Monique Laurent
Centrum voor Wiskunde en Informatica (CWI),
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
monique@cwi.nl

Research

Semidefinite Programming in Combinatorial and Polynomial Optimization

In recent years semidefinite programming has become a widely used tool for designing more efficient algorithms for approximating hard combinatorial optimization problems and, more generally, polynomial optimization problems, which deal with optimizing a polynomial objective function over a basic closed semi-algebraic set. The underlying paradigm is that while testing nonnegativity of a polynomial is a hard problem, one can test efficiently whether it can be written as a sum of squares of polynomials by using semidefinite programming. In this note we sketch some of the main mathematical tools that underlie this approach and illustrate its application to some graph problems dealing with maximum cuts, stable sets and graph colouring.

Linear optimization has become a well established area of applied mathematics that is widely and successfully used for modelling and solving many real-world applications. It is also extensively used for attacking integer or 0/1 linear problems, which are linear problems that arise naturally in combinatorial optimization where the variables are additionally constrained to take integer or 0/1 values respectively. While efficient algorithms exist for solving linear programming problems, most problems become intractable as soon as integrality constraints are added to them. Linear programming techniques are sometimes not powerful enough for designing good and efficient approximation algorithms for 0/1 linear problems. Semidefinite programming, an extension of linear programming where vector variables are replaced by matrix variables constrained to be positive semidefinite, turns out to be a more powerful technique for some problem classes. While semidefinite programming is also widely used in other areas like system and control theory (see for example [3]), we focus here on its application to combinatorial optimization and, more generally, to polynomial optimization. There is a vast amount of information on semidefinite programming in the literature; we now briefly introduce semidefinite programs and refer for example to [19, 42–43] and references therein for a detailed exposition.

Semidefinite programs

Linear programming deals with optimizing a linear function over a set defined by finitely many linear inequalities. Any linear program (LP) can be brought into the form

$$\max\{c^T x \mid a_j^T x = b_j \; (j = 1, \ldots, m) \text{ and } x \geq 0\}, \tag{1}$$

where $c, a_1, \ldots, a_m \in \mathbf{R}^n$ and $b = (b_j)_{j=1}^m \in \mathbf{R}^m$ are given and $x \in \mathbf{R}^n$ is the vector variable, constrained to be nonnegative. A semidefinite program (SDP) is the analogue of the LP (1) where we replace the vector variable $x \in \mathbf{R}^n$ with a matrix variable $X \in \mathbf{R}^{n\times n}$, constrained to be symmetric positive semidefinite. Recall that a symmetric matrix $X \in \mathbf{R}^{n\times n}$ is positive semidefinite, written as $X \succeq 0$, if $u^T X u \geq 0$ for all $u \in \mathbf{R}^n$ or, equivalently, if $X = (v_i^T v_j)_{i,j=1}^n$ for some vectors $v_1, \ldots, v_n \in \mathbf{R}^n$. In other words, a semidefinite program reads

$$\sup\{\mathrm{Tr}(C^T X) \mid \mathrm{Tr}(A_j^T X) = b_j \; (j = 1, \ldots, m) \text{ and } X \succeq 0\}, \tag{2}$$

where $C, A_1, \ldots, A_m \in \mathbf{R}^{n\times n}$ and $b \in \mathbf{R}^m$ are given and X is the matrix variable, required to lie in the cone S_n^+ of positive semidefinite matrices. While the feasible region of (1) is a polyhedron, that of (2) is a convex, in general non-polyhedral, set. Note that the SDP (2) reduces to the LP (1) when all C, A_j are diagonal matrices, and c, a_j denote their main diagonals.

Given an $n \times n$ rational symmetric matrix X, one can test in polynomial time (e.g. using Gaussian elimination) whether X is positive semidefinite and, if not, find a rational vector $u \in \mathbf{R}^n$ for which $u^T X u < 0$, thus giving a hyperplane separating X from the cone S_n^+. In technical terms, one can solve the separation problem over the positive semidefinite cone in polynomial time. Therefore, semidefinite programs can be solved in polynomial time to any fixed precision using the ellipsoid method (see [11]). Algorithms based on the ellipsoid method are however not practical since their running time is prohibitively high. Instead, interior-point algorithms are widely used in practice; they return an approximate optimum solution (to any given precision) in polynomially many iterations and their running time is efficient in practice for medium size problems.

Semidefinite programming in combinatorial optimization

We have chosen to illustrate the use of semidefinite programming in combinatorial optimization on the following basic problems: maximum stable sets, minimum graph colouring and maximum cuts in graphs. For these problems, some milestone results have been obtained in recent years that have spurred intense research activity and results for other optimization problems; we refer to [9, 19, 27, 30] and references therein for a detailed exposition. First we introduce some 'basic' SDP relaxations and then we indicate how to strengthen them and construct hierarchies leading to the full representation of the combinatorial problem at hand.

Maximum stable sets and graph colouring

Consider the problem of determining the *stability number* $\alpha(G)$ of a graph $G = (V, E)$, i.e. the maximum cardinality of a stable set in G, where a stable set is a set of pairwise non-adjacent vertices. A closely related problem is the *graph colouring problem*, which asks for the minimum number $\chi(G)$ of colours that are needed for colouring the nodes in such a way that adjacent nodes receive distinct colours. Thus $\chi(G)$ equals the minimum number of stable sets covering the vertex set V. Note that

$$\chi(G) \geq \omega(G), \tag{3}$$

where $\omega(G)$ is the largest cardinality of a *clique* in G, i.e. a set of pairwise adjacent vertices. Obviously, $\omega(G) = \alpha(\overline{G})$, where $\overline{G}$ is the complement of G, with the same set V of vertices and two distinct vertices being adjacent in $\overline{G}$ precisely when they are not adjacent in G.

