ON TIME COMPLEXITY OF SEMIDEFINITE PROGRAMS ARISING IN POLYNOMIAL OPTIMIZATION *

IGOR KLEP[1,3], JANEZ POVH[2,3], AND ANGELIKA WIEGELE[3]

ABSTRACT. In this paper we investigate matrix inequalities which hold irrespective of the size of the matrices involved, and explain how the search for such inequalities can be implemented as a semidefinite program (SDP). We provide a comprehensive discussion of the time complexity of these SDPs.

1. INTRODUCTION

Starting with Helton's seminal paper [Hel02], *free real algebraic geometry* is being established. Unlike classical real algebraic geometry where real polynomial rings in *commuting* variables are the objects of study, free real algebraic geometry deals with real polynomials in free *noncommuting* (nc) variables. Such polynomials can be evaluated at tuples of matrices giving rise to various notions of positivity. We call an nc polynomial $f(X_1,\ldots,X_n)$ *positive* if $f(A_1,\ldots,A_n)$ is positive semidefinite for all tuples of matrices $A_1,\ldots,A_n$ (of all sizes!).

1.1. **Motivation.** Among the things that make free real algebraic geometry exciting are its many facets of applications. Let us mention just a few. A nice survey on applications to control theory, systems engineering and optimization is given by Helton, McCullough, Oliveira, Putinar [HMdOP08], applications to quantum physics are explained by Pironio, Navascués, Acín [PNA10] who also consider computational aspects related to noncommutative sum of squares. For instance, optimization of nc polynomials has direct applications in quantum information science (to compute upper bounds on the maximal violation of a generic Bell inequality [PV09]), and also in quantum chemistry (e.g. to compute the ground-state electronic energy of atoms or molecules, cf. [Maz04]). Certificates of positivity via sums of squares are often used in the theoretical physics literature to place very general bounds on quantum correlations (cf. [Gla63]). Doherty, Liang, Toner, Wehner [DLTW08] employ free real algebraic geometry to consider the quantum moment problem and multi-prover games.

We developed `NCSOStools` [CKP11] as a consequence of this recent interest in free real algebraic geometry. `NCSOStools` is an open source Matlab toolbox for optimization of nc polynomials using *semidefinite programming* (SDP). As a side product our toolbox implements symbolic computation with noncommuting variables in Matlab. Readers interested in optimization for commuting polynomials are referred to one of the many great existing packages, such as GloptiPoly [HLL09], SOSTOOLS [PPSP05], SparsePOP [WKK+09], or YALMIP [Löf04].

1.2. **Contribution.** The purpose of this article is twofold.

First, we will explain that every noncommutative (nc) polynomial that is positive on certain "semialgebraic" sets of symmetric matrices of all sizes, admits a sum of hermitian squares representation (with weights) and tight degree bounds (the so-called *Nichtnegativstellensatz* 4.6). Loosely speaking, every polynomial matrix inequality which holds independent of the size of the matrices involved, admits a sum of squares certificate; cf. Example 2.1 below for a sample.

Second, by the existence of sharp degree bounds, optima (and even optimizers) for nc polynomials can be computed exactly by solving a *single* semidefinite programming problem (SDP). We discuss the size of this SDP and the time complexity for solving it in §5 below.

Date: December 18, 2012.
2010 *Mathematics Subject Classification.* Primary 90C22, 14P10; Secondary 13J30, 47A57.
Key words and phrases. optimization, sum of squares, semidefinite programming, polynomial, Matlab toolbox, real algebraic geometry.
[1] Research partially supported by the Faculty Research Development Fund (FRDF) of The University of Auckland (project no. 3701119), and the Slovenian Research Agency grants N1-0006 and P1-0222.
[2]Supported by the Slovenian Research Agency - program nr. P1-0297(B).
[3]Supported by the OeAD - Austrian Agency for International Cooperation in Education and Research, project nr. SI 26/2011 and by the Slovene Research agency, project nr. BI-AT/11-12-004.

1.3. **Reader's guide.** The paper starts with a preliminary section fixing notation, introducing terminology and stating some well-known classical results on positive nc polynomials (§2 and §3). We then in §4 explain the Nichtnegativstellensatz and optimization of nc polynomials. That is, we present the construction and properties of the SDP computing the minimum of an nc polynomial. Finally, time complexity of these SDPs is discussed in §5. In Appendix §A we give some fundamental properties and background results on SDP.

We have implemented our algorithms in our open source Matlab toolbox `NCSOStools` freely available at `http://ncsostools.fis.unm.si/`. We refer the reader to [CKP11] and [CKP12] for further examples illustrating the results presented here, and for the use of our computer algebra package.

2. Notation and Preliminaries

2.1. **Words, free algebras and nc polynomials.** Fix $n \in \mathbb{N}$ and let $\langle \underline{X} \rangle$ be the monoid freely generated by $\underline{X} := (X_1, \ldots, X_n)$, i.e., $\langle \underline{X} \rangle$ consists of *words* in the n noncommuting letters $X_1, \ldots, X_n$ (including the empty word denoted by 1). We consider the free algebra $\mathbb{R}\langle \underline{X} \rangle$. The elements of $\mathbb{R}\langle \underline{X} \rangle$ are linear combinations of words in the n letters $\underline{X}$ and are called *noncommutative* (*nc*) *polynomials*. An element of the form aw where $a \in \mathbb{R} \setminus \{0\}$ and $w \in \langle \underline{X} \rangle$ is called a *monomial* and a its *coefficient*. Thus words are monomials with coefficient 1. The length of the longest word in an nc polynomial $f \in \mathbb{R}\langle \underline{X} \rangle$ is the *degree* of f and is denoted by $\deg f$. The set of all words and nc polynomials with degree $\leq d$ will be denoted by $\langle \underline{X} \rangle_d$ and $\mathbb{R}\langle \underline{X} \rangle_d$, respectively. If we are dealing with only two variables, we shall use X, Y instead of X_1, X_2.

