A note on polynomial solvability of the CDT problem
Daniel Bienstock, Columbia University, December 2013

Abstract
We describe a simple polynomial-time algorithm for the CDT problem that relies on a construction of Barvinok.

1 Introduction
The CDT (Celis-Dennis-Tapia) problem [10] concerns the minimization of a quadratic function over the intersection of two ellipsoids. This problem, which generalizes the trust-region subproblem (see [8], [17], [6] for recent complexity results) has long generated interest. See [8], where it was termed the “two trust-region” problem, [1], [5], [15], [19], and references therein; also see [18] and [7], as well as their references. Broadly speaking these papers have sought to exploit the connection between CDT and semidefinite programming; this approach is related to the use of the S-Lemma to solve the classical trust-region subproblem. See [16].

A separate line of work has produced results of a very different flavor that address related problems. Barvinok [2] (“Problem(1.1)”) proved that for each integer \( K \)

\[
\begin{align*}
x^T M_i x & = 0, \quad 1 \leq i \leq K, \\
\|x\|^2 & = 1,
\end{align*}
\]

is feasible, where \( x \in \mathbb{R}^n \) and \( M_i \) is an \( n \times n \) matrix for \( 1 \leq i \leq K \). Related and stronger results were presented by Grigoriev and Pasechnik in [13]. Moreover, it was argued in [14] that the results in [13] imply that a polynomial-time algorithm for a generalization of CDT exists. Also see [3].

Yet another line of research comes from the algebraic and computational geometry communities, in particular the theory of “roadmaps” of semi-algebraic sets, which was started in [9]. This topic appears related to the work cited above on “sampling” algebraic sets. See [4], and [12] for recent research results and additional citations. These research efforts have produced results with applications to diverse problem domains. It is quite possible that other polynomial-time algorithms for CDT can be derived from this work. A nontrivial point concerning the resulting algorithms is the need to argue for polynomial-time complexity in the bit model of computation, as opposed to computation over the reals.

The purpose of this note is to present a simple algorithm that uses a weak version of Barvinok’s construction to obtain a polynomial-time algorithm for a generalization of CDT. This algorithm will use a relaxed version of feasibility, as follows.

**Definition 1.1** Let \( f_i(x) \leq 0, \ 1 \leq i \leq m \) be a system of inequalities where \( x \in \mathbb{R}^n \).

(a) Given \( 0 < \varepsilon < 1 \) a vector \( \hat{x} \in \mathbb{R}^n \) is called \( \varepsilon \)-feasible for the system if \( f_i(\hat{x}) \leq \varepsilon \) for \( 1 \leq i \leq m \). If such a vector exists we will say that the system is \( \varepsilon \)-feasible.

(b) An algorithm that, given any \( 0 < \varepsilon < 1 \) either proves that the system is infeasible, or proves that it is \( \varepsilon \)-feasible will be called a *weak feasibility algorithm*.

The weak version of Barvinok’s construction is as follows:

**Assumption 1.2** We will assume that for each \( K \), there is weak feasibility algorithm for systems of the type (1) with running time polynomial in the size of the data and \( \log \varepsilon^{-1} \).

Barvinok’s method [2] clearly fulfills this role, as does Grigoriev and Pasechnik’s [13]. It is also possible that faster weak feasibility algorithms exist, as opposed to algorithms for strict feasibility as in (1).

To describe our main result, suppose that for \( 0 \leq i \leq p \), \( g_i(x) \) is a quadratic (i.e., a degree \( \leq 2 \) polynomial), over \( x \in \mathbb{R}^n \). We consider the problem

\[
\min \{ g_0(x) : g_i(x) \leq 0, \ 1 \leq i \leq p \},
\]

and prove the following:
Theorem 1.3 For each fixed integer $p \geq 1$ there is an algorithm with the following properties. Given a problem of the form (2), where at least one of the $g_i(x)$ with $i \geq 1$ is strictly convex, and $0 < \varepsilon < 1$, the algorithm either

(1) proves that problem (2) is infeasible,

or

(2) computes an $\varepsilon$-feasible vector $\hat{x}$ such that there exists no feasible $x \in \mathbb{R}^n$ with $g_0(x) < g(\hat{x}) - \varepsilon$.

