SOS is not obviously automatizable, even approximately*

Ryan O'Donnell[1]

1 **Computer Science Department, Carnegie Mellon University, Pittsburgh USA**
odonnell@cs.cmu.edu

Abstract

Suppose we want to minimize a polynomial $p(x) = p(x_1, \dots, x_n)$, subject to some polynomial constraints $q_1(x), \dots, q_m(x) \geq 0$, using the Sum-of-Squares (SOS) SDP hierarchy. Assume we are in the "explicitly bounded" ("Archimedean") case where the constraints include $x_i^2 \leq 1$ for all $1 \leq i \leq n$. It is often stated that the degree-d version of the SOS hierarchy can be solved, to high accuracy, in time $n^{O(d)}$. Indeed, I myself have stated this in several previous works.

The point of this note is to state (or remind the reader) that this is not obviously true. The difficulty comes not from the "r" in the Ellipsoid Algorithm, but from the "R"; a priori, we only know an exponential upper bound on the number of bits needed to write down the SOS solution. An explicit example is given of a degree-2 SOS program illustrating the difficulty.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Sum-of-Squares, semidefinite programming

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Suppose you want to approximately minimize a real polynomial $p(x) = p(x_1, \dots, x_n)$ over the set $K = \{x \in \mathbb{R}^n : q_1(x) \geq 0, \dots, q_m(x) \geq 0\}$, where $q_1, \dots, q_m$ are real polynomials. All of the examples I'll consider will be quite simple: m will be at most $O(n)$; and, the polynomials $p, q_1, \dots, q_m$ will be of degree at most 2 and will have small integer coefficients (magnitude at most $\mathrm{poly}(n)$, say; often at most 2). A good example to keep in mind arises from the "Balanced Separator" problem in combinatorial optimization. There, you're given an n-vertex graph $G = (V, E)$ and the goal is to partition its vertices into two parts, neither of size more than $\frac{2}{3}n$, such that the number of edges crossing between the parts is minimized. Introducing a variable x_i for each vertex, this is equivalent to solving

$$\min \sum_{\{i,j\} \in E} \tfrac{1}{4}(x_i - x_j)^2 \quad \text{subject to} \quad \{x_i^2 = 1 \ \forall i, \ -\tfrac{1}{3}n \leq x_1 + \cdots + x_n \leq \tfrac{1}{3}n\}.$$

Here $x_i^2 = 1$ can be treated as the two inequalities $1 - x_i^2 \geq 0$, $-1 + x_i^2 \geq 0$. Another good example arises from the "Maximum Independent Set" problem on G:

$$\max \sum_{i=1}^{n} x_i \quad \text{subject to} \quad \{x_i^2 = x_i \ \forall i, \ x_i x_j = 0 \ \forall \{i,j\} \in E\}.$$

* This work was partially supported by NSF grant CCF-1618679.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Acces; Article No. 23; pp. 23:1–23:10
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A powerful technique for trying to certify that the minimum is at least $\theta \in \mathbb{R}$ is to find a formal polynomial identity of the form

$$p(x) - \theta = u_0(x) + u_1(x)q_1(x) + \cdots + u_m(x)q_m(x), \tag{1}$$

where each $u_j(x)$ is SOS; i.e., a sum of squares of polynomials. We will refer to this as "SOS-proving" or "SOS-certifying" the statement "$p(x) \geq \theta$". A variation of this technique ("SOS-refutation") is to take $q_{m+1}(x) = (\theta - \epsilon) - p(x) \geq 0$ as an additional constraint, and then try to SOS-prove the statement "$-1 \geq 0$". It's easy to check that we can do this for every $\epsilon > 0$ provided that "$p(x) \geq \theta$" is SOS-provable. So if we aren't concerned with very small additive errors — and I won't be, in this note — the refutation technique is fundamentally stronger (see, e.g., [22] for further discussion). In any case, I'll use the "SOS-certifying" terminology in the rest of the note, since SOS-refutation is just a special case with one extra constraint.

Suppose we now bound the degree of the $u_j(x)$'s, insisting that $\deg(u_0), \deg(u_1q_1), \cdots \leq d$. Then the question of whether the certifying $u_j(x)$'s exist is equivalent to the feasibility of a certain semidefinite program (SDP). This is the "degree-d SOS relaxation", pioneered by Shor [28], Nesterov [20], Grigoriev and Vorobjov [9], Lasserre [16, 17] and Parrilo [23]. See, e.g., [18, 3] for many more details.