By $\mathbb{S}_k$ we denote the set of all symmetric $k \times k$ real matrices and by $\mathbb{S}_k^{\succeq 0}$ we denote the set of all real positive semidefinite $k \times k$ real matrices. Moreover, $\mathbb{S} := \bigcup_{k \in \mathbb{N}} \mathbb{S}_k$ and $\mathbb{S}^{\succeq 0} := \bigcup_{k \in \mathbb{N}} \mathbb{S}_k^{\succeq 0}$. If a real symmetric matrix A is positive semidefinite we denote this by $A \succeq 0$.

2.1.1. *Sums of hermitian squares.* We equip $\mathbb{R}\langle \underline{X} \rangle$ with the *involution* $*$ that fixes $\mathbb{R} \cup \{\underline{X}\}$ pointwise and thus reverses words, e.g. $(X_1 X_2^2 X_3 - 2X_3^3)^* = X_3 X_2^2 X_1 - 2X_3^3$. Hence $\mathbb{R}\langle \underline{X} \rangle$ is the $*$-algebra freely generated by n symmetric letters. Let $\operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle$ denote the set of all *symmetric polynomials*,

$$\operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle := \{f \in \mathbb{R}\langle \underline{X} \rangle \mid f = f^*\}.$$

An nc polynomial of the form $g^* g$ is called a *hermitian square* and the set of all sums of hermitian squares will be denoted by Σ^2. Clearly, $\Sigma^2 \subsetneq \operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle$. The involution $*$ extends naturally to matrices (in particular, to vectors) over $\mathbb{R}\langle \underline{X} \rangle$. For instance, if $V = (v_i)$ is a (column) vector of nc polynomials $v_i \in \mathbb{R}\langle \underline{X} \rangle$, then V^* is the row vector with components v_i^*. We use V^t to denote the row vector with components v_i.

We can stack all words from $\langle \underline{X} \rangle_d$ using the graded lexicographic order into a column vector W_d. The size of this vector will be denoted by $\sigma(d)$, hence

$$\sigma(d) := |W_d| = \sum_{k=0}^{d} n^k = \frac{n^{d+1} - 1}{n - 1}. \tag{1}$$

Every $f \in \mathbb{R}\langle \underline{X} \rangle_{2d}$ can be written (possible nonuniquely) as $f = W_d^* G_f W_d$, where $G_f \in \mathbb{S}_{\sigma(d)}$ is called a *Gram matrix* for f.

Example 2.1. Consider $f = 2 + XYXY + YXYX \in \operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle$. Let

$$W_2 = \begin{bmatrix} 1 & X & Y & X^2 & XY & YX & Y^2 \end{bmatrix}^t.$$

Then there are many $G_f \in \mathbb{S}_7$ satisfying $f = W_2^* G_f W_2$; for instance

$$G_f(u, v) = \begin{cases} 1 & \text{if } u^* v = XYXY \ \vee \ u^* v = YXYX \ \vee \ u^* v = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Obviously $f \notin \Sigma^2$ but we have

$$f = (XY + YX)^*(XY + YX) + (1 - X^2) + Y(1 - X^2)Y + (1 - Y^2) + X(1 - Y^2)X. \tag{2}$$

If $A, B \in \mathbb{S}_k$, then

$$f(A, B) = 2I_k + ABAB + BABA.$$

For example, given

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix},$$

we have

$$f(A, B) = \begin{bmatrix} -6 & 0 \\ 0 & -6 \end{bmatrix}.$$

We point out that (2) certifies positivity of f on the nc polydisc, i.e., if A and B are symmetric contractions (i.e., they satisfy $\|A\|, \|B\| \leq 1$, or equivalently, $I - A^2, I - B^2 \succeq 0$), then

$$f(A, B) = (AB + BA)^*(AB + AB) + (I - A^2) + B(I - A^2)B + (I - B^2) + A(I - B^2)A \succeq 0.$$

Hence for all symmetric contractions A, B, we have $2I + ABAB + BABA \succeq 0$. In fact, as one of the main goals of this note we shall explain that the same type of positivity certificate holds for every (polynomial) matrix inequality which holds irrespective of the size of the matrices; see §4 for details.

3. Positivity of nc polynomials and quadratic modules

In this section we extend the notion of positivity introduced in the previous section.

Definition 3.1. Fix a subset $S \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle$. The n-tuples of symmetric matrices $\underline{A} \in \mathbb{S}^n$ satisfying $s(\underline{A}) \succeq 0$ for all $s \in S$ we denote by $\mathscr{D}_S$. When considering symmetric matrices of a fixed size $k \in \mathbb{N}$, we shall use $\mathscr{D}_S(k) := \mathscr{D}_S \cap \mathbb{S}_k^n$. We note that $\mathscr{D}_S$ is usually called an *nc semialgebraic set*.

Definition 3.2. Given a subset $S \subseteq \operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle$, we introduce

$$\begin{aligned}
\Sigma_S^2 &:= \Big\{ \sum_i h_i^* s_i h_i \mid h_i \in \mathbb{R}\langle \underline{X} \rangle,\, s_i \in S \Big\}, \\
\Sigma_{S,d}^2 &:= \Big\{ \sum_i h_i^* s_i h_i \mid h_i \in \mathbb{R}\langle \underline{X} \rangle,\, s_i \in S,\, \deg(h_i^* s h_i) \leq 2d \Big\}, \\
M_S &:= \Big\{ \sum_{i=1}^N a_i^* s_i a_i \mid N \in \mathbb{N},\, s_i \in S \cup \{1\},\, a_i \in \mathbb{R}\langle \underline{X} \rangle \Big\}, \\
M_{S,d} &:= \Big\{ \sum_i h_i^* s_i h_i \mid h_i \in \mathbb{R}\langle \underline{X} \rangle,\, s_i \in S \cup \{1\},\, \deg(h_i^* s h_i) \leq 2d \Big\},
\end{aligned} \tag{3}$$

and call M_S and $M_{S,d}$ the *quadratic module* and *truncated quadratic module* generated by S, respectively. Note $M_{S,d} = \Sigma_d^2 + \Sigma_{S,d}^2 \subseteq \mathbb{R}\langle \underline{X} \rangle_{2d}$, where $\Sigma_d^2 := M_{\varnothing,d}$ denotes the set of all sums of hermitian squares of polynomials of degree at most d. Observe that $M_{S,d}$ is a convex cone in the $\mathbb{R}$-vector space $\operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle_{2d}$.

Example 3.3.

