^+a - \theta H^{1/2}x^+\| &\le a^T x^+ - \theta Y^+_{00},\\ Y^+ &\succeq 0, \end{aligned} \tag{7}$$

and Y^- satisfies the constraints

$$\begin{aligned} A_i \bullet Y^- &\le 0, \quad i=1,\dots,m,\\ \|X^-a - \theta x^-\| &\le \theta Y^-_{00} - a^T x^-\\ \|H^{1/2}X^-a - \theta H^{1/2}x^-\| &\le \theta Y^-_{00} - a^T x^-\\ Y^- &\succeq 0. \end{aligned} \tag{8}$$

Note that the constraints in (7) and (8) have been fully homogenized through the use of Y^+_{00} and Y^-_{00}, and the constraints $Y^+_{00} = 1$, $Y^-_{00} = 1$ are *not* present. The constraints (7) and (8) initially have $m = 2$, corresponding to the two original inequalities in TTRS, but will ultimately have additional inequalities corresponding to disjunctive cuts that are added to the problem. In order to determine if the constraints (6)-(8) are feasible we will augment them with artificial variables. This can be done in different ways; the approach taken here is similar to what is described for the polyhedral case in [19]. Consider (7) with the added artificial variables $(u^+, v^+, w^+) \in \mathbb{R}^m_+ \times \mathbb{R}^2_+ \times \mathbb{R}_+$:

$$\begin{aligned} A_i \bullet Y^+ &\le u^+_i, \quad i=1,\dots,m,\\ \|X^+a - \theta x^+\| &\le a^T x^+ - \theta Y^+_{00} + v^+_1,\\ \|H^{1/2}X^+a - \theta H^{1/2}x^+\| &\le a^T x^+ - \theta Y^+_{00} + v^+_2,\\ Y^+ + w^+ I &\succeq 0,\ u^+ \ge 0,\ v^+ \ge 0,\ w^+ \ge 0. \end{aligned} \tag{9}$$

Similarly, we augment the system (8) with artificial variables (u^-, v^-, w^-):

$$\begin{aligned} A_i \bullet Y^- &\le u^-_i \quad i=1,\dots,m,\\ \|X^-a - \theta x^-\| &\le \theta Y^-_{00} - a^T x^- + v^-_1\\ \|H^{1/2}X^-a - \theta H^{1/2}x^-\| &\le \theta Y^-_{00} - a^T x^- + v^-_2\\ Y^- + w^- I &\succeq 0,\ u^- \ge 0,\ v^- \ge 0,\ w^- \ge 0. \end{aligned} \tag{10}$$

Lemma 3.1 *Assume that* $\bar{Y} \succeq 0$*,* $\bar{X} - \bar{x}\bar{x}^T \ne 0$*,* $\theta = a^T\bar{x}$*,* $a^T\bar{X}a > \theta^2$*. Consider the optimization problem to minimize* $e^T u^+ + e^T u^- + e^T v^+ + e^T v^- + w^+ + w^-$ *subject to the constraints* (6)*,* (9) *and* (10)*, Let* $\bar{S}$ *be the dual solution matrix for the constraint* (6)*. Then* $\bar{S} \bullet Y \le 0$ *for any* Y *that is in the convex hull of rank-one solutions to* $TTRS_{\mathrm{SDP}}$*.*

Proof: The system of constraints clearly has an interior solution, so strong duality holds [21]. By construction, if $\bar{Y}$ is in the convex hull of rank-one solutions of $\mathrm{TTRS}_{\mathrm{SDP}}$ then the solution objective value is zero. The dual objective is to maximize $S \bullet \bar{Y}$, and therefore if $\bar{Y}$ is in the convex hull of rank-one solutions the dual solution must have $\bar{S} \bullet \bar{Y} = 0$. Now consider the same optimization problem but replacing $\bar{Y}$ with any Y in the convex hull of rank-one solutions. The resulting problem must have solution objective value equal to zero, and $\bar{S}$ is a feasible dual solution for this problem. Therefore $\bar{S} \bullet Y \leq 0$. □

In (9) and (10) the coefficients of all artifical variables are equal to one, but Lemma 1 continues to hold exactly as stated if any positive coefficients are used. The use of unit coefficient for the artifical variables is appropriate if all of the constraints in (7) and (8) are on similar scales.