Under the simple assumptions I mentioned (namely, $m \leq O(n)$, p and q_j's having small coefficients and degree at most 2), the degree-d SOS SDP for (1) can be written down using $N = n^{O(d)}$ bits. It is then quite commonly stated that feasibility can be tested in $\mathrm{poly}(N)$ time, using, say the Ellipsoid Algorithm [15, 11]. This is sometimes referred to as the SOS proof system being "automatizable". Unfortunately, I will now explain why it's not clear whether this is truly the case.

1.0.0.1 Approximation, and the explicitly bounded case.

I should emphasize that I am *not* worried about very small additive errors; i.e., the difference between testing feasibility and near-feasibility. Indeed, most often the caveat is correctly added that semidefinite programming only tests feasibility up to a very small additive error. This caveat is related to the fact that the Ellipsoid Algorithm has a technical requirement, that if the SDP is feasible then it contains a feasible ball of some small radius $r = 2^{-\mathrm{poly}(N)} > 0$. Actually, to talk about additive error only makes sense if there is some notion of "scaling". To continue keeping things simple, I'll henceforth assume that the variables are intended to be in the range $[-1, 1]$; i.e., that K always includes the constraints $x_i^2 \leq 1$ for $1 \leq i \leq n$. (It would actually be fine if we even just had $x_i^2 \leq 2^{\mathrm{poly}(N)}$ for all i.) This is sometimes called the "explicitly bounded" or "Archimedean" case, and it's also known to imply that the SDP has no duality gap [13].

With this issue discussed, let's now again pose the question:

1.0.0.2 Question.

Suppose there is a degree-d SOS proof that $p(x) \geq \theta$ subject to constraints $x_1^2, \ldots, x_n^2 \leq 1$ and $q_1(x), \ldots, q_m(x) \geq 0$, of the form (1). Is there a $\mathrm{poly}(N)$-time algorithm (presumably, a version of the Ellipsoid Algorithm) that finds SOS polynomials $u_0(x), \ldots, u_m(x)$ certifying $p(x) \geq \theta - o_N(1)$?

In a joint work with Yuan Zhou [22, Footnote 2], I wrote that the answer is "yes".[1] However I now see that my reasoning was incomplete, and that the answer is unclear. In fact, I would now guess that the answer is probably "no". Although it's true that the technical "r" parameter in the Ellipsoid Algorithm does not cause real problems in the explicitly bounded case, there is another technical parameter, "R" — and it *does* seem to cause real problems. The Ellipsoid Algorithm is only guaranteed to work correctly in $\mathrm{poly}(N)$ time if the SDP's feasible region (should it exist) intersects a ball of radius $R = 2^{\mathrm{poly}(N)}$. In other words, algorithmically speaking it's not enough for an SOS proof to exist; we also need one to exist in which all the SOS polynomials can be written down with $\mathrm{poly}(N)$ bits. However, in the next section I'll show a simple, explicitly bounded example where an inequality is SOS-provable, but any approximate SOS proof requires integers of size roughly 2^{2^n}. This example is based on the well-known fact (attributed to J. Ramana in [1] and to Khachiyan in [25]) that there are SDPs with n variables and $O(n)$ constraints that are feasible, yet for which every feasible solution requires exponential bit-complexity.

In fact, as pointed out to me by Pablo Parrilo, *every* SDP-feasibility problem can be viewed as an SOS-feasibility problem modulo an ideal; thus, if we ignore the insistence on $x_i^2 \le 1$ constraints, the above **Question** is tantamount to simply asking if the Semidefinite Feasibility Problem (SDFP) is in P. This is a well-known open question; see [25, 24, 30]. The best current upper bound known is PSPACE, by reduction to the existential theory of the reals.[2]

2 SOS-provable, but only with huge coefficients

Let's say we have $2n$ indeterminates $x_1, x_2, \ldots, x_n, y_1, \ldots, y_n$, and the following constraints.

$$\begin{array}{rrrcr} 2x_1y_1 = y_1, & 2x_2y_2 = y_2, & 2x_3y_3 = y_3, & & 2x_ny_n = y_n, \\ x_1^2 = x_1, & x_2^2 = x_2, & x_3^2 = x_3, & \cdots & x_n^2 = x_n, \\ y_1^2 = y_2, & y_2^2 = y_3, & y_3^2 = y_4, & & y_n^2 = 0. \end{array} \tag{K}$$

(These should be read column-wise. Notice the very last constraint, $y_n^2 = 0$, breaks the pattern.)

