KRONECKER PRODUCT CONSTRAINTS WITH AN APPLICATION TO THE TWO-TRUST-REGION SUBPROBLEM

KURT M. ANSTREICHER*

Abstract. We consider semidefinite optimization problems that include constraints of the form $G(x) \succeq 0$ and $H(x) \succeq 0$, where the components of the symmetric matrices $G(\cdot)$ and $H(\cdot)$ are affine functions of $x \in \mathbb{R}^n$. In such a case we obtain a new constraint $K(x, X) \succeq 0$, where the components of $K(\cdot, \cdot)$ are affine functions of x and X, and X is an $n \times n$ matrix that is a relaxation of xx^T. The constraint $K(x, X) \succeq 0$ is based on the fact that $G(x) \otimes H(x) \succeq 0$, where $\otimes$ denotes the Kronecker product. This construction of a constraint based on the Kronecker product generalizes the construction of an RLT constraint from two linear inequality constraints, and also the construction of an SOC-RLT constraint from one linear inequality constraint and a second-order cone constraint. We show how the Kronecker product constraint obtained from two second-order cone constraints can be efficiently used to computationally strengthen the semidefinite programming relaxation of the two-trust-region subproblem.

Key words. Semidefinite programming, Semidefinite optimization, Nonconvex quadratic programming, Trust region subproblem.

AMS subject classifications. 90C20, 90C22, 90C26.

1. Introduction.

Let A and B be $m \times n$ and $p \times q$ matrices. The *Kronecker product* $A \otimes B$ is the $mp \times nq$ matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

Important properties of Kronecker products for our purposes are collected in the following proposition; for details see for example [10, Chapter 4]. We use $A \succeq 0$ to denote that a matrix A is symmetric and positive semidefinite (PSD). For matrices A and B of the same dimensions we use $A \bullet B$ to denote the matrix inner product $A \bullet B = \text{tr}(AB^T)$.

PROPOSITION 1. *If A and C^T have the same number of columns, and B and D^T have the same number of columns, then $(A \otimes B)(C \otimes D) = AC \otimes BD$. Moreover $(A \otimes B)^T = A^T \otimes B^T$, and if $A \succeq 0$ and $B \succeq 0$, then $A \otimes B \succeq 0$.*

We are interested in the situation where a semidefinite optimization problem in the variables $x \in \mathbb{R}^n$ and $X \in \mathbb{R}^{n \times n}$ also includes constraints of the form $G(x) \succeq 0$ and $H(x) \succeq 0$, where the components of $G(\cdot)$ and $H(\cdot)$ are affine functions of x. The matrix X is a relaxation of the rank-one matrix xx^T, and is typically constrained via the semidefinite restriction $X \succeq xx^T$. It is not assumed that the dimensions of the matrices $G(x)$ and $H(x)$ are identical. Since $G(x) \otimes H(x) \succeq 0$ is also a valid constraint by Proposition 1, we can replace every term of the form $x_i x_j$ in $G(x) \otimes H(x)$ with X_{ij} to obtain a valid constraint $K(x, X) \succeq 0$, where the entries of $K(\cdot, \cdot)$ are affine functions of x and X. We refer to a constraint generated in this fashion as a *Kronecker product constraint*.

*Department of Management Sciences, University of Iowa, Iowa City, IA, 52242, USA. Email: kurt-anstreicher@uiowa.edu.

Example 1. Let $G(x) = b - a^T x$, $H(x) = d - c^T x$. Then

$$G(x) \otimes H(x) = (a^T x)(c^T x) - b(c^T x) - d(a^T x) + bd$$
$$K(x, X) = ac^T \bullet X - (bc + da)^T x + bd.$$

In this case $K(x, X) \geq 0$ is exacty the constraint obtained using the well-known Reformulation-Linearization Technique (RLT) [15] applied to the two linear inequalities $a^T x \leq b$, $c^T x \leq d$. In this case the constraint $K(x, X) \geq 0$ is commonly refered to as an ordinary RLT constraint.

Example 2. Let $G(x) = b - a^T x$, and let $H(x)$ be the matrix for the PSD representation of the second-order cone (SOC) constraint $\|A(x - h)\| \leq 1$;

$$H(x) = \begin{pmatrix} I & A(x-h) \\ (x-h)^T A^T & 1 \end{pmatrix} \succeq 0.$$

In this case

$$G(x) \otimes H(x) = \begin{pmatrix} (b - a^T x)I & (b - a^T x)A(x-h) \\ (b - a^T x)(x-h)^T A^T & b - a^T x \end{pmatrix}$$
$$K(x, X) = \begin{pmatrix} (b - a^T x)I & v(x, X) \\ v(x, X)^T & b - a^T x \end{pmatrix},$$

where $v(x, X) = (a^T x - b)Ah + bAx - AXa$. Then $K(x, X) \succeq 0$ is the PSD representation of the constraint formed by replacing xx^T with X in the valid constraint $\|(b - a^T x)A(x - h)\| \leq b - a^T x$. Constraints of this type were introduced in [16], and were subsequently termed "SOC-RLT" constraints in [7].