(a) If $S = \{1\}$ or $S = \varnothing$, then $M_S = \Sigma_S^2 = \Sigma^2$ is exactly the cone containing nc polynomials which allow sum of hermitian squares (sohs) decompositions. Furthermore, $\mathscr{D}_S = \mathbb{S}^n$ is the set of all n-tuples of symmetric matrices. Clearly, $f \in M_S$ implies $f|_{\mathscr{D}_S} \succeq 0$.

(b) If $S = \mathbb{D} := \{1 - X_1^2, \ldots, 1 - X_n^2\}$ then $M_{\mathbb{D},d}$ contains exactly the polynomials f which have a *sohs decomposition over the polydisc*, i.e., can be written as

$$f = \sum_i g_i^* g_i + \sum_{j=1}^n \sum_i h_{i,j}^* \Big(1 - X_j^2\Big) h_{i,j}, \tag{4}$$

where $\deg(g_i) \leq d, \quad \deg(h_{i,j}) \leq d - 1$ for all i, j. We call $\mathscr{D}_{\mathbb{D}}$ the *nc polydisc*. It consists of all n-tuples $\underline{A}$ of symmetric contractions, i.e., matrices $A_j \in \mathbb{S}$ satisfy $\|A_j\| \leq 1$, or equivalently, $I - A_j^2 \succeq 0$.

(c) If $S = \mathbb{B} := \{1-\sum_i X_i^2\}$ then $M_{\mathbb{B},d}$ contains exactly the polynomials f which have a *sohs decomposition over the ball*, i.e., can be written as

$$f = \sum_i g_i^* g_i + \sum_i h_i^* \Big(1 - \sum_j X_j^2\Big) h_i, \tag{5}$$

where $\deg(g_i) \leq d, \quad \deg(h_i) \leq d-1$ for all i. We call $\mathscr{D}_{\mathbb{B}}$ the *nc ball*. It consists of all n-tuples $\underline{A} \in \mathbb{S}^n$ of row contractions, i.e., matrices $A_j \in \mathbb{S}$ satisfy $\big\|\begin{bmatrix} A_1 & \cdots & A_n \end{bmatrix}\big\| \leq 1$, or equivalently, $I - \sum_j A_j^2 \succeq 0$.

We also call a decomposition of the form (4) or (5) a *sohs decomposition with weights*.

Example 3.4. Note the the polynomial f from Example 2.1 has a sohs decomposition (2) over the polydisc.

Example 3.5. Let $f = 2 - X^2 + XY^2X - Y^2 \in \operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle$. Obviously $f \notin \Sigma^2$ but

$$f = 1 + (YX)^* YX + (1 - X^2 - Y^2) \tag{6}$$

is its sohs decomposition over the ball.

We conclude this section with an obvious but important observation:

Proposition 3.6. *Let $S \subseteq \operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle$. If $f \in M_S$, then $f|_{\mathscr{D}_S} \succeq 0$.*

The converse of Proposition 3.6 is false in general, i.e., positivity on an nc semialgebraic set does not imply the existence of a weighted sum of squares certificate, cf. [KS07, Example 3.1]. However, as we shall see below, the converse *does* hold for $S \in \{\varnothing, \mathbb{B}, \mathbb{D}\}$.

4. Separation for quadratic modules and optimization of nc polynomials

In this section we present the core theoretical aspects of the positivity certificates for nc polynomials. The main technical tool employed are linear functionals or, via duality, Hankel matrices, which are the subject of §4.1. In §4.2, §4.3 and §4.4 these technique is then applied to study optimization of nc polynomials.

4.1. Hankel matrices.

Definition 4.1. To each linear functional $L : \mathbb{R}\langle \underline{X}\rangle_{2d} \to \mathbb{R}$ we associate a matrix H_L (called an *nc Hankel matrix*) indexed by words $u, v \in \langle \underline{X}\rangle_d$, with

$$(H_L)_{u,v} = L(u^* v). \tag{7}$$

If L is *positive*, i.e., $L(p^*p) \geq 0$ for all $p \in \mathbb{R}\langle \underline{X}\rangle_d$, then $H_L \succeq 0$. Given $g \in \operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle$, we associate to L the *localizing matrix* $H_{L,g}^{\text{shift}}$ indexed by words $u, v \in \langle \underline{X}\rangle_{d-\deg(g)/2}$ with

$$(H_{L,g}^{\text{shift}})_{u,v} = L(u^* g v). \tag{8}$$

If $L(h^* g h) \geq 0$ for all h with $h^* g h \in \mathbb{R}\langle \underline{X}\rangle_{2d}$ then $H_{L,g}^{\text{shift}} \succeq 0$.

We say that L is *unital* if $L(1) = 1$.

Remark 4.2. Note that a matrix H indexed by words of length $\leq d$ satisfying the *nc Hankel condition* $H_{u_1,v_1} = H_{u_2,v_2}$ whenever $u_1^* v_1 = u_2^* v_2$, gives rise to a linear functional L on $\mathbb{R}\langle \underline{X}\rangle_{2d}$ as in (7). If $H \succeq 0$, then L is positive.

Definition 4.3. Let $A \in \mathbb{R}^{s\times s}$ be a symmetric matrix. A (symmetric) extension of A is a symmetric matrix $\tilde{A} \in \mathbb{R}^{(s+\ell)\times(s+\ell)}$ of the form

$$\tilde{A} = \begin{bmatrix} A & B \\ B^t & C \end{bmatrix}$$

for some $B \in \mathbb{R}^{s\times \ell}$ and $C \in \mathbb{R}^{\ell\times\ell}$. Such an extension is *flat* if $\operatorname{rank} A = \operatorname{rank} \tilde{A}$, or, equivalently, if $B = AZ$ and $C = Z^t AZ$ for some matrix Z.

For later reference we record the following easy linear algebra fact.