At first, I won't include the constraints $x_i^2 \le 1$, $y_i^2 \le 1$; we'll analyze their inclusion later.

We wish to know whether

$$p_n(x, y) = x_1 + x_2 + x_3 + \cdots + x_n - 2y_1$$

is nonnegative subject to these constraints. It's easy for the human mathematician to see the answer is "yes", because "solving" the constraints shows that they are equivalent to

[1] Sorry for talking you into that footnote, Yuan.

[2] Even the special case of deciding whether a given rational multivariate polynomial is SOS is not known to be in P or even in NP. I do not know if the "R" problem is relevant here, but the "r" problem certainly is; according to Scheiderer [26] there are rational polynomials such as $x^4 + xy^3 + y^4 - 3x^2yz - 4xy^2z + 2x^2z^2 + xz^3 + yz^3 + z^4$ that are SOS but don't have a rational SOS representation. However in this work I am less concerned with this kind of example, because I would like to consider the "bounded case" and allow approximation.

$x_1, \dots, x_n \in \{0, 1\}$ and $y_1 = \cdots = y_n = 0$; hence the minimum of $p_n(x)$ is 0. However SOS algorithms do not first try to "solve" or "simplify" the constraints.[3] So we have to see what happens when SOS algorithms are run "generically" on this input.

For simplicity, let's consider the degree-2 SOS algorithm. In this case, whether we consider the constraints as equalities or two-inequalities amounts to the same thing: we get to multiply them by nonnegative reals. Thus the question becomes:

$$p_n(x, y) = \big[\text{degree-2 SOS}\big] \bmod (\mathrm{K})? \tag{2}$$

where the "mod" refers to adding linear multiples of the constraint equations — i.e., adding $a(2x_1y_1 - y_1) + b(x_1^2 - x_1) + c(y_1^2 - y_2) + d(2x_2y_2 - y_2) + \cdots$ for some real constants $a, b, c, d, \dots$. The answer to this question is *also* "yes":

$$p_n(x, y) = (x_1 - 2y_1)^2 + (x_2 - 4y_2)^2 + (x_3 - 16y_3)^2 + (x_4 - 256y_4)^2 + \cdots + (x_n - 2^{2^{n-1}} y_n)^2 \bmod (\mathrm{K}).$$

However, it turns out that *every* way of expressing $p_n(x, y)$ as in (2) has exponential-in-n bit-complexity. This shows that no matter how exactly we formulate the SOS problem as an SDP (e.g., whether we look for homogeneous or non-homogeneous sums of squares, whether we explicitly introduce variables $a, b, c, d, \dots$ to multiply against the constraints or instead work "mod the ideal", etc.), no generic polynomial-time SDP-solving algorithm will find a degree-2 SOS proof of $p_n(x, y) \geq 0$.[4]

Before proving this, two comments: First, this example and its proof are nothing more than a slight rearrangement of the standard example of a feasible SDP whose only feasible solutions are doubly exponential. I'm only putting an SOS spin on it. Second, this argument doesn't really give a negative example for the **Question** from Section 1, because it's conceivable that there is a degree-2 SOS proof with polynomial bit-complexity of "$p_n(x, y) \geq -\epsilon_n$", where $\epsilon_n = o_n(1)$. In Subsections 2.1, 2.2, I'll show that even this is impossible, even when the constraints $x_i^2 \leq 1$, $y_i^2 \leq 1$ are added.

So let's suppose we have an SOS representation of $p_n(x, y)$ as in (2):

$$x_1 + x_2 + \cdots + x_n - 2y_1 = \sum_j \ell_j(x, y)^2 \bmod (\mathrm{K}), \tag{3}$$

where the ℓ_j's denote linear polynomials. In fact, the ℓ_j's must be homogeneous of degree 1. The reason is that if we set all x_i's and y_i's to 0 in (3), the LHS becomes 0 and the RHS becomes the sum of the squares of the constant coefficients of the ℓ_j's. Hence all these constant coefficients must be 0.