Example 3. Consider two strictly convex quadratic constraints expressed in SOC form as $\|x\| \leq 1$ and $\|A(x - h)\| \leq 1$, where A is an $n \times n$ nonsingular matrix. These constraints can be alternatively expressed in PSD form as $G(x) \succeq 0$, $H(x) \succeq 0$, where

$$G(x) = \begin{pmatrix} I & x \\ x^T & 1 \end{pmatrix}, \qquad H(x) = \begin{pmatrix} I & A(x-h) \\ (x-h)^T A^T & 1 \end{pmatrix}.$$

Since $G(x) \succeq 0$ and $H(x) \succeq 0$, it follows that the Kronecker product $G(x) \otimes H(x) \succeq 0$, where

$$G(x) \otimes H(x) = \begin{pmatrix} H(x) & & & x_1 H(x) \\ & \ddots & & \vdots \\ & & H(x) & x_n H(x) \\ x_1 H(x) & \cdots & x_n H(x) & H(x) \end{pmatrix}.$$

To generate a valid constraint on (x, X) we replace any products $x_i x_j$ with X_{ij} in $G(x) \otimes H(x)$. Such products occur in terms of the form $x_j A(x - h) = Ax_j x - x_j Ah$, where $x_j x \to X_j$, the jth column of X. Defining

$$H_j(x, X) = \begin{pmatrix} x_j I & A(X_j - x_j h) \\ (X_j - x_j h)^T A^T & x_j \end{pmatrix},$$

we can write a valid PSD constraint $K(x, X) \succeq 0$, where

$$(1) \qquad K(x, X) = \begin{pmatrix} H(x) & & & H_1(x, X) \\ & \ddots & & \vdots \\ & & H(x) & H_n(x, X) \\ H_1(x, X) & \cdots & H_n(x, X) & H(x) \end{pmatrix}.$$

In this case we refer to the Kronecker product constraint $K(x,X) \succeq 0$ as a "KSOC" constraint. We will consider constraints of this form in more detail in the next section.

2. KSOC constraints. In this section we further study the Kronecker product constraint $K(x,X) \succeq 0$, with $K(\cdot,\cdot)$ as in (1), that is generated from two SOC constraints $\|x\| \le 1$, $\|A(x-h)\| \le 1$. We assume throughout that A is nonsingular. To begin, we note that the problem of generating additional valid constraints on (x,X) that are implied by these two SOC constraints was previously considered in [7]. The approach taken in [7] was based on using a linear constraint $a^T x \le 1$, where $\|a\| = 1$, together with the SOC constraint $\|A(x-h)\| \le 1$, to generate an SOC-RLT constraint as in Example 2 of the previous section. Note that the constraint $a^T x \le 1$ is a supporting hyperplane for the ball $\{x \,|\, \|x\| \le 1\}$ at $x = a$. It is shown in [7] that using all such SOC-RLT constraints, corresponding to different choices of a with $\|a\| = 1$, is equivalent to the use of all possible ordinary RLT constraints generated using supporting hyperplanes for both $\{x \,|\, \|x\| \le 1\}$ and $\{x \,|\, \|A(x-h)\| \le 1\}$. However, the separation problem of finding an a with $\|a\| = 1$ so that the resulting SOC-RLT constaint is currently violated can be efficiently solved as a trust-region subproblem, while the problem of finding two supporting hyperplanes so that the resulting ordinary RLT constraint is violated is bilinear.

We will next show that the KSOC constraint $K(x,X) \succeq 0$ implies all possible SOC-RLT constraints that arise from using an a with $\|a\| = 1$ together with the SOC constraint $\|A(x-h)\| \le 1$. As described in Example 2 of the previous section, such an SOC-RLT constraint has the form $\|(a^T x - 1)Ah + Ax - AXa\| \le 1 - a^T x$.

LEMMA 2. *Suppose that $\|a\| = 1$ and $K(x,X) \succeq 0$, with $K(\cdot,\cdot)$ as in (1). Then $\|(a^T x - 1)Ah + Ax - AXa\| \le 1 - a^T x$.*

Proof. Since $K(x,X) \succeq 0$ it must also be that

$$[(-a^T, 1) \otimes I]\ K(x,X) \left[\begin{pmatrix} -a \\ 1 \end{pmatrix} \otimes I \right] \succeq 0.$$