Under Assumption 1.2 the algorithm runs in polynomial time. More precisely, the algorithm makes a sequence of calls to a weak feasibility algorithm for problems of type (1) with $K = O(p)$; the length of the sequence is polynomial in the number of bits in the data and $\log \varepsilon^{-1}$, as is the size of the coefficients of the matrices $M_i$, and as is all additional work carried out by the algorithm.

This result is proved in several steps in Section 2. Under Assumption 1.2, Theorem 1.3 implies that a polynomial-time algorithm for CDT exists. Throughout, we assume $n \geq 2$.

2 The construction

This section is organized as follows. In Section 2.1 we describe an algorithm to determine if a system of $m$ quadratic inequalities is $\varepsilon$-feasible; this algorithm runs in polynomial time for each fixed $m$ provided that at least one of the quadratics is strictly convex. In Section 2.2 the algorithm in Section 2.1 is used to compute the value of problem (2) within tolerance $\varepsilon$, in polynomial time, under the assumptions in Theorem 1.3. However, this does not yet yield a proof of Theorem 1.3 because the algorithm we describe in Section 2.1 relies on the weak feasibility algorithm in Assumption 1.2 as a subroutine. That algorithm (and in a strict sense, Barvinok’s) may only decide if a system of the form (1) is $\varepsilon$-feasible but without producing an explicit $\varepsilon$-feasible vector. In Section 2.3 we show how to refine our algorithm from Section 2.1 so as to produce an $\varepsilon$-feasible vector in the case that infeasibility is not proved. Together with the results in Section 2.2 this completes the proof of Theorem 1.3.

2.1 Systems of quadratic inequalities

Here we consider a system of quadratic inequalities

$$f_i(x) \leq 0, \quad 1 \leq i \leq m. \quad (3)$$

We write

$$f_i(x) = x^T A_i x + c_i^T x + d_i \quad (4)$$

where $A_i \in \mathbb{R}^{n \times n}$ and symmetric, $c_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$. Such a system describes the feasibility set for a problem of the form (2); more generally we will use the solution of systems of the form (3) as steps in our algorithm for problem (2). Our main result for this section, proved below, is as follows:

**Theorem 2.1** Under Assumption 1.2, for each fixed $m$ there is a polynomial-time weak feasibility algorithm for a system of type (3) if $A_i \succ 0$ for at least one index $i \geq 1$.

We will first prove two technical results, Lemma 2.2 and 2.3, under the assumptions of Theorem 2.1. Thus, assume without loss of generality that

$$f_1(x) = \|x\|^2 - 1. \quad (5)$$

Then, for $2 \leq i \leq m$, there exists (polynomially computable) $U_i > 0$, such that

$$|f_i(x)| \leq U_i, \quad \text{for each } x \in \mathbb{R}^n \text{ with } \|x\|^2 \leq 2.$$
Now consider the following system of quadratic equations on real variables \(v_0, x_1, \ldots, x_n, s_1, \ldots, s_m, w_2, \ldots, w_m\):

\[
\begin{align*}
  x^T A_i x + c_i^T v_0 x + d_i v_0^2 + s_i^2 & = 0 & 1 \leq i \leq m, \quad (6a) \\
  s_i^2 + w_i^2 & = 0 & 2 \leq i \leq m, \quad (6b) \\
  \|x\|^2 + s_1^2 + \sum_{i=2}^{n} \frac{s_i^2 + w_i^2}{U_i} + v_0^2 & = m + 1. \quad (6c)
\end{align*}
\]

**Lemma 2.2** Let \(0 < \delta < 1\) and suppose that

\[
\tilde{x} = (\tilde{v}_0, \tilde{x}_1, \ldots, \tilde{x}_n, \tilde{s}_1, \ldots, \tilde{s}_m, \tilde{w}_2, \ldots, \tilde{w}_m)^T
\]

is a \(\delta\)-feasible solution to (6a)-(6b). Then (i)

\[
m\tilde{v}_0^2 - m\delta \leq \|x\|^2 + \tilde{s}_1^2 + \sum_{i=2}^{n} \frac{s_i^2 + w_i^2}{U_i} \leq m\tilde{v}_0^2 + m\delta,
\]

and (ii)

\[
1 - \delta \leq \tilde{v}_0^2 \leq 1 + \delta.
\]