Lemma 4.4. $\begin{bmatrix} A & B \\ B^t & C \end{bmatrix} \succeq 0$ *if and only if $A \succeq 0$, and there is some Z with $B = AZ$ and $C \succeq Z^t AZ$.*

Suppose $L : \mathbb{R}\langle \underline{X}\rangle_{2d+2} \to \mathbb{R}$ is a linear functional and let $\check{L} : \mathbb{R}\langle \underline{X}\rangle_{2d} \to \mathbb{R}$ denote its restriction. As in Definition 4.1 we associate to L and $\check{L}$ the Hankel matrices H_L and $H_{\check{L}}$, respectively. In block form,

$$H_L = \begin{bmatrix} H_{\check{L}} & B \\ B^t & C \end{bmatrix}. \tag{9}$$

If H_L is flat over $H_{\check{L}}$, we call L (1-step) *flat*.

The following technical proposition is a variant of a Powers-Scheiderer result [PS01, §2].

Proposition 4.5. *If $S \in \{\varnothing, \mathbb{B}, \mathbb{D}\}$, then $M_{S,d}$ is a closed convex cone in the real vector space* $\operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle_{2d}$.

4.2. **Separation and optimization of nc polynomials via SDP.** In this section we explain how SDPs relate to eigenvalue optimization of an nc polynomial. The following theorem encompasses the main results about separation for cones $M_{\varnothing,d}$, $M_{\mathbb{B},d}$ and $M_{\mathbb{D},d}$. Proofs can be found in [KP10, HKM12, CKP12].

Theorem 4.6. *Let $f \in \mathbb{R}\langle \underline{X}\rangle_{2d}$ and let S be one of $\varnothing$, $\mathbb{B}$, $\mathbb{D}$. Then*

$$f|_{\mathscr{D}_S} \succeq 0 \quad \iff \quad f \in M_{S,d+1}. \tag{10}$$

Separation problems for the cones $M_{S,d}$ where $S = \varnothing$, $S = \mathbb{B}$ or $S = \mathbb{D}$ are instances of semidefinite programming feasibility problems. Indeed, $f \in M_{\varnothing,d}$ if and only if there exists a positive semidefinite matrix G of size $\sigma(d)$ such that $f = W_d^* G W_d$. For the proof see [KP10] or the references therein. Similar results hold for $M_{\mathbb{B},d}$ and $M_{\mathbb{B},d}$, as is explained in the following subsections.

Let $S \subseteq \operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle$ be finite and let $f \in \operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle_{2d}$. We are interested in the smallest eigenvalue $f_\star \in \mathbb{R}$ the polynomial f can attain on $\mathscr{D}_S$, i.e.,

$$f_\star := \inf\Big\{ \langle f(\underline{A})\xi, \xi \rangle \mid \underline{A} \in \mathscr{D}_S, \xi \text{ a unit vector}\Big\}. \tag{11}$$

Hence $f_\star$ is the greatest lower bound on the eigenvalues of $f(\underline{A})$ for tuples of symmetric matrices $\underline{A} \in \mathscr{D}_S$, i.e., $(f - f_\star)(\underline{A}) \succeq 0$ for all $\underline{A} \in \mathscr{D}_S$, and $f_\star$ is the largest real number with this property. From Proposition 3.6 it follows that we can bound $f_\star$ from below as follows

$$\begin{array}{rcll} f_\star \ \geq \ f^{(s)}_{\text{sohs}} & := & \sup\ \lambda \\ \text{s. t.} & & f - \lambda \in M_{S,s}, \end{array} \tag{SPSDP$_{\text{eig}-\min}$}$$

for $s \geq d$. For each *fixed* s this is an SDP and leads to the noncommutative version of the Lasserre relaxation scheme, cf. [PNA10]. However, as a consequence of the Nichtnegativstellensatz 4.6, if $S = \varnothing$, $\mathbb{D}$ or $\mathbb{B}$ then we do *not* need *sequences* of SDPs, a *single* SDP suffices: the first step in the noncommutative SDP hierarchy is already exact, see [CKP11, CKP12].

How to compute $f_\star$ when $S = \varnothing$ is explained in [CKP11, Section 2.3]. In the following subsections we present the eigenvalue optimization for the case of the nc ball and polydisc.

4.3. **Optimization of nc polynomials over the ball.** In this subsection we consider $S = \big\{1 - \sum_{i=1}^n X_i^2\big\}$ and the corresponding nc semialgebraic set $\mathbb{B} = \mathscr{D}_S$, the so-called nc ball. From Theorem 4.6 it follows that we can rephrase $f_\star$, the greatest lower bound on the eigenvalues of $f \in \mathbb{R}\langle \underline{X}\rangle_{2d}$ over the ball $\mathbb{B}$, as follows:

$$\begin{array}{rcll} f_\star \ = \ f_{\text{sohs}} & = & \sup\ \lambda \\ \text{s. t.} & & f - \lambda \in M_{\mathbb{B},d+1}. \end{array} \tag{PSDP$_{\text{eig}-\min}$}$$

Remark 4.7. We note that $f_\star > -\infty$ since positive semidefiniteness of a polynomial $f \in \mathbb{R}\langle \underline{X}\rangle_{2d}$ on $\mathbb{B}$ only needs to be tested on the compact set $\mathbb{B}(N)$ for some $N \geq \sigma(d)$ [HM04].

Verifying whether $f \in M_{\mathbb{B},d}$ is a semidefinite programming feasibility problem:

Proposition 4.8 ([CKP12, Proposition 4.2]). *Let $f = \sum_{w \in \langle \underline{X} \rangle_{2d}} f_w w$. Then $f \in M_{\mathbb{B},d}$ if and only there exist positive semidefinite matrices H and G of order $\sigma(d)$ and $\sigma(d-1)$, respectively, such that for all $w \in \langle \underline{X} \rangle_{2d}$,*

$$f_w = \sum_{\substack{u,v \in \langle \underline{X} \rangle_d \\ u^* v = w}} H(u,v) + \sum_{\substack{u,v \in \langle \underline{X} \rangle_{d-1} \\ u^* v = w}} G(u,v) - \sum_{j=1}^{n} \sum_{\substack{u,v \in \langle \underline{X} \rangle_{d-1} \\ u^* X_j^2 v = w}} G(u,v). \tag{12}$$

Remark 4.9. Proposition 4.8 (or its proof) explains how to construct the sohs decomposition with weights for $f \in M_{\mathbb{B},d}$. First we solve semidefinite feasibility problem in the variables $H \in \mathbb{S}^{\succeq 0}_{\sigma(d)}$, $G \in \mathbb{S}^{\succeq 0}_{\sigma(d-1)}$ subject to constraints (12). Then we compute by Cholesky or eigenvalue decomposition vectors $H_i \in \mathbb{R}^{\sigma(d)}$ and $G_i \in \mathbb{R}^{\sigma(d-1)}$ such that $H = \sum_i H_i H_i^t$ and $G = \sum_i G_i G_i^t$. Polynomials h_i and g_i from (5) are then computed as $h_i = H_i^t W_d$ and $g_i = G_i^t W_{d-1}$.