For some graphs the inequality (3) is strict. For instance, it is strict for any circuit C_n of odd length $n \geq 5$, as $\omega(C_n) = 2 < \chi(C_n) = 3$, and for the complement $\overline{C}_n$ of C_n as well. However there are many interesting classes of graphs for which equality $\omega(G) = \chi(G)$ holds. This is the case e.g. for bipartite graphs, line graphs of bipartite graphs, comparability graphs and chordal graphs, and their complements as well. In fact the class of graphs for which equality $\omega(G) = \chi(G)$ holds not only for G but also for all its induced subgraphs, i.e. all those graphs that can be obtained by deleting vertices in G, turns out to be very interesting; following Berge, graphs in this class are called *perfect graphs*. Thus C_n and its complement $\overline{C}_n$ are not perfect for odd $n \geq 5$. Berge conjectured in 1962 that a graph is perfect if and only if its complement is perfect, which was proved a decade later by Lovász [28]. Berge also conjectured that a graph is perfect if and only if it does not contain any odd circuit or its complement of length at least 5 as an induced subgraph, which was proved only recently by Chudnovsky et al. [4] and is known as the *strong perfect graph theorem*. It is intriguing to determine the complexity of computing $\alpha(G)$ and $\chi(G)$ for perfect graphs. As we indicate below this can be done in polynomial time but to show this one has to use semidefinite programming.

Both problems of computing the stability number $\alpha(G)$ and the chromatic number $\chi(G)$ are NP-hard [7]. Lovász [29] introduced his celebrated *theta number* $\vartheta(G)$, which serves as bound for both $\alpha(G)$ and $\chi(G)$. The theta number is defined via the semidefinite program

$$\begin{aligned}\vartheta(G) := \max\{&\mathrm{Tr}(JX) \mid \mathrm{Tr}(X) = 1,\\ &X_{ij} = 0\ (ij \in E),\ X \succeq 0\},\end{aligned} \tag{4}$$

where J denotes the all-ones matrix. Hence it can be computed in polynomial time to any fixed precision. A basic property of the theta number is that it satisfies the so-called *sandwich inequality*

$$\begin{aligned}&\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G}), \text{ or equivalently,}\\ &\omega(G) \leq \vartheta(\overline{G}) \leq \chi(G).\end{aligned} \tag{5}$$

Indeed if $x = \chi^S \in \{0,1\}^V$ is the incidence vector of a stable set S in G (seen as a column vector) then $X := xx^T/|S|$ is feasible for the program (4) with objective value $|S|$, which gives $\alpha(G) \leq \vartheta(G)$. On the other hand, if X is a feasible solution to (4) and $V = C_1 \cup \ldots \cup C_k$ is a partition into $k := \chi(\overline{G})$ cliques of G, then

$$\begin{aligned}0 &\leq \sum_{h=1}^{k} (k\chi^{C_h} - e)^T X (k\chi^{C_h} - e)\\ &= k^2\mathrm{Tr}(X) - ke^T Xe = k\,(k - \mathrm{Tr}(JX))\,,\end{aligned}$$

where e is the all-ones vector, which implies $\mathrm{Tr}(JX) \leq k$ and thus $\vartheta(G) \leq \chi(\overline{G})$.

Hence, for perfect graphs, equality holds throughout in (5), which implies $\alpha(G) = \vartheta(G)$ and $\chi(G) = \vartheta(\overline{G})$. As the theta number can be computed in polynomial time to any fixed precision, the stability number and the chromatic number can be computed in polynomial time for perfect graphs. Moreover, a maximum stable set and a minimum colouring can also be computed in polynomial time for a perfect graph G (by iterated computations of the theta number of certain induced subgraphs of G). These computations thus rely on using semidefinite programming and as of today no alternative efficient algorithm is known.

Lovász' original motivation for introducing the theta number was to bound the *Shannon capacity* of a graph G, which is defined as

$$\Theta(G) := \lim_{k\to\infty} \alpha(G^k)^{\frac{1}{k}}. \tag{6}$$

Here G^k denotes the product of k copies of G, with vertex set V^k and with two distinct vertices $(u_1, \ldots, u_k)$ and $(v_1, \ldots, v_k)$ being adjacent in G^k if $u_h = v_h$ or $u_h v_h \in E$ for each position $h = 1, \ldots, k$. If we view V as an alphabet and adjacent vertices $u, v \in V$ as letters that can be confounded, then $\alpha(G^k)$ is the maximum number of words of length k that cannot be confounded, since for any two of them there is a position h where their hth letters cannot be confounded. One can verify that $\alpha(G^k) \geq \alpha(G)^k$ and $\vartheta(G^k) \leq \vartheta(G)^k$, which implies

$$\alpha(G) \leq \Theta(G) \leq \vartheta(G). \tag{7}$$

Therefore, when G is perfect, $\Theta(G) = \vartheta(G)$ can thus be computed via semidefinite programming. Lovász could also compute the Shannon capacity of the circuit C_5 using the theta number. He showed that $\Theta(C_5) = \sqrt{5}$, which follows from $\alpha(C_5^2) \geq 5$ (easy to verify) and $\vartheta(C_5) = \sqrt{5}$; the latter follows e.g. from the fact that $\vartheta(G)\vartheta(\overline{G}) = |V|$ when G is vertex transitive and that C_5 is vertex transitive and isomorphic to its complement. The exact value of the Shannon capacity of C_n is not known for odd $n \geq 7$.