**Proof.** (i) Note that \(\delta\)-feasibility of \(\tilde{x}\) applied to (6a) for \(i = 1\) states

\[
-\delta \leq \|x\|^2 - \tilde{v}_0^2 + \tilde{s}_1^2 \leq \delta,
\]

and applied to (6b) it states

\[
-\delta \leq \frac{s_i^2 + w_i^2}{U_i} - \tilde{v}_0^2 \leq \delta, \quad 2 \leq i \leq m.
\]

Adding these inequalities yields (7). (ii) Since \(\tilde{x}\) is \(\delta\)-feasible for (6c)

\[
m + 1 - \delta \leq \|x\|^2 + \tilde{s}_1^2 + \sum_{i=2}^{n} \frac{s_i^2 + w_i^2}{U_i} + \tilde{v}_0^2 \leq m + 1 + \delta
\]

which together with (7) implies the desired result. \(\blacksquare\)

Let \(M\) denote the largest absolute value of a coefficient in (6).

**Lemma 2.3** Let \(0 < \epsilon < 1\). (a) Suppose (3) is \(\epsilon\)-feasible. Then (6) is \(m\epsilon\)-feasible. (b) Conversely, if (6) is \(\epsilon\)-feasible, then (3) is \(((2n+1)M+1)\epsilon\)-feasible.

**Proof.** (a) Suppose first that \(\tilde{x}\) is an \(\epsilon\)-feasible solution to (3). Write

\[
\begin{align*}
  \tilde{v}_0 & = 1, \quad (8) \\
  \tilde{s}_i^2 & = \max\{0, -f_i(\tilde{x})\}, \quad 1 \leq i \leq m, \quad (9) \\
  \tilde{w}_i^2 & = U_i - \tilde{s}_i^2, \quad 2 \leq i \leq m. \quad (10)
\end{align*}
\]

Note that (10) is valid since \(\|\tilde{x}\|^2 \leq 1 + \epsilon < 2\). Now we claim that

\[
\tilde{z} = (\tilde{x}_1, \ldots, \tilde{x}_n, \tilde{v}_0, \tilde{s}_1, \ldots, \tilde{s}_m, \tilde{w}_2, \ldots, \tilde{w}_m)^T
\]

is an \(\epsilon\)-feasible solution to (6). To see this, note that \(\tilde{v}_0 = 1\) and (9) imply that \(\tilde{z}\) is \(\epsilon\)-feasible for (6a). Likewise, \(\tilde{z}\) satisfies (6b) by (10) and \(\tilde{v}_0 = 1\). Finally, Lemma 2.2, part (i) (with \(\tilde{x} = \tilde{x}\) and \(\delta = \epsilon\)), together with \(\tilde{v}_0 = 1\) implies that \(\tilde{z}\) is \(m\epsilon\)-feasible for (6c), as desired.

(b) For the converse, suppose

\[
\tilde{z} = (\tilde{x}_1, \ldots, \tilde{x}_n, \tilde{v}_0, \tilde{s}_1, \ldots, \tilde{s}_m, \tilde{w}_2, \ldots, \tilde{w}_m)^T
\]
is \( \varepsilon \)-feasible for (6). By Lemma 2.2 (ii),
\[ 1 - \varepsilon \leq \hat{v}_0 \leq 1 + \varepsilon, \]
This implies
\[ 1 - \varepsilon \leq |v_0| \leq 1 + \varepsilon, \quad (11) \]
and together with is \( \varepsilon \)-feasibility of \( \hat{x} \) for (6a) with \( i = 1 \), it implies
\[ |\hat{x}_j| \leq 1 + \varepsilon, \quad \text{for } 1 \leq j \leq n. \quad (12) \]
Bounds (11) and (12), together with \( \varepsilon \)-feasibility of \( \hat{x} \) for (6a) imply that \( \text{sgn}(\hat{v}_0)(\hat{x}_1, \ldots, \hat{x}_n)^T \) is \((2n + 1)M + 1\)\( \varepsilon \)-feasible for (3).