To compare eigenvector branching to the use of disjunctive cuts we attempted to solve the same TTRS problems considered in the previous section using the disjunctive cuts described in Lemma 3.1. At iteration $k \geq 0$ we have a problem of the form SDP, with $m = 2 + k$ inequality constraints corresponding to the two original inequalities of TTRS_{SDP} and k cuts that have been added on previous iterations. This problem is solved, resulting in a solution $\bar{Y}$, feasible objective value $v(\bar{x})$ and lower bound $z(\bar{x}, \bar{X})$. The BKV is updated if $v(\bar{x}) <$ BKV. Termination occurs if either the current iterate meets the relative gap criterion

$$\frac{\text{BKV} - z(\bar{x}, \bar{X})}{|\text{BKV}|} < 10^{-4}, \tag{11}$$

or $k = 25$. Otherwise a new disjunctive cut $\bar{S} \bullet Y \leq 0$ is generated as in Lemma 3.1 and we go to iteration $k + 1$ with $A_{m+1} = \bar{S}/\|\bar{S}\|$. The criterion (11) is used in place of (3) because the sequence of feasible objective values can be markedly non-monotonic. Also, as suggested in [19], the constraints (9)-(10) used to generate cuts are only periodically updated to include the cuts generated on previous iterations. We experimented with updating these constraints every 3-5 iterations; the results presented below were obtained updating the constraints every 5 iterations.

Before turning to the problems described in Table 1 we again consider the problem (5) from [10]. As described in the previous section this problem has a true solution value of -4. The SDP relaxation generates a bound $z(\bar{x}, \bar{X}) = -4.25$ and a feasible solution with objective value $v(\bar{x}) = -.75$. Applying 25 disjunctive cuts results in the sequence of objective values and lower bounds shown in Figure 5. As shown in the figure the lower bounds rapidly approach the value -4, and after 25 cuts the bound is approximately -4.0001. The sequence of feasible values is very erratic and produces a BKV of -3.9551 on iteration 22, resulting in a final relative gap of about 1.2E-3. The erratic behavior of the feasible values is due to the fact, mentioned earlier, that this problem has two optimal solutions, neither of which can be eliminated using disjunctive cuts. This fact, combined with the use of an interior-point solver, means that we have no control over exactly where iterates will "land" and the disjunctive cuts generated are dependent on the iterates.

Results applying the algorithm using disjunctive cuts on the problems from Table 1 are shown in Figures 6 and 7. In the figures the results are separated for the problems of size 5, 10 and 20. Of the 212 problems, 145 met the relative gap criterion (11) using 25 or fewer cuts (one problem of size 20 met the criterion on the final iteration). From the two figures it is evident that the fraction of problems successfully solved is decreasing with problem size.

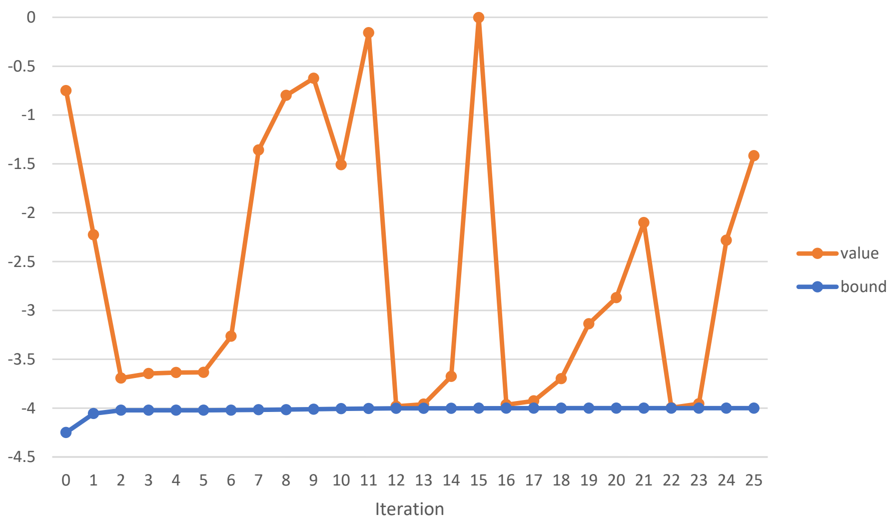

Fig. 5 Feasible values and bounds using disjunctive cuts on instance (5)

The times required are mainly dependent on the number of cuts; note that until the final iteration, each iteration requires the solution of two semidefinite programming problems, the first to generate the iterate $\bar{Y}$ and the second to generate the disjunctive cut matrix $\bar{S}$. The solution time for a problem of size 20 requiring 7 cuts is about 5 seconds. Comparing these results with Table 1, it is noteworthy that the number of problems successfully solved using disjunctive cuts is higher than both the 127 problems solved in [2] and the 79 problems solved in [22]. Comparing the results problem-by problem, however, there is no simple relationship between which problems are solved using disjunctive cuts or the methods in [2] and [22]. We also considered the effect of increasing the iteration limit by running the algorithm for a maximum of 50 cuts. Using up to 50 cuts increases the number of problems that meet the relative gap criterion (11) from 145 to 164, which is higher than the 156 problems that are solved in [2] and [22] put together.

