**Noname manuscript No.**
(will be inserted by the editor)

# DC Decomposition of Nonconvex Polynomials with Algebraic Techniques

**Amir Ali Ahmadi · Georgina Hall**

Received: date / Accepted: date

**Abstract** We consider the problem of decomposing a multivariate polynomial as the difference of two convex polynomials. We introduce algebraic techniques which reduce this task to linear, second order cone, and semidefinite programming. This allows us to optimize over subsets of valid difference of convex decompositions (dcds) and find ones that speed up the convex-concave procedure (CCP). We prove, however, that optimizing over the entire set of dcds is NP-hard.

**Keywords** Difference of convex programming, conic relaxations, polynomial optimization, algebraic decomposition of polynomials

## 1 Introduction

A difference of convex (dc) program is an optimization problem of the form

$$\begin{aligned} &\min f_0(x) \\ &\text{s.t. } f_i(x) \leq 0, i = 1, \ldots, m, \end{aligned} \tag{1}$$

where $f_0, \ldots, f_m$ are difference of convex functions; i.e.,

$$f_i(x) = g_i(x) - h_i(x), i = 0, \ldots, m, \tag{2}$$

The authors are partially supported by the Young Investigator Program Award of the AFOSR.

Amir Ali Ahmadi
ORFE, Princeton University, Sherrerd Hall, Princeton, NJ 08540
E-mail: a_a_a@princeton.edu

Georgina Hall
ORFE, Princeton University, Sherrerd Hall, Princeton, NJ 08540
E-mail: gh4@princeton.edu

and $g_i : \mathbb{R}^n \to \mathbb{R},\ h_i : \mathbb{R}^n \to \mathbb{R}$ are convex functions. The class of functions that can be written as a difference of convex functions is very broad containing for instance all functions that are twice continuously differentiable [18], [22]. Furthermore, any continuous function over a compact set is the uniform limit of a sequence of dc functions; see, e.g., reference [25] where several properties of dc functions are discussed.

Optimization problems that appear in dc form arise in a wide range of applications. Representative examples from the literature include machine learning and statistics (e.g., kernel selection [9], feature selection in support vector machines [23], sparse principal component analysis [30], and reinforcement learning [37]), operations research (e.g., packing problems and production-transportation problems [45]), communications and networks [8],[31], circuit design [30], finance and game theory [17], and computational chemistry [13]. We also observe that dc programs can encode constraints of the type $x \in \{0, 1\}$ by replacing them with the dc constraints $0 \leq x \leq 1, x - x^2 \leq 0$. This entails that any binary optimization problem can in theory be written as a dc program, but it also implies that dc problems are hard to solve in general.

As described in [40], there are essentially two schools of thought when it comes to solving dc programs. The first approach is global and generally consists of rewriting the original problem as a concave minimization problem (i.e., minimizing a concave function over a convex set; see [46], [44]) or as a reverse convex problem (i.e., a convex problem with a linear objective and one constraint of the type $h(x) \geq 0$ where $h$ is convex). We refer the reader to [43] for an explanation on how one can convert a dc program to a reverse convex problem, and to [21] for more general results on reverse convex programming. These problems are then solved using branch-and-bound or cutting plane techniques (see, e.g., [45] or [25]). The goal of these approaches is to return global solutions but their main drawback is scalibility. The second approach by contrast aims for local solutions while still exploiting the dc structure of the problem by applying the tools of convex analysis to the two convex components of a dc decomposition. One such algorithm is the Difference of Convex Algorithm (DCA) introduced by Pham Dinh Tao in [41] and expanded on by Le Thi Hoai An and Pham Dinh Tao. This algorithm exploits the duality theory of dc programming [42] and is popular because of its ease of implementation, scalability, and ability to handle nonsmooth problems.

In the case where the functions $g_i$ and $h_i$ in (2) are differentiable, DCA reduces to another popular algorithm called the Convex-Concave Procedure (CCP) [27]. The idea of this technique is to simply replace the concave part of $f_i$ (i.e., $-h_i$) by a linear overestimator as described in Algorithm 1. By doing this, problem (1) becomes a convex optimization problem that can be solved using tools from convex analysis. The simplicity of CCP has made it an attractive algorithm in various areas of application. These include statistical physics (for minimizing Bethe and Kikuchi free energy functions [48]), machine learning [30],[15],[11], and image processing [47], just to name a few. In addition, CCP enjoys two valuable features: (i) if one starts with a feasible solution, the solution produced after each iteration remains feasible, and (ii) the objective value improves in every iteration, i.e., the method is a descent algorithm. The proof of both claims readily comes out of the description of the algorithm and can be found, e.g., in [30, Section 1.3.], where several other properties of the method are also laid out. Like many iterative algorithms, CCP relies on a stopping criterion to end. This criterion can be chosen amongst a few alternatives. For example, one could stop if the value of the objective does not improve enough, or if the iterates are too close to one another, or if the norm of the gradient of $f_0$ gets small.

**Algorithm 1** CCP

```
Input: $x_0,\ f_i = g_i - h_i, i = 0, \dots, m$
1: $k \leftarrow 0$
2: while stopping criterion not satisfied do
3:     Convexify: $f_i^k(x) := g_i(x) - (h_i(x_k) + \nabla h_i(x_k)^T(x - x_k)),\ i = 0, \dots, m$
4:     Solve convex subroutine: $\min f_0^k(x)$, s.t. $f_i^k(x) \le 0, i = 1, \dots, m$
5:     $x_{k+1} := \underset{f_i^k(x) \le 0}{\operatorname{argmin}} f_0^k(x)$
6:     $k \leftarrow k + 1$
7: end while
Output: $x_k$
```

Convergence results for CCP can be derived from existing results found for DCA, since CCP is a subcase of DCA as mentioned earlier. But CCP can also be seen as a special case of the family of majorization-minimization (MM) algorithms. Indeed, the general concept of MM algorithms is to iteratively upperbound the objective by a convex function and then minimize this function, which is precisely what is done in CCP. This fact is exploited by Lanckriet and Sriperumbudur in [27] and Salakhutdinov et al. in [39] to obtain convergence results for the algorithm, showing, e.g., that under mild assumptions, CCP converges to a stationary point of the optimization problem (1).

### 1.1 Motivation and organization of the paper

Although a wide range of problems already appear in dc form (2), such a decomposition is not always available. In this situation, algorithms of dc programming, such as CCP, generally fail to be applicable. Hence, the question arises as to whether one can (efficiently) compute a difference of convex decomposition (dcd) of a given function. This challenge has been raised several times in the literature. For instance, Hiriart-Urruty [22] states "All the proofs [of existence of dc decompositions] we know are "constructive" in the sense that they indeed yield [$g_i$] and [$h_i$] satisfying (2) but could hardly be carried over [to] computational aspects". As another example, Tuy [45] writes: "The dc structure of a given problem is not always apparent or easy to disclose, and even when it is known explicitly, there remains for the problem solver the hard task of bringing this structure to a form amenable to computational analysis."

Ideally, we would like to have not just the ability to find one dc decomposition, but also to optimize over the set of valid dc decompositions. Indeed, dc decompositions are not unique: Given a decomposition $f = g - h$, one can produce infinitely many others by writing $f = g + p - (h + p)$, for any convex function $p$. This naturally raises the question whether some dc decompositions are better than others, for example for the purposes of CCP.

In this paper we consider these decomposition questions for multivariate polynomials. Since polynomial functions are finitely parameterized by their coefficients, they provide a convenient setting for a computational study of the dc decomposition questions. Moreover, in most practical applications, the class of polynomial functions is large enough for modeling purposes as polynomials can approximate any continuous function on compact sets with arbitrary accuracy. It could also be interesting for future research to explore the potential of dc programming techniques for solving the polynomial optimization problem. This is the problem of minimizing a multivariate polynomial subject to polynomial inequalities and is currently an active area of research with applications throughout engineering and applied mathematics. In the case of quadratic polynomial optimization problems, the dc decomposition approach has already been studied [10],[24].

With these motivations in mind, we organize the paper as follows. In Section 2, we start by showing that unlike the quadratic case, the problem of testing if two given polynomials $g, h$ form a valid dc decomposition of a third polynomial $f$ is NP-hard (Proposition 1). We then investigate a few candidate optimization problems for finding dc decompositions that speed up the convex-concave procedure. In particular, we extend the notion of an undominated dc decomposition from the quadratic case [10] to higher order polynomials. We show that an undominated dcd always exists (Theorem 1) and can be found by minimizing a certain linear function of one of the two convex functions in the decomposition. However, this optimization problem is proved to be NP-hard for polynomials of degree four or larger (Proposition 4). To cope with intractability of finding optimal dc decompositions, we propose in Section 3 a class of algebraic relaxations that allow us to optimize over subsets of dcds. These relaxations will be based on the notions of *dsos-convex, sdsos-convex,* and *sos-convex* polynomials (see Definition 5), which respectively lend themselves to *linear, second order,* and *semidefinite programming.* In particular, we show that a dc decomposition can always be found by linear programming (Theorem 2). Finally, in Section 4, we perform some numerical experiments to compare the scalability and performance of our different algebraic relaxations.

## 2 Polynomial dc decompositions and their complexity

To study questions around dc decompositions of polynomials more formally, let us start by introducing some notation. A multivariate *polynomial* $p(x)$ in variables $x := (x_1, \ldots, x_n)^T$ is a function from $\mathbb{R}^n$ to $\mathbb{R}$ that is a finite linear combination of monomials:

$$p(x) = \sum_{\alpha} c_\alpha x^\alpha = \sum_{\alpha_1,\ldots,\alpha_n} c_{\alpha_1,\ldots,\alpha_n} x_1^{\alpha_1} \cdots x_n^{\alpha_n}, \tag{3}$$

where the sum is over $n$-tuples of nonnegative integers $\alpha_i$. The *degree* of a monomial $x^\alpha$ is equal to $\alpha_1 + \cdots + \alpha_n$. The degree of a polynomial $p(x)$ is defined to be the highest degree of its component monomials. A simple counting argument shows that a polynomial of degree $d$ in $n$ variables has $\binom{n+d}{d}$ coefficients. A *homogeneous polynomial* (or a *form*) is a polynomial where all the monomials have the same degree. An $n$-variate form $p$ of degree $d$ has $\binom{n+d-1}{d}$ coefficients.