However

$$\begin{aligned}
&[(-a^T, 1) \otimes I]\ K(x,X) \left[\begin{pmatrix} -a \\ 1 \end{pmatrix} \otimes I \right] \\
&= (-a_1 I, \dots, -a_n I, I) \begin{pmatrix} H(x) & & & H_1(x,X) \\ & \ddots & & \vdots \\ & & H(x) & H_n(x,X) \\ H_1(x,X) & \cdots & H_n(x,X) & H(x) \end{pmatrix} \begin{pmatrix} -a_1 I \\ \vdots \\ -a_n I \\ I \end{pmatrix} \\
&= (-a_1 I, \dots, -a_n I, I) \begin{pmatrix} -a_1 H(x) + H_1(x,X) \\ \vdots \\ -a_n H(x) + H_n(x,X) \\ H(x) - \sum_{j=1}^n a_j H_j(x,X) \end{pmatrix} \\
&= (1 + a^T a) H(x) - 2 \sum_{j=1}^n a_j H_j(x,X) \\
&= 2[H(x) - \sum_{j=1}^n a_j H_j(x,X)],
\end{aligned}$$

implying that $H(x) - \sum_{j=1}^n a_j H_j(x,X) \succeq 0$. But

$$
\begin{aligned}
&H(x) - \sum_{j=1}^n a_j H_j(x,X) \\
&= \begin{pmatrix} I & A(x-h) \\ (x-h)^T A^T & 1 \end{pmatrix} - \sum_{j=1}^n a_j \begin{pmatrix} x_j I & A(X_j - x_j h) \\ (X_j - x_j h)^T A^T & x_j \end{pmatrix} \\
&= \begin{pmatrix} (1-a^T x) I & A(x-h) - \sum_{j=1}^n a_j (AX_j - x_j Ah) \\ (x-h)^T A^T - \sum_{j=1}^n a_j (AX_j - x_j Ah)^T & (1-a^T x) \end{pmatrix} \\
&= \begin{pmatrix} (1-a^T x) I & (a^T x - 1)Ah + Ax - AXa \\ (a^T x - 1) h^T A^T + x^T A^T - a^T X^T A^T & (1-a^T x) \end{pmatrix},
\end{aligned}
$$

so $H(x) - \sum_{j=1}^n a_j H_j(x,X) \succeq 0$ is exactly the PSD representation of the SOC-RLT constraint $\|(a^T x - 1)Ah + Ax - AXa\| \le 1 - a^T x$. ∎

Lemma 2 shows that the use of the KSOC constraint $K(x,X) \succeq 0$ implies all possible SOC-RLT constraints used in [7], but it does not show that the constraint $K(x,X) \succeq 0$ is actually stronger. In the computational results of the next section we will demonstrate that in at least some cases the constraint $K(x,X) \succeq 0$ is in fact stronger than all possible SOC-RLT constraints used in [7].

From a computational standpoint, one difficulty with the constraint $K(x,X) \succeq 0$ is that the size of the matrix $K(\cdot,\cdot)$ is $(n+1)^2 \times (n+1)^2$, and therefore even a modest underlying dimension n will result in a very large PSD constraint. Fortunately it is possible to use the block structure of $K(\cdot,\cdot)$ from (1) to express $K(x,X) \succeq 0$ as semidefiniteness of an $(n+1) \times (n+1)$ matrix. The fact that this much smaller matrix can be efficiently computed facilitates the use of a cut-generation scheme for enforcing $K(x,X) \succeq 0$.

Lemma 3. *Suppose that $K(x,X) \succeq 0$, with $K(\cdot,\cdot)$ as in (1). Then $\|x\| \le 1$ and $\|A(x-h)\| \le 1$. In addition, if either $\|x\| = 1$ or $\|A(x-h)\| = 1$ then $X = xx^T$.*

Proof. That $K(x,X) \succeq 0$ implies $H(x) \succeq 0$ is obvious since $H(x)$ occurs as a principal submatrix of $K(x,X)$. However $G(x)$ is also a principal submatrix of $K(x,X)$, corresponding to the rows and colums indexed by the $(n+1, n+1)$ components of each diagonal block of $K(x,X)$, so $K(x,X) \succeq 0$ also implies that $G(x) \succeq 0$. To prove the remainder of the lemma, consider a nonsingular symmetric transformation of the form

$$
K'(x,X) = \begin{pmatrix} V(x)^T & & & \\ & \ddots & & \\ & & V(x)^T & \\ -x_1 I & \cdots & -x_n I & I \end{pmatrix} K(x,X) \begin{pmatrix} V(x) & & & -x_1 I \\ & \ddots & & \\ & & V(x) & -x_n I \\ & & & I \end{pmatrix},
$$

where

$$
V(x) = \begin{pmatrix} I & -A(x-h) \\ & 1 \end{pmatrix}.
$$

Substituting in the definition of $K(x,X)$ from (1), we obtain

$$
K'(x,X) = \begin{pmatrix} T(x) & & & W_1(x,X) \\ & \ddots & & \vdots \\ & & T(x) & W_n(x,X) \\ W_1(x,X)^T & \cdots & W_n(x,X)^T & Z(x,X) \end{pmatrix},
$$

where

$$T(x) = V(x)^T H(x) V(x) = \begin{pmatrix} I & \\ & t(x) \end{pmatrix}, \qquad t(x) = 1 - \|A(x-h)\|^2,$$

$$W_j(x,X) = V(x)^T[H_j(x,X) - x_j H(x)] = \begin{pmatrix} 0 & A(X_j - x_j x) \\ (X_j - x_j x)^T A^T & (h-x)^T A^T A(X_j - x_j x) \end{pmatrix},$$

$$\begin{aligned} Z(x,X) &= (1+\|x\|^2)H(x) - 2\sum_{j=1}^n x_j H_j(x,X) \\ &= (1+\|x\|^2)\begin{pmatrix} I & A(x-h) \\ (x-h)^T A^T & 1 \end{pmatrix} \\ &\quad -2\sum_{j=1}^n x_j \begin{pmatrix} x_j I & A(X_j - x_j h) \\ (X_j - x_j h)^T A^T & x_j \end{pmatrix}. \end{aligned}$$

