Copositive Formulations of the DOMINATING SET Problem and Applications

Master Thesis

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in the
Department of Mathematics
Faculty of Science

Jan Hendrik Rolfes
University of Cologne
Winter 2014

Contents

1 Introduction

The game of chess has inspired mathematicians for a long time. A well-known example for a chess problem is the so-called "Eight Queens Puzzle". In 1848 the german mathematician M. Bezzel asked for the number of possibilities to place eight queens on a chessboard, without two queens threatening each other. A related problem was covered in G. Berge's famous book "The Theory of graphs" and is known as the "Five Queens Puzzle". This problem deals with the question of finding the minimum number of queens needed to attack or occupy every square on a chess board. The general case of this problem is known as the DOMINATING SET Problem or DS Problem and belongs to the fundamental problems in graph theory.

In 1990 S. T. Hedetniemi and R. C. Laskar (see [17]) noted that essential research on Domination in graphs started with the graph theory texts of D. König (1950), G. Berge (1958) and O. Ore (1962). From the mid-1970s onwards the number of domination papers grew quickly. The authors attribute this to mainly three factors. Two of them are the diversity of applications and the interest in finding polynomial time solutions to domination problems in special classes of graphs, which have motivated the approach worked out in this thesis .

A dominating set for an undirected graph $G = (V, E)$ is a subset D of V, such that every vertex in V is connected to at least one member of D. Now we are looking for the domination number $\gamma(G)$ which is the number of vertices in a smallest dominating set of G.

To classify the hardness of DOMINATING SET we consider the so-called decision version of DOMINATING SET, which means, given an integer k, to decide whether a dominating set of size $\leq k$ exists in a given graph. Due to the fact that V itself is always a dominating set in a graph $G = (V, E)$, we can reduce the decision problem to integers $k \leq |V|$. It has been shown by M. Garey and D. Johnson in 1979 ([10]), that this decision version of DOMINATING SET is NP-complete. Under the hypothesis that P$\neq$ NP, there is no efficient algorithm to solve this problem.

The approach to the DS Problem worked out in this thesis is to formulate DOMINATING SET as a so-called copositive program, which is a certain optimization problem.

The field of Mathematical Optimization is a very active area of research in modern mathematics. Generally speaking it is about finding a point in a set of feasible points which maximizes or minimizes an objective function. For the purpose of this thesis we mostly stick to conic programs, where the set of feasible points can be expressed with the help of convex cones and linear constraints. Examples for convex cones are the non-negative orthant or the cone of positive semidefinite matrices, common constraints are linear (in)equalities or integrality constraints. Copositive programs are certain conic programs, where the considered cone is the cone of copositive matrices. In Chapter 3 we develope two specific copositive formulations for DOMINATING SET.

Because solving copositive programs in general remains NP-hard, we approximate the resulting program with a series of simpler programs developed by P. Parrilo in [21]. This allows us to develope algorithms to approximate any copositive program, such as the formulations given in Chapter 3 within polynomial time. This means we provide two new approximation algorithms for DOMINATING SET.
Furthermore, a simplification for semidefinite programs, developed by C. Bachoc, D. C. Gijswijt, A. Schrijver and F. Vallentin [1], can be applied to Parrilo's hierarchy in certain instances of DOMINATING SET. This increases the performance of the algorithms significantly for large instances, providing a useful tool to find lower bounds for the domination number $\gamma(G)$.

2 Notations, General Theory and Complexity

In this chapter we give the theoretical background that we need to obtain a copositive formulation of DOMINATING SET. We start with some general definitions and properties for convex sets, continue with theory on conic programming and close with specific definitions and statements about graph theory, necessary to be able to express DOMINATING SET as an optimization problem. For this we follow mainly the script of M. Laurent and F. Vallentin [18]. Later on, the complexity of DOMINATING SET and its approximations is covered. The last section overviews a couple of applications in different fields of science.

2.1 Definitions and Basic Properties

Optimization problems differ in terms of hardness, due to differences in the type of the objective function and the underlying set of feasible points. For our purpose it is sufficient to limit oneself to convex sets of feasible points in the Euclidian Space.

Definition 2.1. A set $C \subseteq \mathbb{R}^n$ is called a *convex* set, if for every pair of elements $x, y \in C$ the entire line segment between x and y is contained in C. The *line segment* between the elements x and y is defined as

$$[x, y] = \{(1-\lambda)x + \lambda y : 0 \leq \lambda \leq 1\}.$$

An important special case for the following chapters are convex sets of real matrices.

Remark 2.2. Since the spaces $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{n^2}$ are isomorphic, matrices $A \in \mathbb{R}^{n\times n}$ can be regarded as elements of convex sets in the sense of the definition above.

Definition 2.3. A *convex combination* in a set S is a linear combination of elements in S, where the non-negative coefficients of the elements sum up to 1, i.e.

$$\sum_{i=1}^{N} \lambda_i x_i : \ x_1, ..., x_N \in S, \ \lambda_1, ..., \lambda_N \in \mathbb{R}_{\geq 0}, \ \sum_{i=1}^{N} \lambda_i = 1.$$

Convex sets are closed under convex combinations of their elements.

Definition 2.4. The *convex hull* of $S \subseteq \mathbb{R}^n$ is the smallest convex set containing S, defined by

$$\operatorname{conv}(S) := \left\{ \sum_{i=1}^{N} \lambda_i x_i : N \in \mathbb{N}, \ x_1, ..., x_N \in S, \ \lambda_1, ..., \lambda_N \in \mathbb{R}_{\geq 0}, \ \sum_{i=1}^{N} \lambda_i = 1 \right\}.$$

As mentioned before a very useful class of convex sets is the class of convex cones. Roughly speaking the main difference between the two classes is that we omit the last constraint, which guarantees the parameters λ_i to be bounded. This implies that convex cones are in general unlimited.

Definition 2.5. A nonempty subset $\mathcal{K}$ of $\mathbb{R}^n$ is called a *convex cone*, if it is closed under non-negative linear combinations, so-called *conic combinations*, i.e.

$$\sum_{i=1}^{N} \lambda_i x_i \in \mathcal{K} \ \forall x_1, ..., x_N \in \mathcal{K}, \ \forall \lambda_1, ..., \lambda_N \in \mathbb{R}_{\geq 0}.$$

Definition 2.6. Similarly the *conic hull* of $S \subseteq \mathbb{R}^n$ is the smallest cone containing S, defined by

$$\operatorname{cone}(S) := \left\{ \sum_{i=1}^{N} \lambda_i x_i : N \in \mathbb{N}, \ x_1, ..., x_N \in S, \ \lambda_1, ..., \lambda_N \in \mathbb{R}_{\geq 0} \right\}.$$

Another interesting hull of a set is the affine hull, obtained by setting the nonnegativity of the parameters λ_i aside.

Definition 2.7. The *affine hull* of a set $S \subseteq \mathbb{R}^n$ is

$$\operatorname{aff}(S) := \left\{ \sum_{i=1}^{N} \alpha_i x_i : N \in \mathbb{N}, \ x_1, ..., x_N \in S, \ \alpha_1, ..., \alpha_N \in \mathbb{R}, \ \sum_{i=1}^{N} \alpha_i = 1 \right\}.$$

In particular the definitions above define objects in the space of real matrices. To be able to represent conic programs on this space we further provide the definition of the Frobenius inner product.

Definition 2.8. Given two matrices $A, B \in \mathbb{R}^{n \times n}$ the Frobenius inner product of these matrices is:

$$\langle A, B \rangle := \operatorname{Tr}(A^T B) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} B_{ij}.$$

With the help of these basic facts and definitions we can express a conic program as follows: Let $\mathcal{K}$ denote a cone. A conic program over this cone is a program which asks besides of linear restrictions for a membership in $\mathcal{K}$. A possible representation of these kinds of programs is

$$\begin{aligned} \inf \ & \langle C, X \rangle \\ \text{s.t. } & \langle A_i, X \rangle = b_i \ \forall i \in \{1, ..., m\} \\ & X \in \mathcal{K}. \end{aligned} \tag{Prim}$$

Example 2.9. We will see later on that the domination number $\gamma(G)$ of a graph G mentioned in the introduction (see also Definition 2.31) can be expressed as

$$\begin{aligned} \gamma(G) = \inf \ & \langle C, X \rangle \\ \text{s.t. } & \langle A_i, X \rangle = b_i \ \forall i \in \{1, ..., m\} \\ & X \in \mathcal{K}, \end{aligned}$$

where the matrices A_i, C and the vector b depend on the graph G. The underlying cone $\mathcal{K}$ is the completely positive cone (see Definition 2.11).

Similar to linear programming every conic program of the form (Prim) has a dual program that we denote by (Du):

$$\begin{aligned} &\sup\ b^T y \\ &\text{s.t. } C - \sum_{i=1}^{m} y_i A_i \in \mathcal{K}^*. \end{aligned} \tag{Du}$$

In this program $\mathcal{K}^*$ denotes the so-called dual cone to the cone $\mathcal{K}$:

Definition 2.10. Let $\mathcal{K} \subseteq \mathbb{R}^{n\times n}$ be a cone, then its dual cone $\mathcal{K}^*$ is defined by

$$\mathcal{K}^* := \left\{ Y \in \mathbb{R}^{n\times n} :\ \langle X, Y\rangle \geq 0\ \forall X \in \mathcal{K} \right\}.$$

Common examples of conic programs are linear or semidefinite programs, where $\mathcal{K}$ is the non-negative orthant respectively the cone of positive semidefinite symmetric matrices. These examples have the convenient property that they are self-dual, i.e. $\mathcal{K} = \mathcal{K}^*$. In this thesis our main concern will be with the case that $\mathcal{K}$ denotes the copositive cone. As we will see later on (see Chapter 3, formulations (DVV) and (D2V)) DOMINATING SET can be expressed as a copositive, conic program.

Definition 2.11. The closed, full-dimensional, convex cone of $n \times n$ copositive matrices is defined as

$$\mathcal{C}_n := \left\{ X \in \mathcal{S}^n :\ v^T X v \geq 0\ \forall\ v \in \mathbb{R}^n_{\geq 0} \right\},$$

and its dual is the closed, full-dimensional, convex cone of $n \times n$ completely positive matrices

$$\mathcal{C}_n^* := \operatorname{cone}\left\{ vv^T :\ v \in \mathbb{R}^n_{\geq 0} \right\}.$$

Remark 2.12. In S. Burer's paper [3] an equivalent definition of $\mathcal{C}_n^*$ is used, which is more convenient for the proofs of the upcoming theorems:

$$\mathcal{C}_n^* = \left\{ X \in \mathbb{R}^{n\times n} : X = \sum_{k\in K} z^k (z^k)^T \text{ for some finite } \{z^k\}_{k\in K} \subset \mathbb{R}^n_+ \setminus \{0\} \right\} \cup \{0\}.$$

To check whether a given matrix is copositive or not turns out to be very costly in terms of computation time. Thus in general copositive programs remain very hard, i.e. NP-hard to solve. To illustrate this we introduce a method to check copositivity, mentioned by W. Kaplan [15].

Definition 2.13. A principal submatrix of a matrix $A \in \mathbb{R}^{n\times n}$ is a matrix constructed by selecting some of the rows and columns of A simultaneously. This means for $I \subset \{1, ..., n\}$ with $I \neq \emptyset$ the corresponding principal submatrix is the matrix $A_{I,I} = (A_{ij})_{i,j\in I}$.