Maximum cuts

Another successful application of semidefinite programming to combinatorial optimization is the celebrated 0.878-approximation algorithm of Goemans and Williamson [10] for the max-cut problem, which we briefly sketch below.

Given a graph $G = (V, E)$ and edge weights $w \in \mathbf{R}^E_+$, a *cut* is a set of edges of the form $\delta_G(S) := \{ij \in E \mid i \in S, j \in V \setminus S\}$ for some $S \subseteq V$, and its weight is $w(\delta_G(S)) = \sum_{ij \in \delta_G(S)} w_{ij}$. The *max-cut problem* asks for a cut of maximum total weight, whose weight is then denoted as $\mathrm{mc}(G)$. While a minimum weight nonempty cut can be found in polynomial time (using flow algorithms), the max-cut problem is NP-hard [7].

Erdös proposed in 1967 the following simple algorithm for constructing a cut of weight at least half the optimum cut. Colour the vertices $v_1, \ldots, v_n$ of G with two colours blue and red as follows: first colour v_1 with blue. Assuming $v_1, \ldots, v_i$ are already coloured, colour v_{i+1} with blue if the total weight of the edges joining v_{i+1} to the red vertices in $\{v_1, \ldots, v_i\}$ is more than the total weight of the edges joining v_{i+1} to the blue vertices in this set; otherwise colour v_{i+1} red. Then the cut formed by the edges connecting blue and red vertices has weight at least $w(E)/2$ and thus at least $\mathrm{mc}(G)/2$. This simple algorithm is thus an efficient 1/2-approximation algorithm for max-cut. There is an even easier randomized 1/2-approximation algorithm. Namely colour randomly each node blue or red independently, with probability 1/2. The probability that an edge belongs to the cut determined by this partition into blue and red vertices is 1/2 and thus the expected weight of this cut is $w(E)/2$. Can one construct in polynomial time a cut achieving a better approximation ratio? Goemans and Williamson [10] showed that this is indeed possible. For this they use a semidefinite program as relaxation for the max-cut problem and a suitable rounding of its optimum solution to a cut. To start with, they model the max-cut problem using ± 1-valued variables as

$$\mathrm{mc}(G) = \max \left\{ \sum_{ij \in E} w_{ij}(1 - x_i x_j)/2 \mid x \in \{\pm 1\}^V \right\}. \tag{8}$$

Observe that, for $x \in \{\pm 1\}^V$, the matrix $X := xx^T$ can be characterized by the constraints: (i) $X \succeq 0$, (ii) $X_{ii} = 1 \;\; \forall i \in V$ and (iii) $\mathrm{rank}(X) = 1$. If we omit the rank condition (iii) then we find the semidefinite relaxation

$$\mathrm{sdp}(G) := \max \left\{ \sum_{ij \in E} w_{ij}(1 - X_{ij})/2 \mid X \succeq 0,\ X_{ii} = 1 \ (i \in V) \right\}. \tag{9}$$

Let X be an optimum solution to (9). Goemans and Williamson propose the following random rounding procedure for constructing a good cut from X. Compute the Cholesky decomposition of X, i.e. vectors v_i ($i \in V$) such that $X_{ij} = v_i^T v_j \; \forall i, j \in V$. Select a random unit vector $r \in \mathbf{R}^n$. The hyperplane with normal r splits the vectors v_i into two sets, depending on the sign of $r^T v_i$. Let $S := \{i \in V \mid r^T v_i \geq 0\}$. As the probability that an edge ij lies in the cut $\delta_G(S)$ is equal to $\frac{1}{\pi} \arccos(v_i^T v_j)$, the expected weight of the cut $\delta_G(S)$ is equal to

$$\begin{aligned}\sum_{ij \in E} w_{ij} \frac{\arccos(v_i^T v_j)}{\pi} &= \sum_{ij \in E} w_{ij} \frac{1 - v_i^T v_j}{2} \, \frac{2}{\pi} \frac{\arccos v_i^T v_j}{1 - v_i^T v_j} \\ &\geq \alpha_{\mathrm{GW}} \,\mathrm{sdp}(G) \geq 0.878567 \,\mathrm{mc}(G),\end{aligned}$$

after setting $\alpha_{\mathrm{GW}} := \min_{0 < \vartheta \leq \pi} \frac{2}{\pi} \frac{\vartheta}{1 - \cos\vartheta}$ and observing that $\alpha_{\mathrm{GW}} > 0.878567$. This randomized algorithm can be derandomized to yield in polynomial time a deterministic cut achieving the same performance ratio.

Much research has been done trying to improve the Goemans-Williamson approximation algorithm for max-cut and to extend and apply it to other problems (see for example the survey [27] and references therein). However, although improved algorithms could be designed for special graph classes, no better approximation ratio could yet be shown for the general max-cut problem. It is in fact proved that α_{GW} is the best possible approximation ratio for max-cut that can be achieved in polynomial time (if P $\neq$ NP) under the so-called *Unique Games Conjecture* (see [17] and [18]). On the negative side, Håstad [15] proved that if P $\neq$ NP then no polynomial time approximation algorithm exists for max-cut with performance guarantee better than $16/17 \sim 0.94117$.