Recall that a symmetric matrix $A$ is positive semidefinite (psd) if $x^T Ax \geq 0$ for all $x \in \mathbb{R}^n$; this will be denoted by the standard notation $A \succeq 0$. Similarly, a polynomial $p(x)$ is said to be *nonnegative* or positive semidefinite if $p(x) \geq 0$ for all $x \in \mathbb{R}^n$. For a polynomial $p$, we denote its Hessian by $H_p$. The second order characterization of convexity states that $p$ is convex if and only if $H_p(x) \succeq 0$, $\forall x \in \mathbb{R}^n$.

**Definition 1** We say a polynomial $g$ is a *dcd* of a polynomial $f$ if $g$ is convex and $g - f$ is convex.

Note that if we let $h := g - f$, then indeed we are writing $f$ as a difference of two convex functions $f = g - h$. It is known that any polynomial $f$ has a (polynomial) dcd $g$. A proof of this is given, e.g., in [47], or in Section 3.2, where it is obtained as corollary of a stronger theorem (see Corollary 1). By default, all dcds considered in the sequel will be of even degree. Indeed, if $f$ is of even degree $2d$, then it admits a dcd $g$ of degree $2d$. If $f$ is of odd degree $2d - 1$, it can be viewed as a polynomial $\tilde{f}$ of even degree $2d$ with highest-degree coefficients which are 0. The previous result then remains true, and $\tilde{f}$ admits a dcd of degree $2d$.

Our results show that such a decomposition can be found efficiently (e.g., by linear programming); see Theorem 3. Interestingly enough though, it is not easy to check if a candidate $g$ is a valid dcd of $f$.

**Proposition 1** *Given two $n$-variate polynomials $f$ and $g$ of degree 4, with $f \neq g$, it is strongly NP-hard* [1] *to determine whether $g$ is a dcd of $f$.*[2]

*Proof* We will show this via a reduction from the problem of testing nonnegativity of biquadratic forms, which is already known to be strongly NP-hard [29], [4]. A biquadratic form $b(x, y)$ in the variables $x = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_m)^T$ is a quartic form that can be written as

$$b(x; y) = \sum_{i \leq j, k \leq l} a_{ijkl} x_i x_j y_k y_l.$$

[1] For a strongly NP-hard problem, even a pseudo-polynomial time algorithm cannot exist unless P=NP [16].

[2] If we do not add the condition on the input that $f \neq g$, the problem would again be NP-hard (in fact, this is even easier to prove). However, we believe that in any interesting instance of this question, one would have $f \neq g$.

Given a biquadratic form $b(x;y)$, define the $n \times n$ polynomial matrix $C(x,y)$ by setting $[C(x,y)]_{ij} := \frac{\partial b(x;y)}{\partial x_i \partial y_j}$, and let $\gamma$ be the largest coefficient in absolute value of any monomial present in some entry of $C(x,y)$. Moreover, we define

$$r(x;y) := \frac{n^2\gamma}{2} \sum_{i=1}^{n} x_i^4 + \sum_{i=1}^{n} y_i^4 + \sum_{1 \leq i < j \leq n} x_i^2 x_j^2 + \sum_{1 \leq i < j \leq n} y_i^2 y_j^2.$$

It is proven in [4, Theorem 3.2.] that $b(x;y)$ is nonnegative if and only if

$$q(x,y) := b(x;y) + r(x,y)$$

is convex. We now give our reduction. Given a biquadratic form $b(x;y)$, we take $g = q(x,y) + r(x,y)$ and $f = r(x,y)$. If $b(x;y)$ is nonnegative, from the theorem quoted above, $g - f = q$ is convex. Furthermore, it is straightforward to establish that $r(x,y)$ is convex, which implies that $g$ is also convex. This means that $g$ is a dcd of $f$. If $b(x;y)$ is not nonnegative, then we know that $q(x,y)$ is not convex. This implies that $g - f$ is not convex, and so $g$ cannot be a dcd of $f$. $\square$

Unlike the quartic case, it is worth noting that in the quadratic case, it is easy to test whether a polynomial $g(x) = x^T Gx$ is a dcd of $f(x) = x^T Fx$. Indeed, this amounts to testing whether $F \succeq 0$ and $G - F \succeq 0$ which can be done in $O(n^3)$ time.

As mentioned earlier, there is not only one dcd for a given polynomial $f$, but an infinite number. Indeed, if $f = g - h$ with $g$ and $h$ convex then any convex polynomial $p$ generates a new dcd $f = (g + p) - (h + p)$. It is natural then to investigate if some dcds are better than others, e.g., for use in the convex-concave procedure.

Recall that the main idea of CCP is to upperbound the non-convex function $f = g - h$ by a convex function $f^k$. These convex functions are obtained by linearizing $h$ around the optimal solution of the previous iteration. Hence, a reasonable way of choosing a good dcd would be to look for dcds of $f$ that minimize the curvature of $h$ around a point. Two natural formulations of this problem are given below. The first one attempts to minimize the *average*[3] curvature of $h$ at a point $\bar{x}$ over all directions:

$$\begin{aligned} &\min_g \text{Tr } H_h(\bar{x}) \\ &\text{s.t. } f = g - h, g, h \text{ convex.} \end{aligned} \tag{4}$$

The second one attempts to minimize the *worst-case*[3] curvature of $h$ at a point $\bar{x}$ over all directions:

$$\begin{aligned} &\min_g \lambda_{\max} H_h(\bar{x}) \\ &\text{s.t. } f - g - h, g, h \text{ convex.} \end{aligned} \tag{5}$$

[3] Note that Tr $H_h(\bar{x})$ (resp. $\lambda_{\max} H_h(\bar{x})$) gives the average (resp. maximum) of $y^T H_h(\bar{x}) y$ over $\{y \mid ||y|| = 1\}$.

A few numerical experiments using these objective functions will be presented in Section 4.2.

Another popular notion that appears in the literature and that also relates to finding dcds with minimal curvature is that of *undominated dcds.* These were studied in depth by Bomze and Locatelli in the quadratic case [10]. We extend their definition to general polynomials here.

**Definition 2** Let $g$ be a dcd of $f$. A dcd $g'$ of $f$ is said to dominate $g$ if $g - g'$ is convex and nonaffine. A dcd $g$ of $f$ is *undominated* if no dcd of $f$ dominates $g$.

Arguments for chosing undominated dcds can be found in [10], [12, Section 3]. One motivation that is relevant to CCP appears in Proposition 2[4]. Essentially, the proposition shows that if we were to start at some initial point and apply one iteration of CCP, the iterate obtained using a dc decomposition $g$ would always beat an iterate obtained using a dcd dominated by $g$.

**Proposition 2** *Let $g$ and $g'$ be two dcds of $f$. Define the convex functions $h := g - f$ and $h' := g' - f$, and assume that $g'$ dominates $g$. For a point $x_0$ in $\mathbb{R}^n$, define the convexified versions of $f$*

$$f_g(x) := g(x) - (h(x_0) + \nabla h(x_0)^T(x - x_0)),$$
$$f_{g'}(x) := g'(x) - (h'(x_0) + \nabla h'(x_0)^T(x - x_0)).$$

*Then, we have*

$$f_g'(x) \leq f_g(x), \forall x.$$

*Proof* As $g'$ dominates $g$, there exists a nonaffine convex polynomial $c$ such that $c = g - g'$. We then have $g' = g - c$ and $h' = h - c$, and

$$\begin{aligned} f_g'(x) &= g(x) - c(x) - h(x_0) + c(x_0) - \nabla h(x_0)^T(x - x_0) + \nabla c(x_0)^T(x - x_0) \\ &= f_g(x) - (c(x) - c(x_0) - \nabla c(x_0)^T(x - x_0)). \end{aligned}$$

The first order characterization of convexity of $c$ then gives us

$$f_{g'}(x) \leq f_g(x), \forall x. \quad \square$$

In the quadratic case, it turns out that an optimal solution to (4) is an undominated dcd [10]. A solution given by (5) on the other hand is not necessarily undominated. Consider the quadratic function

$$f(x) = 8x_1^2 - 2x_2^2 - 8x_3^2$$

and assume that we want to decompose it using (5). An optimal solution is given by $g^*(x) = 8x_1^2 + 6x_2^2$ and $h^*(x) = 8x_2^2 + 8x_3^2$ with $\lambda_{\max} H_h = 8$. This is clearly dominated by $g'(x) = 8x_1^2$ as $g^*(x) - g'(x) = 6x_2^2$ which is convex.

[4] A variant of this proposition in the quadratic case appears in [10, Proposition 12].

When the degree is higher than 2, it is no longer true however that solving (4) returns an undominated dcd. Consider for example the degree-4 polynomial

$$f(x) = x^{12} - x^{10} + x^6 - x^4.$$

A solution to (4) with $\bar{x} = 0$ is given by $g(x) = x^{12} + x^6$ and $h(x) = x^{10} + x^4$ (as $\mathrm{Tr} H_h(0) = 0$). This is dominated by the dcd $g(x) = x^{12} - x^8 + x^6$ and $h(x) = x^{10} - x^8 + x^4$ as $g - g' = x^8$ is clearly convex.

It is unclear at this point how one can obtain an undominated dcd for higher degree polynomials, or even if one exists. In the next theorem, we show that such a dcd always exists and provide an optimization problem whose optimal solution(s) will always be undominated dcds. This optimization problem involves the integral of a polynomial over a sphere which conveniently turns out to be an explicit linear expression in its coefficients.