Collecting terms, we obtain

$$(2)\quad Z(x,X) = \begin{pmatrix} s(x)I & 2A(x - Xx) - s(x)A(x+h) \\ 2(x-Xx)^T A^T - s(x)(x+h)^T A^T & s(x) \end{pmatrix},$$

where $s(x) = 1 - \|x\|^2$. Since $K'(x,X) \succeq 0$, $t(x) = 0$ implies that each row and column corresponding to the diagonal entries of $K'(x,X)$ equal to $t(x)$ must be zero, and therefore $A(X_j - x_j x) = 0$ for each j. But then $A(X - xx^T) = 0$, implying $X = xx^T$ since A is nonsingular. Similarly if $s(x) = 0$ then $Z(x,X) = 0$ and $W_j(x,X) = 0$ for each j, again implying that $A(X - xx^T) = 0$ and therefore $X = xx^T$. ∎

The first result in Lemma 3 is reminiscent of the well-known fact [15] that when generating ordinary RLT constraints from linear inequalities, the set of all possible RLT constraints implies the original inequality constraints. Note that if $X = xx^T$, then the vector $2A(x - Xx) - s(x)A(x+h)$ in the upper right and lower left blocks of $Z(x,X)$ is equal to $s(x)A(x-h)$. In this case $W_j(x,X) = 0$ for each j, and $Z(x,X) = s(x)H(x)$.

We next consider further elimination of the off-diagonal blocks in $K'(x,X)$ when $X \neq xx^T$. From Lemma 3 we know that if $K(x,X) \succeq 0$ then we would certainly have $s(x) > 0$ and $t(x) > 0$. However we are ultimately interested in generating a cut when in fact $K(x,X) \not\succeq 0$. In the context of interest we will enforce the constraint $X \succeq xx^T$, and also the constraints $\mathrm{tr}(X) \le 1$ and $A^T A \bullet X - 2h^T A^T Ax \le 1 - h^T A^T Ah$ obtained from the original SOC constraints $\|x\| \le 1$ and $\|A(x-h)\| \le 1$. We omit the easy proof of the following result.

Lemma 4. *Assume that* $X \succeq xx^T$, $\mathrm{tr}(X) \le 1$ *and* $A^T A \bullet X - 2h^T A^T Ax \le 1 - h^T A^T Ah$. *Then* $s(x) = 1 - \|x\|^2 \ge 0$ *and* $t(x) = 1 - \|A(x-h)\|^2 \ge 0$. *Moreover if* $X \neq xx^T$ *then* $s(x) > 0$ *and* $t(x) > 0$.

Assuming that $t(x) > 0$, let $\overline{W}_j(x,X) = T(x)^{-1} W_j(x,X)$, and define

$$K''(x,X) = U(x,X)^T K'(x,X)\, U(x,X) = \begin{pmatrix} T(x) & & & \\ & \ddots & & \\ & & T(x) & \\ & & & Z'(x,X) \end{pmatrix},$$

where

$$U(x,X) = \begin{pmatrix} I & & & -\overline{W}_1(x,X) \\ & \ddots & & \\ & & I & -\overline{W}_n(x,X) \\ & & & I \end{pmatrix}$$

and

$$Z'(x,X) = Z(x,X) - \sum_{j=1}^{n} W_j(x,X)^T T(x)^{-1} W_j(x,X). \tag{3}$$

Using the definition of $W_j(x,X)$, and letting $\widehat{X} = X - xx^T$, it is straightforward to compute that $W_j(x,X)^T T(x)^{-1} W_j(x,X)$ is equal to

$$\begin{pmatrix} \frac{1}{t(x)} A\widehat{X}_j\widehat{X}_j^T A^T & \frac{1}{t(x)} A\widehat{X}_j\widehat{X}_j^T A^T A(h-x) \\ \frac{1}{t(x)}(h-x)^T A^T A\widehat{X}_j\widehat{X}_j^T A^T & \widehat{X}_j^T A^T A\widehat{X}_j + \frac{1}{t(x)}\left((h-x)^T A^T A\widehat{X}_j\right)^2 \end{pmatrix},$$

and therefore $\sum_{j=1}^{n} W_j(x,X)^T T(x)^{-1} W_j(x,X)$ is equal to

$$\frac{1}{t(x)} \begin{pmatrix} A\widehat{X}^2 A^T & A\widehat{X}^2 A^T A(h-x) \\ (h-x)^T A^T A\widehat{X}^2 A^T & t(x)\operatorname{tr}(\widehat{X} A^T A \widehat{X}) + \|\widehat{X} A^T A(h-x)\|^2 \end{pmatrix}. \tag{4}$$

Substituting (4) and (2) into (3) we obtain a complete expression for $Z'(x,X)$.