Theorem 2.14. *The matrix A is copositive if and only if all principal submatrices of A have no non-negative eigenvector with negative eigenvalue.*

Proof. see [15] Theorem 2. □

Due to the fact that each matrix in the n-dimensional Euclidean Space has 2^n principal submatrices, this characterization provides exponentially many eigenvalues we have to compute. To achieve more convenient formulations for convex optimization problems it is sometimes useful to have a closer look on the boundary of the underlying feasible set. In linear programming for example, the powerful simplex algorithm is based on a method of searching for optimal solutions on the boundary of the feasible set. In the case of copositive programming we talk about objects on the boundary of the underlying convex set. To this end we give the following definitions:

Definition 2.15. A point $x \in C$ in a closed, convex set C is defined as *extreme* if x cannot be written in the form $x = (1-\lambda)y + \lambda z$ with $y, z \in C$ and $\lambda \in (0,1)$. The set of all these points will be denoted $\mathrm{Extp}(C)$.

Definition 2.16. The set of extremal rays $\mathrm{Ext}(\mathcal{K})$ of a closed, convex cone $\mathcal{K} \subseteq V$ in a Euclidean Vector Space V consists of the conic hulls $\mathrm{cone}(z)$ of the elements $z \in \mathcal{K}$ for which the equation $z = x + y$ for all $x, y \in \mathcal{K}$ implies

$$\exists \lambda \in [0,1] : x = \lambda z, y = (1-\lambda)z.$$

Remark 2.17. A pointed convex cone $\mathcal{K}$, i.e. a cone with the property

$$x, -x \in \mathcal{K} \Rightarrow x = 0$$

has only one extreme point in $x = 0$ but maybe a lot of extremal rays due to its shape. The non-negative orthant, the positive semidefinite cone and the copositive cone are pointed.

Definition 2.18. The interior of a closed, convex set $C \subseteq V$, denoted $\mathrm{Int}(C)$ is defined as

$$\mathrm{Int}(C) := \{x \in C : \ \exists \ \epsilon > 0 : \ B_\epsilon(x) \subseteq C\}.$$

Likewise the set of boundary points of this set is defined as $\partial C := C \setminus \mathrm{Int}(C)$.

Remark 2.19. A characterization for the interior of dual cones $\mathcal{K}^*$ of cones of symmetric matrices $\mathcal{K} \subseteq \mathcal{S}$ was given by Dür and Still [9]:

$$\mathrm{Int}(\mathcal{K}^*) := \{Y \in \mathcal{S} : \langle Y, X \rangle > 0 \ \forall X \in \mathcal{K} \setminus \{0\}\}.$$

For our purpose it is convenient to consider the interior of the copositive cone. The characterization above can be applied to this cone as follows:

Lemma 2.20. *The interior of the n-dimensional copositive cone is the set of strictly copositive matrices, i.e.*

$$\mathrm{Int}(\mathcal{C}_n) = \left\{A \in \mathcal{S} | x^T A x > 0 \ \forall x \in \mathbb{R}^n_{\geq 0} \setminus \{0\}\right\}.$$

Proof. The above characterization can be found e.g. in Dür and Still [9]. □

Remark 2.21. M. Hall and M. Newman [14] proposed the following characterization for the extremal rays of the completely positive cone, obtained from Definition 2.11:

$$\mathrm{Ext}(\mathcal{C}^*_n) = \left\{aa^T : a \in \mathbb{R}^n_{\geq 0}\right\}.$$

For the reader's convenience we further provide a few basic properties of convex sets and their boundaries, which will be useful to reformulate DOMINATING SET. Especially Lemma 2.25 will play a major role in proving the main theorems in Chapter 3. Important for these properties are so-called hyperplanes.

Definition 2.22. A *hyperplane* H at a point $x \in \mathbb{R}^n$ is a $(n-1)$-dimensional subspace of the form

$$H := \left\{y \in \mathbb{R}^n : \; a^T x = a^T y\right\},$$

where $a \in \mathbb{R}^n \setminus \{0\}$ is the normal of H. This hyperplane divides $\mathbb{R}^n$ into two *halfspaces*

$$H^+ := \left\{y \in \mathbb{R}^n : \; a^T x \geq a^T y\right\}, \; H^- := \left\{y \in \mathbb{R}^n : \; a^T x \leq a^T y\right\}.$$

A hyperplane H is a *supporting hyperplane* of a convex set C at $z \in C$ if $z \in H$ and either $C \subseteq H^+$ or $C \subseteq H^-$.

Lemma 2.23. *[18] Let $C \subseteq \mathbb{R}^n$ be a closed convex set and let $x \in \partial C$ be a point lying on the boundary of C. Then there is a hyperplane that supports C at x.*

Lemma 2.24. *[18] Let $C \subseteq \mathbb{R}^n$ be a convex set. If $\operatorname{Int}(C) = \emptyset$ then the dimension of its affine closure is at most $n-1$.*

Lemma 2.25. *Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a closed, convex cone and H be a hyperplane in $\mathbb{R}^n$, with $\mathcal{K} \cap H$ being a compact set. Then,*

$$\operatorname{Extp}(\mathcal{K} \cap H) = \operatorname{Ext}(\mathcal{K}) \cap H.$$

Proof. To show "$\operatorname{Extp}(\mathcal{K} \cap H) \subseteq \operatorname{Ext}(\mathcal{K}) \cap H$", consider:

$$x \in \operatorname{Extp}(\mathcal{K} \cap H) \Rightarrow x \in \mathcal{K} \cap H \Rightarrow x \in \mathcal{K} \text{ and } x \in H$$

This means it suffices to show that $x \in \operatorname{Ext}(\mathcal{K})$. We prove this by induction on the dimension n. If $n = 0$ then $\mathcal{K} = \{0\}$ and the result follows immediately.
If we assume the interior of $\mathcal{K}$ to be empty, we have that $\mathcal{K}$ lies in a $n-1$-dimensional subspace A of $\mathbb{R}^n$. Because of the compactness of $\mathcal{K} \cap H$ we know that H cannot be identified with A except for the case $\mathcal{K} = \{0\}$, because otherwise $\mathcal{K} \cap H = \mathcal{K}$ and thus compactness would not hold. We can therefore consider the $n-2$-dimensional intersection $H_A := H \cap A$ and the $n-1$-dimensional cone $\mathcal{K} = \mathcal{K} \cap A$. The induction hypothesis implies the statement of the theorem in this case. This means we can assume that we have a nonempty interior of $\mathcal{K}$.
If x lies in the interior $\operatorname{Int}(\mathcal{K})$ of $\mathcal{K}$, we obtain without loss of generality $H = x + \lambda_1 y_1 + \ldots + \lambda_{n-1} y_{n-1}$, where y_1 and x are linearly independent. If we define $y := x + \frac{\epsilon}{2} \frac{y_1}{\|y_1\|}$, $z := x - \frac{\epsilon}{2} \frac{y_1}{\|y_1\|}$ with ϵ is sufficiently small and $\lambda := \frac{1}{2} \in (0,1)$, we obtain

$$x = (1-\lambda)y + \lambda z, \text{ where } y, z \in B_\epsilon(x) \subseteq \operatorname{Int}(\mathcal{K}) \subseteq \mathcal{K}, \; y, z \in H.$$

This implies that x can be written as a strict convex combination of y and z, i.e. $x \notin \operatorname{Extp}(\mathcal{K} \cap H)$, which is a contradiction.
Thus x lies on the boundary $\partial \mathcal{K}$ of $\mathcal{K}$. Lemma 2.23 tells us that there is a supporting hyperplane $\tilde{H}$ for $\mathcal{K}$. This means x lies in a $n-1$ dimensional cone $F := \tilde{H} \cap \mathcal{K}$ in a

$n-1$ dimensional subspace $\operatorname{aff}(F)$. With similar arguments as in the case $\operatorname{Int}(\mathcal{K}) = \emptyset$ we obtain that H cannot be identified with $\operatorname{aff}(F)$. Thus $H_F := H \cap \operatorname{aff}(F)$ defines a $n-2$ dimensional hyperplane. Applying the induction hypothesis on this $n-1$ dimensional cone F and the $n-2$ dimensional hyperplane H_F leads to

$$x \in \operatorname{Extp}(F \cap H) \subseteq \operatorname{Ext}(F) \cap H.$$

We finally show that $\operatorname{Ext}(F) \subseteq \operatorname{Ext}(\mathcal{K})$.
Let c denote the normal vector of $\tilde{H}$. Because 0 lies in $\tilde{H}$ we can denote $\tilde{H} = \{y \in \mathbb{R}^n : c^T y = 0\}$ and w.l.o.g $\mathcal{K} \subseteq \tilde{H}^+$.
If we consider the equation $x = x_1 + x_2$ for any $x_1, x_2 \in \mathcal{K}$, implying w.l.o.g $c^T x_1,\ c^T x_2 \geq 0$, we obtain

$$\begin{aligned} x \in \operatorname{Ext}(F) &\Rightarrow c^T x = 0 \\ &\Rightarrow c^T x_1 + c^T x_2 = 0,\ x_1, x_2 \in \mathcal{K} \\ &\Rightarrow c^T x_1 = 0,\ c^T x_2 = 0 \\ &\Rightarrow x_1,\ x_2 \in \tilde{H} \\ &\Rightarrow x_1,\ x_2 \in F. \end{aligned}$$

The fact that $x \in \operatorname{Ext}(F)$ implies for x_1, x_2 that there is a $\lambda \in [0,1]$, such that $x = \lambda x_1,\ x = (1-\lambda)x_2$ and thus we conclude that $x \in \operatorname{Ext}(\mathcal{K})$.
For the inclusion "$\operatorname{Extp}(\mathcal{K} \cap H) \supseteq \operatorname{Ext}(\mathcal{K}) \cap H$", we use contraposition

$$(\mathcal{K} \cap H) \setminus \operatorname{Extp}(\mathcal{K} \cap H) \subseteq (\mathcal{K} \cap H) \setminus (\operatorname{Ext}(\mathcal{K}) \cap H).$$

Let $x \notin \operatorname{Extp}(\mathcal{K} \cap H)$ then we can express x as follows:

$$x = \lambda y + (1-\lambda)z \text{ for } y, z \in \mathcal{K} \cap H,\ \lambda \in (0,1). \tag{2.1.1}$$

If we choose $x_1 = \lambda y$ and $x_2 = (1-\lambda)z$ we know because $\mathcal{K}$ is closed under non-negative scalar multiplication that

$$x = x_1 + x_2 :\ x_1, x_2 \in \mathcal{K}.$$

Let $H = \{y \in \mathbb{R}^n :\ c^T y = b\}$ with $b \neq 0$ because otherwise $\mathcal{K} \cap H$ is either a non-compact cone or the zero cone. If we assume that $x \in \operatorname{Ext}(\mathcal{K})$ we obtain

$$\begin{aligned} x_1 = \mu x,\ x_2 = (1-\mu)x &\Rightarrow y = \frac{\mu}{\lambda} x,\ z = \frac{1-\mu}{1-\lambda} x \text{ for } x \in \mathcal{K} \cap H \\ &\Rightarrow c^T y = \frac{\mu}{\lambda} c^T x = \frac{\mu}{\lambda} b \neq b \text{ for } \mu \neq \lambda \\ &\Rightarrow y \notin H \end{aligned}$$

This is a contradiction to equation (2.1.1) and thus $x \notin \operatorname{Ext}(\mathcal{K})$. Finally this implies $x \notin \operatorname{Ext}(\mathcal{K}) \cap H$. □

Lemma 2.26. *[18] Let $C \subseteq \mathbb{R}^n$ be a compact and convex set. Then,*

$$C = \operatorname{conv}(\operatorname{Extp}(C)).$$

With the help of the above definitions and basic properties we are able to formulate a central theorem for conic programming. Let opt (Prim) and opt (Du) denote the optimal values of (Prim) and (Du).