**Proposition 3 ([14])** *Let $S^{n-1}$ denote the unit sphere in $\mathbb{R}^n$. For a monomial $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$, define $\beta_j := \frac{1}{2}(\alpha_j + 1)$. Then*

$$\int_{S^{n-1}} x_1^{\alpha_1} \ldots x_n^{\alpha_n} d\sigma = \begin{cases} 0 & \text{if some } \alpha_j \text{ is odd,} \\ \frac{2\Gamma(\beta_1)\ldots\Gamma(\beta_n)}{\Gamma(\beta_1+\ldots+\beta_n)} & \text{if all } \alpha_j \text{ are even,} \end{cases}$$

where $\Gamma$ denotes the gamma function, and $\sigma$ is the rotation invariant probability measure on $S^{n-1}$.

**Theorem 1** *Let $f \in \tilde{H}_{n,2d}$. Consider the optimization problem*

$$\begin{aligned} \min_{g \in \tilde{H}_{n,2d}} \ & \frac{1}{\mathcal{A}_n} \int_{S^{n-1}} Tr\ H_g d\sigma \\ s.t.\ & g \text{ convex}, \\ & g - f \text{ convex}, \end{aligned} \tag{6}$$

*where $\mathcal{A}_n = \frac{2\pi^{n/2}}{\Gamma(n/2)}$ is a normalization constant which equals the area of $S^{n-1}$. Then, an optimal solution to (6) exists and any optimal solution is an undominated dcd of $f$.*

Note that problem (6) is exactly equivalent to (4) in the case where $n = 2$ and so can be seen as a generalization of the quadratic case.

*Proof* We first show that an optimal solution to (6) exists. As any polynomial $f$ admits a dcd, (6) is feasible. Let $\tilde{g}$ be a dcd of $f$ and define $\gamma := \int_{S^{n-1}} \mathrm{Tr}\ H_{\tilde{g}} d\sigma$. Consider the optimization problem given by (6) with the additional constraints:

$$\begin{aligned} \min_{g \in \tilde{H}_{n,2d}} \ & \frac{1}{\mathcal{A}_n} \int_{S^{n-1}} \mathrm{Tr}\ H_g d\sigma \\ \text{s.t. } & g \text{ convex and with no affine terms} \\ & g - f \text{ convex}, \\ & \int_{S^{n-1}} \mathrm{Tr}\ H_g d\sigma \leq \gamma. \end{aligned} \tag{7}$$

Notice that any optimal solution to (7) is an optimal solution to (6). Hence, it suffices to show that (7) has an optimal solution. Let $\mathcal{U}$ denote the feasible set of (7). Evidently, the set $\mathcal{U}$ is closed and $g \to \int_{S^{n-1}} \text{Tr } H_g d\sigma$ is continuous. If we also show that $\mathcal{U}$ is bounded, we will know that the optimal solution to (7) is achieved. To see this, assume that $\mathcal{U}$ is unbounded. Then for any $\beta$, there exists a coefficient $c_g$ of some $g \in \mathcal{U}$ that is larger than $\beta$. By absence of affine terms in $g$, $c_g$ features in an entry of $H_g$ as the coefficient of a nonzero monomial. Take $\bar{x} \in S^{n-1}$ such that this monomial evaluated at $\bar{x}$ is nonzero: this entails that at least one entry of $H_g(\bar{x})$ can get arbitrarily large. However, since $g \to \text{Tr } H_g$ is continuous and $\int_{S^{n-1}} \text{Tr } H_g d\sigma \leq \gamma$, $\exists \bar{\gamma}$ such that $\text{Tr } H_g(x) \leq \bar{\gamma}$, $\forall x \in S^{n-1}$. This, combined with the fact that $H_g(x) \succeq 0$ $\forall x$, implies that $||H_g(x)|| \leq \bar{\gamma}$, $\forall x \in S^{n-1}$, which contradicts the fact that an entry of $H_g(\bar{x})$ can get arbitrarily large.

We now show that if $g^*$ is any optimal solution to (6), then $g^*$ is an undominated dcd of $f$. Suppose that this is not the case. Then, there exists a dcd $g'$ of $f$ such that $g^* - g'$ is nonaffine and convex. As $g'$ is a dcd of $f$, $g'$ is feasible for (6). The fact that $g^* - g'$ is nonaffine and convex implies that

$$\int_{S^{n-1}} \text{Tr } H_{g^*-g'} d\sigma > 0 \Leftrightarrow \int_{S^{n-1}} \text{Tr} H_{g^*} d\sigma > \int_{S^{n-1}} \text{Tr } H_{g'} d\sigma,$$

which contradicts the assumption that $g^*$ is optimal to (6). □

Although optimization problem (6) is guaranteed to produce an undominated dcd, we show that unfortunately it is intractable to solve.

**Proposition 4** *Given an n-variate polynomial $f$ of degree 4 with rational coefficients, and a rational number $k$, it is strongly NP-hard to decide whether there exists a feasible solution to (6) with objective value $\leq k$.*

*Proof* We give a reduction from the problem of deciding convexity of quartic polynomials. Let $q$ be a quartic polynomial. We take $f = q$ and $k = \frac{1}{\mathcal{A}_n} \int_{S^{n-1}} \text{Tr } H_q(x)$. If $q$ is convex, then $g = q$ is trivially a dcd of $f$ and

$$\frac{1}{\mathcal{A}_n} \int_{S^{n-1}} \text{Tr } H_g d\sigma \leq k. \tag{8}$$

If $q$ is not convex, assume that there exists a feasible solution $g$ for (6) that satisfies (8). From (8) we have

$$\int_{S^{n-1}} \text{Tr } H_g(x) \leq \int_{S^{n-1}} \text{Tr } H_f d\sigma \Leftrightarrow \int_{S^{n-1}} \text{Tr } H_{f-g} d\sigma \geq 0. \tag{9}$$

But from (6), as $g - f$ is convex, $\int_{S^{n-1}} \text{Tr } H_{g-f} d\sigma \geq 0$. Together with (9), this implies that

$$\int_{S^{n-1}} \text{Tr } H_{g-f} d\sigma = 0$$

which in turn implies that $g - f = 0 \Leftrightarrow g = f$. This is not possible as $g$ is convex and $f$ is not. □

We remark that solving (6) in the quadratic case (i.e., $2d = 2$) is simply a semidefinite program.

## 3 Alegbraic relaxations and more tractable subsets of the set of convex polynomials

We have just seen in the previous section that for polynomials with degree as low as four, some basic tasks related to dc decomposition are computationally intractable. In this section, we identify three subsets of the set of convex polynomials that lend themselves to polynomial-time algorithms. These are the sets of *sos-convex, sdsos-convex*, and *dsos-convex* polynomials, which will respectively lead to semidefinite, second order cone, and linear programs. The latter two concepts are to our knowledge new and are meant to serve as more scalable alternatives to sos-convexity. All three concepts certify convexity of polynomials via explicit algebraic identities, which is the reason why we refer to them as algebraic relaxations.

### 3.1 DSOS-convexity, SDSOS-convexity, SOS-convexity

To present these three notions we need to introduce some notation and briefly review the concepts of sos, dsos, and sdsos polynomials.

We denote the set of nonnegative polynomials (resp. forms) in $n$ variables and of degree $d$ by $\tilde{PSD}_{n,d}$ (resp. $PSD_{n,d}$). A polynomial $p$ is a *sum of squares* (sos) if it can be written as $p(x) = \sum_{i=1}^{r} q_i^2(x)$ for some polynomials $q_1, \ldots, q_r$. The set of sos polynomials (resp. forms) in $n$ variables and of degree $d$ is denoted by $\tilde{SOS}_{n,d}$ (resp. $SOS_{n,d}$). We have the obvious inclusion $\tilde{SOS}_{n,d} \subseteq \tilde{PSD}_{n,d}$ (resp. $SOS_{n,d} \subseteq PSD_{n,d}$), which is strict unless $d = 2$, or $n = 1$, or $(n, d) = (2, 4)$ (resp. $d = 2$, or $n = 2$, or $(n, d) = (3, 4)$) [20], [38].

Let $\tilde{z}_{n,d}(x)$ (resp. $z_{n,d}(x)$) denote the vector of all monomials in $x = (x_1, \ldots, x_n)$ of degree up to (resp. exactly) $d$; the length of this vector is $\binom{n+d}{d}$ (resp. $\binom{n+d-1}{d}$). It is well known that a polynomial (resp. form) $p$ of degree $2d$ is sos if and only if it can be written as $p(x) = \tilde{z}_{n,d}^T(x) Q \tilde{z}_{n,d}(x)$ (resp. $p(x) = z_{n,d}^T(x) Q z_{n,d}(x)$), for some psd matrix $Q$ [35], [34]. The matrix $Q$ is generally called the Gram matrix of $p$. An SOS optimization problem is the problem of minimizing a linear function over the intersection of the convex cone $SOS_{n,d}$ with an affine subspace. The previous statement implies that SOS optimization problems can be cast as semidefinite programs.

We now define dsos and sdsos polynomials, which were recently proposed by Ahmadi and Majumdar [3], [2] as more tractable subsets of sos polynomials. When working with dc decompositions of $n$-variate polynomials, we will end up needing to impose sum of squares conditions on polynomials that have $2n$ variables (see Definition 5). While in theory the SDPs arising from sos conditions are of polynomial size, in practice we rather quickly face a scalability challenge. For this reason, we also consider the class of dsos and sdsos polynomials, which while more restrictive than sos polynomials, are considerably more tractable. For example, Table 2 in Section 4.2 shows that when $n = 14$, dc decompositions using these concepts are about 250 times faster than an sos-based approach. At $n = 18$ variables, we are unable to run the sos-based approach on our machine. With this motivation in mind, let us start by recalling some concepts from linear algebra.