By Proposition 4.8 and Remark 4.9, the problem ($\text{PSDP}_{\text{eig-min}}$) is an SDP; it can be reformulated as

$$\begin{array}{rcl} f_{\text{sohs}} & = & \sup\ f_1 - \langle E_{1,1}, H \rangle - \langle E_{1,1}, G \rangle \\ \text{s.t.} & & f_w = \displaystyle\sum_{\substack{u,v \in \langle \underline{X} \rangle_{d+1} \\ u^* v = w}} H(u,v) + \sum_{\substack{u,v \in \langle \underline{X} \rangle_{d} \\ u^* v = w}} G(u,v) - \sum_{j=1}^{n} \sum_{\substack{u,v \in \langle \underline{X} \rangle_{d} \\ u^* X_j^2 v = w}} G(u,v), \\ & & \quad \text{for all } 1 \neq w \in \langle \underline{X} \rangle_{2d+2}, \\ & & H \in \mathbb{S}^{\succeq 0}_{\sigma(d+1)},\ G \in \mathbb{S}^{\succeq 0}_{\sigma(d)}. \end{array} \tag{PSDP'$_{\text{eig-min}}$}$$

The dual semidefinite program to ($\text{PSDP}_{\text{eig-min}}$) and ($\text{PSDP'}_{\text{eig-min}}$) is:

$$\begin{array}{rcl} L_{\text{sohs}} & = & \inf L(f) \\ \text{s.t.} & & L : \operatorname{Sym} \mathbb{R}\langle \underline{X} \rangle_{2d+2} \to \mathbb{R} \quad \text{is linear} \\ & & L(1) = 1 \\ & & L(q^* q) \geq 0 \quad \text{for all } q \in \mathbb{R}\langle \underline{X} \rangle_{d+1} \\ & & L(h^*(1 - \sum_j X_j^2) h) \geq 0 \quad \text{for all } h \in \mathbb{R}\langle \underline{X} \rangle_d. \end{array} \tag{DSDP$_{\text{eig-min}})_{d+1}$}$$

Proposition 4.10 ([CKP12]). *($\text{DSDP}_{\text{eig-min}})_{d+1}$ admits Slater points.*

Remark 4.11. Having Slater points for ($\text{DSDP}_{\text{eig-min}})_{d+1}$ is important for the clean duality theory of SDP to kick in [VB96, dK02]. In particular, there is no duality gap, so $L_{\text{sohs}} = f_{\text{sohs}} (= f_\star)$. Since also the optimal value $f_{\text{sohs}} > -\infty$ (cf. Remark 4.7), f_{sohs} is attained. Furthermore, L_{sohs} is attained. This is important as it enables us to extract optimizers $\underline{A}, \xi$ for $f_\star$ (see [CKP12, §5] for details).

4.4. **Optimization of NC polynomials over the polydisc.** In this section we consider $S = \{1 - X_1^2, \ldots, 1 - X_n^2\}$ and the corresponding nc semialgebraic set

$$\mathbb{D} = \mathscr{D}_S = \bigcup_{k \in \mathbb{N}} \{\underline{A} = (A_1, \ldots, A_n) \in \mathbb{S}_k^n \mid 1 - A_1^2 \succeq 0, \ldots, 1 - A_n^2 \succeq 0\},$$

the so-called nc polydisc.

The truncated quadratic module tailored for this S is

$$M_{\mathbb{D},d} = \Big\{ \sum_i h_i^* s_i h_i \mid h_i \in \mathbb{R}\langle \underline{X} \rangle,\ s_i \in S \cup \{1\},\ \deg(h_i^* s_i h_i) \leq 2d \Big\}.$$

Consider

$$\begin{array}{rcl} f_\star = f_{\text{sohs}} & = & \sup\ \lambda \\ \text{s.t.} & & f - \lambda \in M_{\mathbb{D},d+1}. \end{array} \tag{PSDP$_{\text{eig-min}}$}$$

Remark 4.12. As in Remark 4.7, $f_\star > -\infty$ since positive semidefiniteness of a polynomial $f \in \mathbb{R}\langle \underline{X} \rangle_{2d}$ on $\mathbb{D}$ only needs to be tested on the compact set $\mathbb{D}(N)$ for $N \geq \sigma(d)$ [HM04].

Theorem 4.6 implies that the problem ($\text{PSDP}_{\text{eig-min}}$) yields also the greatest lower bound on the eigenvalues of an nc polynomial f over the polydisc. Furthermore, verifying whether $f \in M_{\mathbb{D},d}$ is a semidefinite programming feasibility problem [CKP12, Proposition 4.6].

Proposition 4.13. *Let $f = \sum_{w\in\langle \underline{X}\rangle_{2d}} f_w w$. Then $f \in M_{\mathbb{D},d}$ if and only there exists a positive semidefinite matrix H of order $\sigma(d)$, and positive semidefinite matrices G_i, $1 \le i \le n$ of order $\sigma(d-1)$ such that*

$$f_w = \sum_{\substack{u,v\in\langle \underline{X}\rangle_d \\ u^*v=w}} H(u,v) + \sum_i \sum_{\substack{u,v\in\langle \underline{X}\rangle_{d-1} \\ u^*v=w}} G_i(u,v) - \sum_{i=1}^{n} \sum_{\substack{u,v\in\langle \underline{X}\rangle_{d-1} \\ u^*X_i^2 v=w}} G_i(u,v), \quad \text{for all } w \in \langle \underline{X}\rangle_{2d}. \tag{13}$$

Remark 4.14. Proposition 4.13 (or its proof, cf. [CKP12, Proposition 4.6]) explains how to construct the sohs decomposition with weights for $f \in M_{\mathbb{D},d}$. First we solve semidefinite feasibility problem in the variables $H \in \mathbb{S}^{\succeq 0}_{\sigma(d)}$, $G_j \in \mathbb{S}^{\succeq 0}_{\sigma(d-1)}$ subject to constraints (13). Then we compute by Cholesky or eigenvalue decomposition vectors $H_i \in \mathbb{R}^{\sigma(d)}$ and $G_{i,j} \in \mathbb{R}^{\sigma(d-1)}$ such that $H = \sum_i H_i H_i^t$ and $G_j = \sum_i G_{i,j} G_{i,j}^t$. Defining polynomials h_i and $g_{i,j}$ as $h_i = H_i^t W_d$ and $g_{i,j} = G_{i,j}^t W_{d-1}$ yields (4).