With $t(x) > 0$, we have by construction $K(x,X) \succeq 0 \iff K''(x,X) \succeq 0 \iff Z'(x,X) \succeq 0$. If the latter does not hold for values $(x,X) = (\bar{x}, \overline{X})$, there is a vector $a \in \mathbb{R}^{n+1}$ with $a^T Z'(\bar{x}, \overline{X}) a < 0$. It then follows that $b^T K(\bar{x}, \overline{X}) b < 0$, where

$$\begin{aligned} b &= \begin{pmatrix} V(\bar{x}) & & & -\bar{x}_1 I \\ & \ddots & & \\ & & V(\bar{x}) & -\bar{x}_n I \\ & & & I \end{pmatrix} \begin{pmatrix} I & & & -\overline{W}_1(\bar{x}, \overline{X}) \\ & \ddots & & \\ & & I & -\overline{W}_n(\bar{x}, \overline{X}) \\ & & & I \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a \end{pmatrix} \\ &= \begin{pmatrix} -V(\bar{x})\overline{W}_1(\bar{x}, \overline{X})a - \bar{x}_1 a \\ \vdots \\ -V(\bar{x})\overline{W}_n(\bar{x}, \bar{x})a - \bar{x}_n a \\ a \end{pmatrix} = \begin{pmatrix} B_1 \\ \vdots \\ B_n \\ a \end{pmatrix}. \end{aligned}$$

Then $b^T K(x,X) b \geq 0$ is a valid, linear constraint on (x,X) that is violated at $(\bar{x}, \overline{X})$. Using the definition of $K(x,X)$ from (1), we have

$$b^T K(x,X) b = a^T H(x) a + \sum_{i=1}^{n} B_i^T H(x) B_i + 2\sum_{i=1}^{n} a^T H_i(x,X) B_i. \tag{5}$$

Letting

$$a = \begin{pmatrix} \bar{a} \\ \alpha \end{pmatrix}, \qquad B_i = \begin{pmatrix} \overline{B}_i \\ \beta_i \end{pmatrix}, \quad i = 1, \dots, n,$$

we have

$$\begin{aligned} a^T H(x) a &= \|\bar{a}\|^2 + 2\alpha \bar{a}^T A(x-h) + \alpha^2 \\ B_i^T H(x) B_i &= \|\overline{B}_i\|^2 + 2\beta_i \overline{B}_i^T A(x-h) + \beta_i^2 \\ a^T H_i(x,X) B_i &= \bar{a}^T \overline{B}_i x_i + \beta_i \bar{a}^T A(X_i - x_i h) + \alpha \overline{B}_i^T A(X_i - x_i h) + \alpha\beta_i x_i. \end{aligned}$$

Substituting these expressions into (5) and collecting terms, we obtain a valid linear inequality (cut) of the form $C \bullet X + c^T x + \delta \geq 0$, where $C \bullet \overline{X} + c^T \bar{x} + \delta < 0$. We refer to a linear constraint obtained in this fashion as a "KSOC cut."

3. Computational results. In this section we consider the application of Kronecker product constraints to instances of the two-trust-region subproblem (TTRS). The TTRS, also referred to as the Celis-Dennis-Tapia (CDT) problem [8], arises as a direction-finding subproblem in certain trust-region based methods for nonlinear optimization [9]. The TTRS has the form

$$\begin{aligned} \text{TTRS}: \ & \min x^T Q x + c^T x \\ & \text{s.t. } \|x\| \leq 1, \quad \|A(x-h)\| \leq 1 \end{aligned}$$

where Q is an $n \times n$ symmetric matrix that is not assumed to be PSD. TTRS is a heavily studied problem. Optimality conditions for TTRS are considered in [5], [6], [11] and [13]. In some cases these conditions can provide a constructive proof that a locally optimal solution for TTRS is in fact the global optimum. A convergent trajectory-following method for TTRS is described in [18]. This method is not provably polynomial-time, but a polynomial-time algorithm for TTRS based on methods for polynomial equations [2] is described in [4].