**Definition 3** A symmetric matrix $M$ is said to be *diagonally dominant (dd)* if $m_{ii} \geq \sum_{j \neq i} |m_{ij}|$ for all $i$, and *strictly diagonally dominant* if $m_{ii} > \sum_{j \neq i} |m_{ij}|$ for all $i$. We say that $M$ is *scaled diagonally dominant (sdd)* if there exists a diagonal matrix $D$, with positive diagonal entries, such that $DAD$ is dd.

We have the following implications from Gershgorin's circle theorem

$$\text{M } dd \Rightarrow \text{M } sdd \Rightarrow \text{M } psd. \tag{10}$$

Furthermore, notice that requiring $M$ to be dd can be encoded via a linear program (LP) as the constraints are linear inequalities in the coefficients of $M$. Requiring that $M$ be sdd can be encoded via a second order cone program (SOCP). This follows from the fact that $M$ is sdd if and only if

$$M = \sum_{i<j} M^{ij}_{2\times 2},$$

where each $M^{ij}_{2\times 2}$ is an $n \times n$ symmetric matrix with zeros everywhere except four entries $M_{ii}, M_{ij}, M_{ji}, M_{jj}$ which must make the $2 \times 2$ matrix $\begin{pmatrix} M_{ii} & M_{ij} \\ M_{ji} & M_{jj} \end{pmatrix}$ symmetric positive semidefinite [3]. These constraints are *rotated quadratic cone constraints* and can be imposed via SOCP [7].

**Definition 4 ([3])** A polynomial $p \in \tilde{\mathcal{H}}_{n,2d}$ is said to be

- *diagonally-dominant-sum-of-squares (dsos)* if it admits a representation $p(x) = \tilde{z}^T_{n,d}(x) Q \tilde{z}_{n,d}(x)$, where $Q$ is a dd matrix.
- *scaled-diagonally-dominant-sum-of-squares (sdsos)* it it admits a representation $p(x) = \tilde{z}^T_{n,d}(x) Q \tilde{z}_{n,d}(x)$, where $Q$ is an sdd matrix.

Identical conditions involving $z_{n,d}$ instead of $\tilde{z}_{n,d}$ define the sets of dsos and sdsos forms.

The following implications are again straightforward:

$$p(x) \text{ dsos} \Rightarrow p(x) \text{ sdsos} \Rightarrow p(x) \text{ sos} \Rightarrow p(x) \text{ nonnegative}. \tag{11}$$

Given the fact that our Gram matrices and polynomials are related to each other via linear equalities, it should be clear that optimizing over the set of dsos (resp. sdsos, sos) polynomials is an LP (resp. SOCP, SDP).

Let us now get back to convexity.

**Definition 5** Let $y = (y_1, \ldots, y_n)^T$ be a vector of variables. A polynomial $p := p(x)$ is said to be

- *dsos-convex* if $y^T H_p(x) y$ is dsos.
- *sdsos-convex* if $y^T H_p(x) y$ is sdsos.

– *sos-convex* if $y^T H_p(x) y$ is sos.[5]

We denote the set of dsos-convex (resp. sdsos-convex, sos-convex, convex) forms in $\mathcal{H}_{n,2d}$ by $\Sigma_D C_{n,2d}$ (resp. $\Sigma_S C_{n,2d}$, $\Sigma C_{n,2d}$, $C_{n,2d}$). Similarly, $\tilde{\Sigma}_D C_{n,2d}$ (resp. $\tilde{\Sigma}_S C_{n,2d}$, $\tilde{\Sigma} C_{n,2d}$, $\tilde{C}_{n,2d}$) denote the set of dsos-convex (resp. sdsos-convex, sos-convex, convex) polynomials in $\tilde{\mathcal{H}}_{n,2d}$.

The following inclusions

$$\Sigma_D C_{n,2d} \subseteq \Sigma_S C_{n,2d} \subseteq \Sigma C_{n,2d} \subseteq C_{n,2d} \tag{12}$$

are a direct consequence of (11) and the second-order necessary and sufficient condition for convexity which reads

$$p(x) \text{ is convex } \Leftrightarrow H_p(x) \succeq 0, \forall x \in \mathbb{R}^n \Leftrightarrow y^T H_p(x) y \geq 0, \forall x, y \in \mathbb{R}^n.$$

Optimizing over $\Sigma_D C_{n,2d}$ (resp. $\Sigma_S C_{n,2d}$, $\Sigma C_{n,2d}$) is an LP (resp. SOCP, SDP). The same statements are true for $\tilde{\Sigma}_D C_{n,2d}$, $\tilde{\Sigma}_S C_{n,2d}$ and $\tilde{\Sigma} C_{n,2d}$.

Let us draw these sets for a parametric family of polynomials

$$p(x_1, x_2) = 2x_1^4 + 2x_2^4 + a x_1^3 x_2 + b x_1^2 x_2^2 + c x_1 x_2^3. \tag{13}$$

Here, $a, b$ and $c$ are parameters. It is known that for bivariate quartics, all convex polynomials are sos-convex; i.e., $\Sigma C_{2,4} = C_{2,4}$.[6] To obtain Figure 1, we fix $c$ to some value and then plot the values of $a$ and $b$ for which $p(x_1, x_2)$ is s/d/sos-convex. As we can see, the quality of the inner approximation of the set of convex polynomials by the sets of dsos/sdsos-convex polynomials can be very good (e.g., $c = 0$) or less so (e.g., $c = 1$).

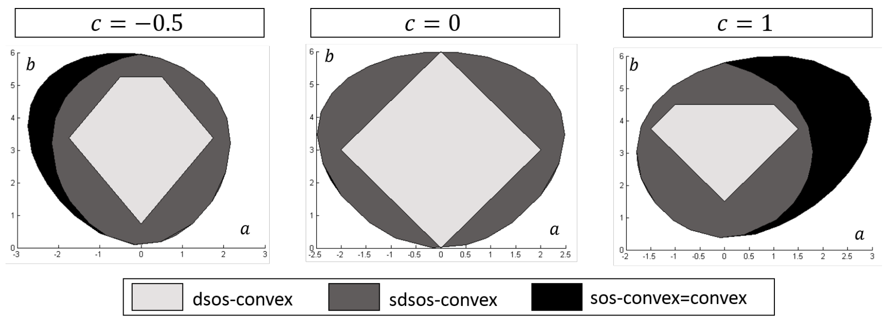

Fig. 1: The sets $\Sigma_D C_{n,2d}, \Sigma_S C_{n,2d}$ and $\Sigma C_{n,2d}$ for the parametric family of polynomials in (13)

[5] The notion of sos-convexity has already appeared in the study of semidefinite representability of convex sets [19] and in applications such as shaped-constrained regression in statistics [32].

[6] In general, constructing polynomials that are convex but not sos-convex seems to be a nontrivial task [5]. A complete characterization of the dimensions and degrees for which convexity and sos-convexity are equivalent is given in [6].

## 3.2 Existence of difference of s/d/sos-convex decompositions of polynomials

The reason we introduced the notions of s/d/sos-convexity is that in our optimization problems for finding dcds, we would like to replace the condition

$$f = g - h, \;\; g, h \text{ convex}$$

with the computationally tractable condition

$$f = g - h, \;\; g, h \text{ s/d/sos-convex.}$$

The first question that needs to be addressed is whether for any polynomial such a decomposition exists. In this section, we prove that this is indeed the case. This in particular implies that a dcd can be found efficiently.

We start by proving a lemma about cones.

**Lemma 1** *Consider a vector space $E$ and a full-dimensional cone $K \subseteq E$. Then, any $v \in E$ can be written as $v = k_1 - k_2$, where $k_1, k_2 \in K$.*

*Proof* Let $v \in E$. If $v \in K$, then we take $k_1 = v$ and $k_2 = 0$. Assume now that $v \notin K$ and let $k$ be any element in the interior of the cone $K$. As $k \in int(K)$, there exists $0 < \alpha < 1$ such that $k' := (1 - \alpha)v + \alpha k \in K$. Rewriting the previous equation, we obtain

$$v = \frac{1}{1-\alpha}k' - \frac{\alpha}{1-\alpha}k.$$

By taking $k_1 := \frac{1}{1-\alpha}k'$ and $k_2 := \frac{\alpha}{1-\alpha}k$, we observe that $v = k_1 - k_2$ and $k_1, k_2 \in K$. □

The following theorem is the main result of the section.

**Theorem 2** *Any polynomal $p \in \tilde{\mathcal{H}}_{n,2d}$ can be written as the difference of two dsos-convex polynomials in $\tilde{\mathcal{H}}_{n,2d}$.*

**Corollary 1** *Any polynomial $p \in \tilde{\mathcal{H}}_{n,2d}$ can be written as the difference of two sdsos-convex, sos-convex, or convex polynomials in $\tilde{\mathcal{H}}_{n,2d}$.*

*Proof* This is straightforward from the inclusions

$$\tilde{\Sigma}_D C_{n,2d} \subseteq \tilde{\Sigma}_S C_{n,2d} \subseteq \tilde{\Sigma} C_{n,2d} \subseteq \tilde{C}_{n,2d}. \;\; \square$$

In view of Lemma 1, it suffices to show that $\tilde{\Sigma}_D C_{n,2d}$ is full dimensional in the vector space $\tilde{\mathcal{H}}_{n,2d}$ to prove Theorem 2. We do this by constructing a polynomial in $int(\tilde{\Sigma}_D C_{n,2d})$ for any $n, d$.