By Proposition 4.13, the problem of computing $f_\star$ over the polydisc is an SDP. Its dual SDP is:

$$\begin{array}{rcll} L_{\text{sohs}} & = & \inf L(f) & \\ \text{s. t.} & & L : \operatorname{Sym}\mathbb{R}\langle \underline{X}\rangle_{2d+2} \to \mathbb{R} \quad \text{is linear} & \\ & & L(1) = 1 & \\ & & L(q^*q) \ge 0 \quad \text{for all } q \in \mathbb{R}\langle \underline{X}\rangle_{d+1} & \\ & & L(h^*(1-X_j^2)h) \ge 0 \quad \text{for all } h \in \mathbb{R}\langle \underline{X}\rangle_d,\ 1 \le j \le n. & \end{array} \tag{$\text{DSDP}_{\text{eig-min}})_{d+1}$$

For implementational purposes, problem $(\text{DSDP}_{\text{eig-min}})_{d+1}$ is more conveniently given as

$$\begin{array}{rcll} L_{\text{sohs}} & = & \inf \langle H_L, G_f \rangle & \\ \text{s. t.} & & H_L(u,v) = H_L(w,z), \text{ if } u^*v = w^*z, \text{ where } u,v,w,z \in \langle \underline{X}\rangle_{d+1} & \\ & & H_L(1,1) = 1,\ H_L \in \mathbb{S}^{\succeq 0}_{\sigma(d+1)},\ H_L^j \in \mathbb{S}^{\succeq 0}_{\sigma(d)},\ \forall j & \\ & & H_L^j(u,v) = H_L(u,v) - H_L(X_j u, X_j v), \text{ for all } u,v \in \langle \underline{X}\rangle_d,\ 1 \le j \le n & \end{array} \tag{$\text{DSDP'}_{\text{eig-min}})_{d+1}$$

where G_f is a Gram matrix for f, and H_L^j represents L acting on nc polynomials of the form $u^*(1-X_j^2)v$, i.e., H_L^j is what we call a localizing matrix for $1 - X_j^2$.

Proposition 4.15 ([CKP12]). *$(\text{DSDP}_{\text{eig-min}})_{d+1}$ admits Slater points.*

As before, having Slater points for $(\text{DSDP}_{\text{eig-min}})_{d+1}$ is important from both practical and theoretical point of view, cf. Remark 4.11 above.

5. Computational complexity

In this section we present the complexity of: (i) detecting if f is a member of $M_{S,d}$ for $S = \varnothing$, $\mathbb{B}$ or $\mathbb{D}$; (ii) finding the minimum eigenvalue of f.

5.1. Complexity of separation for $M_{S,d}$.

Verifying if f is positive on $\mathbb{S}^n$, i.e., testing membership of f in $M_{\varnothing,d}$ amounts to solving an instance of a semidefinite programming problem, as mentioned in the paragraph after Theorem 4.6 (or see [CKP11, §2.2]). This problem is a SDP feasibility problem with matrix variable of order $\sigma(d)$, see (1). If for example $n = 4$ and $d = 5$ then $\sigma(d) = 1365$ which is already on the feasibility edge for all general SDP solvers. We point out that these problems also have lots of constraints, i.e., the number of constraints m is exactly

$$m = \operatorname{card}\{w \in W_{2d} \mid w^* = w\} + \frac{1}{2}\operatorname{card}\{w \in W_{2d} \mid w^* \ne w\}.$$

Since W_d contains all words in $\langle \underline{X} \rangle$ of degree $\leq d$, we have

$$\frac{n^{2d+1}-1}{2(n-1)} = \frac{1}{2}\sigma(2d) < m < \sigma(2d) = \frac{n^{2d+1}-1}{n-1},$$

so $m \approx n^{2d}$. Fortunately, detecting if $f \in M_{\varnothing,d}$ can be done much faster. The Newton chip method presented in [KP10] constructs a basis vector W which can replace W_d and has length at most kd, where k is the number of hermitian squares appearing in f.

Unfortunately, for the other two separation instances (that is, testing membership $f \in M_{\mathbb{D},d}$ or $f \in M_{\mathbb{B},d}$) we have to solve the semidefinite programs in full dimension. Finding sohs decomposition with weights is for the case of the nc ball an SDP feasibility problem with one matrix variable of order $\sigma(d)$ and one of order $\sigma(d-1)$. For the case of the nc polydisc it is even more demanding: the corresponding SDP feasibility problem (13) consists of one matrix variable of order $\sigma(d)$ and n matrix variables of order $\sigma(d-1)$.

The number of linear constraints is equal for both problems: $m = \text{card}\{w \in W_{2d} \mid w^* = w\} + \frac{1}{2}\text{card}\{w \in W_{2d} \mid w^* \neq w\}$. The only meaningful reduction of the size of matrix variables and the number of constraints for the case of the nc polydisc and ball we are aware of is the following: if there is a word w of degree d such that w^*w does not appear in f, then w can be eliminated from W_d.

5.2. **Eigenvalue optimization of NC polynomials.** Finding the minimum eigenvalue $f_\star$ and the corresponding minimizer of f over the set of all n-tuple of symmetric matrices, i.e., solving (11) for the case $S = \varnothing$ amounts to solving an instance of a semidefinite programming problem, as mentioned in the paragraph after Theorem 4.6 and in [CKP11, Subsection 2.3.1]. This SDP in primal form has one matrix variable of order $\sigma(d)$ with the number of constraints being $\text{card}\{w \in W_{2d} \mid w^* = w\} + \frac{1}{2}\text{card}\{w \in W_{2d} \mid w^* \neq w\} - 1$. Unfortunately – for extracting optimizers – we cannot employ the Newton chip method for these problems, so this is the real complexity that can not be reduced.