Hierarchies of semidefinite programming relaxations

We saw above how to define in a natural way a semidefinite relaxation for the maximum stable set problem (via the SDP (4)) and for the max-cut problem (via the SDP (9)). Several procedures have been proposed for constructing stronger SDP relaxations (discussed in [22, 27, 31] and references therein). We now describe a simple method for constructing a hierarchy of SDP relaxations, which finds the exact representation of the combinatorial problem at hand in finitely many steps. We present it for simplicity on the instance of the stable set problem.

Given a graph $G = (V, E)$, let P_G denote the convex hull of the incidence vectors of all stable sets in G; in other words,

$$P_G = \mathrm{conv}\left\{ x \in \{0,1\}^V \mid x_i + x_j \leq 1 \ (ij \in E) \right\},$$

called the *stable set polytope* of G. Then maximizing the linear function $\sum_{i \in V} x_i$ over P_G gives the stability number $\alpha(G)$, while maximizing it over a relaxation of P_G gives an upper bound on $\alpha(G)$. The basic idea is to 'lift' a vector $x \in \{0,1\}^V$ to the higher dimensional vector

$$x^{(t)} = \left(x_I := \prod_{i \in I} x_i \right)_{I \in P_t(V)}$$

indexed by $P_t(V) = \{I \subseteq V \mid |I| \leq t\}$

and to consider the matrix

$$X = x^{(t)} \left(x^{(t)}\right)^T.$$

Here are some obvious conditions satisfied by X: (i) $X \succeq 0$ and (ii) any (I,J)-entry of X depends only on the union $I \cup J$ (as $X_{I,J} = x_{I\cup J}$).

A matrix indexed by $P_t(V)$ satisfying (ii) is of the form

$$C_t(y) := (y_{I\cup J})_{I,J\in P_t(V)} \text{ for some } y \in \mathbf{R}^{P_{2t}(V)}; \qquad (10)$$

then $C_t(y)$ is called the *combinatorial moment matrix* of order t of y.

Summarizing, we just saw that, if $y = x^{(2t)}$ for some $x \in \{0,1\}^V$, then its combinatorial moment matrix satisfies the SDP condition $C_t(y) \succeq 0$. Moreover, $y_\emptyset = 1$ and, if x is the incidence vector of a stable set in G, then y satisfies the edge equations $y_{ij} = 0$ for all $ij \in E$. This motivates the following definition. For any integer $t \geq 1$, consider the set

$$\left\{y \in \mathbf{R}^{P_{2t}(V)} \mid C_t(y) \succeq 0,\ y_\emptyset = 1,\ y_{ij} = 0\ (ij \in E)\right\} \qquad (11)$$

and its projection onto the space $\mathbf{R}^V$, denoted $P_G^{(t)}$. As $P_G \subseteq P_G^{(t+1)} \subseteq P_G^{(t)}$, we obtain a hierarchy of SDP relaxations for the stable set polytope P_G. It finds P_G in $\alpha(G)$ steps, i.e. $P_G^{(t)} = P_G$ for $t \geq \alpha(G)$. Optimizing the function $\sum_{i\in V} x_i$ over $P_G^{(t)}$ yields an upper bound on $\alpha(G)$, which coincides with $\alpha(G)$ for $t \geq \alpha(G)$. This upper bound can be computed in polynomial time (to any precision) when t is fixed, since it is expressed via an SDP involving a matrix of size $O(n^t)$. Moreover, for $t = 1$, one can verify that this upper bound coincides with the theta number $\vartheta(G)$ from (4). Therefore, the above construction is a systematic procedure for producing a hierarchy of upper bounds for the stability number, starting with the theta number.

As t grows we obtain a tighter approximation of $\alpha(G)$, however at a higher computational cost. More economical block-diagonal variations of the above hierarchy have been proposed, which are based on considering, instead of the full matrix $C_t(y)$, a number of smaller blocks arising from principal submatrices of it. Computational experiments for the stable set and graph colouring problems show that such relaxations can give approximations for $\alpha(G)$ and $\chi(G)$, which may improve substantially the theta number (see [12–14, 20, 25]). When G is a Hamming graph, with vertex set $\{0,1\}^n$ and with edges the pairs of nodes with Hamming distance below a prescribed value, $\alpha(G)$ corresponds to the maximum cardinality of a code correcting a prescribed number of errors, $\vartheta(G)$ corresponds to the well-known LP bound of Delsarte [6], and the next bounds in the hierarchy are studied e.g. in [8, 25, 39]; as G has a large number of vertices, a crucial ingredient for the practical computation of these bounds is exploiting symmetry in the SDP formulations and using the explicit block-diagonalization of the Terwilliger algebra given in [39].