Next, let us express each ℓ_j as $\sum_{k=1}^n \ell_{jk}$, where each ℓ_{jk} is of the form $a_{jk}x_k + b_{jk}y_k$. It would of course be incorrect to say that $\ell_j^2 = \sum_{k=1}^n \ell_{jk}^2$ — to neglect the cross-terms is the so-called "freshman's dream". Notice, though, that any nonzero cross-term contains a monomial of the form $x_k x_{k'}$, $x_k y_{k'}$, or $y_k y_{k'}$ ($k \neq k'$), and no such monomial appears on the left in (3). Furthermore, such monomials are not affected by the "mod (K)", and thus they

[3] Otherwise, the well-known SOS lower bounds for "Knapsack" [8] and "kXOR" [7, 27] would be invalid. In particular, applying a Gröbner basis algorithm to the constraints is not a good idea in general, since it has exponential complexity even for zero-dimensional ideals [12]. For example, the size of the Gröbner basis for the very simple "Max-Bisection" ideal, $\{x_1^2 = \cdots = x_{2n}^2 = 1, x_1 + \cdots + x_{2n} = 0\}$, is $\widetilde{\Theta}(2^n)$.

[4] Note that it doesn't matter whether we ask the algorithm to find a PSD matrix representing the SOS polynomial, or the actual sums of squares. Since Cholesky (LDL) decomposition can be done in polynomial time (see Section 4), if there were a rational PSD matrix of polynomial bit-complexity representing the SOS polynomial, we could extract from it an explicit rational sum-of-squares representation with polynomial bit-complexity.

must be canceled via cross-terms arising from other squares $\ell_{j'}^2$ in the sum. Hence following the "freshman's dream" in $\sum_j \ell_j^2$ actually gives the same identity in (3). In other words, we may assume without loss of generality that (3) is of the form

$$\sum_{i=1}^{n}\sum_{j}(a_{ij}x_i + b_{ij}y_i)^2 = \sum_{i=1}^{n}(A_i^2x_i^2 + 2M_ix_iy_i + B_i^2y_i^2), \tag{4}$$

where $A_i = \sqrt{\sum_j a_{ij}^2}$, $B_i = \sqrt{\sum_j b_{ij}^2}$, and $M_i = \sum_j a_{ij}b_{ij}$. Cauchy–Schwarz implies

$$|M_i| = \sum_j a_{ij}b_{ij} \le A_iB_i. \tag{5}$$

For (4) to equal the LHS of (3) mod (K), we'll need to use all of the equality constraints, thereby obtaining

$$\sum_{i=1}^{n}(A_i^2x_i + M_iy_i + B_i^2y_{i+1}),$$

with y_{n+1} denoting 0. Equating coefficients with LHS(3), we deduce

$$A_i = 1\ \forall i, \quad M_1 = -2, \quad M_{i+1} = -B_i^2\ \forall 1 < i < n.$$

Combining this with (5), we get $B_i \ge |M_i|$ for all i, and hence

$$B_1 \ge 2, \quad B_{i+1} \ge B_i^2\ \forall 1 < i < n.$$

Thus $B_n \ge 2^{2^{n-1}}$; i.e., the sum of the squares of the coefficients on y_n in any representation (3) is at least 2^{2^n}. So indeed any solution to (2) has exponential bit-complexity.

2.1 Even approximately

I'll now show that even getting a degree-2 SOS proof of $p_n(x,y) \ge -o_n(1)$ is impossible without exponential bit-complexity. So suppose we have

$$x_1 + x_2 + \cdots + x_n - 2y_1 + \epsilon = \sum_j \ell_j(x,y)^2 \bmod (\mathrm{K}), \tag{6}$$

where $\epsilon \le .01$, say. Now we can't deduce that the ℓ_j's are homogeneous, but the reasoning concerning the "freshman's dream" still holds. So the SOS part must be of the form

$$\sum_{i=1}^{n}\sum_{j}(a_{ij}x_i + b_{ij}y_i + c_{ij})^2 = \sum_{i=1}^{n}(A_i^2x_i^2 + 2M_ix_iy_i + B_i^2y_i^2 + 2U_ix_i + 2V_iy_i + C_i^2),$$