Theorem 2.27 (Strong Duality Theorem for conic Programming)**.**
(i) If there exists an interior feasible solution $X \in \operatorname{Int}(\mathcal{K}^*)$ *of* (Prim) *and a feasible solution* y *of* (Du)*, then*

$$\operatorname{opt}(Prim) = \operatorname{opt}(Du)$$

(ii) If there exists an interior feasible solution y *of* (Du)*, i.e.*

$$C - \sum_{i=1}^{m} y_i A_i \in \operatorname{Int}(\mathcal{K}),$$

and a feasible solution X *of* (Prim)*, then*

$$\operatorname{opt}(Prim) = \operatorname{opt}(Du)$$

Proof. A proof is given in J. Renegar's paper [24]. □

Applying this theorem to the case $\mathcal{K} = \mathcal{C}$ will help us to provide a copositive formulation for DOMINATING SET and closes the optimization theoretical part of this chapter. To give a first formulation of the DS Problem we provide a few basic concepts in graph theory. Consider an undirected and connected graph $G = (V, E)$ with V the set of vertices and E the set of edges.

Remark 2.28. We can limit ourselves to connected graphs because, if we have a disconnected graph, we can consider all of its connected components separately. Then we solve the DS Problem for all of these connected components and add up the results to get the domination number for the original graph.

Definition 2.29. We define the *adjacency matrix* A_G of a graph G by

$$A_G := \begin{cases} (A_G)_{uv} = 1 & \{u, v\} \in E \\ (A_G)_{uv} = 0 & \{u, v\} \notin E \end{cases}$$

and we denote by $(A_G)_i$ the i-th column of A_G.

Definition 2.30. A *dominating set* is a set $D \subseteq V$, if for all $v \in V$, we have $v \in D$ or there is a node $u \in D$ such that $\{u, v\} \in E$.

Definition 2.31. If $G = (V, E)$ is an undirected graph. The *domination number* $\gamma(G)$ is defined as

$$\gamma(G) := \min\{|D| : D \subseteq V \text{ is a dominating set}\}.$$

Remark 2.32. [18] The domination number $\gamma(G)$ can be reformulated as the following binary optimization problem i.e. a problem, where the decision variables can only take two values

$$\gamma(G) = \min \left\{ \sum_{v \in V} x_v : \ x \in \{0,1\}^V, x_v + \sum_{u:\{u,v\} \in E} x_u \geq 1 \ \forall v \in V \right\}.$$

2.2 Complexity of DOMINATING SET

To get an impression of the complexity of the DS Problem we give an introduction similar to the one in T. W. Haynes et al. [11]. Given a graph $G = (V, E)$ with $|V| = n$, the domination number lies in the range $1 \leq \gamma(G) \leq n$. Thus to determine a dominating set $D \subseteq V$ we could check all the 2^n given subsets of V and take the one with the smallest cardinality. The construction of this brute force algorithm is quite easy but this algorithm would need exponentially many steps. Thus one goal is to obtain an algorithm with significantly faster, for example polynomial run time. The theory of NP-completeness suggests that it is not likely that we are able to construct an polynomial algorithm.

In the theory of NP-completeness we restrict our attention to the so-called decision problems. The answer to these kinds of problems is always a "yes" or a "no". For the DS Problem this means we seek an algorithm which, given a graph G and a positive integer k, can decide whether G has a dominating set of size at most k.

Let P denote the class of all decision problems that can be solved in polynomial time, which means polynomial in the length of the inputs for every instance of the problem.

NP denotes the class of all decision problems that have a certificate which is checkable in polynomial time by a deterministic Turing machine. For details see [10] or the famous paper of S. A. Cook [5].

For DOMINATING SET we want to know if a given graph G has a dominating set D with $|D| \leq k$ for some given positive integer k. A nondeterministic Turing machine has the ability to make a guess if $v_i \in D$ or not for every $1 \leq i \leq n$. Then it is possible to verify in polynomial time whether the resulting D is a dominating set and $|D| \leq k$ or not.

The fundamental open question and also one of the famous Millennium Prize Problems (see [12]) is whether P equals NP or not. To characterize the complexity of DOMINATING SET we use Cook's (see [10] or [5]) class of NP-complete problems and check it with the help of the following definitions.

Definition 2.33. P_1 is *polynomial-time reducible* to a problem P_2 if

1. there exists a function f which maps any instance of P_1 to an instance of P_2 in such a way that: I_1 is a "yes" instance of $P_1 \Leftrightarrow f(I_1)$ is a "yes" instance of P_2.

2. for any instance I_1, the instance $f(I_1)$ can be constructed in polynomial time.

Notation: $P_1 \leq_p P_2$

This implies if P_2 is solvable in polynomial time so is P_1, or roughly speaking: P_2 is at least as hard as P_1 in terms of run time.

Definition 2.34. P is NP-*complete* if

1. $P \in \text{NP}$

2. $P' \in \text{NP} \Rightarrow P' \leq_p P$

Remark 2.35. If condition 2 is satisfied but not necessarily condition 1 we call the problem *NP-hard*.

Because of the transitivity of $\leq_p$ a method to show that a given problem P (in our case DOMINATING SET) is in NP is the following:

1. show $P \in$ NP, and
2. show that there is a NP-complete problem P' such that $P' \leq_p P$

We use this format, as established by M. Garey and D. Johnson in [10], to be able to handle the following form for the basic complexity question concerning the decision problem for the domination number.

DOMINATING SET
Instance: A graph $G = (V, E)$ and a positive integer k
Question: Does G have a dominating set of size $\leq k$?

To prove the NP-completeness of DOMINATING SET we will use the reduction techniques given in [10]. Thus we need the definition and the NP-completeness of the following decision problem.

VERTEX COVER
Instance: A graph $G = (V, E)$ and a positive integer k
Question: Is there a *vertex cover* of size k or less for G, that is, a subset $V' \subseteq V$ such that $|V'| \leq k$ and, for each edge $(u, v) \in E$, at least one of u and v belongs to V'?

Lemma 2.36. *VERTEX COVER is NP-complete.*

Proof. see [10] for a reduction from the well known 3-SAT problem, which is one of R. M. Karp's famous 21 NP-complete problems (see [16]).
In [10] the additional restriction $k \leq |V|$ is given. But for $k > |V|$ the answer to VERTEX COVER is "yes" because V itself is always a vertex cover of V, this means we can omit the additional restriction. □

Lemma 2.37. *DOMINATING SET is NP-complete.*

Proof. We use the idea of A. Paz and S. Moran [22] (Examples 5.3.(v)) and reduce VERTEX COVER polynomially to DOMINATING SET.
First we need to show that DOMINATING SET $\in$ NP.

Consider a "yes" instance of DOMINATING SET, that is, for a graph $G = (V, E)$ a positive integer k and an arbitrary set $S \subseteq V$ with $|S| \leq k$. To verify in polynomial time whether S is a dominating set can be done for example by a brute force algorithm by checking the connections of all $v \in V$ to all $v \in S$.

Second we will show that VERTEX COVER $\leq_p$ DOMINATING SET.
We use the abbreviations VC for a vertex cover and DS for a dominating set and transform VERTEX COVER to DOMINATING SET. Let $G = (V, E)$ and $k \in \mathbb{N}$

be any instance of VERTEX COVER. We need to construct a graph $G' = (V', E')$ with $k' \in \mathbb{N}$ such that

$$G \text{ has a VC } S \text{ of size } |S| \leq k \Leftrightarrow G' \text{ has a DS } S' \text{ of size } |S'| \leq k'.$$

For this we choose G' and k', where each undirected edge e connecting nodes u and v is denoted by exactly one tuple (u, v) as follows:

$$V' = V \cup E,\ k' = k,$$

$$E' = \{(i,j)|\ i,j \in V\} \cup \{(i,e)|\ i \in V,\ e \in E,\ i \text{ incident to } e\}.$$

$" \Leftarrow "$
We will show that $S := S' \cup \{u_e \in V : e = (u_e, v_e) \in E \cap S'\} \setminus E \cap S'$ is a VC of G with $|S| \leq k$, where u_e is the left node of the tuple (u_e, v_e) corresponding to edge e, i.e. we take exactly one node for each edge e.

G' has a DS S' $\Rightarrow$ $\forall\ v \in V'$ either $v \in S'$ or $\exists\ u \in S' :\ (u, v) \in E'$
Consider an arbitrary edge $e \in E$, then

$$e \in E \Rightarrow e \in V' \overset{DS}{\Rightarrow} e \in S'\ \vee\ \exists\ u \in S' :\ (u, e) \in E'$$

Case 1:

$$e \in S' \Rightarrow u_e \in S \Rightarrow \exists\ u \in S :\ u \in e$$

Case 2:

$$\begin{aligned}\exists\ u \in S' :\ (u, e) \in E' &\Rightarrow \exists\ u \in S' :\ u \in V,\ u \in e \\ &\Rightarrow \exists\ u \in S' :\ u \notin E \cap S',\ u \in e \\ &\Rightarrow \exists\ u \in S :\ u \in e\end{aligned}$$

This means S is a VC of G and with

$$\begin{aligned}|S| &= |S' \cup \{u_e \in V : e = (u_e, v_e) \in E \cap S'\} \setminus E \cap S'| \\ &\overset{E\cap S'\subseteq S'}{=} |S' \cup \{u_e \in V : e = (u_e, v_e) \in E \cap S'\}| - |E \cap S'| \\ &\leq |S'| + |\{u_e \in V : e = (u_e, v_e) \in E \cap S'\}| - |E \cap S'| \\ &\leq |S'| \leq k' = k\end{aligned}$$

the first inclusion follows.
$" \Rightarrow "$
We will show that $S' := S$ is a DS of size $|S'| \leq k'$.

G has a VC S such that $|S| \leq k \Rightarrow \forall\ (u, v) \in E \Rightarrow u \in S\ \vee\ v \in S$. Consider an arbitrary vertex $v' \in V'$ then $v' \in V\ \vee\ v' \in E$.
Case 1:

$$v' \in V \Rightarrow v' \in S\ \vee\ v' \notin S \overset{\text{see } E'}{\Rightarrow} v' \in S'\ \vee\ \exists\ u \in S' :\ (u, v') \in E'$$

Case 2:

$$(u_{v'}, w_{v'}) := v' \in E \overset{\text{VC}}{\Rightarrow} u_{v'} \in S\ \vee\ w_{v'} \in S \Rightarrow \exists\ u \in S' :\ (u, v') \in E'$$

This means S' is a DS of G' and with $|S'| = |S| \leq k = k'$ the second inclusion follows. □

2.2.1 Complexity of approximations

To give some relevant statements for approximations of DOMINATING SET we use the equivalent problem SET COVER, where equivalency means we can map instances of the problems to each other (for a construction see [22]).
SET COVER
Instance: Collection C of subsets of a finite set S, positive integer $k \leq |C|$.
Question: Does C contain a cover for S of size k or less, i.e. a subset $C' \subseteq C$ with $|C'| \leq k$ such that every element of S belongs to at least one member of C'?