Recall that $z_{n,2d}$ (resp. $\tilde{z}_{n,2d}$) denotes the vector of all monomials in $x = (x_1, \ldots, x_n)$ of degree exactly (resp. up to) $d$. If $y = (y_1, \ldots, y_n)$ is a vector of variables of length $n$, we define

$$w_{n,d}(x, y) := y \cdot z_{n,d}(x),$$

where $y \cdot z_{n,d}(x) = (y_1 z_{n,d}(x), \ldots, y_n z_{n,d}(x))^T$. Analogously, we define

$$\tilde{w}_{n,d}(x, y) := y \cdot \tilde{z}_{n,d}(x).$$

**Theorem 3** *For all $n, d$, there exists a polynomial $p \in \tilde{\mathcal{H}}_{n,2d}$ such that*

$$y^T H_p(x) y = \tilde{w}_{n,d-1}^T(x,y) Q \tilde{w}_{n,d-1}(x,y), \tag{14}$$

*where $Q$ is strictly dd.*

Any such polynomial will be in $int(\tilde{\Sigma}_D C_{n,2d})$. Indeed, if we were to pertub the coefficients of $p$ slightly, then each coefficient of $Q$ would undergo a slight perturbation. As $Q$ is strictly dd, $Q$ would remain dd, and hence $p$ would remain dsos-convex.

We will prove Theorem 3 through a series of lemmas. First, we show that this is true in the homogeneous case and when $n = 2$ (Lemma 2). By induction, we prove that this result still holds in the homogeneous case for any $n$ (Lemma 3). We then extend this result to the nonhomogeneous case.

**Lemma 2** *For all $d$, there exists a polynomial $p \in \tilde{\mathcal{H}}_{2,2d}$ such that*

$$y^T H_p(x) y = w_{2,d-1}^T(x,y) Q w_{2,d-1}(x,y), \tag{15}$$

*for some strictly dd matrix $Q$.*

*Proof* First, if $2d = 2$, we simply take $p(x_1, x_2) = x_1^2 + x_2^2$ as $y^T H_p(x) y = 2y^T I y$. Now, assume $2d > 2$. We consider two cases depending on whether $d$ is divisible by 2.

In the case that it is, we construct $p$ as

$$p(x_1,x_2) := a_0 x_1^{2d} + a_1 x_1^{2d-2} x_2^2 + a_2 x_1^{2d-4} x_2^4 + \ldots + a_{d/2} x_1^d x_2^d + \ldots + a_1 x_1^2 x_2^{2d-2} + a_0 x_2^{2d},$$

with the sequence $\{a_k\}_{k=0,\ldots,\frac{d}{2}}$ defined as follows

$$\begin{aligned} a_1 &= 1 \\ a_{k+1} &= \left(\frac{2d-2k}{2k+2}\right) a_k,\ k = 1, \ldots, \frac{d}{2} - 1 \text{ (for } 2d > 4) \\ a_0 &= \frac{1}{d} + \frac{d}{2(2d-1)} a_{\frac{d}{2}}. \end{aligned} \tag{16}$$

Let

$$\begin{aligned} \beta_k &= a_k(2d-2k)(2d-2k-1), k = 0, \ldots, \frac{d}{2} - 1, \\ \gamma_k &= a_k \cdot 2k(2k-1), k = 1, \ldots, \frac{d}{2}, \\ \delta_k &= a_k(2d-2k) \cdot 2k, k = 1, \ldots, \frac{d}{2}. \end{aligned} \tag{17}$$

We claim that the matrix $Q$ defined as

$$\left(\begin{array}{c|c}
\begin{array}{cccccc} \beta_0 &&&&& \\ & \ddots &&&& \\ && \beta_k &&& \\ &&& \ddots && \\ &&&& \beta_{\frac{d}{2}-2} & \\ &&&&& \beta_{\frac{d}{2}-1} \end{array}
&
\begin{array}{cccccccc} 0 & \delta_1 &&&&&& \delta_{\frac{d}{2}} \\ & \ddots & \ddots &&&&& \\ && 0 & \delta_{k+1} &&&& \\ &&& \ddots & \ddots &&& \\ &&&& \ddots & \delta_{\frac{d}{2}-1} && \\ &&&&& 0 & 0 & \end{array}
\\ \hline
\begin{array}{ccccc} \gamma_{\frac{d}{2}} &&&& \\ & \ddots &&& \\ && \gamma_k && \\ &&& \ddots & \\ &&&& \gamma_1 \end{array}
&
\begin{array}{cccccc} 0 & \delta_{\frac{d}{2}-1} &&&& \\ & \ddots & \ddots &&& \\ && 0 & \delta_{k-1} && \\ &&& \ddots & \ddots & \\ &&&& 0 & \end{array}
\\ \hline\hline
\begin{array}{cccccc} 0 &&&&& \\ \ddots & \ddots &&&& \\ & \delta_{k-1} & 0 &&& \\ && \ddots & \ddots && \\ &&& \delta_{\frac{d}{2}-1} & 0 & \end{array}
&
\begin{array}{ccccc} \gamma_1 &&&& \\ & \ddots &&& \\ && \gamma_k && \\ &&& \ddots & \\ &&&& \gamma_{\frac{d}{2}} \end{array}
\\ \hline
\begin{array}{cccccccc} && 0 & 0 &&&& \\ &&& \delta_{\frac{d}{2}-1} & 0 &&& \\ &&&& \ddots & \ddots && \\ &&&&& \delta_{k+1} & 0 & \\ &&&&&& \ddots & \ddots \\ \delta_{\frac{d}{2}} &&&&&& \delta_1 & 0 \end{array}
&
\begin{array}{cccccc} \beta_{\frac{d}{2}-1} &&&&& \\ & \beta_{\frac{d}{2}-2} &&&& \\ && \ddots &&& \\ &&& \beta_k && \\ &&&& \ddots & \\ &&&&& \beta_0 \end{array}
\end{array}\right)$$

is strictly dd and satisfies (15) with $w_{2,d-1}(x,y)$ ordered as

$$\left(y_1x_1^{d-1}, y_1x_1^{d-2}x_2, \ldots, y_1x_1x_2^{d-2}, y_1x_2^{d-1}, y_2x_1^{d-1}, y_2x_1^{d-2}x_2, \ldots, y_2x_1x_2^{d-2}, y_2x_2^{d-1}\right)^T.$$

To show (15), one can derive the Hessian of $p$, expand both sides of the equation, and verify equality. To ensure that the matrix is strictly dd, we want all diagonal coefficients to be strictly greater than the sum of the elements on the row. This translates to the following inequalities

$$\begin{aligned}
\beta_0 &> \delta_1 + \delta_{\frac{d}{2}} \\
\beta_k &> \delta_{k+1}, \forall k = 1, \ldots, \frac{d}{2} - 2 \\
\beta_{\frac{d}{2}-1} &> 0, \gamma_1 > 0 \\
\gamma_{k+1} &> \delta_k, \forall k = 1, \ldots, \frac{d}{2} - 1.
\end{aligned}$$

Replacing the expressions of $\beta_k, \gamma_k$ and $\delta_k$ in the previous inequalities using (17) and the values of $a_k$ given in (16), one can easily check that these inequalities are satisfied.

We now consider the case where $d$ is not divisable by 2 and take

$$p(x_1,x_2) := a_0 x_1^{2d} + a_1 x_1^{2d-2}x_2^2 + \ldots + a_{(d-1)/2} x_1^{d+1} x_2^{d-1} + a_{(d-1)/2} x_1^{d-1} x_2^{d+1} + \ldots + a_1 x_1^2 x_2^{2d-2} + a_0 x_2^{2d},$$

with the sequence $\{a_k\}_{k=0,\ldots,\frac{d-1}{2}}$ defined as follows

$$\begin{aligned} a_1 &= 1 \\ a_{k+1} &= \left(\frac{2d-2k}{2k+2}\right) a_k,\ k=1,\ldots,\frac{d-3}{2} \\ a_0 &= 1 + \frac{2(2d-2)}{2d(2d-1)}. \end{aligned} \tag{18}$$

Again, we want to show existence of a strictly dd matrix $Q$ that satisfies (15). Without changing the definitions of the sequences $\{\beta_k\}_{k=1,\ldots,\frac{d-3}{2}}$, $\{\gamma_k\}_{k=1,\ldots,\frac{d-1}{2}}$ and $\{\delta_k\}_{k=1,\ldots,\frac{d-1}{2}}$, we claim this time that the matrix $Q$ defined as

$$\left(\begin{array}{cccccccccc|cccccccccc}
\beta_0 &&&&&&&&&& 0 & \delta_1 &&&&&&&& \mathbf{0} \\
& \ddots &&&&&&&&&& \ddots & \ddots &&&&&&& \\
&& \beta_k &&&&&&&&&& 0 & \delta_{k+1} &&&&&& \\
&&& \ddots &&&&&&&&&& \ddots & \ddots &&&&& \\
&&&& \beta_{\frac{d-3}{2}} &&&&&&&&&& 0 & \delta_{\frac{d-1}{2}} &&&& \\
\hline
&&&&& \gamma_{\frac{d-1}{2}} &&&&&&&&&& 0 & \delta_{\frac{d-1}{2}-1} &&& \\
&&&&&& \ddots &&&&&&&&&& \ddots & \ddots && \\
&&&&&&& \gamma_k &&&&&&&&&& 0 & \delta_{k-1} & \\
&&&&&&&& \ddots &&&&&&&&&& \ddots & \ddots \\
&&&&&&&&& \gamma_1 &&&&&&&&&& 0 \\
\hline\hline
0 &&&&&&&&&& \gamma_1 &&&&&&&&& \\
\ddots & \ddots &&&&&&&&&& \ddots &&&&&&&& \\
& \delta_{k-1} & 0 &&&&&&&&&& \gamma_k &&&&&&& \\
&& \ddots & \ddots &&&&&&&&&& \ddots &&&&&& \\
&&& \delta_{\frac{d-1}{2}-1} & 0 &&&&&&&&&& \gamma_{\frac{d-1}{2}} &&&&& \\
\hline
&&&& \delta_{\frac{d-1}{2}} & 0 &&&&&&&&&& \beta_{\frac{d-3}{2}} &&&& \\
&&&&& \ddots & \ddots &&&&&&&&&& \ddots &&& \\
&&&&&& \delta_{k+1} & 0 &&&&&&&&&& \beta_k && \\
&&&&&&& \ddots & \ddots &&&&&&&&&& \ddots & \\
\mathbf{0} &&&&&&&& \delta_1 & 0 &&&&&&&&&& \beta_0
\end{array}\right)$$

satisfies (15) and is strictly dd. Showing (15) amounts to deriving the Hessian of $p$ and checking that the equality is verified. To ensure that $Q$ is strictly dd, the inequalities that now must be verified are

$$\begin{aligned} &\beta_k > \delta_{k+1}, \forall k = 0,\ldots,\frac{d-1}{2}-1 \\ &\gamma_k > \delta_{k-1}, \forall k = 2,\ldots,\frac{d-1}{2} \\ &\gamma_1 > 0. \end{aligned}$$