The complexity of solving the semidefinite program in primal form ($\text{PSDP}_{\text{eig}-\min}$) underlying $f_\star$ for the case of $\mathbb{B}$ is determined by the size of two matrix variables: the first variable is of order $\sigma(d+1)$ while the second is of order $\sigma(d)$ (recall that we are actually operating above $M_{\mathbb{B},d+1}$. The number of linear constraints in this semidefinite program is $\text{card}\{w \in W_{2d+2} \mid w^* = w\} + \frac{1}{2}\text{card}\{w \in W_{2d+2} \mid w^* \neq w\} - 1$. When we compute $f_\star$ for the case $\mathbb{D}$ we obtain semidefinite programming problem with $n+1$ matrix variables, one of order $\sigma(d+1)$ and n variables of order $\sigma(d)$. The number of linear constraints is the same as for $\mathbb{B}$..

Semidefinite programming problems are tractable problems, see Appendix for details about the theoretical complexity for semidefinite programming. As mentioned there we can efficiently deal with semidefinite programs if the matrix variables are of order up to 1 000 and if there is not more than 10 000 linear constraints. This implies that in the case of separation over $M_{\varnothing,d}$ we can find sohs decompositions for polynomials with many variables and with reasonable degree, i.e., with more than 10 variables and with degree higher than 10, if the number of hermitian square is below 100. For the other problems listed in this paper we are currently very limited with the number of variables and with the degree, e.g. we can manage problems with degree d less that 5 and with less than 5 variables.

6. Concluding remarks

In this paper we have shown how to effectively detect if given noncommutative (nc) polynomial is sum of hermitian squares (with weights) and how to compute the smallest (or biggest eigenvalue) the nc polynomial can attain on the set of all n tuples of symmetric matrices or on the ball $\mathbb{B}$ or polydisc $\mathbb{D}$. Our algorithm is based on sums of hermitian squares and yields an exact solution with a *single* semidefinite program (SDP). To prove exactness, we investigated the solution of the dual SDP and used to compute the exact bound.

It is clear that the Nichtnegativstellensatz 4.6 works not only for S being one of $\varnothing$, $\mathbb{B}$, $\mathbb{D}$, but also for all nc semialgebraic sets obtained from these via invertible linear change of variables. What is less clear (and has been established after we have obtained Theorem 4.6), is that this result can be slightly strengthened. Namely, its conclusion holds for all *convex* nc semialgebraic sets (or, equivalently [HM12], nc *LMI domains* $\mathscr{D}_L$). However, this requires a different and more involved proof. For details we refer the reader to [HKM12].

APPENDIX A. SEMIDEFINITE PROGRAMMING (SDP)

Semidefinite programming (SDP) is a subfield of convex optimization concerned with the optimization of a linear objective function over the intersection of the cone of positive semidefinite matrices with an affine space [Nem07, BTN01, VB96]. Linear programming is included as a special case of semidefinite programming, or said differently, semidefinite programming is a generalization of linear programming. As opposed to linear programming, the feasible set of a semidefinite problem is not polyhedral and there is no method like the simplex algorithm that could be applied to solve semidefinite programs. The duality theory of linear programming carries over naturally to SDP, but becomes more profound. Weak duality always holds. Strong duality, however, does not hold in general but requires some sort of regularity of the feasible regions. The same requirements are needed to guarantee attainment of the optima. For a thorough discussion we refer to the handbook of semidefinite programming [WSV00].

The importance of semidefinite programming was spurred by the development of efficient methods which can find an ε-optimal solution in a polynomial time in s, m and $\log \varepsilon$, where s is the order of the matrix variables, and m is the number of linear constraints. The most prominent methods for solving semidefinite programs nowadays are *interior-point methods*. Several variants of this Newton-method based algorithm have been developed over the past few decades ([dK02, NT08]) and several open sources packages are available, e.g., SDPT3 [TTT], SeDuMi [Stu99], CSDP [Bor99]. Interior-point methods are capable of solving semidefinite programs of medium size (i.e., $s \leq 1000$ and $m \leq 10.000$). Solving large-scale programs (matrix dimension s being huge or having a vast number of linear constraints) is not practical using interior-point algorithms due to dense linear algebra operations.

Various applications lead to large-scale semidefinite programming relaxations. This advanced in the past few years the development of algorithms for solving larger semidefinite programs. Several ideas to get rid of the semidefiniteness constraint have been investigated and some variants of first order methods have been developed. The *spectral bundle method* [HR00] transforms the SDP into an Eigenvalue optimization problem. This method can be applied to semidefinite programs for which the so-called constant trace property holds. Matrices of sizes around $s \approx 10000$ can be handled by this method. *Projection methods* rely on the fact that a symmetric matrix can be projected onto the cone of semidefinite matrices through a spectral decomposition. This has been done in different ways by several authors ([BV06, MPRW09, WGY10, JR08, ZST08]). Another idea for solving an SDP is by exploiting the factorization $X = RR^t$ that holds for any positive semidefinite matrix X. Methods from non-linear programming can then be used to solve the optimization problem in R. However, note that the resulting problem is non-convex. *SDPLR* [BM03] is an implementation of this low-rank method.

For a comprehensive list of state of the art SDP solvers see [Mit03]. However, in contrast to linear programming where algorithms are easily accessible even for non-experts, using algorithms for solving SDP still requires insight into the theory of semidefinite programming. There are no standard methods for solving an SDP in a routine way and one has to carefully choose the right tool depending on the characteristics of the semidefinite programs to be solved.

REFERENCES

[BM03] S. Burer and R.D.C. Monteiro (2003), "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization", *Mathematical Programming* (Series B), 95, 329–357.

[Bor99] B. Borchers (1999), "CSDP, a C library for semidefinite programming" *Optim. Methods Softw*., 11/12(1-4), 613–623.

[BTN01] A. Ben-Tal and A. Nemirovski (2001), *Lectures on modern convex optimization*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

[BV06] S. Burer and D. Vandenbussche (2006), “Solving lift-and-project relaxations of binary integer programs“, *SIAM Journal on Optimization*, 16(3), 726–750.