Although the results using disjunctive cuts are better than those in either [2] or [22], they are clearly worse than the results using eigenvalue branching in the previous section. Since all of the 212 problems are solved using eigenvalue branching, we can use those results to better understand what is happening in the instances that are not solved using disjunctive cuts. Returning to the results using a maximum of 25 cuts, it turns out that in all 67 cases where the relative gap criterion (11) was not met, this was due to failure to generate a feasible solution close enough to optimality and not failure to generate a good enough lower bound. As already described in the context of the example (5), any cutting-plane method using an interior-point solver on a problem of the form SDP with multiple optimal solutions may have difficulty generating

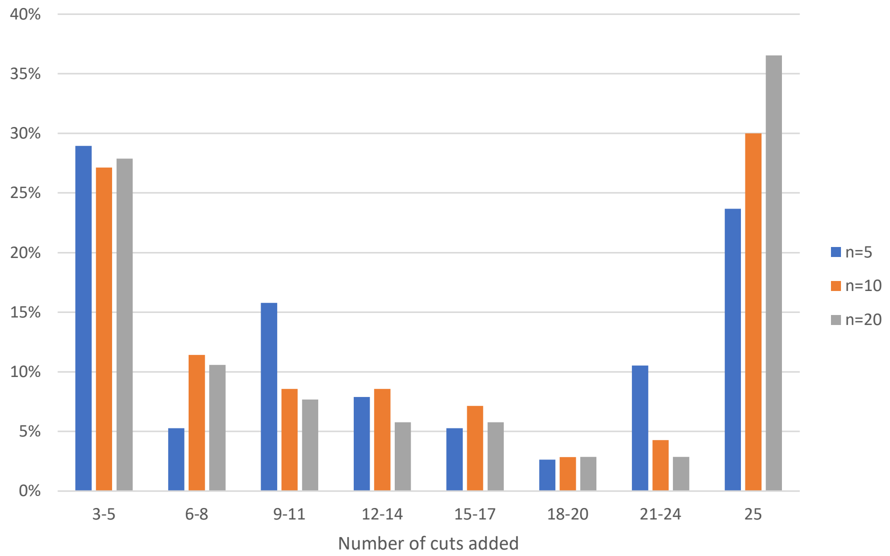

Fig. 6 Number of disjunctive cuts

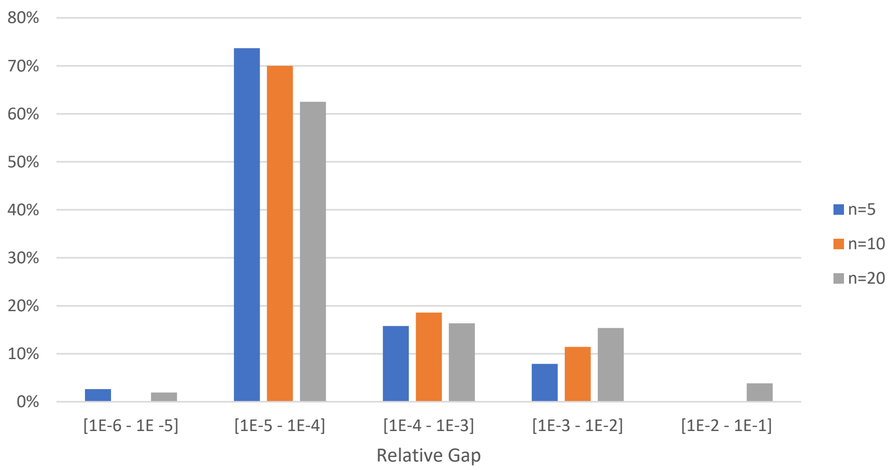

Fig. 7 Relative gap using disjunctive cuts

an optimal solution rank-one solution. The problems from Table 1 are not expected to have multiple optimal solutions, but it is possible that one side of the disjunction used to generate a cut has the optimal solution while the other has a near-optimal solution. Consider for example instance_10_607, whose solution tree using eigenvalue branching is shown in Figure 1. This problem is one of the instances that is not solved using disjunctive cuts, even when the number