These inequalities can all be shown to hold using (18). □

**Lemma 3** *For all $n, d$, there exists a form $p_{n,2d} \in \mathcal{H}_{n,2d}$ such that*

$$y^T H_{p_{n,2d}}(x) y = w_{n,d-1}^T(x,y) Q_{p_{n,2d}} w_{n,d-1}(x,y)$$

*and $Q_{p_{n,2d}}$ is a strictly dd matrix.*

*Proof* We proceed by induction on $n$ with fixed and arbitrary $d$. The property is verified for $n = 2$ by Lemma 2. Suppose that there exists a form $p_{n,2d} \in \mathcal{H}_{n,2d}$ such that

$$y^T H_{p_{n,2d}} y = w_{n,d-1}^T(x,y) Q_{p_{n,2d}} w_{n,d-1}(x,y), \tag{19}$$

for some strictly dd matrix $Q_{p_{n,2d}}$. We now show that

$$p_{n+1,2d} := q + \alpha v$$

with

$$\begin{aligned} q &:= \sum_{\{i_1,\ldots,i_n\} \in \{1,\ldots,n+1\}^n} p_{n,2d}(x_{i_1}, \ldots, x_{i_n}) \\ v &:= \sum_{2i_1+\ldots 2i_{n+1}=2d, i_1,\ldots,i_{n+1}>0} x_1^{2i_1} x_2^{2i_2} \ldots x_{n+1}^{2i_{n+1}}, \end{aligned} \tag{20}$$

and $\alpha > 0$ small enough, verifies

$$y^T H_{p_{n+1,2d}} y = w_{n+1,d-1}^T(x,y) Q_{p_{n+1,2d}} w_{n+1,d-1}(x,y), \tag{21}$$

for some strictly dd matrix $Q_{p_{n+1,2d}}$. Equation (21) will actually be proved using an equivalent formulation that we describe now. Let $\hat{w}_n$ (resp. $\hat{w}_{n+1}$) be a vector containing all monomials from $w_{n+1,d-1}$ that include up to $n$ variables (resp. exactly $n+1$ variables) in $x = (x_1, \ldots, x_n)$. Obviously, $w_{n+1,d-1}$ is equal to

$$\hat{w} := \begin{pmatrix} \hat{w}_n \\ \hat{w}_{n+1} \end{pmatrix}$$

up to a permutation of its entries. If we show that there exists a strictly dd matrix $\hat{Q}$ such that

$$y^T H_{p_{n+1,2d}}(x) y = \hat{w}^T(x,y) \hat{Q} \hat{w}(x,y) \tag{22}$$

then (21) will hold, as the rows of $Q_{p_{n+1,2d}}$ will be a permutation of the rows of $\hat{Q}$.

We claim the following:

- there exists a strictly dd matrix $\hat{Q}_q$ such that

$$y^T H_q(x) y = \begin{pmatrix} \hat{w}_n \\ \hat{w}_{n+1} \end{pmatrix}^T \begin{pmatrix} \hat{Q}_q & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{w}_n \\ \hat{w}_{n+1} \end{pmatrix}. \tag{23}$$

– there exist a symmetric matrix $\hat{Q}_v$, and $q_1, \ldots, q_m > 0$ (where $m$ is the length of $\hat{w}_{n+1}$) such that

$$y^T H_v(x) y = \begin{pmatrix} \hat{w}_n \\ \hat{w}_{n+1} \end{pmatrix}^T \begin{pmatrix} \hat{Q}_v & 0 \\ 0 & \operatorname{diag}(q_1, \ldots, q_m) \end{pmatrix} \begin{pmatrix} \hat{w}_n \\ \hat{w}_{n+1} \end{pmatrix}. \tag{24}$$

Using these two claims and linearity of the Hessian, we have

$$\hat{Q} = \begin{pmatrix} \hat{Q}_q + \alpha \hat{Q}_v & 0 \\ 0 & \alpha \operatorname{diag}(q_1, \ldots, q_m) \end{pmatrix}.$$

As $\hat{Q}_q$ is strictly dd, we can pick $\alpha > 0$ small enough such that $\hat{Q}_q + \alpha \hat{Q}_v$ is strictly dd. This entails that $\hat{Q}$ is strictly dd, and (22) holds.

We now give a proof of the two claims.

To prove the first claim, notice that if we denote by

$$M = \bigcup_{(i_1, \ldots, i_n) \in \{1, \ldots, n+1\}^n} \{\text{monomials in } w_{n,d-1}(x_{i_1}, \ldots, x_{i_n}, y)\},$$

then $M$ is exactly equal to $\hat{M} = \{\text{monomials in } \hat{w}_n(x, y)\}$.

The fact that $M \subseteq \hat{M}$ shows that there exists some $\hat{Q}_q$ such that $y^T H_q y = \hat{w}_n^T \hat{Q}_q \hat{w}_n$. Furthermore, it implies that each row of $\hat{Q}_q$ is a sum of rows of $Q_{p_{n,2d}(x_{i_1}, \ldots, x_{i_n})}$ (padded out with zeros). The fact that $\hat{M} \subseteq M$ shows that each row of $\hat{Q}_q$ contains at least one nonzero element. These last two facts combined with the induction hypothesis imply that $\hat{Q}_q$ is strictly dd.

We now prove the second claim. Let $I := \{i_1, \ldots, i_n \mid i_1 + \ldots + i_{n+1} = d, i_1, \ldots, i_{n+1} > 0\}$ and $\hat{w}_{n+1}^i$ be the $i^{th}$ element of $\hat{w}_{n+1}$. To prove (24), we need to show that

$$\begin{aligned} y^T H_v(x) y = & \sum_{i_1, \ldots, i_n \in I} \sum_{k=1}^{n+1} 2 i_k (2 i_k - 1) x_1^{2 i_1} \ldots x_k^{2 i_k - 2} \ldots x_{n+1}^{2 i_{n+1}} y_k^2 \\ & + 4 \sum_{i_1, \ldots, i_m \in I} \sum_{j \neq k} i_k i_j x_1^{2 i_1} \ldots x_j^{2 i_j - 1} \ldots x_k^{2 i_k - 1} \ldots x_{n+1}^{2 i_{n+1}} y_j y_k. \end{aligned} \tag{25}$$

can equal

$$\hat{w}_n^T(x, y) \hat{Q}_v \hat{w}_n^T(x, y) + \sum_{i=1}^m q_i (\hat{w}_{n+1}^i)^2 \tag{26}$$

for some symmetric matrix $\hat{Q}_v$ and positive scalars $q_1, \ldots, q_m$. We first argue that all monomials contained in $y^T H_v(x) y$ appear in the expansion (26). This means that we do not need to use any other entry of the Gram matrix in (24). It is clear that any monomial in the first term of (25) will feature in (26) via the diagonal terms of the matrix $\hat{Q}_v$ and the entries $q_1, \ldots, q_m$. Furthermore, it is not hard to see from the structure of $v$ in (20) that each $q_i > 0$.

Consider now any monomial contained in the second term of (25). We claim that it can be written as the product of two monomials $m'$ and $m''$ with $n$ or less variables in $x$. Indeed, at least two variables in the monomial must have degree less than $d-1$. Placing one variable in $m'$ and the other variable in $m''$ and then filling up $m'$ and $m''$ with the remaining variables (in any fashion as long as the degrees at $m'$ and $m''$ equal $d-1$) yields the desired result. This entails that the monomials in the second term of (25) appear in the first term of expansion (26). □

*Proof (of Theorem 3)* Let $p_{n,2k} \in \mathcal{H}_{n,2k}$ be the form constructed in the proof of Lemma 3 which is in the interior of $\Sigma_D C_{n,2k}$. Let $Q_k$ denote the strictly diagonally dominant matrix which was constructed to satisfy

$$y^T H_{p_{n,2k}} y = w_{n,2k}^T(x,y) Q_k w_{n,2k}.$$

To prove Theorem 3, we take

$$p := \sum_{k=1}^{d} p_{n,2k} \in \tilde{H}_{n,2d}.$$

We have

$$\begin{aligned} y^T H_p(x) y &= \begin{pmatrix} w_{n,1}(x,y) \\ \vdots \\ w_{n,d-1}(x,y) \end{pmatrix}^T \begin{pmatrix} Q_1 & & \\ & \ddots & \\ & & Q_d \end{pmatrix} \begin{pmatrix} w_{n,1}(x,y) \\ \vdots \\ w_{n,d-1}(x,y) \end{pmatrix} \\ &= \tilde{w}_{n,d-1}(x,y)^T Q \tilde{w}_{n,d-1}(x,y). \end{aligned}$$

We observe that $Q$ is strictly dd, which shows that $p \in int(\tilde{\Sigma}_D C_{n,2d})$. □

*Remark 1* If we had only been interested in showing that any polynomial in $\tilde{H}_{n,2d}$ could be written as a difference of two sos-convex polynomials, this could have been easily done by showing that $p(x) = \left(\sum_i x_i^2\right)^d \in int(\Sigma C_{n,2d})$. However, this form is not dsos-convex or sdsos-convex for all $n, d$ (e.g., for $n = 3$ and $2d = 8$). We have been unable to find a simpler proof for existence of sdsos-convex dcds that does not go through the proof of existence of dsos-convex dcds.