Semidefinite Programming in Polynomial Optimization

We now turn to the application of semidefinite programming to polynomial optimization. Given $p, g_1, \ldots, g_m \in \mathbf{R}[\mathbf{x}]$ the ring of polynomials in n variables $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, consider the problem

$$p^{\min} := \inf\{p(x) \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0\} \qquad (12)$$

of minimizing the polynomial p over the basic closed semi-algebraic set

$$K := \{x \in \mathbf{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}. \qquad (13)$$

This is a hard problem. For instance, it contains $0/1$ linear programming, as $0/1$ variables can be modelled by the quadratic equations $\mathbf{x}_i^2 = \mathbf{x}_i\ \forall i$. It also contains the max-cut problem (8) where the objective and the constraints are quadratic polynomials (expressing $x_i = \pm 1$ by $\mathbf{x}_i^2 = 1$).

We fix some notation. For $\alpha \in \mathbf{N}^n$, $\mathbf{x}^\alpha = \mathbf{x}_1^{\alpha_1} \cdots \mathbf{x}_n^{\alpha_n}$ is the monomial with exponent α, whose degree is $|\alpha| = \sum_i \alpha_i$. For an integer d, $\mathbf{N}_d^n = \{\alpha \in \mathbf{N}^n \mid |\alpha| \leq d\}$ corresponds to the set of monomials of degree at most d. For $g = \sum_\alpha g_\alpha \mathbf{x}^\alpha \in \mathbf{R}[\mathbf{x}]$, set $d_g := \lceil \deg(g)/2 \rceil$ and let $\vec{g} = (g_\alpha)_\alpha$ denote the vector of coefficients of g. Finally, for K as in (13), set

$$d_K := \max\{d_{g_1}, \ldots, d_{g_m}\}.$$

Several authors (see [21, 32, 34, 40]) have proposed approximating the problem (12) by convex (semidefinite) relaxations, obtained by using sums of squares representations for nonnegative polynomials and the dual theory of moments. We give below a brief sketch of this approach and refer e.g. to the survey [26] and references therein for more details. The basic idea underlying this approach is that, while testing whether a polynomial is nonnegative is a hard problem, the relaxed problem of testing whether it can be written as a sum of squares of polynomials is much easier since it can be reformulated as a semidefinite program.

Of course, as Hilbert already realized in 1888, not every nonnegative polynomial p can be written as a sum of squares of polynomials. This is true only in the following three exceptional cases: when p is univariate (in which case one can easily verify that p is a sum of two squares), when p is quadratic (which corresponds to the fact that a positive semidefinite matrix A can be written as BB^T for some matrix B) and when p is a quartic polynomial in 2 variables (in which case Hilbert proved that p can be written as a sum of three squares - a non-trivial result). In all other cases Hilbert proved that there exists a nonnegative polynomial that is not a sum of squares of polynomials. His proof was not constructive. Concrete examples of such polynomials were found only much later; for instance, the following polynomial $\mathbf{x}_1^2\mathbf{x}_2^2(\mathbf{x}_1^2 + \mathbf{x}_2^2 - 3) + 1$ is due to Motzkin (see [41] for a detailed account). Hilbert asked at the 1900 International Congress of Mathematicians in Paris whether every nonnegative polynomial can be written as a sum of squares of *rational* functions, known as Hilbert's 17th problem. This was settled in the affirmative by Artin in 1927, whose work laid the foundations for the field of real algebraic geometry. See for example [35–36] for a detailed exposition.

Sums of squares of polynomials and semidefinite programming

We first recall how to test whether a polynomial can be written as a sum of squares of polynomials using semidefinite programming: a polynomial $p = \sum_\alpha p_\alpha \mathbf{x}^\alpha$ of degree $2d$ is a sum of squares of polynomials (s.o.s. for short), i.e. $p = \sum_{j=1}^m u_j^2$ for some $u_j \in \mathbf{R}[\mathbf{x}]$, if and only if the SDP

$$X \succeq 0, \quad \sum_{\substack{\beta,\gamma\in\mathbf{N}^n_d \\ \beta+\gamma=\alpha}} X_{\beta,\gamma} = p_\alpha \quad (\alpha \in \mathbf{N}^n_{2d}) \tag{14}$$

is feasible, where the matrix variable X is indexed by $\mathbf{N}^n_d$. Indeed, setting $\mathbf{z} := (\mathbf{x}^\alpha)_{\alpha\in\mathbf{N}^n_d}$, we have

$$\sum_{j=1}^m u_j^2 = \sum_{j=1}^m (\mathbf{z}^T\vec{u}_j)^2 = \mathbf{z}^T \left(\underbrace{\sum_{j=1}^m \vec{u}_j\vec{u}_j^T}_{=:\, X\succeq 0} \right) \mathbf{z}$$

$$= \sum_{\beta,\gamma\in\mathbf{N}^n_d} \mathbf{x}^\beta\mathbf{x}^\gamma X_{\beta,\gamma} = \sum_{\alpha\in\mathbf{N}^n_{2d}} \mathbf{x}^\alpha \left(\sum_{\substack{\beta,\gamma\in\mathbf{N}^n_d \\ \beta+\gamma=\alpha}} X_{\beta,\gamma} \right),$$

which shows that the s.o.s. decompositions for p correspond to the solutions X of (14).