where we're now introducing the notation $U_i = \sum_j a_{ij}c_{ij}$, $V_i = \sum_j b_{ij}c_{ij}$, and $C_i = \sqrt{\sum_j c_{ij}^2}$. Cauchy–Schwarz still implies (5), and also

$$|U_i| \le A_iC_i, \quad |V_i| \le B_iC_i. \tag{7}$$

Again, equating coefficients and reducing mod (K) yields

$$\epsilon = \sum_i C_i^2, \quad A_i^2 + 2U_i = 1\ \forall i, \quad M_1 + 2V_1 = -2, \quad M_{i+1} + 2V_{i+1} = -B_i^2\ \forall 1 < i < n. \tag{8}$$

As $\epsilon \leq .01$, the first equation implies $C_i \leq .1$ for all i. Thus (7) implies $|U_i| \leq .1A_i$, $|V_i| \leq .1B_i$. Substituting these into the above yields the following:

$$A_i^2 - .2A_i \leq 1 \implies A_i \leq 1.2 \;\forall i; \qquad |M_1| \geq 2 - .2B_1; \qquad |M_{i+1}| \geq B_i^2 - .2B_{i+1} \;\forall 1 < i < n.$$

As we still have (5), the first inequality above yields $1.2B_i \geq |M_i|$ for all i. Combining this with the second and third inequalities above gives:

$$1.4B_1 \geq 2 \implies B_1 \geq 1.42;$$
$$1.2B_{i+1} \geq |M_{i+1}| \geq B_i^2 - .2B_{i+1} \implies 1.4B_{i+1} \geq B_i^2 \;\forall 1 < i < n.$$

Together, the above yield $B_n \geq 1.4(1.42/1.4)^{2^{n-1}}$, and we again see that exponential bit-complexity is required for a degree-2 SOS proof of $p_n(x,y) \geq -.01 \bmod (\mathrm{K})$. (Incidentally, this also rules out the possibility of "SOS-refuting" the statement $p_n(x,y) < -o_n(1)$ with degree 2.)

2.2 Even with the "Archimedean" constraints

Finally, it's easy to see that the conclusion doesn't change even if we add to (K) the additional constraints $x_i^2 \leq 1$ and $y_i^2 \leq 1$ for all i, making the domain "Archimedean" ("explicitly bounded"). We know these constraints are actually redundant, so still $p_n(x,y)$ has minimal value 0. As for the effect on degree-2 SOS proofs, the new constraints allow us to also add terms $D_i(1 - x_i^2)$ and $E_i(1 - y_i^2)$ on the RHS of (6) for nonnegative constants D_i, E_i. In turn, this changes (8) to

$$\epsilon = \sum_i (C_i^2 + D_i + E_i), \quad A_i^2 + 2U_i - D_i = 1 \;\forall i, \quad M_1 + 2V_1 = -2,$$
$$M_{i+1} + 2V_{i+1} = -B_i^2 + E_i \;\forall 1 < i < n.$$

The first constraint implies $D_i, E_i \leq .01$ for all i. Given $D_i \leq .01$, we can still deduce $A_i^2 - .2A_i \leq 1.01$, which still implies $A_i \leq 1.2$. The condition $E_i \leq .01$ changes $|M_{i+1}| \geq B_i^2 - .2B_{i+1}$ to $|M_{i+1}| \geq B_i^2 - .2B_{i+1} - .01$, and hence we only get $1.4B_{i+1} \geq B_i^2 - .01$ for all $1 < i < n$. But this is still enough to conclude B_n is doubly-exponential in n, as before. In summary:

▶ **Theorem 1.** *Subject to* (K) *and* $x_i^2 \leq 1$, $y_i^2 \leq 1$, *there is a degree-*2 *SOS proof that* $p_n(x,y) \geq 0$. *However any degree-*2 *SOS proof even of* $p_n(x,y) \geq -.01$ *requires bit-complexity* $\Theta(2^n)$.