Lemma 2.38. *DOMINATING SET is approximable within* $1 + log(|V|)$ *by using a polynomial algorithm*

Proof. see the paper of D. Johnson [13] for the approximation of SET COVER and conclude the above lemma by using Johnson's algorithm for

$$S = V,\ C := \{S_1, ..., S_{|V|}\}, \text{ where } S_i := \{i\} \cup \{j | (i,j) \in E\} \subseteq S$$

(see [22] Examples 5.3. (iv)). □

Lemma 2.39. *For any* $0 < c < \frac{1}{8}$*, DOMINATING SET cannot be approximated within a factor of* $c \cdot log(|V|)$ *in polynomial time unless* NP*-complete problems can be solved in* $\mathcal{O}(n^{poly(log(n))})$ *(i.e.* $\text{NP} \subseteq DTIME(n^{poly(log(n))})$*).*

Proof. Assume that we knew a polynomial algorithm which would be able to approximate DOMINATING SET within a factor of $c \cdot log(|V|)$. We could apply the solution of this algorithm to get a solution for DOMINATING SET with $G = (V, E)$, where $V := \{1, ..., n, x_1, ..., x_m\}$ and $E = \{(i,j)|\ 1 \leq i < j \leq n\} \cup \{(i, x_i)|\ x_i \in S_i\}$ (see [22] Examples 5.3. (iii)).
Thus we obtain an approximation of SET COVER with $C = \{S_1, ..., S_n\}$ and $S = \bigcup_{i=1}^{n} S_i = \{x_1, ..., x_m\}$ within a factor of $c \cdot log(n + m)$. For a connected graph and $m \geq 3$ we can conclude $n+m \leq 2m+1 \leq m^2$ and thus $c \cdot log(n+m) \leq 2c \cdot log(m)$. For $0 < c < \frac{1}{8}$ we would be able to approximate SET COVER within a factor of $\frac{1}{4} log(m)$. This would contradict a theorem of C. Lund and M. Yannakakis (Theorem 3.3 in [19]) or NP-complete problems would be solvable in $\mathcal{O}(n^{poly(log(n))})$. □

2.3 Applications of DOMINATING SET

These statements are valid for graphs with finite node and edge sets. Examples for these kinds of graphs are well known, for example we could think of cluster heads in a network. A rather popular example to apply DOMINATING SET are the famous British "football pools" and its continental counterpart the "toto" competitions.

2.3.1 Football Pools

A "football pool" is a betting pool based on predicting the outcome of football matches. Invented in the United Kingdom similar football pools known as "toto" competitions became very popular in Continental Europe. In these games players predict the outcome of typically thirteen given football matches from the upcoming matchday. The player marks a home win with 1, an away win with 2 and a draw

with 0 (sometimes with an X or N).
It is of great interest for the players to know, how many combinations or *codes* $\omega \in \{1, 0, 2\}^{13}$ a player has to cover to predict at least 12 out of 13 matches correctly, i.e. to guarantee quite a large amount of money. In fact players have to cover 59049 (see J. G. Mauldon [20]) of these codes. But if we don't stick to $n = 13$ matches to choose from, or to $R = 1$ differences between the codes, the problem is generally unsolved even today. A possibility to reformulate this problem is DOMINATING SET for a graph $G = (V, E)$ with node set $V = \{1, 2, 0\}^n$ and $E = \{\{u, v\} \in V \times V : |i : u_i \neq v_i| \leq R\}$. We will refer to this problem later on.

2.3.2 Radio Stations

T. W. Haynes, S. T. Hedetniemi and P. J. Slater [11] gave another example of DOMINATING SET. If we suppose a remote part of the world like the Outback in Australia or Siberia and we want to locate radio stations in some of the very rare villages in these regions we need to use several stations to ensure that each village can receive the radio program, due to limited broadcasting range. Since radio stations are expensive we want to construct as few as possible. Let the broadcasting range be 50 kilometres, the villages be represented by a vertex and two villages be adjacent if the distance between them is less or equal to 50 kilometres. The domination number of this graph gives us the least number of radio stations needed to broadcast to every village.

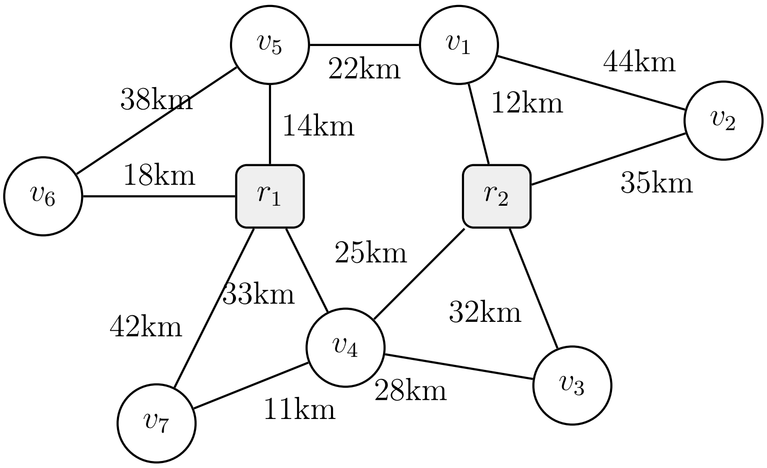

Figure 2.1: Two radio stations broadcasting to 9 villages $v_1, ..., v_7, r_1, r_2$.

So far we have only considered applications for DOMINATING SET, given finite node and edge sets. To convince the reader that the concept of dominating sets is also useful in infinite graphs we provide some useful applications when there are infinite node and edge sets.

2.3.3 Applications of the Spherical Covering problem

More applications of DOMINATING SET can be found for the following parameters of our problem. Let $V := S^{n-1} = \{x \in \mathbb{R}^n : \ x^T x = 1\}$, $E_R := \{\{x, y\} :$

$\|x - y\| \leq R\}$, where $R > 0$ and consider the DS Problem of the graph $G = (V, E)$. This special case of DOMINATING SET is called Spherical Covering problem because it describes for a fixed radius R the minimal amount N of points $x_1, ..., x_N$, which is needed to cover S^{n-1} via so-called *spherical caps* with angle γ. These caps are defined by

$$C(x, \gamma) := \{y \in S^{n-1} : \ \|x - y\| \leq R, \ \} = \{y \in S^{n-1} : \ x^T y \geq \cos(\gamma)\},$$

where γ is defined by $\gamma := \arccos(1 - \frac{1}{2}R^2)$.

2.3.4 Coverings with Transmission Towers

Similar to covering an area with radio stations we can cover a whole planet with transmission towers for e.g. a GPS network. A planet's surface can approximately be assumed as shaped like a sphere and we want to design a network that covers the whole surface of the planet. Each transmission tower has fixed expenses and fixed range R. To obtain the cheapest network we want to know how to place the towers on the planet. We design the problem as a Spherical Covering problem where the radius of range R assigns the spherical caps.

2.3.5 Surveys in astronomy

A similar application of these spherical caps is given in [2]: "Large surveys using multiobject spectrographs require automated methods for deciding how to efficiently point observations and how to assign targets to each pointing. The Sloan Digital Sky Survey (SDSS) will observe around 10^6 spectra from targets distributed over an area of about 10,000 deg^2, using a multiobject fiber spectrograph that can simultaneously observe 640 objects in a circular field of view (referred to as "tile") $1.49°$ in radius. No two fibers can be placed closer than $55''$ during the same observation; multiple targets closer than this distance are said to "collide"."
If we assume γ to be $1.49°$, these circular fields can be considered as the caps in our model and the firmament we can consider as the node set $V = S^2$. Let us assume further that we have enough fibers to cover each object in the caps (or tiles), so we neglect the restrictions given above. A solution to our problem would provide the minimal amount of spectrographs needed to cover the sphere and thus to create an efficient survey to observe the whole firmament.

3 Formulation of DOMINATING SET as a Copositive Program

As seen before approximating DOMINATING SET remains a hard problem. In 2009 S. Burer [3] provided a general result to achieve a completely positive reformulation of binary programs such as DOMINATING SET. Because of duality reasons approximating copositive programs should have the same computational complexity as approximating completely positive programs. However, a formal proof for this statement still has to be found. Nevertheless for certain classes of matrices checking copositivity is cheap in terms of computational run time. For example the copositivity of diagonal matrices is verifiable in linear time. This leads to the impression that copositive programs are promising candidates to approximate binary programs like DOMINATING SET. The ideas of Burer's approach will be used in this chapter to obtain several completely positive formulations for DOMINATING SET. Afterwards by dualizing we obtain corresponding copositve formulations that are approximable with the methods given in Chapter 4. Recall that we have the following formulation of DOMINATING SET (see Remark 2.32):

$$\begin{aligned}\gamma(G) &= \min \left\{ \sum_{v \in V} x_v : \ x \in \{0,1\}^V, x_v + \sum_{u:\{u,v\} \in E} x_u \geq 1 \ \forall v \in V \right\} \\ &= \min \left\{ \mathbb{1}^T x : \ x \in \{0,1\}^V, \quad (I_V + A_G)x \geq \mathbb{1} \right\}.\end{aligned}$$

Introducing a slack vector $s := (s_1, s_2, ..., s_n) \geq 0$ changes the inequalities to equalities. Let us further denote $A := I_V + A_G$. We obtain

$$\gamma(G) = \min \left\{ \mathbb{1}^T x : \ x \in \{0,1\}^V, \ s \geq 0, \ Ax - s = \mathbb{1} \right\}.$$

Defining $V = \{1, ..., n\}$ leads to the below formulation of this program. We will refer to this formulation as the standard form of the binary program for DOMINATING SET.

$$\begin{aligned} \min \ & \mathbb{1}^T x && && \text{(P)} \\ \text{s.t. } & A_i^T x = \ 1 + s_i && \forall i \in V && \\ & x, s \geq \ 0 && && (3.0.1) \\ & x_i \in \ \{0,1\} && \forall i \in V && \end{aligned}$$

A natural completely positive formulation of (P), due to S. Burer [3] is the following:

$$\begin{aligned}
\min\ & \mathbb{1}^T x && && \text{(C)}\\
\text{s.t. } & A_i^T x = 1 + s_i && \forall\, i \in V && (3.0.2)\\
& \langle A_i A_i^T, X\rangle = (1+s_i)^2 && \forall\, i \in V && (3.0.3)\\
& x_i = X_{ii} && \forall\, i \in V && (3.0.4)\\
& s \geq 0 && && \\
& \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathcal{C}^*_{n+1} && && (3.0.5)
\end{aligned}$$

For convenience of the reader we provide a proof for this formulation in Theorem 3.5. For this we need some further preliminaries.

Definition 3.1.

$$\mathrm{Feas}(P) := \left\{(x,s) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^n_{\geq 0} : (x,s) \text{ is feasible for (P)}\right\}$$

$$\mathrm{Feas}(C) := \left\{(x,s,X) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n\times n} : (x,s,X) \text{ is feasible for (C)}\right\}$$

$$\mathrm{Feas}^+(P) := \left\{(s,X) : X \in \mathrm{conv}\left\{\begin{pmatrix} 1 \\ x \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}^T\right\},\ (x,s) \in \mathrm{Feas}(P)\right\}$$

$$\mathrm{Feas}^+(C) := \left\{\left(s, \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}\right) :\ (x,s,X) \in \mathrm{Feas}(C)\right\}$$

Definition 3.2.

$$\mathrm{opt}^+(P) := \left\{(s,X) : X \in \mathrm{conv}\left\{\begin{pmatrix} 1 \\ x \end{pmatrix}\begin{pmatrix} 1 \\ x \end{pmatrix}^T\right\},\ (x,s) \text{ is optimal for (P)}\right\}$$

$$\mathrm{opt}^+(C) := \left\{\left(s, \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix}\right) :\ (x,s,X) \text{ is optimal for (C)}\right\}$$

Lemma 3.3. Feas(P) *and* Feas(C) *are compact.*

Proof. Consider (P). The constraint $x_i \in \{0,1\}$ restricts x in each component and thus via $s_i = A_i^T x - 1$, s is restricted in each component. This means we have a finite set of points in $\mathbb{R}^n$, which is closed and restricted and thus compact.