We now introduce some SDP relaxations based on sums of squares for the polynomial optimization problem (12). Observe first that (12) can be rewritten as

$$p^{\min} = \sup\{\lambda \mid p(x) - \lambda \geq 0 \;\forall x \in K\}. \tag{15}$$

Then define, for any integer $t \geq \max(d_K, d_p)$, the parameter

$$p_t^{\text{sos}} := \sup \Big\{ \lambda \mid p - \lambda = s_0 + \sum_{j=1}^m s_j g_j \\ \text{such that } s_0, s_j \text{ s.o.s. with } \deg(s_0), \deg(s_j g_j) \leq 2t \Big\}, \tag{16}$$

which is obviously a lower bound for $p^{\min}$. Moreover, it follows from the above that p_t^{sos} can be computed via semidefinite programming. As $p_t^{\text{sos}} \leq p_{t+1}^{\text{sos}} \leq p^{\min}$, we obtain a hierarchy of SDP bounds for (12).

Positive semidefinite moment matrices and polynomial optimization

We now give a 'dual' SDP hierarchy for $p^{\min}$ in terms of moment matrices. For this let us go back to problem (12) and observe that it can be reformulated as

$$p^{\min} = \inf\Big\{ y^T\vec{p} \mid \exists\, \mu \text{ probability measure on } K \text{ such that} \\ y_\alpha = \int_K x^\alpha \mu(dx) \;\forall\alpha \Big\}; \tag{17}$$

here the variable y is constrained to have a representing measure μ, in which case the quantity $\int_K x^\alpha\mu(dx)$ is called its *moment* of order α. Indeed, if μ is a probability measure on K then $\int_K p(x)\mu(dx) \geq \int_K p^{\min}\mu(dx) = p^{\min}$, giving inf(17) $\geq p^{\min}$. On the other hand, if $x_0 \in K$ and μ is the Dirac measure at x_0, then $p(x_0) = \int_K p(x)\mu(dx) \geq$ inf(17), thus giving the reverse inequality $p^{\min} \geq$ inf(17).

Characterizing the sequences y having a representing measure on K is the object of classical moment theory. Well-known necessary conditions include (i) $M_t(y) \succeq 0$, and the *localizing conditions* (ii) $M_{t-d_{g_j}}(g_j y) \succeq 0$ $(j \leq m)$ for any $t \geq d_K$. Here $M_t(y) := (y_{\beta+\gamma})_{\beta,\gamma\in\mathbf{N}^n_t}$ is the *moment matrix* of order t of y and, for a polynomial $g = \sum_\alpha g_\alpha\mathbf{x}^\alpha$, $gy \in \mathbf{R}^{\mathbf{N}^n}$ is the sequence with αth entry $\sum_\beta g_\beta y_{\alpha+\beta}$. Hence, for any $t \geq \max(d_K, d_p)$, the parameter

$$p_t^{\text{mom}} := \inf\{ y^T\vec{p} \mid y_0 = 1,\; M_t(y) \succeq 0, \\ M_{t-d_{g_j}}(g_j y) \succeq 0 \;(j = 1,\ldots,m)\} \tag{18}$$

is an SDP lower bound for (12). The two programs (14) and (17) give 'dual' formulations for $p^{\min}$, corresponding to the known duality between the cone of nonnegative polynomials on K and the cone of sequences having a nonnegative representing measure on K, while the two programs (16) and (18) are dual SDPs (see [21] for details). We have $p_t^{\text{sos}} \leq p_t^{\text{mom}} \leq p^{\min}$, with equality $p_t^{\text{mom}} = p_t^{\text{sos}}$, e.g. when K has a nonempty interior. We see below some conditions under which the SDP relaxations are exact, i.e. equality $p_t^{\text{mom}} = p_t^{\text{sos}} = p^{\min}$ holds.

Convergence, optimality certificate and extracting global minimizers

We group here some basic properties of the SDP hierarchies (14) and (17), regarding convergence and extraction of a global minimizer for the original problem (12).

Assume that the quadratic module $M_K := \{s_0 + \sum_{j=1}^m s_j g_j \mid s_0, s_j \text{ s.o.s}\}$ is Archimedean, i.e. $\forall p \in \mathbf{R}[\mathbf{x}]$ $N \pm p \in M_K$ for some $N \in \mathbf{N}$. As shown by Schmüdgen [38], M_K is Archimedean if and only if the set $\{x \in \mathbf{R}^n \mid u(x) \geq 0\}$ is compact for some $u \in M_K$. Thus M_K Archimedean implies K compact. On the other hand, if K is compact and if we know an explicit ball of radius R containing K, then it suffices to add the quadratic constraint $R^2 - \sum_i \mathbf{x}_i^2 \geq 0$ to the description of K to make M_K Archimedean. The important fact for our treatment here is that if M_K is Archimedean then there is asymptotic convergence of p_t^{sos} (and thus of p_t^{mom}) to $p^{\min}$ as $t \to \infty$. As pointed out in [21], this follows directly from the following representation result of Putinar [37]: if M_K is Archimedean then any polynomial that is positive on K belongs to M_K.

Sometimes there is even *finite* convergence to $p^{\min}$. For instance, $p_t^{\text{sos}} = p_t^{\text{mom}} = p^{\min}$ (or $p_t^{\text{mom}} = p^{\min}$) for t large enough when the description of K contains a set of equations having finitely many common complex (or real) roots (see [24, 26]). Finite convergence occurs in particular in the 0/1 case considered earlier, corresponding to the presence of the equations $\mathbf{x}_i^2 = \mathbf{x}_i$ $(i = 1,\ldots,n)$. Note that, in the presence of these equations, one can eliminate all variables y_α with some $\alpha_i \geq 2$ in the moment matrices $M_t(y)$ in (18), so that we find again the combinatorial moment matrices $C_t(y)$ considered in (10).