As a further remark, in the "explicitly bounded" case it's known that there is no SDP duality gap. So instead of trying to use semidefinite programming to get an SOS proof of $p_n(x,y) \geq -o_n(1)$, we might try using it to find a "pseudoexpectation" $\widetilde{\mathbf{E}}[\cdot]$ that satisfies the constraints and minimizes $\widetilde{\mathbf{E}}[p_n(x,y)]$. (See [3] for more on this terminology.) In this dual case, there won't be any "R problem", but instead we'll get an "r problem". The Ellipsoid Algorithm might be used to produce an $\widetilde{\mathbf{E}}[\cdot]$ that satisfies all the constraints up to doubly-exponentially small tolerance; e.g., the *genuine* distribution $x_i \equiv 0$, $y_i \equiv 2^{-2^i}$ satisfies all constraints except for $y_n^2 = 0$, which it satisfies to doubly-exponentially small tolerance. As constructors of SDP hierarchy integrality gaps know, the step of massaging an almost-satisfying solution to an exactly-satisfying solution is often non-obvious and problem-specific.

3 Discussion

I think that Theorem 1 gives a particularly simple example of things going wrong. It's not too much different from, say, the SOS formulation of Maximum Independent Set. Undoubtedly there are generic extensions of the SOS method that will handle this one specific example. For example, the Gröbner basis technique will not have exponential complexity in this case, and will in fact lead to an efficient degree-2 SOS-proof of $p_n(x, y) \geq 0$. We also did not analyze how degree-*four* SOS behaves on this instance. But the point is that it's not so easy to think of generic SOS extensions that will always work in polynomial time. Nor is it easy to think of additional structural constraints on instances that may help, yet that are not too restrictive.

An obvious candidate for additional structure is the constraint that every variable is not just bounded in $[-1, 1]$ but *Boolean*. This is at least a common scenario in combinatorial optimization. One still has to be careful though. For example, there is a well-known trick for converting inequality constraints to equality constraints in SOS: replace $q_i(x) \geq 0$ with $q_i(x) = z^2$ where z is a new variable. However this new variable wouldn't be constrained to be Boolean.

I don't know whether constraining every variable to be Boolean will cause feasible SOS SDPs to always have solutions of polynomial bit-complexity. However in the next section I'll observe that if these are the *only* constraints, we are in good shape. Nevertheless, this seems to me a somewhat rare situation; e.g., it's not satisfied in the Balanced Separator or Independent Set examples. And as noted earlier, although you may succeed in SOS-proving $p(x) \geq \theta$ subject to $x_i^2 = 1\ \forall i$, it's always fundamentally better to try SOS-proving $-1 \geq 0$ subject to the extra constraint $p(x) \leq \theta - \epsilon$. But then you have an additional constraint-inequality in addition to Booleanness.

4 Automatizing SOS proofs subject only to Booleanness

Here I'll record the known observation that, in $n^{O(d)}$ time, we can approximately test if some $p(x)$ is a degree-d sum of squares, modulo $\{x_i^2 = 1,\ \forall i\}$. (To avoid an extra parameter, I'll assume the coefficients of $p(x)$ are rationals expressible with $n^{O(d)}$ bits.) Specifically, if indeed $p(x)$ is degree-d SOS, then the algorithm will find a degree-d SOS representation of $p(x) + \epsilon$ for some $0 \leq \epsilon \leq 2^{-N^{O(d)}}$. I emphasize that there is no mathematical innovation in this section; all the details herein are known.

The typical way to formulate this problem as an SDP is to consider real symmetric matrices X with rows and columns indexed by the $N = n^{O(d)}$ subsets $S \subseteq [n]$, $|S| \leq d/2$. Then $p(x)$ is degree-d SOS mod $\{x_i^2 = 1,\ \forall i\}$ if and only if

$$\exists X \succeq 0 \quad \text{such that} \quad \sum_{\substack{|S|,|T| \leq d/2 \\ S \Delta T = U}} X_{S,T} = p_U\ \forall\, U \subseteq [n],\ |U| \leq d, \tag{SDP}$$

where p_U denotes the coefficient of $p(x)$ on $\prod_{i \in U} x_i$. (We may assume without loss of generality that $p(x)$ is multilinear.) This SDP feasibility problem can be written down using $\mathrm{poly}(N)$ bits, and we want to argue it can be decided (approximately) in $\mathrm{poly}(N)$ time.