To show that Feas(C) is compact, consider the closed convex cone $\mathbb{R}^n_{\geq 0} \times \mathcal{C}^*_{n+1}$. Feas$(C)$ is a projection of this cone on the intersection of the hyperplanes (3.0.2), (3.0.3) and (3.0.4). Projections of closed sets remain closed and thus Feas(C) is closed. To show that for Feas(C) is restricted, observe that for $(x,s,X) \in \mathrm{Feas}(C)$ the components x and X define a matrix $\begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \{Y \in \mathcal{C}^*_{n+1} :\ Y_{11} = 1,\ Y_{ii} = X_{1i},\ i \geq 2\} \subseteq \{Y \in \mathcal{S}^{n+1} :\ Y_{11} = 1,\ Y_{ii} = Y_{1i},\ i \geq 2\}$, which is a compact set. This is because each principal minor of a matrix in the last set has to be positive semidefinite, implying

$$(1,-1)\begin{pmatrix} 1 & Y_{1i} \\ Y_{i1} & Y_{ii} \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = 1 - 2Y_{1i} + Y_{ii} = 1 - Y_{ii} \geq 0\ \forall i \in V.$$

This implies that $0 \leq Y_{ii} \leq 1$ for each $i \in V$. If we use a similar argumentation for the remaining minors we conclude that each entry of Y is restricted and thus $x^T = (Y_{1,2}, ..., Y_{1,n+1})$, $X = \begin{pmatrix} Y_{2,2} & \cdots \\ \cdots & Y_{n+1,n+1} \end{pmatrix}$ and $s = Ax - \mathbb{1}$ are restricted. □

Remark 3.4. Observe that $\mathrm{opt}(P) \geq 0$ and $\mathrm{opt}(C) \geq 0$ because of restrictions (3.0.1) and (3.0.5). Also we observe that none of the above sets is empty, because $x = \mathbb{1},\ s = Ax - \mathbb{1},\ X = xx^T$ always provides a solution. The compactness of $\mathrm{Feas}^+(P)$ and $\mathrm{Feas}^+(C)$ implies that we do not have to distinguish between inf and min.

With these preliminaries we can prove the equivalence between (P) and (C).

Theorem 3.5. *(C) is equivalent to (P), i.e.:*
(i) $\mathrm{opt}(C) = \mathrm{opt}(P)$
(ii) (x^*, s^*, X^*) *is optimal for (C)* $\Rightarrow$ (x^*, s^*) *is in the convex hull of optimal solutions for (P).*

Proof. For (i):
Let $\mathbb{1}$ be of dimension n. We define the linear function

$$L: \ \mathcal{S}^{n+1} \longrightarrow \mathbb{R}$$
$$L(Y) := \langle \begin{pmatrix} 0 & \frac{1}{2}\mathbb{1}^T \\ \frac{1}{2}\mathbb{1} & 0 \end{pmatrix}, Y \rangle.$$

We can express $\mathrm{opt}(P)$ and $\mathrm{opt}(C)$ as follows:

$$\mathrm{opt}(P) = \min_{(t,Y) \in \mathrm{Feas}^+(P)} L(Y) = \min_{(t,Y) \in \mathrm{opt}^+(P)} L(Y)$$
$$\mathrm{opt}(C) = \min_{(t,Y) \in \mathrm{Feas}^+(C)} L(Y) = \min_{(t,Y) \in \mathrm{opt}^+(C)} L(Y).$$

" $\mathrm{opt}(C) \leq \mathrm{opt}(P)$"

It can be easily checked that $\left(s, \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \right) \in \mathrm{Feas}^+(P)$ provides a feasible solution $\left(x, s, xx^T\right)$ for (C). Thus we have $\mathrm{Feas}^+(P) \subseteq \mathrm{Feas}^+(C)$, which implies $\mathrm{opt}(C) \leq \mathrm{opt}(P)$.

" $\mathrm{opt}(C) \geq \mathrm{opt}(P)$"

The proof of this inequality is organized in two steps. First we show that each optimal solution of (C) can be expressed as a convex combination of extreme points of a subset of $\mathrm{Feas}^+(C)$. Second we show that these extreme points are in $\mathrm{Feas}^+(P)$ and conclude the inequality.

1. We consider the convex cone $\bar{C} := \left\{ X \in \mathcal{C}^*_{n+1} : \ X_{ii} = X_{1i} \ \forall i \in \{2, ..., n+1\} \right\}$ and the hyperplane $H := \left\{ X \in \mathcal{C}^*_{n+1} : \ X_{11} = 1 \right\}$. Let us define $\tilde{C} := \bar{C} \cap H$, which is compact and convex. Lemma 2.25 implies that the extreme points of $\tilde{C}$ are of the form $\begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$, where $x_i = x_i^2$ and thus $x_i \in \{0, 1\}$.

If we consider an optimal solution $\left(s^*, \begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} \right) \in \mathrm{opt}^+(C)$, we have in particular

$$\begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} \in \tilde{C}.$$

Lemma 2.26 implies that we can regard this matrix as a convex combination of the extreme points of $\tilde{C}$. These points are denoted with the help of a finite set, say S, of vectors x^k, whose support corresponds to a node set of the graph as follows:

$$\begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} = \sum_{k\in S} \lambda_k \begin{pmatrix} 1 \\ x^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \end{pmatrix}^T, \ \lambda_k > 0, \sum_{k\in S} \lambda_k = 1, \begin{pmatrix} 1 \\ x^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \end{pmatrix}^T \in \mathrm{Extp}(\tilde{C}).$$

2. The two first feasibility restrictions provide

$$\sum_{k\in S} \lambda_k A_i^T x^k = 1 + s_i^*, \ \sum_{k\in S} \lambda_k \langle A_i A_i^T, x^k (x^k)^T \rangle = (1 + s_i^*)^2.$$

This implies

$$\left(\sum_{k\in S} \lambda_k A_i^T x^k \right)^2 = \sum_{k\in S} \lambda_k (A_i^T x^k)^2.$$

Recall that $\sum_{k\in S} \lambda_k = 1$, if we denote $z^k = \sqrt{\lambda_k} x^k$, we obtain

$$\left(\sum_{k\in S} \sqrt{\lambda_k} A_i^T z^k \right)^2 = \left(\sum_{k\in S} \lambda_k \right) \left(\sum_{k\in S} (A_i^T z^k)^2 \right).$$

Because of the equality case of the Cauchy-Schwarz inequality we know that there are δ_i for which we have

$$\delta_i \sqrt{\lambda_k} = A_i^T z^k \ \forall k \in S, \forall i \in V.$$

To calculate these δ_i we use

$$\delta_i = \sum_{k\in S} \lambda_k \delta_i = \sum_{k\in S} \sqrt{\lambda_k} A_i^T z^k = \sum_{k\in S} \lambda_k A_i^T x^k = 1 + s_i^*,$$

which leads to

$$1 + s_i^* = \frac{A_i^T z^k}{\sqrt{\lambda_k}} = A_i^T x^k \ \forall k \in S, \ \forall i \in V.$$

Thus for each $k \in S$ we have $A_i^T x^k = 1 + s_i^*$ and $\begin{pmatrix} 1 \\ x^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \end{pmatrix}^T \in \mathrm{Ext}(\tilde{C})$, which means $x^k \in \{0, 1\}^V$. This implies $(x^k, s^*) \in \mathrm{Feas}(P)$, implying

$$\left(s^*, \begin{pmatrix} 1 \\ x^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \end{pmatrix}^T \right) \in \mathrm{Feas}^+(P).$$

And thus

$$\begin{aligned} \mathrm{opt}(C) = L\left(\begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} \right) &= \sum_{k\in S} \lambda_k L\left(\begin{pmatrix} 1 \\ x^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \end{pmatrix}^T \right) \\ &\geq \sum_{k\in S} \lambda_k \min_{(t,Y)\in \mathrm{Feas}^+(P)} L(Y) = \mathrm{opt}(P). \end{aligned}$$

For (ii):
If we consider the solutions $(x^k, s^*, x^k(x^k)^T)$ from part i), we know $\mathbb{1}^T x^k \geq \text{opt}(P)$ because of feasibility of (x^k, s^*) for (P). With the help of i) we conclude

$$\text{opt}(P) \stackrel{(i)}{=} \text{opt}(C) = \sum_{k \in S} \lambda_k \mathbb{1}^T x^k, \; \mathbb{1}^T x^k \geq \text{opt}(P) \; \forall \; k \in S.$$

Thus we have $\mathbb{1}^T x^k = \text{opt}(P)$ for each $k \in S$, which means that (x^k, s^*) are optimal solutions of (P). Finally the equation $(x^*, s^*) = \sum_{k \in S} \lambda_k (x^k, s^*)$ finishes the proof. □

The biggest obstacle left in (C) is the quadratic restriction (3.0.3). One cannot neglect this restriction due to the following counterexample.

Example 3.6. Consider a circle with four nodes as the underlying graph G.

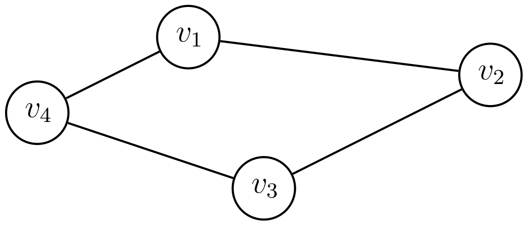

Figure 3.1: Circle C_4

Let $G = C_4 = (V, E)$ be a graph with extended adjacency matrix

$$A = I + A_G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

The domination number can be derived as $\gamma(G) = 2$. We will show that

$$(x^*, s^*, t^*, X^*) = \left(\frac{5}{12}\mathbb{1}, \frac{1}{4}\mathbb{1}, \frac{3}{4}\mathbb{1}, \frac{1}{12} \begin{pmatrix} 5 & 2 & 2 & 2 \\ 2 & 5 & 2 & 2 \\ 2 & 2 & 5 & 2 \\ 2 & 2 & 2 & 5 \end{pmatrix} \right)$$

solves the following relaxation of (C) with a better value than $\gamma(G) = 2$.

$$\begin{aligned} \min \;& \mathbb{1}^T x && \text{(CEx)} \\ \text{s.t.} \;& A_i^T x = 1 + s_i && \forall \; i \in V \\ & \langle A_i A_i^T, X \rangle = 1 + 2s_i + t_i && \forall \; i \in V \\ & x_i = X_{ii} && \forall \; i \in V \\ & s, t \geq 0 \\ & \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathcal{C}^*_{n+1} \end{aligned}$$

First we calculate the objective value:

$$\mathbb{1}^T x^* = \mathbb{1}^T \frac{5}{12}\mathbb{1} = \frac{5}{12}\mathbb{1}^T\mathbb{1} = \frac{5}{12} \cdot 4 = \frac{5}{3} < 2.$$

Second we show that (x^*, s^*, t^*, X^*) is feasible for (CEx).