**Corollary 2.4** A system (3) is feasible if and only if the corresponding system (6) is feasible.

Now we can present the proof of the main result.

**Proof of Theorem 2.1.** Consider the corresponding system (6), and let \( \delta \doteq \varepsilon \frac{2n}{2n + 1} \). Using the method in Assumption 1.2, we terminate in polynomial time with a proof that system (6) is infeasible (in which case (3) is infeasible (by Corollary 2.4)) or, using part (2) of Lemma 2.3, with a proof that (3) is \( \varepsilon \)-feasible.

### 2.2 Computing the value of problem (2)

Write
\[ G^* \doteq \min \{ g_0(x) : g_i(x) \leq 0, \ 1 \leq i \leq p \}. \quad (13) \]

In this section we show how to compute \( G^* \), within tolerance \( \varepsilon \), in polynomial time. Since we assume that at least one of the \( g_i(x) \), for \( i \geq 1 \), is positive definite, we can compute, in polynomial time, a quantity \( U > 0 \) such that \( |G^*| \leq U \).

Let \( 0 < \varepsilon < 1 \) be given. Suppose we apply binary search to problem (2). In a number of iterations which is \( O(\log U + \log \varepsilon^{-1}) \) (which is polynomial in the number of bits in the data) we will compute at termination a value \( V \) such that
\[ V \leq G^* \leq V + \varepsilon. \]

Each iteration of the binary search will require the solution of the weak feasibility problem, with tolerance \( \varepsilon/4 \), for a system of the form
\[ \begin{align*}
g_i(x) & \leq 0 \quad \text{for } 1 \leq i \leq p, \quad (14a) \\
g_0(x) & \leq U \quad (14b)
\end{align*} \]

where the total number of bits in \( U \) is polynomial in the input data. We can determine whether such a system is \( \varepsilon/4 \)-feasible (or infeasible) as in Section 2.1. If infeasible, we have proved \( U \) is a lower bound on \( G^* \) and otherwise \( U + \varepsilon/4 \) is an upper bound. We continue until the gap between the lower and upper bounds is at most \( \varepsilon \).

### 2.3 Computing explicit solutions in polynomial time

Here we will show that for each fixed \( m \) there is a polynomial-time algorithm that, given a system of the form
\[ \begin{align*}
f_i(x) & \leq 0, \quad 1 \leq i \leq q, \\
\|x\|^2 & = 1, \quad (15a)
\end{align*} \]

where the \( f_i(x) \) are quadratics, and \( 0 < \varepsilon < 1 \), either proves the system is infeasible, or computes an explicit \( \varepsilon \)-feasible solution. Any system of the form (6) used in the algorithm in Section 2.1 can be reduced, by scaling, to an equivalent system (15) with \( q = O(m) \) and \( n \) appropriately redefined.
Given a system (15) and \(0 < \varepsilon < 1\), Algorithm C (below) in polynomial time will decide that the system is infeasible or compute an \(\varepsilon\)-feasible rational vector \(\hat{x} \in \mathbb{R}^n\).

In preparation for the algorithm choose \(\gamma = \gamma(\varepsilon) < 1\) so that whenever \(x, y \in \mathbb{R}^n\) are such that \(|x| < 2\) and \(|x_j - y_j| \leq \gamma\) for \(1 \leq j \leq n\), then \(|f_i(y) - f_i(x)| \leq \varepsilon\) for all \(1 \leq i \leq q\), and also \(||x||^2 - ||y||^2\) \(\leq \varepsilon\). Such a value \(\gamma\) can be computed in polynomial time (in the number of bits in the data, and in \(\log \varepsilon^{-1}\)).

The algorithm given next will compute, for \(k = 1, 2, \ldots, n\), the value \(\hat{x}_k\). We will prove in Corollary 2.10 that at termination the vector \(\hat{x}\) is \(\varepsilon\)-feasible for (15).