*Remark 2* If we solve problem (6) with the convexity constraint replaced by a dsos-convexity (resp. sdsos-convexity, sos-convexity) requirement, the same arguments used in the proof of Theorem 1 now imply that the optimal solution $g^*$ is not dominated by any dsos-convex (resp. sdsos-convex, sos-convex) decomposition.

## 4 Numerical results

In this section, we present a few numerical results to show how our algebraic decomposition techniques affect the convex-concave procedure. The objective function $p \in \tilde{\mathcal{H}}_{n,2d}$ in all of our experiments is generated randomly following the ensemble of [36, Section 5.1.]. This means that

$$p(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i^{2d} + g(x_1, \ldots, x_n),$$

where $g$ is a random polynomial of total degree $\leq 2d-1$ whose coefficients are random integers uniformly sampled from $[-30, 30]$. An advantage of polynomials generated in this fashion is that they are bounded below and that their minimum $p^*$ is achieved over $\mathbb{R}^n$. The starting point of CCP was generated randomly from a zero-mean Gaussian distribution.

One nice feature of our decomposition techniques is that all the polynomials $f_i^k, i = 0, \ldots, m$ in line 4 of Algorithm 1 in the introduction are sos-convex. This allows us to solve the convex subroutine of CCP exactly via a single SDP [26, Remark 3.4.], [28, Corollary 2.3.]:

$$\begin{aligned} &\min \gamma \\ &\text{s.t. } f_0^k - \gamma = \sigma_0 + \sum_{j=1}^{m} \lambda_j f_j^k \\ &\sigma_0 \text{ sos}, \;\; \lambda_j \geq 0, j = 1, \ldots, m. \end{aligned} \tag{27}$$

The degree of $\sigma_0$ here is taken to be the maximum degree of $f_0^k, \ldots, f_m^k$. We could have also solved these subproblems using standard descent algorithms for convex optimization. However, we are not so concerned with the method used to solve this convex problem as it is the same for all experiments. All of our numerical examples were done using MATLAB, the polynomial optimization library SPOT [33], and the solver MOSEK [1].

## 4.1 Picking a good dc decomposition for CCP

In this subsection, we consider the problem of minimizing a random polynomial $f_0 \in \tilde{\mathcal{H}}_{8,4}$ over a ball of radius $R$, where $R$ is a random integer in $[20, 50]$. The goal is to compare the impact of the dc decomposition of the objective on the performance of CCP. To monitor this, we decompose the objective in 4 different ways and then run CCP using the resulting decompositions. These decompositions are obtained through different SDPs that are listed in Table 1.

| Feasibility | $\lambda_{\max} H_h(x_0)$ | $\lambda_{\max,B} H_h$ | Undominated |
|---|---|---|---|
| $\min 0$ <br> s.t. $f_0 = g - h$, <br> $g, h$ sos-convex | $\min t$ <br> s.t. $f_0 = g - h$, <br> $g, h$ sos-convex <br> $tI - H_h(x_0) \succeq 0$ | $\min_{g,h} t$ <br> s.t. $f_0 = g - h$, <br> $g, h$ sos-convex <br> $y^T(tI - H_h(x) + f_1\tau(x))y$ sos <br> $y^T\tau(x)y$ [7] sos | $\min \frac{1}{\mathcal{A}_n} \int \mathrm{Tr} H_g d\sigma$ <br> s.t. $f_0 = g - h$, <br> $g, h$ sos-convex |

Table 1: Different decomposition techniques using sos optimization

The first SDP in Table 1 is simply a feasibility problem. The second SDP minimizes the largest eigenvalue of $H_h$ at the initial point $x_0$ inputed to CCP. The third minimizes the largest eigenvalue of $H_h$ over the ball $B$ of radius $R$. Indeed, let $f_1 := \sum_i x_i^2 - R^2$. Notice that $\tau(x) \succeq 0, \forall x$ and if $x \in B$, then $f_1(x) \leq 0$. This implies that $tI \succeq H_h(x), \forall x \in B$. The fourth SDP searches for an undominated dcd.

Once $f_0$ has been decomposed, we start CCP. After 4 mins of total runtime, the program is stopped and we recover the objective value of the last iteration. This procedure is repeated on 30 random instances of $f_0$ and $R$, and the average of the results is presented in Figure 2. From the figure, we can see

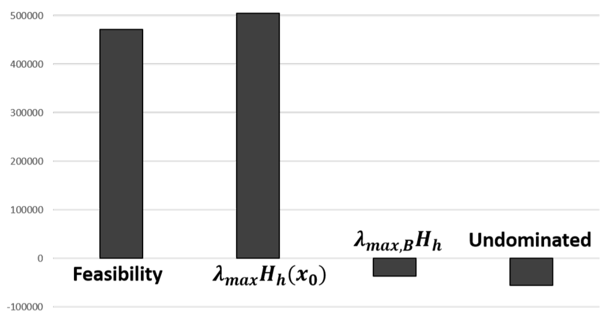

Fig. 2: Impact of choosing a good dcd on CCP ($n = 8, 2d = 4$)

that the choice of the initial decomposition impacts the performance of CCP considerably, with the region formulation of $\lambda_{\max}$ and the undominated decomposition giving much better results than the other two. It is worth noting that all formulations have gone through roughly the same number of iterations of CCP (approx. 400). Furthermore, these results seem to confirm that it is best to pick an undominated decomposition when applying CCP.

## 4.2 Scalibility of s/d/sos-convex dcds and the multiple decomposition CCP

While solving the last optimization problem in Table 1 usually gives very good results, it relies on an sos-convex dc decomposition. However, this choice is only reasonable in cases where the number of variables and the degree of the polynomial that we want to decompose are low. When these become too high, obtaining an sos-convex dcd can be too time-consuming. The concepts of dsos-convexity and sdsos-convexity then become interesting alternatives to sos-convexity. This is illustrated in Table 2, where we have reported the time

[7] Here, $\tau(x)$ is an $n \times n$ matrix where each entry is in $\tilde{H}_{n,2d-4}$

taken to solve the following decomposition problem:

$$\begin{aligned} &\min \frac{1}{\mathcal{A}_n} \int_{S^{n-1}} \text{Tr } H_g d\sigma \\ &\text{s.t. } f = g - h, g, h \text{ s/d/sos-convex} \end{aligned} \tag{28}$$

In this case, $f$ is a random polynomial of degree 4 in $n$ variables. We also report the optimal value of (28) (we know that (28) is always guaranteed to be feasible from Theorem 3). Notice that for $n = 18$, it takes over 30 hours to

| | n=6 | | n=10 | | n=14 | | n=18 | |
|---|---|---|---|---|---|---|---|---|
| | Time | Value | Time | Value | Time | Value | Time | Value |
| dsos-convex | $< 1s$ | 62090 | <1s | 168481 | 2.33s | 136427 | 6.91s | 48457 |
| sdsos-convex | $< 1s$ | 53557 | 1.11 s | 132376 | 3.89s | 99667 | 12.16s | 32875 |
| sos-convex | $< 1s$ | 11602 | 44.42s | 18346 | 800.16s | 9828 | 30hrs+ | —— |

Table 2: Time and optimal value obtained when solving (28)

obtain an sos-convex decomposition, whereas the run times for s/dsos-convex decompositions are still in the range of 10 seconds. This increased speed comes at a price, namely the quality of the decomposition. For example, when $n = 10$, the optimal value obtained using sos-convexity is nearly 10 times lower than that of sdsos-convexity.

Now that we have a better quantitative understanding of this tradeoff, we propose a modification to CCP that leverages the speed of s/dsos-convex dcds for large $n$. The idea is to modify CCP in such a way that one would compute a new s/dsos-convex decomposition of the functions $f_i$ after each iteration. Instead of looking for dcds that would provide good global decompositions (such as undominated sos-convex dcds), we look for decompositions that perform well locally. From Section 2, candidate decomposition techniques for this task can come from formulations (4) and (5) that minimize the maximum eigenvalue of the Hessian of $h$ at a point or the trace of the Hessian of $h$ at a point. This modified version of CCP is described in detail in Algorithm 2. We will refer to it as *multiple decomposition CCP*.

We compare the performance of CCP and multiple decomposition CCP on the problem of minimizing a polynomial $f$ of degree 4 in $n$ variables, for varying values of $n$. In Figure 3, we present the optimal value (averaged over 30 instances) obtained after 4 mins of total runtime. The "SDSOS" columns correspond to multiple decomposition CCP (Algorithm 2) with sdsos-convex decompositions at each iteration. The "SOS" columns correspond to classical CCP where the first and only decomposition is an undominated sos-convex dcd. From Figure 2, we know that this formulation performs well for small values of $n$. This is still the case here for $n = 8$ and $n = 10$. However, this approach performs poorly for $n = 12$ as the time taken to compute the initial decomposition is too long. In contrast, multiple decomposition CCP combined with sdsos-convex decompositions does slightly worse for $n = 8$ and $n = 10$, but significantly better for $n = 12$.