The key observation is that the $U = \emptyset$ constraint of (SDP) is precisely "$\mathrm{tr}(X) = p_\emptyset$". The bit-complexity of $p_\emptyset$ is $\mathrm{poly}(N)$, by assumption (in general, it's bounded by the input size). Thus any feasible PSD solution X has the sum of its eigenvalues at most $\mathrm{poly}(N)$, and hence squared Frobenius norm at most $\mathrm{poly}(N)$. This means we can take the "R" in the Ellipsoid

Algorithm to be poly(N), overcoming the main difficulty described in this note. (Note that we could still run in poly(N) time even if R were $2^{\text{poly}(N)}$.)

We now recall how to take care of the "r" for the Ellipsoid Algorithm. The linear constraints in (SDP) obviously preclude any feasible region from containing a ball of positive radius. So we relax the "$= p_U$" equality constraints to two-sided "$\in [p_U - \epsilon', p_U + \epsilon']$" inequalities, where $\epsilon' = 2^{-N^c}$ for some constant c. (Note that this preserves feasibility, and the bit-complexity of ϵ' is just poly(N).) Actually, we're still not done because of the additional symmetry requirement $X_{S,T} = X_{T,S}$, but we can take care of this as in the original paper by Grötschel, Lovász, and Schrijver [10] by not introducing variables for the below-diagonal elements of X, treating them implicitly. We can now take the Ellipsoid Algorithm's "r" parameter to be $2^{-\text{poly}(N)}$, as needed.

Finally, we have a poly(N)-time "strong separation oracle" for the relaxed form of (SDP). This follows immediately from the fact that testing whether a matrix of rationals is PSD can be done exactly in polynomial time, as noted in [10].[5] Thus the Ellipsoid Method, as described thoroughly in [11], will find a solution to the relaxed form of (SDP) in poly(N) time, provided one exists.

By performing $LDL^\top$ decomposition on the solution (see, e.g., [21]), in poly(N) time we get an exact SOS representation $p'(x) = \sum_{j=1}^n c_j r_j(x)^2$, where $p'(x)$ is a polynomial with the property that $|p_U - p'_U| \leq 2^{-N^c}$ for all U. (Here c_j and the coefficients of $r_j(x)$ are rational, and the degree of each $r_j(x)$ is at most $d/2$.) Writing $\Delta(x) = p'(x) - p(x)$, we have a degree-d SOS representation of $p(x) + \Delta(x)$, where Δ has degree at most d and all coefficients bounded by ϵ. Now for each monomial δx^U in $\Delta(x)$, we can get a degree-d SOS proof of $\delta x^U \leq |\delta|$ by using either $x^U = -1 + \frac{1}{2}(x^V + x^W)^2$ or $x^U = 1 - \frac{1}{2}(x^V - x^W)^2$, where V and W partition U into two sets, each of cardinality at most $d/2$. Adding these in for each of the roughly N^2 potential monomials of $\Delta(x)$ therefore gives a degree-$2d$ SOS representation of $p(x) + \epsilon$ for $\epsilon \lessapprox N^2 2^{-N^c}$, and we can make the constant c as large as we want.

5 Conclusion

Several papers have shown that certain "hard-seeming instances" of combinatorial optimization problems — like Unique-Games or Balanced-Separator — are not hard for the constant-degree SOS proof system. Optimistically, this might be evidence that there are better polynomial-time approximation algorithms for the problems than those currently known. But in the end, if we want to show that certain approximation tasks are literally in P in the Turing machine model, we'll have to treat some of the details discussed in this paper.

A good open problem is to establish useful conditions under which this treatment can be done automatically. E.g., does the **Question** from Section 1 have a positive answer if the constraints $x_i^2 \leq 1$ are upgraded to $x_i^2 = 1$?

[5] I've found that the correct proof of this fact appears extremely rarely in the literature; indeed, I've only seen it in the work of Grötschel, Lovász, and Schrijver [10, 11] and in a survey article by Lovász [19]. You certainly can't just "compute the eigenvalues of the matrix and check if they're nonnegative". It's also a somewhat common misconception [29, 4] that X is PSD if and only if its N leading principal minors are nonnegative. The correct proof from [19] involves seeing whether the Cholesky (LDL) decomposition on X succeeds. (This is essentially the same as the proof in [10], which involves finding the image of X, then checking if X is strictly positive definite on the image by testing if the leading principal minors are strictly positive.) In turn, this relies on the old but nonobvious [6] fact due to Edmonds [5] that Gaussian Elimination is in polynomial time.