$$A_i^T x^* = A_i^T \frac{5}{12}\mathbb{1} = \frac{5}{12}A_i^T\mathbb{1} = \frac{5}{12} \cdot 3 = \frac{5}{4} = 1 + \frac{1}{4} = 1 + s_i^* \ \forall i \in \{1,2,3,4\}$$

$$\begin{aligned}\langle A_i A_i^T, X^* \rangle &= \frac{1}{12}\langle A_i A_i^T, \begin{pmatrix} 5 & 2 & 2 & 2 \\ 2 & 5 & 2 & 2 \\ 2 & 2 & 5 & 2 \\ 2 & 2 & 2 & 5 \end{pmatrix} \rangle = \frac{15+12}{12} = \frac{27}{12} \\ &= 1 + \frac{1}{2} + \frac{9}{12} = 1 + 2 \cdot \frac{1}{4} + \frac{3}{4} = 1 + 2s_i^* + t_i^* \ \forall i \in \{1,2,3,4\}\end{aligned}$$

$$x_i^* = \frac{5}{12} = X_{ii}^* \ \forall i \in \{1,2,3,4\}$$

$$s_i^* = \frac{1}{4} \geq 0, \ t_i^* = \frac{3}{4} \geq 0 \ \forall i \in \{1,2,3,4\}$$

$$\begin{aligned}\begin{pmatrix} 1 & (x^*)^T \\ x^* & X^* \end{pmatrix} &= \frac{1}{12}\begin{pmatrix} 12 & 5 & 5 & 5 & 5 \\ 5 & 5 & 2 & 2 & 2 \\ 5 & 2 & 5 & 2 & 2 \\ 5 & 2 & 2 & 5 & 2 \\ 5 & 2 & 2 & 2 & 5 \end{pmatrix} = \frac{1}{6}\begin{pmatrix} 4 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} + \frac{1}{12}\begin{pmatrix} 4 & 3 & 3 & 3 & 3 \\ 3 & 3 & 2 & 2 & 2 \\ 3 & 2 & 3 & 2 & 2 \\ 3 & 2 & 2 & 3 & 2 \\ 3 & 2 & 2 & 2 & 3 \end{pmatrix} \\ &= \frac{1}{6}\sum_{i=1}^{4}\begin{pmatrix} 1 \\ e_i \end{pmatrix}\begin{pmatrix} 1 \\ e_i \end{pmatrix}^T + \frac{1}{12}\sum_{i=1}^{4}\begin{pmatrix} 1 \\ \mathbb{1} - e_i \end{pmatrix}\begin{pmatrix} 1 \\ \mathbb{1} - e_i \end{pmatrix}^T \in \mathcal{C}_{n+1}^*\end{aligned}$$

Now we can conclude that C_4 is a suitable counterexample for the equivalence between (CEx) and (C). This means we cannot relax the quadratic constraint (3.0.3) in this way.

A completely positive program like (C) is though not very promising in terms of run time, due to the quadratic restriction (3.0.3). Hence a program which avoids this constraint is more convenient. A program satisfying this property is

$$\begin{aligned} \min \ & \mathbb{1}^T x & & \text{(C2V)} \\ \text{s.t.} \ & (A_i^T, -e_i^T)\begin{pmatrix} x \\ s \end{pmatrix} = 1 & & \forall\, i \in V \\ & \langle \begin{pmatrix} A_i \\ -e_i \end{pmatrix}\begin{pmatrix} A_i \\ -e_i \end{pmatrix}^T, X \rangle = 1 & & \forall\, i \in V \\ & x_i = X_{ii} & & \forall\, i \in V \\ & \begin{pmatrix} 1 & (x^T, s^T) \\ \begin{pmatrix} x \\ s \end{pmatrix} & X \end{pmatrix} \in \mathcal{C}_{2n+1}^*. & & \end{aligned}$$

Similar to the proof of Theorem 3.5 we need some further definitions to describe the feasible respectively optimal solutions of (P) and (C2V).

Definition 3.7.

$$\mathrm{Feas}_2^+(P) := \mathrm{conv}\left\{ \begin{pmatrix} 1 \\ x \\ s \end{pmatrix} \begin{pmatrix} 1 \\ x \\ s \end{pmatrix}^T : (x,s) \in \mathrm{Feas}(P) \right\}$$

$$\mathrm{Feas}^+(C2V) := \left\{ \begin{pmatrix} 1 & (x^T, s^T) \\ \begin{pmatrix} x \\ s \end{pmatrix} & X \end{pmatrix} : \ (x,s,X) \in \mathrm{Feas}(C2V) \right\}$$

$$\mathrm{opt}_2^+(P) := \mathrm{conv}\left\{ \begin{pmatrix} 1 \\ x \\ s \end{pmatrix} \begin{pmatrix} 1 \\ x \\ s \end{pmatrix}^T : (x,s) \text{ is optimal for (P)} \right\}$$

$$\mathrm{opt}^+(C2V) := \left\{ \begin{pmatrix} 1 & (x^T, s^T) \\ \begin{pmatrix} x \\ s \end{pmatrix} & X \end{pmatrix} : \ (x,s,X) \text{ is optimal for (C2V)} \right\}$$

Theorem 3.8. *(C2V) is equivalent to (P), i.e.:*
(i) $\mathrm{opt}(C2V) = \mathrm{opt}(P)$
(ii) (x^*, s^*, X^*) *is optimal for* $(C2V) \Rightarrow (x^*, s^*)$ *is in the convex hull of optimal solutions for (P).*

Proof. For (i):
Let $\mathbb{1}$ be of dimension n. We define the linear function

$$L: \ \mathcal{S}^{2n+1} \longrightarrow \mathbb{R}$$

$$L(Y) := \langle \begin{pmatrix} 0 & (\frac{1}{2}\mathbb{1}^T, 0^T) \\ \begin{pmatrix} \frac{1}{2}\mathbb{1} \\ 0 \end{pmatrix} & 0 \end{pmatrix}, Y \rangle.$$

We can express $\mathrm{opt}(P)$ and $\mathrm{opt}(C2V)$ as follows:

$$\mathrm{opt}(P) = \min_{Y \in \mathrm{Feas}_2^+(P)} L(Y) = \min_{Y \in \mathrm{opt}_2^+(P)} L(Y)$$

$$\mathrm{opt}(C2V) = \min_{Y \in \mathrm{Feas}^+(C2V)} L(Y) = \min_{Y \in \mathrm{opt}^+(C2V)} L(Y).$$

" $\mathrm{opt}(C2V) \le \mathrm{opt}(P)$"

It can be easily checked that $\begin{pmatrix} 1 \\ x \\ s \end{pmatrix} \begin{pmatrix} 1 \\ x \\ s \end{pmatrix}^T \in \mathrm{Feas}_2^+(P)$ is a feasible solution for (C2V). Thus we have $\mathrm{Feas}_2^+(P) \subseteq \mathrm{Feas}^+(C2V)$, which implies $\mathrm{opt}(C2V) \le \mathrm{opt}(P)$.

" $\mathrm{opt}(C2V) \ge \mathrm{opt}(P)$"

This inequality will be proved as follows: First we show that each optimal solution of (C2V) can be expressed as a convex combination of extreme points of a subset of $\mathrm{Feas}^+(C2V)$. Second we show that these extreme points are in $\mathrm{Feas}^+(P)$ and

conclude the inequality.
1. We consider the convex cone

$$\bar{C} := \left\{ X \in \mathcal{C}^*_{2n+1} : \begin{array}{r} A_i^T(X_{2,1}, ..., X_{n+1,1})^T - X_{n+1+i,1} = X_{11} \ \forall i \in V \\ \langle \begin{pmatrix} A_i \\ -e_i \end{pmatrix} \begin{pmatrix} A_i \\ -e_i \end{pmatrix}^T, X_{\{2,...,2n+1\}} \rangle = X_{11} \ \forall i \in V \\ X_{i+1,i+1} = X_{1,i+1} \ \forall i \in V \end{array} \right\},$$

where $X_{\{2,...,2n+1\}}$ is the bottom $2n \times 2n$ principal submatrix of X and the hyperplane $H := \{X \in \mathcal{C}^*_{2n+1} : \ X_{11} = 1\}$. Let the compact, convex set $\tilde{C} := \bar{C} \cap H$, where compactness can be shown similar to the proof of Lemma 3.3. Lemma 2.25 implies that the extreme points of $\tilde{C}$ are of the form $\begin{pmatrix} 1 \\ x \\ s \end{pmatrix} \begin{pmatrix} 1 \\ x \\ s \end{pmatrix}^T$, where $x_i = x_i^2$ and $A_i^T x = 1 + s_i$.

If we consider an optimal solution $\begin{pmatrix} 1 & ((x^*)^T, (s^*)^T) \\ \begin{pmatrix} x^* \\ s^* \end{pmatrix} & X^* \end{pmatrix} \in \mathrm{opt}^+(C2V)$, we have $\begin{pmatrix} 1 & ((x^*)^T, (s^*)^T) \\ \begin{pmatrix} x^* \\ s^* \end{pmatrix} & X^* \end{pmatrix} \in \tilde{C}$. This implies that we can regard this matrix as a convex combination of the extreme points of $\tilde{C}$. These points are denoted with the help of a finite set, say S, of vectors $\begin{pmatrix} x^k \\ s^k \end{pmatrix}$, whose support corresponds to a node set of the graph, i.e.

$$\begin{pmatrix} 1 & ((x^*)^T, (s^*)^T) \\ \begin{pmatrix} x^* \\ s^* \end{pmatrix} & X^* \end{pmatrix} = \sum_{k \in S} \lambda_k \begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix}^T,$$

where $\lambda_k > 0, \sum_{k \in S} \lambda_k = 1$ and $\begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix}^T \in \mathrm{Extp}(\tilde{C})$.

2. Thus we have $A_i^T x^k = 1 + s_i^k$, $x_i^k \in \{0, 1\}$ and $s_i^k \geq 0$. This implies $(x^k, s^k) \in \mathrm{Feas}(P)$, which on the other hand implies $\begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix}^T \in \mathrm{Feas}_2^+(P)$. For the optimal value this leads to

$$\begin{aligned} \mathrm{opt}(C2V) = L \left(\begin{pmatrix} 1 & ((x^*)^T, (s^*)^T) \\ \begin{pmatrix} x^* \\ s^* \end{pmatrix} & X^* \end{pmatrix} \right) &= \sum_{k \in S} \lambda_k L \left(\begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix} \begin{pmatrix} 1 \\ x^k \\ s^k \end{pmatrix}^T \right) \\ &\geq \sum_{k \in S} \lambda_k \min_{Y \in \mathrm{Feas}_2^+(P)} L(Y) = \mathrm{opt}(P). \end{aligned}$$

For (ii):
If we consider the solutions $(x^k, s^k) \in \mathrm{Feas}(P)$ from part i), we know $\mathbb{1}^T x^k \geq \mathrm{opt}(P)$ because of the feasibility of (x^k, s^k) for (P). With the help of i) we conclude

$$\mathrm{opt}(P) \overset{(i)}{=} \mathrm{opt}(C2V) = \sum_{k\in S} \lambda_k \mathbb{1}^T x^k,\ \mathbb{1}^T x^k \geq \mathrm{opt}(P)\ \forall k \in S.$$

Thus we have $\mathbb{1}^T x^k = \mathrm{opt}(P)$ for each $k \in S$, which means that (x^k, s^k) are optimal solutions of (P). Finally the equation $(x^*, s^*) = \sum_{k\in S} \lambda_k (x^k, s^k)$ finishes the proof. □

Due to this equivalency we obtain a completely positive reformulation of DOMINATING SET. To get a more convenient, e.g. copositive, formulation we need to dualize the above program (C2V).