[CKP11] K. Cafuta, I. Klep, and J. Povh (2011), “NCSOStools: a computer algebra system for symbolic and numerical computation with noncommutative polynomials“, *Optimization Methods and Softw*are, 26(3), 363–380. http://ncsostools.fis.unm.si/.

[CKP12] K. Cafuta, I. Klep, and J. Povh (2012), „Constrained polynomial optimization problems with noncommuting variables. SIAM Journal of Optimization, 22(2):363–383.

[dK02] E. de Klerk (2002), *Aspects of semidefinite programming*, volume 65 of Applied Optimization, Kluwer Academic Publishers, Dordrecht.

[DLTW08] A.C. Doherty, Y.-C. Liang, B. Toner, and S. Wehner (2008), „The quantum moment problem and bounds on entangled multi-prover games“, In Twenty-Third Annual IEEE Conference on Computational Complexity, 199–210, IEEE Computer Soc., Los Alamitos, CA.

[Gla63] R.J. Glauber (1963), “The quantum theory of optical coherence“, *Physical Rev*iew, 130(6), 2529–2539.

[Hel02] J.W. Helton (2002), “Positive noncommutative polynomials are sums of squares“, Annals of Mathematics (2), 156(2), 675–694.

[HKM12] J.W. Helton, I. Klep, and S. McCullough (2012), “The convex positivstellensatz in a free algebra“, *Adv. Math*., 231(1), 516–534,

[HLL09] D. Henrion, J.-B. Lasserre, and J. Löfberg (2009), „GloptiPoly 3: moments, optimization and semidefinite programming“, *Optim. Methods Softw*., 24(4-5), 761–779. 2009.

[HM04] J.W. Helton and S.A. McCullough (2004), “A Positivstellensatz for non-commutative polynomials“, *Trans. Amer. Math. Soc*., 356(9), 3721–3737.

[HM12] J.W. Helton and S. McCullough (2012), „Every free basic convex semi-algebraic set has an LMI representation“, *Ann. of Math*. (2), 176(2), 979–1013.

[HMdOP08] J.W. Helton, S. McCullough, M.C. de Oliveira, and M. Putinar (2008), „Engineering systems and free semi-algebraic geometry“, In Emerging Applications of Algebraic Geometry, volume 149 of IMA Vol. Math. Appl., pages 17–62. Springer.

[HR00] C. Helmberg and F. Rendl (2000), “A spectral bundle method for semidefinite programming“, *SIAM Journal on Optimization*, 10(3), 673–696.

[JR08] F. Jarre and F. Rendl (2008), “An augmented primal-dual method for linear conic problems“, *SIAM Journal on Optimization*.

[KP10] I. Klep and J. Povh (2010), “Semidefinite programming and sums of hermitian squares of noncommutative polynomials“, *Journal of Pure and Applied Algebra*, 214, 740–749.

[KS07] I. Klep and M. Schweighofer (2007), “A nichtnegativstellensatz for polynomials in noncommuting variables“, Israel Jornal of Mathematics, 161, 17–27.

[Löf04] J. Löfberg (2004), “YALMIP: A toolbox for modeling and optimization in MATLAB“, In Proceedings of the CACSD Conference, Taipei, Taiwan.

[Maz04] D.A. Mazziotti (2004), “Realization of quantum chemistry without wave functions through first-order semidefinite programming“, *Physical Review Letters*, 93(21), 213001.

[Mit03] D. Mittelmann (2003), “An independent benchmarking of SDP and SOCP solvers“, Mathematical Programming, B, 95, 407–430.

[MPRW09] J. Malick, J. Povh, F. Rendl, and A. Wiegele (2009), "Regularization methods for semidefinite programming", *SIAM Journal of Optimization*, 20(1), 336–356.

[Nem07] A. Nemirovski (2007), "Advances in convex optimization: conic programming". In International Congress of Mathematicians. Vol. I, pages 413–444. Eur. Math. Soc., Zurich.

[NT08] A. S. Nemirovski and M. J. Todd (2008), "Interior-point methods for optimization", Acta Numer., 17, 191–234.

[PNA10] S. Pironio, M. Navascues, and A. Acın (2010), "Convergent relaxations of polynomial optimization problems with noncommuting variables", *SIAM Journal of Optimization*, 20(5), 2157–2180.

[PPSP05] S. Prajna, A. Papachristodoulou, P. Seiler, and P.A. Parrilo (2005), "SOSTOOLS and its control applications. In Positive polynomials in control, volume 312 of Lecture Notes in Control and Inform. Sci., 273–292, Springer, Berlin.

[PS01] V. Powers and C. Scheiderer (2001), "The moment problem for non-compact semialgebraic sets", Adv. Geom., 1(1), 71–88.

[PV09] K.F. Pal and T. Vertesi (2009), "Quantum bounds on Bell inequalities", *Phys. Rev*. A (3), 79(2), 022120, 12.

[Stu99] J.F. Sturm (1999), "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones", *Optiization Methods and Softw*are, 11/12(1-4), 625–653.

[TTT] K.-C. Toh, M.J. Todd, and R.-H. Tutuncu. "SDPT3 version 4.0 (beta) – aMATLAB software for semidefinite-quadratic-linear programming", http://www.math.nus.edu.sg/~mattohkc/sdpt3.html.

[VB96] L. Vandenberghe and S. Boyd (1996), "Semidefinite programming", *SIAM Rev*., 38(1), 49–95.

[WGY10] Z. Wen, D. Goldfarb, and W. Yin (2010), "Alternating direction augmented lagrangian methods for semidefinite programming", *Math. Prog. Comp*., 2, 203–230.

[WKK+09] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto (2009), "Algorithm 883: sparsePOP—a sparse semidefinite programming relaxation of polynomial optimization problems" *ACM Trans. Math. Software*, 35(2), Art. 15, 13.

[WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe (2000), *Handbook of Semidefinite Programming*, Kluwer.

[ZST08] X. Zhao, D. Sun, and K. Toh (2008), "A Newton-CG augmented Lagrangian method for semidefinite programming", Technical Report, National University of Singapore.