The basic SDP (Shor) relaxation for TTRS is

$$\begin{aligned} \text{TTRS}_{\text{SDP}}: \ & \min Q \bullet X + c^T x \\ & \text{s.t. } A^T A \bullet X - 2h^T A^T A x \leq 1 - h^T A^T A h \\ & \quad\ \operatorname{tr}(X) \leq 1, \quad X \succeq x x^T. \end{aligned}$$

The PSD constraint $X \succeq xx^T$ can be enforced by requiring that $Y(x, X) \succeq 0$, where

$$Y(x, X) = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}.$$

It is well known that the Shor relaxation TTRS_{SDP} can have a nonzero optimality gap, unlike the simpler trust-region subproblem TRS (TTRS without the second constraint $\|A(x-h)\| \leq 1$), for which the Shor relaxation is tight [14]. Note that if (x, X) is feasible in TTRS_{SDP} then x is feasible in TTRS, so an optimal solution (x^*, X^*) of TTRS_{SDP} provides both a feasible objective value $v(x^*) = {x^*}^T Q x^* + c^T x^*$ as well as a lower bound $z(x^*, X^*) = Q \bullet X^* + c^T x^*$. There are a variety of results (see for example [1] and [3]) that give conditions under which the Shor relaxation for TTRS is in fact tight, i.e. $v(x^*) = z(x^*, X^*)$; one example [18, Section 2.2] is the "homogenous" case where $c = h = 0$. There are also cases where optimality conditions for TTRS can establish global optimality for problems where the Shor relaxation is not tight; see for example [5, 6].

For the general case of TTRS, in which the Shor relaxation may not be tight, the approach taken in [7] is to start with TTRS_{SDP}, and then add violated SOC-RLT constraints based on the second-order cone constraints of the problem as in Example 2 of Section 1. After each constraint addition the problem is re-solved and an attempt is made to generate another violated SOC-RLT constraint. This process continues until either no violated constraint can be found, or 25 SOC-RLT constraints are added. At termination, an instance is considered to be solved if the relative gap satisfies

$$\gamma(x^*, X^*) = \frac{v(x^*) - z(x^*, X^*)}{|v(x^*)|} < 10^{-4}, \tag{6}$$

TABLE 1
Comparison of results using KSOC cuts versus Yang and Burer (2016)

		Number of instances solved by:			
n	Instances	KSOC only	YB only	KSOC and YB	Neither
5	38	8	8	12	10
10	70	34	7	14	15
20	104	35	14	24	31
	212	77	29	50	56

where (x^*, X^*) is the optimal solution of TTRS_{SDP} with the added SOC-RLT constraints. This approach is applied to instances of TTRS that are generated based on a theorem of Martínez [12] that are likely to have a gap for TTRS_{SDP} (that is, have $\gamma(x^*, X^*) > 0$ for the solution (x^*, X^*) of TTRS_{SDP}). Using the approach of generating SOC-RLT cuts and a test set consisting of 1000 problems each of dimension 5, 10 and 20, the numbers of unsolved instances are then 41, 70 and 104, respectively.

The results of [7] are improved on by [17]. The methodology of [17] 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 also be extended heuristically to higher dimensions[1]. When applied to test problems from [7], the algorithm of [17] also solves some of the instances that are unsolved using 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 in [17].

The approach we consider here is to again start with the Shor relaxation TTRS_{SDP} but to add cuts based on the Kronecker product constraint $K(x, X) \succeq 0$ as described in the previous section. After each cut addition the problem is re-solved and an attempt is made to generate a new violated constraint. We continue until either $K(x^*, X^*) \succeq 0$, in which case no constraint can be generated, or 25 KSOC cuts have been added. We apply this procedure to the TTRS problems from [7] that were reported as *not* solved using SOC-RLT cuts in both [7] and [17]; these are the 38 problems with $n = 5$ reported as unsolved in [17] and the 70 (respectively 104) problems with $n = 10$ (respectively $n = 20$) reported as unsolved in [7]. Note that by Lemma 3 the condition $K(x, X) \succeq 0$ implies all of the SOC-RLT cuts that could be added, so the problems that were successfully solved using SOC-RLT cuts would also be solved using the approach based on adding the KSOC constraint $K(x, X) \succeq 0$. We verified that all of these problems are also solved by the procedure that adds up to 25 KSOC cuts.

In Table 1 we give a comparison of the results from [17] versus results using cuts based on the KSOC constraint on the instances from [7] that were not previously solved using SOC-RLT cuts. As shown in Table 1, overall results based on the KSOC constraint are better than those from [17], but neither method dominates the other. In all cases a problem is considered to be solved if the relative gap criterion (6) is satisfied at termination. Our 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. Solution times for the problem sizes considered here were quite modest; for

[1]The addition of $K(x, X) \succeq 0$ to TTRS_{SDP} is not sufficient to give an exact representation of TTRS for $n = 2$. This can be demonstrated by numerically solving the example given in [7, Section 5.2]. Adding the constraint $K(x, X) \succeq 0$ reduces the gap obtained using SOC-RLT cuts in [7], but is not sufficient to give the true optimal value of the problem.