**Algorithm C**

**Initialization.** Set \(k = 1\).

**Step 1.** Let \(z\) denote the vector \((\hat{x}_1 + \delta_1, \ldots, \hat{x}_{k-1} + \delta_{k-1}, x_k, \ldots, x_n)^T\) where for \(1 \leq j \leq k - 1\) the \(\delta_j\) indicate new variables and the \(\hat{x}_j\) are values computed in prior iterations. Let

\[
Z^k = \left\{ z : f_i(z) \leq 0 \quad \text{for} \quad 1 \leq i \leq q, \quad ||z||^2 \leq 1, \quad \sum_{j=1}^{k-1} \delta_j^2 \leq (k - 1) \frac{\gamma^2}{n^4} \right\}.
\]

For \(h = k, k + 1, \ldots, n\) use the algorithm in 2.2 to produce one of the following two outcomes (1a) and (1b), in polynomial time:

(1a) Decide that \(Z^h = \emptyset\). If so, stop, and declare (15) infeasible.

(1b) Decide that \(Z^h\) is \(\frac{\gamma^2}{3^m}\)-feasible and compute rationals \(\bar{P}_{k,h}\) and \(\bar{M}_{k,h}\) such that

\[
|\max\{z_h : z \in Z^h\} - \bar{P}_{k,h}| \leq \frac{\gamma^2}{3n^4} \quad \text{and} \quad |\min\{z_h : z \in Z^h\} - \bar{M}_{k,h}| \leq \frac{\gamma^2}{3n^4}.
\]

**Step 2.** For \(h = k, k + 1, \ldots, n\), if \(\bar{P}_{k,h} < \bar{M}_{k,h}\) we reset \(\bar{M}_{k,h} = \bar{P}_{k,h}\).

**Step 3.** Let \(m_k\) be the maximum of all values \(\bar{P}_{k,h}\) and \(-\bar{M}_{k,h}\) over \(k \leq h \leq n\). If \(m_k = \bar{P}_{k,h}\) (for some \(h\)) then we set \(\hat{x}_k = m_k\); else we set \(\hat{x}_k = -m_k\).

**Step 4.** If \(k < n\) and \(\sum_{j=1}^{k} \delta_j^2 \leq 1 - \gamma\), set \(k \leftarrow k + 1\) and go to Step 1. Otherwise, define \(k^* = k\) and set \(\hat{x}_h = 0\) for \(k + 1 \leq h \leq n\). and stop the procedure.

We now analyze Algorithm C.

**Lemma 2.5** For \(k \leq k^*\) and \(k \leq h \leq n\), \(|\min\{z_h : z \in Z^k\} - \bar{M}_{k,h}| \leq \gamma^2/(3n^4)\).

**Proof.** Clearly this property holds before executing Step 2 at iteration \(k\), and if \(\bar{M}_{k,h}\) is reset in Step 2 then it holds because \(\min\{z_h : z \in Z^k\} \leq \max\{z_h : z \in Z^k\}\).

**Remark 2.6** Using Step 2 we have that for \(k \leq k^*\) and \(k \leq h \leq n\), if \(0 \leq \bar{M}_{k,h}\) then \(0 \leq \bar{M}_{k,h} \leq \bar{P}_{k,h}\) and if \(\bar{P}_{k,h} \leq 0\) then \(0 \leq -\bar{P}_{k,h} = -\bar{M}_{k,h}\). Hence, in both cases, Lemma 2.5 yields that \(\max\{\bar{P}_{k,h}, -\bar{M}_{k,h}\}\) approximates \(\max\{|z_h| : z \in Z^k\}\).

**Lemma 2.7** Suppose system (15) is feasible. Then for any \(k \leq k^*\), \(Z^k \neq \emptyset\). As a corollary if the algorithm stops at Step 1a, (15) is infeasible.