Theorem 3.9. *The dual formulation for* (C2V) *is:*

$$\sup \mathbb{1}^T \begin{pmatrix} 2X_{1,n+2} \\ \dots \\ 2X_{1,2n+1} \end{pmatrix} - \mathbb{1}^T \begin{pmatrix} X_{n+2,n+2} \\ \dots \\ X_{2n+1,2n+1} \end{pmatrix} - X_{11}\ \ s.t. \tag{D2V}$$

$$\begin{aligned}
&X_{1+i,1+i} + 2 \sum_{j:\{i,j\}\in E} X_{1,n+1+j} + 2X_{1,1+i} - \sum_{j:\{i,j\}\in E} X_{n+1+j,n+1+j} = 1\ \forall i \in V \\
&X_{1+i,1+j} - \sum_{k:\{i,k\}\in E,\{j,k\}\in E} X_{n+1+k,n+1+k} = 0 && \forall i,j \in V,\ i<j \\
&X_{1+i,n+1+j} + X_{n+1+j,n+1+j} = 0 && \forall \{i,j\} \in E \\
&X_{1+i,n+1+j} = 0 && \forall \{i,j\} \notin E \\
&X_{n+1+i,n+1+j} = 0 && \forall i,j \in V,\ i<j \\
&X \in \mathcal{C}_{2n+1}
\end{aligned}$$

Proof. Recall the primal formulation of (C2V):

$$\begin{aligned}
\min\ & \mathbb{1}^T x \\
\text{s.t.}\ & \left(A_i^T, -e_i^T\right) \begin{pmatrix} x \\ s \end{pmatrix} = 1 && \forall\, i \in V \\
& \langle \begin{pmatrix} A_i \\ -e_i \end{pmatrix} \begin{pmatrix} A_i \\ -e_i \end{pmatrix}^T, X \rangle = 1 && \forall\, i \in V \\
& x_i = X_{ii} && \forall\, i \in V \\
& \begin{pmatrix} 1 & (x^T, s^T) \\ \begin{pmatrix} x \\ s \end{pmatrix} & X \end{pmatrix} \in \mathcal{C}^*_{2n+1}
\end{aligned} \tag{C2V}$$

We can reformulate (C2V) by using the Frobenius inner product for the restrictions and redefine $X := \begin{pmatrix} 1 & (x^T, s^T) \\ \begin{pmatrix} x \\ s \end{pmatrix} & X \end{pmatrix}$.

$$\min \langle \begin{pmatrix} 0 & (\frac{1}{2}\mathbb{1}^T, 0^T) \\ \begin{pmatrix} \frac{1}{2}\mathbb{1} \\ 0 \end{pmatrix} & 0 \end{pmatrix}, X \rangle \tag{C2V'}$$

$$\begin{aligned} \text{s.t. } \langle \frac{1}{2} \begin{pmatrix} 0 & (A_i^T, -e_i^T) \\ \begin{pmatrix} A_i \\ -e_i \end{pmatrix} & 0 \end{pmatrix}, X \rangle &= 1 && \forall\, i \in V \\ \langle \begin{pmatrix} 0 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} A_i \\ -e_i \end{pmatrix} \begin{pmatrix} A_i \\ -e_i \end{pmatrix}^T \end{pmatrix}, X \rangle &= 1 && \forall\, i \in V \\ \langle \begin{pmatrix} 0 & (e_i^T, 0^T) \\ \begin{pmatrix} e_i \\ 0 \end{pmatrix} & \begin{pmatrix} -2e_ie_i^T & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix}, X \rangle &= 0 && \forall\, i \in V \\ \langle \begin{pmatrix} 1 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 \end{pmatrix}, X \rangle &= 1 \\ X &\in \mathcal{C}^*_{2n+1} \end{aligned}$$

Consider the dual program:

$$\sup \sum_{i=1}^{2n} y_i + y_{3n+1}$$

$$\begin{aligned} \text{s.t. } & \begin{pmatrix} 0 & (\frac{1}{2}\mathbb{1}^T, 0^T) \\ \begin{pmatrix} \frac{1}{2}\mathbb{1} \\ 0 \end{pmatrix} & 0 \end{pmatrix} - \sum_{i=1}^{n} y_i \frac{1}{2} \begin{pmatrix} 0 & (A_i^T, -e_i^T) \\ \begin{pmatrix} A_i \\ -e_i \end{pmatrix} & 0 \end{pmatrix} \\ & - \sum_{i=1}^{n} y_{i+n} \begin{pmatrix} 0 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} A_i \\ -e_i \end{pmatrix} \begin{pmatrix} A_i \\ -e_i \end{pmatrix}^T \end{pmatrix} - \sum_{i=1}^{n} y_{i+2n} \begin{pmatrix} 0 & (e_i^T, 0^T) \\ \begin{pmatrix} e_i \\ 0 \end{pmatrix} & \begin{pmatrix} -2e_ie_i^T & 0 \\ 0 & 0 \end{pmatrix} \end{pmatrix} \\ & - y_{3n+1} \begin{pmatrix} 1 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & 0 \end{pmatrix} \in \mathcal{C}_{2n+1} \end{aligned}$$

If we define the vectors $y_0 := -y_{3n+1}$, $y^1 := (y_1, ..., y_n)$, $y^2 := (-y_{n+1}, ..., -y_{2n})$ and $y^3 := 2(y_{2n+1}, ..., y_{3n})$ the dual program can be written in a more convenient form:

$$\sup \mathbb{1}^T y^1 - \mathbb{1}^T y^2 - y_0 \tag{D2V'}$$

$$\text{s.t. } \begin{pmatrix} y_0 & \frac{1}{2}(\mathbb{1} - y^3 - Ay^1)^T & \frac{1}{2}(y^1)^T \\ \frac{1}{2}(\mathbb{1} - y^3 - Ay^1) & \begin{matrix} \operatorname{diag}(y^3) + A \operatorname{diag}(y^2) A \\ - \operatorname{diag}(y^2) A \end{matrix} & \begin{matrix} -A \operatorname{diag}(y^2) \\ \operatorname{diag}(y^2) \end{matrix} \\ \frac{1}{2} y^1 & & \end{pmatrix} \in \mathcal{C}_{2n+1}.$$

To verify that for (C2V') and (D2V') strong duality holds and thus the objective values of the two programs do not differ, we need to check the conditions of Theorem 2.27. In this proof we set back ourselves to the dual formulation and prove strong duality in Lemma 3.10. We continue with the formulation of (D2V'), where we define $X \in \mathcal{C}_{2n+1}$:

$$\sup \mathbb{1}^T y^1 - \mathbb{1}^T y^2 - y_0 \tag{D2V'}$$

$$\text{s.t. } X = \begin{pmatrix} y_0 & \frac{1}{2}(\mathbb{1} - y^3 - Ay^1)^T & \frac{1}{2}(y^1)^T \\ \frac{1}{2}(\mathbb{1} - y^3 - Ay^1) & \operatorname{diag}(y^3) + A\operatorname{diag}(y^2)A & -A\operatorname{diag}(y^2) \\ \frac{1}{2}y^1 & -\operatorname{diag}(y^2)A & \operatorname{diag}(y^2) \end{pmatrix} \in \mathcal{C}_{2n+1}.$$

Let $X_{1,1}, ..., X_{1,2n+1}$ and $X_{n+2,n+2}, ..., X_{2n+1,2n+1}$ be free variables. This leaves the other variables restricted with the constraints:

$$\begin{pmatrix} X_{2,2} & ... & X_{n+1,2} \\ & ... & \\ X_{2,n+1} & ... & X_{n+1,n+1} \end{pmatrix} = \operatorname{diag}(y^3) + A\operatorname{diag}(y^2)A$$

$$\begin{pmatrix} X_{2,n+2} & ... & X_{2,2n+1} \\ & ... & \\ X_{n+1,n+2} & ... & X_{n+1,2n+1} \end{pmatrix} = -A\operatorname{diag}(y^2)$$

$$\begin{pmatrix} X_{n+2,n+2} & ... & X_{n+2,2n+1} \\ & ... & \\ X_{2n+1,n+2} & ... & X_{2n+1,2n+1} \end{pmatrix} = \operatorname{diag}(y^2),$$

which are equivalent to

$$\begin{pmatrix} X_{2,2} & ... & X_{n+1,2} \\ & ... & \\ X_{2,n+1} & ... & X_{n+1,n+1} \end{pmatrix} = I - 2\operatorname{diag}(X_{1,2}, ..., X_{1,n+1})$$

$$- 2\operatorname{diag}(A(X_{1,n+2}, ..., X_{1,2n+1})^T) + A\operatorname{diag}(X_{n+2,n+2}, ..., X_{2n+1,2n+1})A$$

$$\begin{pmatrix} X_{2,n+2} & ... & X_{2,2n+1} \\ & ... & \\ X_{n+1,n+2} & ... & X_{n+1,2n+1} \end{pmatrix} = -A\operatorname{diag}(X_{n+2,n+2}, ..., X_{2n+1,2n+1})$$

$$\begin{pmatrix} X_{n+2,n+2} & ... & X_{n+2,2n+1} \\ & ... & \\ X_{2n+1,n+2} & ... & X_{2n+1,2n+1} \end{pmatrix} = \operatorname{diag}(X_{n+2,n+2}, ..., X_{2n+1,2n+1}).$$

Due to symmetry reasons this leads to:

$$\begin{aligned} & X_{1+i,1+i} + 2\sum_{j:\{i,j\}\in E} X_{1,n+1+j} + 2X_{1,1+i} - \sum_{j:\{i,j\}\in E} X_{n+1+j,n+1+j} = 1 \ \forall i \in V \\ & X_{1+i,1+j} - \sum_{k:\{i,k\}\in E,\{j,k\}\in E} X_{n+1+k,n+1+k} = 0 && \forall i,j \in V,\ i<j \\ & X_{1+i,n+1+j} + X_{n+1+j,n+1+j} = 0 && \forall \{i,j\} \in E \\ & X_{1+i,n+1+j} = 0 && \forall \{i,j\} \notin E \\ & X_{n+1+i,n+1+j} = 0 && \forall i,j \in V,\ i<j \end{aligned}$$

The objective function translates to:

$$\mathbb{1}^T y^1 - \mathbb{1}^T y^2 - y_0 = \mathbb{1}^T \begin{pmatrix} 2X_{1,n+2} \\ ... \\ 2X_{1,2n+1} \end{pmatrix} - \mathbb{1}^T \begin{pmatrix} X_{n+2,n+2} \\ ... \\ X_{2n+1,2n+1} \end{pmatrix} - X_{11}$$

□

As mentioned before we need to verify that strong duality between (C2V') and (D2V') (and thus for (C2V) and (D2V)) holds.