Another interesting case of (finite) convergence is for the problem (12) of minimizing a polynomial p over its gradient variety

$$K_p := \{x \in \mathbf{R}^n \mid \partial p/\partial x_i = 0 \;\forall i = 1,\ldots,n\},$$

which follows from the following result of Nie et al. [33]: if p is positive on K_p then p is an s.o.s. modulo its gradient ideal I_p, defined as the ideal generated by $\partial p/\partial x_i$ $(i = 1, \dots, n)$; moreover the same conclusion holds when p is nonnegative on K_p and I_p is a radical ideal.

Henrion and Lasserre [16] give the following optimality criterion for the SDP hierarchy (18): if y is an optimum solution to (18) satisfying

$$\begin{aligned} &\mathrm{rank} M_s(y) = \mathrm{rank} M_{s-d_K}(y) \\ &\text{for some } \max(d_K, d_p) \le s \le t \end{aligned} \tag{19}$$

then equality $p_t^{\mathrm{mom}} = p^{\mathrm{min}}$ holds and, moreover, all common roots to the polynomials lying in the kernel of $M_s(y)$ are global minimizers of p over the set K. Therefore one can compute these roots (e.g. using the so-called eigenvalue method for solving polynomial equations) and thus obtain global minimizers for the original problem (12). Here is a brief sketch of the proof for this optimality criterion. It relies on the following results of [5] for moment matrices: firstly, if $\mathrm{rank} M_s(y) = \mathrm{rank} M_{s-1}(y)$ then y can be extended to $\tilde{y} \in \mathbf{R}^{\mathbf{N}^n}$ in such a way that $\mathrm{rank} M_\infty(\tilde{y}) = \mathrm{rank} M_s(y)$. Secondly, if $M_\infty(\tilde{y}) \succeq 0$ with finite rank then $\tilde{y}$ has a representing measure. Combining these two results one can derive that, under the rank condition (19), y has a representing measure μ on K up to order $2s$; this implies that $p_t^{\mathrm{mom}} = y^T \vec{p} = \int_K p(x)\mu(dx) \ge p^{\mathrm{min}}$ and thus equality $p_t^{\mathrm{mom}} = p^{\mathrm{min}}$ holds and, moreover, the support of μ is contained in the set of global minimizers. See for example [26] for a detailed exposition.

Conclusions

We have given here a brief sketch of how to use semidefinite programming for designing hierarchies of convex relaxations for polynomial optimization problems, which include 0/1 linear optimization problems as special instances. The underlying paradigm is that, while testing whether a polynomial is nonnegative is a hard problem, one can test whether it can be written as a sum of squares efficiently using semidefinite programming. The duality between nonnegative polynomials and moment theory leads to dual SDPs in terms of sums of squares and in terms of positive semidefinite moment matrices, the latter lending themselves to possible extraction of global optimizers. There are many further interesting aspects that were not discussed here. To name just a few: how often do positive polynomials admit s.o.s. decompositions? Various answers may be given depending whether one lets the number of variables or the degree vary; how do you reduce the size of the SDPs using structural properties of the problem, like equations, sparsity or symmetries? This is indeed crucial as SDPs that are too large could not be handled by the current SDP solvers; and how do these hierarchies (based on Putinar's representation theorem) compare to other hierarchies based on other representation results, like e.g. Pólya's representation theorem for positive homogeneous polynomials on the standard simplex?

Finally let us mention some recent work showing that semidefinite programming combined with invariant theory and harmonic analysis can also be very useful for attacking various problems on the unit sphere. In particular, Bachoc and Vallentin [1] obtain the best upper bounds for the famous kissing number in dimension up to 10, while Bachoc et al. [2] introduce an analogue of the theta number for compact metric spaces, leading e.g. to new lower bounds for the measurable chromatic number of distance graphs on the unit sphere.

Thanks This work was supported by the Netherlands Organization for Scientific Research under grant NWO 639.032.203.

References

1 C. Bachoc and F. Vallentin, 'New upper bounds for kissing numbers from semidefinite programming', *Journal of the American Mathematical Society* 21(**3**): 909–924, 2008

2 C. Bachoc, G. Nebe, F. Mario de Oliveira Filho, F. Vallentin, 'Lower bounds for measurable chromatic numbers', arXiv:0801.1059, 2008

3 S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics, volume 15, SIAM, 1994

4 M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, 'The strong perfect graph theorem', *Annals of Mathematics* **164**:51–229, 2006

5 R.E. Curto and L.A. Fialkow, 'Solution of the truncated complex moment problem for flat data', *Memoirs of the American Mathematical Society*, vol. 119, n. 568, 1996

6 P. Delsarte, *An Algebraic Approach to the Association Schemes of Coding Theory*, Philips Research Reports Supplements (1973) No. 10, Philips Research Laboratories, Eindhoven, 1973

7 M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, W.H. Freeman & Company, Publishers, 1979

8 D. Gijswijt, *Matrix Algebras and Semidefinite Programming Techniques for Codes*, PhD thesis, Univ. Amsterdam, 2005

9 M.X. Goemans, 'Semidefinite Programming in Combinatorial Optimization', *Mathematical Programming* **79**:143–161, 1997

10 M.X. Goemans and D. Williamson, 'Improved approximation algorithms for maximum cuts and satisfiability problems using semidefinite programming', *Journal of the ACM* **42**:1115–1145, 1995