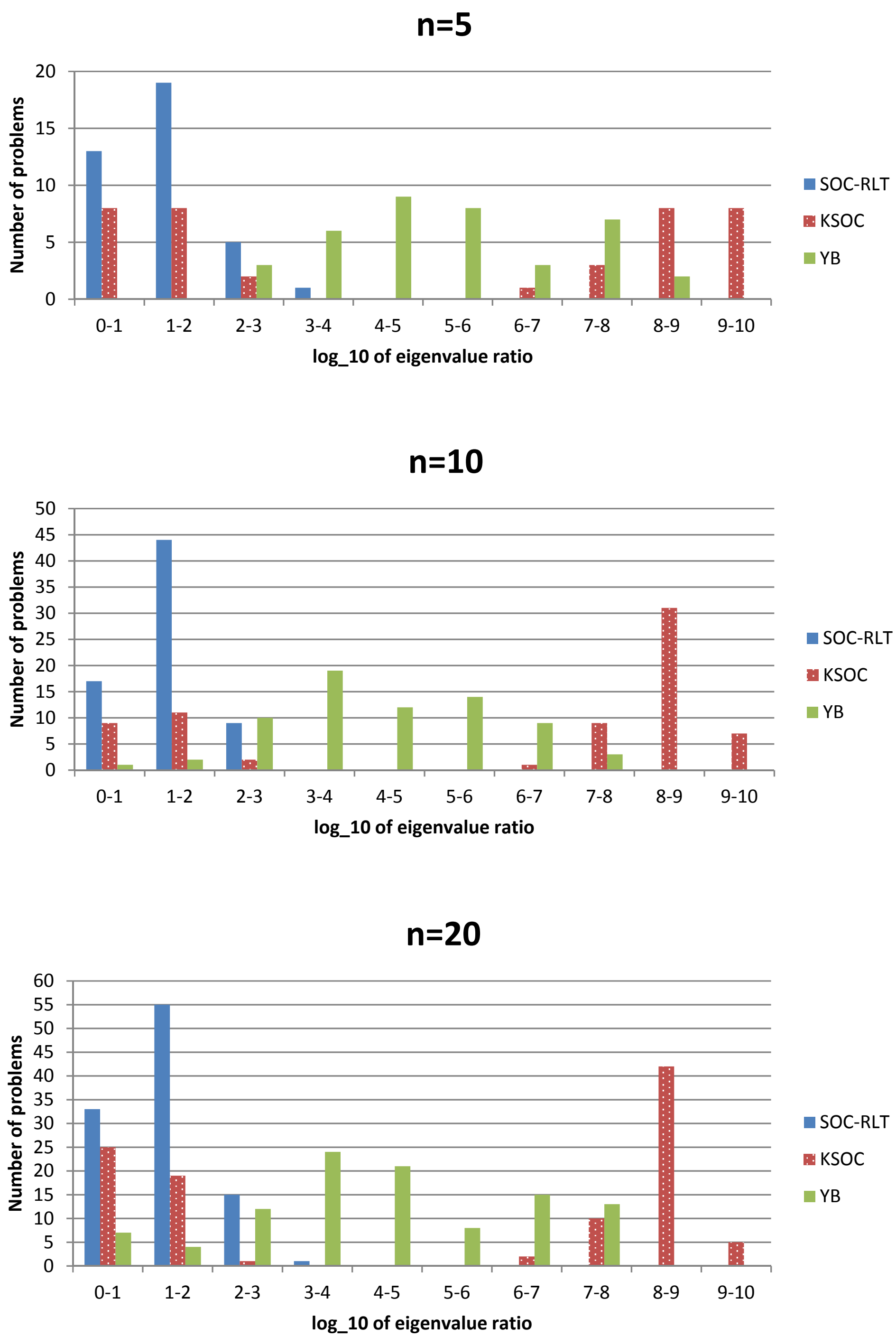

Fig. 1. *Results on TTRS instances not solved using SOC-RLT cuts*

example a problem with $n = 20$, using the maximum of 25 KSOC cuts, requires less that 2 seconds of CPU time, with over 95% of the time dedicated to the SeDuMi solver.

In addition to the relative gap criterion (6), [7] considers a measure of the rank of the solution matrix $Y(x^*, X^*)$. Letting $\lambda_1 \le \lambda_2 \le \dots \lambda_{n+1}$ be the eigenvalues of $Y(x^*, X^*)$, this measure is the eigenvalue ratio λ_{n+1}/λ_n. In [7] it is shown that empirically the eigenvalue ratio is closely related to the relative gap $\gamma(x^*, X^*)$, and there is a gap in the observed eigenvalue ratios around 10^4 that naturally separates "solved" and "unsolved" problems. In Figure 1 we illustrate the distributions of the eigenvalue ratios obtained on our suite of test problems using SOC-RLT cuts, KSOC cuts and the cuts used by Yang and Burer [17]. It is interesting to note that the total number of problems for which the eigenvalue ratio satisfies $\lambda_{n+1}/\lambda_n \ge 10^4$ using KSOC cuts is almost identical to the number of instances that satisfy $\lambda_{n+1}/\lambda_n \ge 10^4$ using YB cuts. However, it is clear from Figure 1 that the distributions of eigenvalue ratios obtained using KSOC cuts is quite different from the distribution obtained using YB cuts. In particular, using KSOC cuts there are no problems with eigenvalue ratios between 10^3 and 10^6, while the results using YB cuts have many problems with eigenvalue ratios in this range. It should also be noted that the limit of 25 KSOC cuts used here is not a critical design factor; we find that problems are typically either solved using a small number of cuts, or alternatively will continue to generate cuts but not substantially improve measures such as the eigenvalue ratio λ_{n+1}/λ_n and relative gap $\gamma(x^*, X^*)$. In the instances considered in Table 1, all but one of the 85 problems that were unsolved using KSOC cuts reached the limit of 25 cuts; one problem of size $n = 20$ terminated with $K(x^*, X^*) \succeq 0$ after 23 cuts were added. On the other hand, of the 127 instances that were solved using KSOC cuts, the average number of cuts required was 8.8 and all but five (three with $n = 5$ and two with $n = 20$) terminated with $K(x^*, X^*) \succeq 0$ before 25 cuts were added.