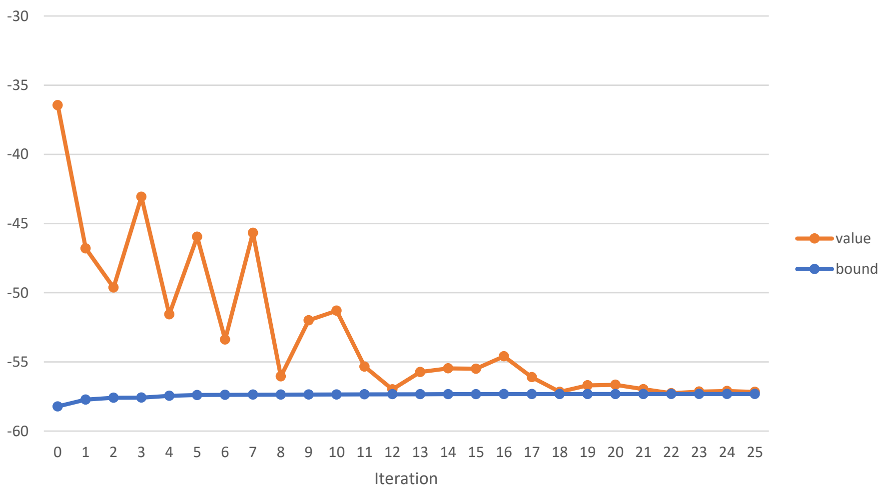

Fig. 8 Feasible values and bounds using disjunctive cuts on instance_10_607

of cuts is increased to 50. Note that the solution for the problem, obtained at node 4 in Figure 1, has objective value −57.319. However the solution at node 1 is also numerically rank-one, with an objective value of −57.251. The disjunctive cut applied at the root node will likely be able to cut off the solution $(\bar{x}, \bar{X})$ there, but the rank-one solution at node 1 cannot be removed by any disjunctive cut. Figure 8 shows the sequence of objective values and lower bounds obtained when the algorithm using disjunctive cuts is applied to instance_10_607. Using 25 iterations, the BKV of −57.2624 is obtained on iteration 22, resulting in a final relative gap of about 1E-3. The time to run 25 iterations is about 11 seconds. Figure 9 shows the sequence of objective values for the feasibility problem from Lemma 3.1 on these same iterations. The solution values fall off relatively quickly and are all below 0.0005 after iteration 16.

4 Conclusion

Eigenvector branching is a new spatial branching method for semidefinite optimization problems that is directly motivated by the rank-one condition $X = xx^T$. We obtain excellent computational results applying eigenvector branching to difficult instances of the two-trust-region subproblem. A related methodology based on disjunctive cuts obtains computational results on the same problems that compare favorably with previous literature, but the results using disjunctive cuts are not as good as those obtained using eigenvector branching. Eigenvector branching appears to be a promising technique to strengthen the initial semidefinite relaxation of problems with a nonconvex quadratic objective and/or constraints.

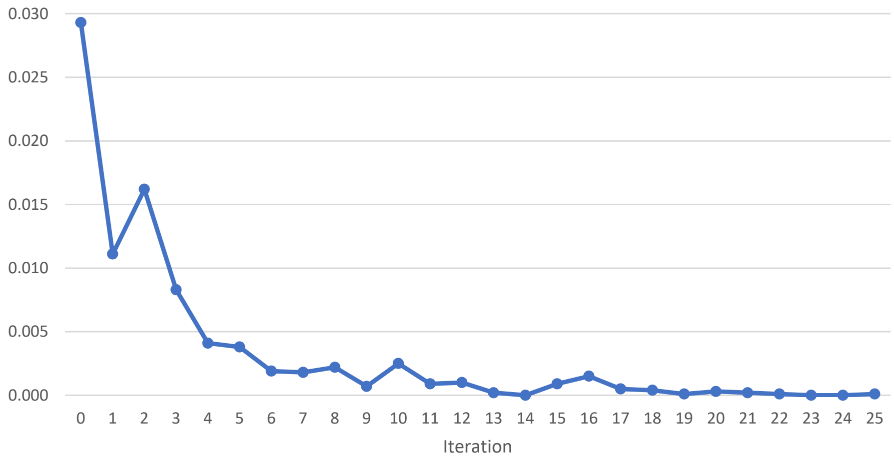

Fig. 9 Value in feasibility problem for disjunctive cuts on instance_10_607

Acknowledgement

I am grateful to Sam Burer for helpful discussions on the topic of this paper.