**Proof.** By induction on \(k\). For \(k = 1\) the result follows since \(Z^1\) is the set of points feasible for system (15). We will prove next that if \(Z^k \neq \emptyset\) and \(k < k^*\), then \(Z^{k+1} \neq \emptyset\). To see this, we have by construction that there is a vector

\[
\tilde{z} = (\hat{x}_1 + \tilde{\delta}_1, \ldots, \hat{x}_{k-1} + \tilde{\delta}_{k-1}, \hat{x}_k, \ldots, \hat{x}_n)^T,
\]
such that \( \tilde{z} \in Z^k \) and
\[
|\hat{x}_k^2 - \tilde{x}_k^2| = |\hat{x}_k - \tilde{x}_k||\hat{x}_k + \tilde{x}_k| \leq \frac{\gamma^2}{n^4}.
\]
It follows that by setting \( \delta_k = \hat{x}_k - \tilde{x}_k \), the vector
\[
(\hat{x}_1 + \delta_1, \ldots, \hat{x}_k + \delta_k, \tilde{x}_{k+1}, \ldots, \tilde{x}_n)^T
\]
is contained in \( Z^{k+1} \).

**Lemma 2.8** Let \( k \leq k^* \), and suppose that the algorithm does not stop at Step 1a in iteration \( k \). Then \( \hat{x}_k^2 \geq \frac{\gamma}{4n} \).

**Proof.** Since the algorithm has not terminated by iteration \( k - 1 \), \( \sum_{j=1}^{k-1} \tilde{x}_j^2 < \gamma \). Since at iteration \( k \) the algorithm does not stop in Step 1a, it follows that there is a vector \( \hat{z} \) that is \( \tilde{x}^2 \)-feasible for \( Z^k \).
Thus
\[
\sum_{j=1}^{k-1} (\tilde{x}_j + \delta_j)^2 + \sum_{j=k}^n \tilde{x}_j^2 \geq 1 - \frac{\gamma^2}{3n^4}.
\]

Thus, in Step 3 the maximum of all the values \( \tilde{R}_{k,h} \) and \( -M_{k,h} \) must be at least
\[
\left[ \frac{1}{n-k+1} \frac{1}{1 - (1 - \gamma^2/n^2)^2} \right]^{1/2} - \frac{\gamma^2}{3n^4} \geq \left[ \frac{\gamma - 2\gamma/n^2 - \gamma^2/n^3}{n} \right]^{1/2} - \frac{\gamma^2}{3n^4} > \left[ \frac{\gamma}{4n} \right]^{1/2},
\]
using simple estimates.

**Lemma 2.9** Let \( \bar{z} \) be any point in \( Z^{k^*} \). (a) For any \( 1 \leq k \leq k^* \), \( |\hat{x}_k^2 - \tilde{x}_k^2| \leq \frac{\gamma}{8n^2} \tilde{x}_k^2 \). (b) Suppose \( k^* < n \). Then \( \sum_{j=k+1}^n \tilde{x}_j^2 \leq \gamma/n^2 \).

**Proof.** (a) Proceeding as in the proof of Lemma 2.7, \( \sum_{j=1}^{k^*} |\hat{x}_k^2 - \tilde{x}_k^2| \leq k^* \frac{\gamma^2}{n^2} \), from which the result follows, using Lemma 2.8.
(b) By the conditions for termination of the algorithm, \( \sum_{j=1}^{k^*} \hat{x}_j^2 \geq 1 - \gamma \). Thus, by (a),
\[
\sum_{j=1}^{k^*} \tilde{x}_j^2 \geq (1 - \frac{\gamma}{8n^2})(1 - \gamma) > 1 - \frac{\gamma}{n^2}
\]
again by simple estimates.

**Corollary 2.10** Algorithm C terminates in time polynomial in the number of bits in the data and \( \log \varepsilon^{-1} \) and at termination the vector \( \hat{x} \) is \( \varepsilon \)-feasible for (15).

**Proof.** The computation of the quantities \( \tilde{R}_{k,h} \) and \( \tilde{M}_{k,h} \) in each execution of Step 1 is performed in polynomial time using the algorithm in Section 2.2. Moreover, by Lemma 2.7, \( Z^{k^*} \neq \emptyset \). Thus, given any \( \bar{z} \in Z^{k^*} \), by Lemma 2.9 we have \( |x_j - \bar{z}_j| < \gamma \) for any \( 1 \leq j \leq n \).