Since the methodology based on the KSOC constraint used here is completely different from that used in [17], and neither method solves some of the problems from [7], it is reasonable to consider simultaneously applying both classes of cuts. Sam Burer (private communication) implemented such a "combined" method by adding the separation routine for KSOC cuts to the algorithm of [17]. It turns out that of the 56 problems that could not be solved using either KSOC or YB cuts alone, three problems (all with $n = 20$) can be solved when both classes of cuts are implemented together. Burer also reports that the separation problem for KSOC cuts is solved much faster than that for YB cuts.

Acknowledgement. I am grateful to Sam Burer for providing details of the computational results from [17] and for testing the "combined" method with KSOC cuts added to the algorithm of [17], and to two anonymous referees for a number of suggestions that improved the paper.

REFERENCES

[1] W. Ai and S. Zhang, *Strong duality for the CDT subproblem: a necessary and sufficient condition*, SIAM J. Optim., 19 (2008), pp. 1735–1756, http://dx.doi.org/10.1137/07070601X.

[2] A. I. Barvinok, *Feasibility testing for systems of real quadratic equations*, Discrete Comput. Geom., 10 (1993), pp. 1–13, http://dx.doi.org/10.1007/BF02573959.

[3] A. Beck and Y. C. Eldar, *Strong duality in nonconvex quadratic optimization with two quadratic constraints*, SIAM J. Optim., 17 (2006), pp. 844–860, http://dx.doi.org/10.1137/050644471.

[4] D. Bienstock, *A note on polynomial solvability of the CDT problem*, SIAM J. Optim., 26

(2016), pp. 488–498, http://dx.doi.org/10.1137/15M1009871.

[5] I. M. Bomze, *Copositive relaxation beats Lagrangian dual bounds in quadratically and linearly constrained quadratic optimization problems*, SIAM J. Optim., 25 (2015), pp. 1249–1275, http://dx.doi.org/10.1137/140987997.

[6] I. M. Bomze and M. L. Overton, *Narrowing the difficulty gap for the Celis-Dennis-Tapia problem*, Math. Prog., 151 (2015), pp. 459–476, http://dx.doi.org/10.1007/s10107-014-0836-3.

[7] S. Burer and K. M. Anstreicher, *Second-order-cone constraints for extended trust-region subproblems*, SIAM J. Optim., 23 (2013), pp. 432–451, http://dx.doi.org/10.1137/110826862.

[8] M. R. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization, 1984 (Boulder, Colo., 1984), SIAM, Philadelphia, PA, 1985, pp. 71–82.

[9] A. Conn, N. Gould, and P. Toint, *Trust Region Methods*, Society for Industrial and Applied Mathematics, 2000, http://dx.doi.org/10.1137/1.9780898719857.

[10] R. Horn and C. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.

[11] V. Jeyakumar, G. Lee, and G. Li, *Alternative theorems for quadratic inequality systems and global quadratic optimization*, SIAM J. Optim., 20 (2009), pp. 983–1001, http://dx.doi.org/10.1137/080736090.

[12] J. M. Martínez, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM J. Optim., 4 (1994), pp. 159–176, http://dx.doi.org/10.1137/0804009.

[13] J.-M. Peng and Y.-X. Yuan, *Optimality conditions for the minimization of a quadratic with two quadratic constraints*, SIAM J. Optim., 7 (1997), pp. 579–594, http://dx.doi.org/10.1137/S1052623494261520.

[14] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Prog., 77 (1997), pp. 273–299, http://dx.doi.org/10.1007/BF02614438.

[15] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer, 1997.

[16] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Math. Oper. Res., 28 (2003), pp. 246–267, http://dx.doi.org/10.1287/moor.28.2.246.14485.

[17] B. Yang and S. Burer, *A two-variable approach to the two-trust-region subproblem*, SIAM J. Optim., 26 (2016), pp. 661–680, http://dx.doi.org/10.1137/130945880.

[18] Y. Ye and S. Zhang, *New results on quadratic minimization*, SIAM J. Optim., 14 (2003), pp. 245–267, http://dx.doi.org/10.1137/S105262340139001X.