References

1. Ai, W., Zhang, S.: Strong duality for the CDT subproblem: a necessary and sufficient condition. SIAM J. Optim. **19**(4), 1735–1756 (2008). DOI 10.1137/07070601X
2. Anstreicher, K.M.: Kronecker product constraints with an application to the two-trust-region subproblem. SIAM J. Optim. **27**, 368–378 (2017). DOI 10.1137/16M1078859
3. Barvinok, A.I.: Feasibility testing for systems of real quadratic equations. Discrete Comput. Geom. **10**(1), 1–13 (1993). DOI 10.1007/BF02573959
4. Beck, A., Eldar, Y.C.: Strong duality in nonconvex quadratic optimization with two quadratic constraints. SIAM J. Optim. **17**(3), 844–860 (2006). DOI 10.1137/050644471
5. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed-integer nonlinear optimization. Acta Numerica **22**, 1–131 (2013). DOI 10.1017/S0962492913000032
6. Belotti, P., Lee, J., Liberti, L., Margot, F., Wachter, A.: Branching and bounds tightening techniques for non-convex MINLP. Optim. Meth. Software **24**, 597–634 (2009). DOI 10.1080/10556780903087124
7. Bienstock, D.: A note on polynomial solvability of the CDT problem. SIAM J. Optim. **26**, 488–498 (2016). DOI 10.1137/15M1009871
8. Bomze, I.M., Overton, M.L.: Narrowing the difficulty gap for the Celis-Dennis-Tapia problem. Math. Prog. **151**(2), 459–476 (2015). DOI 10.1007/s10107-014-0836-3
9. Burer, S.: Private communication (2022)
10. Burer, S., Anstreicher, K.M.: Second-order-cone constraints for extended trust-region subproblems. SIAM J. Optim. **23**(1), 432–451 (2013). DOI 10.1137/110826862
11. Celis, M.R., Dennis, J.E., Tapia, R.A.: A trust region strategy for nonlinear equality constrained optimization. In: Numerical Optimization, 1984 (Boulder, Colo., 1984), pp. 71–82. SIAM, Philadelphia, PA (1985)
12. Conn, A., Gould, N., Toint, P.: Trust Region Methods. Society for Industrial and Applied Mathematics (2000). DOI 10.1137/1.9780898719857
13. Couenne: https://www.coin-or.org/Couenne/

14. Fu, M., Luo, Z.Q., Ye, Y.: Approximation algorithms for quadratic programming. J. Comb. Optim. **2**(1), 29–50 (1998). DOI 10.1023/A:1009739827008
15. Nemirovski, A., Roos, C., Terlaky, T.: On maximization of a quadratic form over the intersection of ellipsoids with common center. Math. Prog. **86**, 463–473 (1999). DOI 10.1007/s101079900099
16. Peng, J.M., Yuan, Y.X.: Optimality conditions for the minimization of a quadratic with two quadratic constraints. SIAM J. Optim. **7**(3), 579–594 (1997). DOI 10.1137/S1052623494261520
17. Rendl, F., Wolkowicz, H.: A semidefinite framework for trust region subproblems with applications to large scale minimization. Math. Prog. **77**(1), 273–299 (1997). DOI 10.1007/BF02614438
18. Sahinidis, N.: BARON: a general purpose global optimization software package. J. Global Optim. **8**, 201–205 (1996). DOI 10.1007/BF00138693
19. Saxeena, A., Bonami, P., Lee, J.: Convex relaxations of non-convex mixed integer quadratically constrained programs: extended formulations. Math. Prog. Ser. B **124**, 383–411 (2010). DOI 10.1007/s10107-010-0371-9
20. Sherali, H., Adams, W.: A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems. Kluwer (1998)
21. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Review **38**, 49–95 (1996). DOI 10.1137/1038003
22. Yang, B., Burer, S.: A two-variable approach to the two-trust-region subproblem. SIAM J. Optim. **26**(1), 661–680 (2016). DOI 10.1137/130945880
23. Ye, Y., Zhang, S.: New results on quadratic minimization. SIAM J. Optim. **14**(1), 245–267 (2003). DOI 10.1137/S105262340139001X