**Remark.** Lemma 2.9 proves that not only is \( \hat{x} \) a close approximation to \( \bar{z} \), but that whenever \( \hat{x}_k \neq 0 \) we have that the relative error incurred in estimating \( \tilde{x}_k \) using \( \delta_k \) is small.

**Acknowledgment.** This work was partly supported by ONR award N00014-13-1-0042.

**References**

[1] W. Ai and S. Zhang, Strong duality for the CDT subproblem: a necessary and sufficient condition, *SIAM J. Optim.* 19 (2008), 1735-1756.
[2] A. I. BARVINOK, Feasibility Testing for Systems of Real Quadratic Equations, *Disc. Comput. Geometry* **10** (1993), 1 – 13.

[3] S. BASE, D. PASECHNIK AND M.-F. ROY, Bounding the Betti numbers and computing the Euler-Poincare characteristic of semi-algebraic sets defined by partly quadratic systems of polynomials, arXiv.org preprint math.AG/0708.3522 (shortened version in *J. European Math. Soc.* **12** (2010), 529 - 553).

[4] S. BASU, R. POLLACK AND M.-F. ROY, Algorithms in real algebraic geometry, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, second edition (2006).

[5] A. BECK AND Y. C. ELDAR, Strong duality in nonconvex quadratic optimization with two quadratic constraints, *SIAM J. Optim.* **17** (2006), 844-860.

[6] D. BIENSTOCK AND A. MICHALKA, Polynomial solvability of variants of the trust-region subproblem. To appear, *Proc. 2014 ACM-SIAM Symposium on Discrete Algorithms (SODA ’14).* http://www.optimization-online.org/DB_HTML/2013/07/3951.html.

[7] I.M. BOMZE AND M.L. OVERTON, Narrowing the difficulty gap for the Celis-Dennis-Tapia problem. http://www.optimization-online.org/DB_HTML/2013/11/4121.html.

[8] S. Burer AND K. ANSTREICHER, Second-order cone constraints for extended trust-region subproblems, *SIAM Journal on Optimization* **23** (2013) 432–451.

[9] J. CANNY, The complexity of robot motion planning. PhD thesis, MIT (1987).

[10] M. R. CELIS, J. E. DENNIS AND R. A. TAPIA, A trust region strategy for nonlinear equality constrained optimization. In P.T. Boggs, R.H. Byrd and R.B. Shnabel, editors, *Numerical Optimization*, 71-82, SIAM, Philadelphia, PA, 1985.

[11] X.-D. CHEN AND Y.-X. YUAN, Strong duality for the CDT subproblem: a necessary and sufficient condition, *J. Comp. Math.* **19** (2009) 113-124.

[12] M. SAFEY EL DIN AND É. SCHOST, A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. http://arxiv.org/abs/1307.7836.

[13] D. GRIGORIEV AND D.V. PASECHNIK, Polynomial-time computing over quadratic maps I. Sampling in real algebraic sets, *Computational Complexity* **14** (2005), 20–52.

[14] D. GRIGORIEV, D.V. PASECHNIK AND E. DE KLERK, Quadratic optimization subject to a fixed number of quadratic constraints is polynomial-time. Slides for a talk (2002).

[15] J.-M. PENG AND Y.-X. YUAN, Optimality conditions for the minimization of a quadratic with two quadratic constraints, *SIAM J. Optim.* **7** (1997), 579-594.

[16] I. PÓLIK AND T. TERLAKY, A survey of the S-lemma, *SIAM Review* **49** (2007), 371 – 418.

[17] B. YANG AND S. BURER, The Trust Region Subproblem with Non-Intersecting Linear Constraints. To appear, *Math. Programming.* http://www.optimization-online.org/DB_HTML/2013/02/3789.html.

[18] B. YANG AND S. BURER, A Two-Variable Analysis of the Two-Trust-Region Problem. http://www.optimization-online.org/DB_HTML/2013/11/4126.html.

[19] Y. YE AND S. ZHANG, New results on quadratic minimization, *SIAM J. Optim.* **14** (2003), 245 – 267.