Regarding errata for my own works: In [22, 14] we showed that certain explicit families of combinatorial optimization instances have low-degree SOS analyses. I did not yet verify that these SOS proofs also have the necessarily small bit-complexity that would allow an efficient algorithm to (approximately) find them. However we didn't formally claim that these algorithms exist; the theorems in these works were just evidence that SOS *might* be successful on all instances. In [2] we claimed that SOS algorithms could efficiently refute random instances of certain CSPs with certain parameters. I'm confident that this statement is true, but rather than prove it I'll simply say that the SOS-ability here is essentially just a side comment. It's clear that there is *some* efficient algorithm: all that's ultimately needed for refutation is the certification that a certain symmetric matrix A has $\|A\| \leq O(\text{small})$. This is equivalent to $O(\text{small}) \cdot I - A \succeq 0$, and as noted in Section 4, testing semidefiniteness of rational matrices can be done efficiently.

Acknowledgments I would like to thank Boaz Barak, Sangxia Huang, Etienne de Klerk, Pravesh Kothari, James Lee, Pablo Parrilo, Dima Pasechnik, David Steurer, and especially David Witmer for discussions. I would also like to thank Dieter van Melkebeek for pointing out the incorrect justification of (4) in an earlier draft of this work.

References

1. Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.
2. Sarah Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.
3. Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. In *Proceedings of the 2014 International Congress of Mathematicians*. International Mathematical Union, 2014.
4. Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(1–3):557–574, 1993.
5. Jack Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards*, 71B:241–245, 1967.
6. Jeff Erickson. What is the actual time complexity of Gaussian elimination? `http://cstheory.stackexchange.com/questions/3921/what-is-the-actual-time-complexity-of-gaussian-elimination`, 2010.
7. Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. Technical Report IHES/M/99/68, Insitut des Hautes Études Scientifiques, 1999.
8. Dima Grigoriev. Complexity of Positivstellensatz proofs for the knapsack. *Computational Complexity*, 10(2):139–154, 2001.
9. Dima Grigoriev and Nicolai Vorobjov. Complexity of Null- and Positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1):153–160, 2001.
10. Martin Grötschel, László Lovász, and Alexander Schrijver. The Ellipsoid Method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
11. Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer–Verlag, 1988.
12. Amir Hashemi and Daniel Lazard. Sharper complexity bounds for zero-dimensional gröbner bases and polynomial system solving. *International Journal of Algebra and Computation*, 21(05):703–713, 2011.
13. Cédric Josz and Didier Henrion. Strong duality in Lasserre's hierarchy for polynomial optimization. *Optimization Letters*, 10(1):3–10, 2016.

14 Manuel Kauers, Ryan O'Donnell, Li-Yang Tan, and Yuan Zhou. Hypercontractive inequalities via SOS, and the Frankl-Rödl graph. *Discrete Analysis*, 4, 2016.

15 Leonid Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

16 Jean Lasserre. Optimisation globale et théorie des moments. *Comptes Rendus de l'Académie des Sciences*, 331(11):929–934, 2000.

17 Jean Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

18 Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. *Emerging Applications of Algebraic Geometry*, 149:157–270, 2009.

19 László Lovász. Semidefinite programs and combinatorial optimization. In *Recent advances in algorithms and combinatorics*, pages 137–194. 2003.

20 Yurii Nesterov. *Squared functional systems and optimization problems*, chapter 17, pages 405–440. Kluwer Academic Publishers, 2000.

21 Ryan O'Donnell and Franklin Ta. Linear programming and semidefinite programming lecture 10 notes, 2011. `http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture10.pdf`.

22 Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1537–1556, 2013.

23 Pablo Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, 2000.

24 Lorant Porkolab and Leonid Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, 1997.

25 Motakuri Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77(1):129–162, 1997.

26 Claus Scheiderer. Sums of squares of polynomials with rational coefficients. *Journal of the European Mathematical Society*, 18(7):1495–1513, 2016.

27 Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602, 2008.

28 Naum Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

29 Kuduvally Swamy. On Sylvester's criterion for positive-semidefinite matrices. *IEEE Transactions on Automatic Control*, 18(3):306–306, 1973.

30 Sergey Tarasov and Mikhail Vyalyi. Semidefinite programming and arithmetic circuit evaluation. *Discrete Applied Mathematics*, 156(11):2070–2078, 2008.