**Algorithm 2** Multiple decomposition CCP ($\lambda_{\max}$ version)

```
Input: $x_0,\ f_i, i = 0, \dots, m$
1: $k \leftarrow 0$
2: while stopping criterion not satisfied do
3:     Decompose: $\forall i$ find $g_i^k, h_i^k$ s/d/sos-convex that min. $t$, s.t. $tI - H_{h_i^k}(x_k)$ s/dd[8] and $f_i = g_i^k - h_i^k$
4:     Convexify: $f_i^k(x) := g_i^k(x) - (h_i^k(x_k) + \nabla h_i^k(x_k)^T(x - x_k)),\ i = 0, \dots, m$
5:     Solve convex subroutine: $\min f_0^k(x)$, s.t. $f_i^k(x) \leq 0, i = 1, \dots, m$
6:     $x_{k+1} := \underset{f_i^k(x) \leq 0}{\operatorname{argmin}} f_0^k(x)$
7:     $k \leftarrow k + 1$
8: end while
Output: $x_k$
```

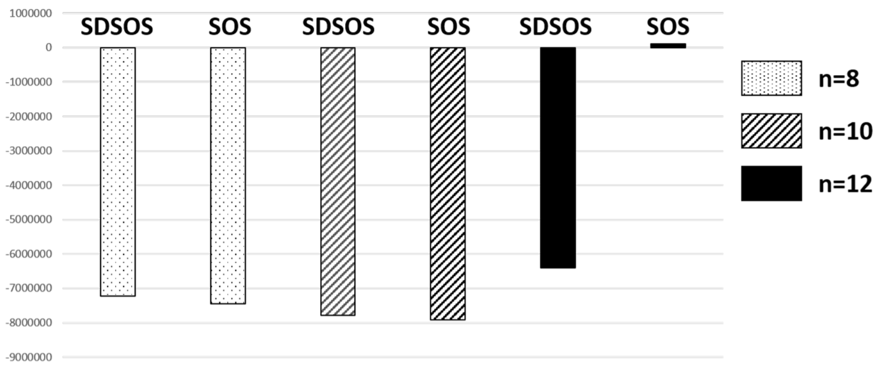

Fig. 3: Comparing multiple decomposition CCP using sdsos-convex decompositions against CCP with a single undominated sos-convex decomposition

In conclusion, our overall observation is that picking a good dc decomposition noticeably affects the perfomance of CCP. While optimizing over all dc decompositions is intractable for polynomials of degree greater or equal to 4, the algebraic notions of sos-convexity, sdsos-convexity and dsos-convexity can provide valuable relaxations. The choice among these options depends on the number of variables and the degree of the polynomial at hand. Though these classes of polynomials only constitute subsets of the set of convex polynomials, we have shown that even the smallest subset of the three contains dcds for any polynomial.

**Acknowledgements** We would like to thank Pablo Parrilo for insightful discussions and Mirjam Dür for pointing out reference [10].

[8] Here dd and sdd matrices refer to notions introduced in Definition 3. Note that any $t$ which makes $tI - A$ dd or sdd gives an upperbound on $\lambda_{\max}(A)$. By formulating the problem this way (instead of requiring $tI \succeq A$) we obtain an LP or SOCP instead of an SDP.

## References

1. MOSEK reference manual (2013). Version 7. Latest version available at `http://www.mosek.com/`
2. Ahmadi, A.A., Majumdar, A.: DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization. In: Proceedings of the 48th Annual Conference on Information Sciences and Systems. Princeton University (2014)
3. Ahmadi, A.A., Majumdar, A.: DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization (2015). In preparation
4. Ahmadi, A.A., Olshevsky, A., Parrilo, P.A., Tsitsiklis, J.N.: NP-hardness of deciding convexity of quartic polynomials and related problems. Mathematical Programming **137**(1-2), 453–476 (2013)
5. Ahmadi, A.A., Parrilo, P.A.: A convex polynomial that is not sos-convex. Mathematical Programming **135**(1-2), 275–292 (2012)
6. Ahmadi, A.A., Parrilo, P.A.: A complete characterization of the gap between convexity and sos-convexity. SIAM Journal on Optimization **23**(2), 811–833 (2013). Also available at arXiv:1111.4587
7. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Mathematical Programming **95**(1), 3–51 (2003)
8. Alvarado, A., Scutari, G., Pang, J.: A new decomposition method for multiuser dc-programming and its applications. Signal Processing, IEEE Transactions on **62**(11), 2984–2998 (2014)
9. Argyriou, A., Hauser, R., Micchelli, C.A., Pontil, M.: A DC-programming algorithm for kernel selection. In: Proceedings of the 23rd international conference on Machine learning, pp. 41–48. ACM (2006)
10. Bomze, I., Locatelli, M.: Undominated dc decompositions of quadratic functions and applications to branch-and-bound approaches. Computational Optimization and Applications **28**(2), 227–245 (2004)
11. Chapelle, O., Do, C.B., Teo, C.H., Le, Q.V., Smola, A.J.: Tighter bounds for structured estimation. In: Advances in Neural Information Processing Systems, pp. 281–288 (2009)
12. Dür, M.: A parametric characterization of local optimality. Mathematical Methods of Operations Research **57**(1), 101–109 (2003)
13. Floudas, C., Pardalos, P.: Optimization in computational chemistry and molecular biology: local and global approaches, vol. 40. Springer Science & Business Media (2013)
14. Folland, G.: How to integrate a polynomial over a sphere. American Mathematical Monthly pp. 446–448 (2001)
15. Fung, G., Mangasarian, O.: Semi-supervised support vector machines for unlabeled data classification. Optimization methods and software **15**(1), 29–44 (2001)
16. Garey, M.R., Johnson, D.S.: Computers and Intractability. W. H. Freeman and Co., San Francisco, Calif. (1979)
17. Gulpinar, N., Hoai An, L.T., Moeini, M.: Robust investment strategies with discrete asset choice constraints using DC programming. Optimization **59**(1), 45–62 (2010)
18. Hartman, P.: On functions representable as a difference of convex functions. Pacific J. Math **9**(3), 707–713 (1959)
19. Helton, J.W., Nie, J.: Semidefinite representation of convex sets. Mathematical Programming **122**(1), 21–64 (2010)
20. Hilbert, D.: Über die Darstellung Definiter Formen als Summe von Formenquadraten. Math. Ann. **32** (1888)
21. Hillestad, R., Jacobsen, S.: Reverse convex programming. Applied Mathematics and Optimization **6**(1), 63–78 (1980)
22. Hiriart-Urruty, J.B.: Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. In: Convexity and duality in optimization, pp. 37–70. Springer (1985)
23. Hoai An, L.T., Le, H.M., Tao, P.D., et al.: A DC programming approach for feature selection in support vector machines learning. Advances in Data Analysis and Classification **2**(3), 259–278 (2008)
24. Hoai An, L.T., Tao, P.D.: Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. Journal of Global Optimization **11**(3), 253–285 (1997)

25. Horst, R., Thoai, N.: DC programming: overview. Journal of Optimization Theory and Applications **103**(1), 1–43 (1999)
26. de Klerk, E., Laurent, M.: On the Lasserre hierarchy of semidefinite programming relaxations of convex polynomial optimization problems (2010). Available at `http://www.optimization-online.org/DB-FILE/2010/11/2800.pdf`
27. Lanckriet, G., Sriperumbudur, B.: On the convergence of the concave-convex procedure. In: Advances in neural information processing systems, pp. 1759–1767 (2009)
28. Lasserre, J.B.: Convexity in semialgebraic geometry and polynomial optimization. SIAM Journal on Optimization **19**(4), 1995–2014 (2008)
29. Ling, C., Nie, J., Qi, L., Ye, Y.: Biquadratic optimization over unit spheres and semidefinite programming relaxations. SIAM Journal on Optimization **20**(3), 1286–1310 (2009)
30. Lipp, T., Boyd, S.: Variations and extensions of the convex-concave procedure (2014). Available at `http://web.stanford.edu/~boyd/papers/cvx_ccv.html`
31. Lou, Y., Osher, S., Xin, J.: Computational Aspects of Constrained $l_1 - l_2$ Minimization for Compressive Sensing. In: Modelling, Computation and Optimization in Information Systems and Management Sciences, pp. 169–180. Springer (2015)
32. Magnani, A., Lall, S., Boyd, S.: Tractable fitting with convex polynomials via sum of squares. In: Proceedings of the $44^{th}$ IEEE Conference on Decision and Control (2005)
33. Megretski, A.: SPOT: systems polynomial optimization tools (2013)
34. Parrilo, P.A.: Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, California Institute of Technology (2000)
35. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming **96**(2, Ser. B), 293–320 (2003)
36. Parrilo, P.A., Sturmfels, B.: Minimizing polynomial functions. Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **60**, 83–99 (2003)
37. Piot, B., Geist, M., Pietquin, O.: Difference of convex functions programming for reinforcement learning. In: Advances in Neural Information Processing Systems, pp. 2519–2527 (2014)
38. Reznick, B.: Some concrete aspects of Hilbert's 17th problem. In: Contemporary Mathematics, vol. 253, pp. 251–272. American Mathematical Society (2000)
39. Salakhutdinov, R., Roweis, S., Ghahramani, Z.: On the convergence of bound optimization algorithms. In: Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, pp. 509–516. Morgan Kaufmann Publishers Inc. (2002)
40. Tao P. D.and Hoai An, L.T.: Convex analysis approach to dc programming: Theory, algorithms and applications. Acta Mathematica Vietnamica **22**(1), 289–355 (1997)
41. Tao, P.D.: Duality in dc (difference of convex functions) optimization. Subgradient methods. In: Trends in Mathematical Optimization, pp. 277–293. Springer (1988)
42. Toland, J.: On subdifferential calculus and duality in non-convex optimization. Mémoires de la Société Mathématique de France **60**, 177–183 (1979)
43. Tuy, H.: A general deterministic approach to global optimization via dc programming. North-Holland Mathematics Studies **129**, 273–303 (1986)
44. Tuy, H.: Global minimization of a difference of two convex functions. In: Nonlinear Analysis and Optimization, pp. 150–182. Springer (1987)
45. Tuy, H.: DC optimization: theory, methods and algorithms. In: Handbook of global optimization, pp. 149–216. Springer (1995)
46. Tuy, H., Horst, R.: Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and dc optimization problems. Mathematical Programming **41**(1-3), 161–183 (1988)
47. Wang, S., Schwing, A., Urtasun, R.: Efficient inference of continuous markov random fields with polynomial potentials. In: Advances in Neural Information Processing Systems, pp. 936–944 (2014)
48. Yuille, A., Rangarajan, A.: The concave-convex procedure (CCCP). Advances in neural information processing systems **2**, 1033–1040 (2002)