11 M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988

12 N. Gvozdenović. *Approximating the Stability Number and the Chromatic Number of a Graph via Semidefinite Programming*, PhD thesis, Univ. Amsterdam, 2008

13 N. Gvozdenović and M. Laurent, 'Computing semidefinite programming lower bounds for the (fractional) chromatic number via block-diagonalization', *SIAM Journal on Optimization* 19 (**2**): 592–615, 2008

14 N. Gvozdenović, M. Laurent and F. Vallentin, 'Block-diagonal semidefinite programming hierarchies for 0/1 programming', arXiv:0712.3079, 2007

15 J. Håstad, 'Some optimal inapproximability results', in *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pp. 1–10, 1997

16 D. Henrion and J.-B. Lasserre, 'Detecting global optimality and extracting solutions in GloptiPoly', in *Positive Polynomials in Control*, D. Henrion and A. Garulli (eds.), LNCIS **312**:293–310, 2005

17 S. Khot, 'On the power of unique 2-prover 1-round games', in *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, ACM, New York, pp. 767–775, 2002

18 S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, 'Optimal inapproximability results for Max-Cut and other 2-variable CSPs?', in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, IEEE, Washington, pp. 146–154, 2004

19 E. de Klerk, *Aspects of Semidefinite Programming - Interior Point Algorithms and Selected Applications*, Kluwer, 2002

20 E. De Klerk, D.V. Pasechnik, 'A note on the stability number of an orthogonality graph' *European Journal of Combinatorics* **28**:1971–1979, 2007

21 J.B. Lasserre, 'Global optimization with polynomials and the problem of moments', *SIAM Journal on Optimization* **11**:796–817, 2001

22 J.B. Lasserre, 'An explicit exact SDP relaxation for nonlinear $0-1$ programs', in K. Aardal and A.M.H. Gerards, eds., *Lecture Notes in Computer Science* **2081**:293–303, 2001

23 M. Laurent, 'A comparison of the Sherali-Adams', Lovász-Schrijver and Lasserre relaxations for $0-1$ programming', *Mathematics of Operations Research* **28(3)**:470–496, 2003

24 M. Laurent, 'Semidefinite representations for finite varieties', *Mathematical Programming* **109**:1–26, 2007

25 M. Laurent, 'Strengthened semidefinite programming bounds for codes', *Mathematical Programming* **109(2-3)**:239-261, 2007

26 M. Laurent, 'Sums of squares, moment matrices and optimization over polynomials', in *IMA volume Emerging Applications of Algebraic Geometry*, M. Putinar and S. Sullivant (eds.), 2008

27 M. Laurent and F. Rendl, 'Semidefinite Programming and Integer Programming', in *Handbook on Discrete Optimization*, K. Aardal, G. Nemhauser, R. Weismantel (eds.), pp. 393-514, Elsevier B.V., 2005

28 L. Lovász, 'Normal hypergraphs and the perfect graph conjecture', *Discrete Mathematics* **2**:253–267, 1972

29 L. Lovász, 'On the Shannon capacity of a graph', *IEEE Transactions on Information Theory* **IT-25**:1–7, 1979

30 L. Lovász, 'Semidefinite programs and combinatorial optimization', in *Recent Advances in Algorithms and Combinatorics*, B.A. Reed and C.L. Sales (eds.), Springer, 2003

31 L. Lovász and A. Schrijver, 'Cones of matrices and set-functions and $0-1$ optimization', *SIAM Journal on Optimization* **1**:166–190, 1991

32 Y. Nesterov, 'Squared functional systems and optimization problems', in J.B.G. Frenk, C. Roos, T. Terlaky and S. Zhang (eds.), *High Performance Optimization*, Kluwer, pp. 405–440, 2000

33 J. Nie, J. Demmel and B. Sturmfels, 'Minimizing polynomials via sums of squares over the gradient ideal', *Mathematical Programming* **106**:587–606, 2006

34 P.A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, PhD thesis, California Institute of Technology, 2000

35 V.V. Prasolov, *Polynomials*, Springer, 2004

36 A. Prestel and C.N. Delzell, *Positive Polynomials - From Hilbert's 17th Problem to Real Algebra*, Springer, 2001

37 M. Putinar, 'Positive polynomials on compact semi-algebraic sets', *Indiana University Mathematics Journal* **42**:969–984, 1993

38 K. Schmüdgen, 'The K-moment problem for compact semi-algebraic sets', *Mathematische Annalen* **289**:203–206, 1991

39 A. Schrijver, 'New code upper bounds from the Terwilliger algebra and semidefinite programming', *IEEE Transactions on Information Theory* **51**:2859–2866, 2005

40 N.Z. Shor, 'An approach to obtaining global extremums in polynomial mathematical programming problems', *Kibernetika* **5**:102–106, 1987

41 B. Reznick, 'Some concrete aspects of Hilbert's 17th problem', In *Real Algebraic Geometry and Ordered Structures*, C.N. Delzell and J.J. Madden, eds., *Contemporary Mathematics* 253: 251–272, 2000

42 L. Vandenberghe and S. Boyd, 'Semidefinite Programming', *SIAM Review* **38(1)**:49–95, 1996

43 H. Wolkowicz, R. Saigal, L. Vandeberghe (eds.), *Handbook of Semidefinite Programming*, Kluwer, 2000