Lemma 3.10. *If we choose $y_0 = 1$, $y^1 = 0$, $y^2 = \mathbb{1}$ and $y^3 = \mathbb{1}$ then,*

$$\begin{pmatrix} y_0 & \frac{1}{2}(\mathbb{1} - y^3 - Ay^1)^T & \frac{1}{2}(y^1)^T \\ \frac{1}{2}(\mathbb{1} - y^3 - Ay^1) & \operatorname{diag}(y^3) + A\operatorname{diag}(y^2)A & -A\operatorname{diag}(y^2) \\ \frac{1}{2}y^1 & -\operatorname{diag}(y^2)A & \operatorname{diag}(y^2) \end{pmatrix} \in \operatorname{Int}(\mathcal{C}_{2n+1}).$$

Proof.

$$\begin{pmatrix} y_0 & \frac{1}{2}(\mathbb{1} - y^3 - Ay^1)^T & \frac{1}{2}(y^1)^T \\ \frac{1}{2}(\mathbb{1} - y^3 - Ay^1) & \operatorname{diag}(y^3) + A\operatorname{diag}(y^2)A & -A\operatorname{diag}(y^2) \\ \frac{1}{2}y^1 & -\operatorname{diag}(y^2)A & \operatorname{diag}(y^2) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & \frac{1}{2}\mathbb{1}^T - \frac{1}{2}\mathbb{1}^T & 0^T \\ \frac{1}{2}\mathbb{1} - \frac{1}{2}\mathbb{1} & I + A^2 & -A \\ 0 & -A & I \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0^T & 0^T \\ 0 & I + A^2 & -A \\ 0 & -A & I \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0^T & 0^T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0^T & 0^T \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} A \\ -I \end{pmatrix}\begin{pmatrix} A \\ -I \end{pmatrix}^T \end{pmatrix} =: M$$

To show that this matrix is strictly copositive we have to show that $x^T M x > 0$ holds for all $x \geq 0,\ x \neq 0$. We will show the contrapositive. Due to the fact that each of these matrices is positive semidefinite we have $x^T M x \geq 0$ and thus we can assume $x^T M x = 0$ instead of $x^T M x \leq 0$ and show that this implies that $x = 0$.

$$x^T \left(\begin{pmatrix} 1 & 0^T & 0^T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0^T & 0^T \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} A \\ -I \end{pmatrix}\begin{pmatrix} A \\ -I \end{pmatrix}^T \end{pmatrix} \right) x = 0$$

Also because of positive semidefiniteness we can conclude:

$$x^T \begin{pmatrix} 1 & 0^T & 0^T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} x = 0 \tag{3.0.6}$$

$$x^T \begin{pmatrix} 0 & 0^T & 0^T \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} x = 0 \tag{3.0.7}$$

$$x^T \begin{pmatrix} 0 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} A \\ -I \end{pmatrix}\begin{pmatrix} A \\ -I \end{pmatrix}^T \end{pmatrix} x = 0. \tag{3.0.8}$$

From (3.0.6) and (3.0.7) we obtain $x_1 = ... = x_{|V|+1} = 0$ and thus for (3.0.8) we conclude

$$x^T \begin{pmatrix} 0 & (0^T, 0^T) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} A \\ -I \end{pmatrix}\begin{pmatrix} A \\ -I \end{pmatrix}^T \end{pmatrix} x = \begin{pmatrix} x_{|V|+2} \\ \dots \\ x_{2|V|+1} \end{pmatrix}^T I \begin{pmatrix} x_{|V|+2} \\ \dots \\ x_{2|V|+1} \end{pmatrix} = 0.$$

This means we also have $x_{|V|+2} = ... = x_{2|V|+1} = 0$ and have shown the contrapositive. □

We can conclude with Lemma 3.10 and Theorem 2.27 that $\gamma(G)$ can be formulated e.g. with the copositive program (D2V). As an example we can consider the equivalent copositive program (D2V'), where the underlying graph is just a simple graph with two connected nodes.

Example 3.11. Let G be a graph with two nodes connected via an edge, i.e. $A_G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and thus $A = I + A_G = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. The domination number $\gamma(G) \leq 1$ because we can choose one of the nodes and the other one is connected via the edge to obtain a dominating set. We consider (D2V') with $y_0 = 1$, $y^1 = \frac{3}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $y^2 = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $y^3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. We obtain

$$\mathbb{1}^T y^1 - \mathbb{1}^T y^2 - y_0 = 3 - 1 - 1 = 1 = \gamma(G)$$

and

$$\begin{pmatrix} y_0 & \frac{1}{2}(\mathbb{1} - y^3 - Ay^1)^T & \frac{1}{2}(y^1)^T \\ \frac{1}{2}(\mathbb{1} - y^3 - Ay^1) & \operatorname{diag}(y^3) + A\operatorname{diag}(y^2)A & -A\operatorname{diag}(y^2) \\ \frac{1}{2}y^1 & -\operatorname{diag}(y^2)A & \operatorname{diag}(y^2) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & \frac{1}{2}\mathbb{1}^T - \frac{3}{4}(1,1)\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & \frac{3}{4}(1,1) \\ \frac{1}{2}\mathbb{1} - \frac{3}{4}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} & \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} I \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & -\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} I \\ \frac{3}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix} & -\frac{1}{2} I \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & \frac{1}{2} I \end{pmatrix}$$

$$= \begin{pmatrix} 1 & -\mathbb{1}^T & \frac{3}{4}(1,1) \\ -\mathbb{1} & \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & -\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \\ \frac{3}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix} & -\frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} & \frac{1}{2} I \end{pmatrix} = \begin{pmatrix} 1 & -1 & -1 & \frac{3}{4} & \frac{3}{4} \\ -1 & 1 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ -1 & 1 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ \frac{3}{4} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{3}{4} & -\frac{1}{2} & -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} \in \mathcal{C}.$$

To prove that this matrix is copositive can be done by using Theorem 2.14. This is quite elaborate and thus is done in the appendix. The solution provides a lower bound $\mathbb{1}^T y^1 - \mathbb{1}^T y^2 - y_0 = 3 - 1 - 1 = 1 \leq \gamma(G)$ and thus $\gamma(G) = 1$.

The program (D2V) is a convenient copositive program with the major drawback, that the dimension of the matrix X is more than twice of the input size of the original problem. This means every solver has to use big matrices that decrease the performance significantly in the long run. So far it is unknown whether this approach is a good choice for medium size instances. To avoid the problems in the long run we use a third possibility to reformulate (P) as a completely positive program.

$$
\begin{aligned}
\min\ & \mathbb{1}^T x && \text{(CVV)}\\
\text{s.t.}\ & \left(A_i^T, -1\right)\begin{pmatrix} x \\ s_i \end{pmatrix} = 1 && \forall\, i \in V\\
& \left\langle \begin{pmatrix} A_i \\ -1 \end{pmatrix}\begin{pmatrix} A_i \\ -1 \end{pmatrix}^T, \begin{pmatrix} X & S_i \\ S_i^T & \tilde{S}_i \end{pmatrix} \right\rangle = 1 && \forall\, i \in V\\
& x_i = X_{ii} && \forall\, i \in V\\
& \begin{pmatrix} 1 & x^T & s_i \\ x & X & S_i \\ s_i & S_i^T & \tilde{S}_i \end{pmatrix} \in \mathcal{C}^*_{n+2}, && \forall\, i \in V
\end{aligned}
$$

where $x,\ s,\ S_i \in \mathbb{R}^V,\ \tilde{S}_i \in \mathbb{R}$ and $X \in \mathcal{C}^*_n$.
Again similar to the proof of Theorem 3.5, we need to define the sets of feasible and optimal solutions of the programs (P) and (CVV) to prove the formulation (CVV).

Definition 3.12. Let $Z(x, s, X, S, \tilde{S})_i := \begin{pmatrix} 1 & x^T & s_i \\ x & X & S_i \\ s_i & S_i^T & \tilde{S}_i \end{pmatrix}$ and $W(x, s)_i := \begin{pmatrix} 1 \\ x \\ s_i \end{pmatrix}\begin{pmatrix} 1 \\ x \\ s_i \end{pmatrix}^T$ then we define:

$$
\text{LFeas}_3(P) := \operatorname{conv}\left\{ \begin{pmatrix} W(x,s)_1 & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & W(x,s)_n \end{pmatrix} :\ (x, s) \in \text{Feas}(P) \right\}
$$

$$
\text{LFeas}(CVV) :=
\left\{ \begin{pmatrix} Z(x,s,X,S,\tilde{S})_1 & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & Z(x,s,X,S,\tilde{S})_n \end{pmatrix} : (x, s, X, S, \tilde{S}) \in \text{Feas}(CVV) \right\}
$$

$$
\text{LOpt}_3(P) := \operatorname{conv}\left\{ \begin{pmatrix} W(x,s)_1 & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & W(x,s)_n \end{pmatrix} : (x, s) \text{ is optimal for (P)} \right\}
$$

$$
\begin{aligned}
\text{LOpt}(CVV) := \Bigg\{ & \begin{pmatrix} Z(x,s,X,S,\tilde{S})_1 & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & Z(x,s,X,S,\tilde{S})_n \end{pmatrix} : \\
& (x, s, X, S, \tilde{S}) \text{ is optimal for (CVV)} \Bigg\}
\end{aligned}
$$

Theorem 3.13. *(CVV) is equivalent to (P), i.e.:*
(i) $\text{opt}(CVV) = \text{opt}(P)$
(ii) $(x^*, s^*, X^*, S^*, \tilde{S}^*)$ *is optimal for* $(CVV) \Rightarrow (x^*, s^*)$ *is in the convex hull of optimal solutions for (P).*

Proof. For (i):
Let $\mathbb{1}$ be of dimension n. We define the linear function

$$L: \ \mathcal{S}^{n(n+2)} \longrightarrow \mathbb{R}$$
$$L(Y) := \langle \begin{pmatrix} 0 & \frac{1}{2}\mathbb{1}^T & 0 \\ \frac{1}{2}\mathbb{1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, Y \rangle.$$

We can express $\mathrm{opt}(CVV)$ and $\mathrm{opt}(P)$ as follows:

$$\mathrm{opt}(CVV) = \min_{Y \in \mathrm{LFeas}(CVV)} L(Y) = \min_{Y \in \mathrm{LOpt}(CVV)} L(Y)$$

$$\mathrm{opt}(P) = \min_{Y \in \mathrm{LFeas}_3(P)} L\left(Y\right) = \min_{Y \in \mathrm{LOpt}_3(P)} L\left(Y\right)$$

" $\mathrm{opt}(CVV) \leq \mathrm{opt}(P)$"
Let us denote $s^2 := (s_1^2, ..., s_n^2)$, $Z_i := Z(x, s, X, S, \tilde{S})_i$ and $W_i := W(x, s)_i$. It can be easily checked that $\begin{pmatrix} W_1 & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & W_n \end{pmatrix} \in \mathrm{LFeas}_3(P)$ encodes a feasible solution (x, s, xx^T, xs^T, s^2) for (CVV) and thus $\mathrm{LFeas}_3(P)$ is contained in $\mathrm{LFeas}(CVV)$. This implies $\mathrm{opt}(CVV) \leq \mathrm{opt}(P)$.

" $\mathrm{opt}(CVV) \geq \mathrm{opt}(P)$"
The proof of the other inequality is organized as follows: We will show that for a fixed j the Z_j^* part of an optimal solution for (CVV) is in the convex hull of feasible solutions for certain relaxations of (P). In a second step we use this property to obtain a convex combination for the whole optimal solution of (CVV) with elements in $\mathrm{LFeas}_3(P)$ and therefore obtain the wanted inequality.
1. First we consider the convex cone

$$\bar{C}_j := \left\{ X \in \mathcal{C}^*_{V+2} : \quad \begin{aligned} A_j^T(X_{2,1}, ..., X_{n+1,1})^T - X_{n+2,1} &= X_{11} \\ \langle \begin{pmatrix} A_j \\ -1 \end{pmatrix} \begin{pmatrix} A_j \\ -1 \end{pmatrix}^T, X_{\{2,...,n+2\}} \rangle &= X_{11} \\ X_{i+1,i+1} &= X_{1,i+1} \ \forall i \in V \end{aligned} \right\},$$

where $X_{\{2,...,n+2\}}$ is the bottom $(n+1) \times (n+1)$ submatrix of X and the hyperplane $H := \left\{ X \in \mathcal{C}^*_{V+2} : \ X_{11} = 1 \right\}$. Consider the compact, convex set $\tilde{C}_j := \bar{C}_j \cap H$, where the compactness can be verified again by using the methods of the proof for Lemma 3.3. Referring to Lemma 2.25 the set of extreme points of $\tilde{C}_j$ is given by the intersection of the extremal rays of $\bar{C}_j$ with the hyperplane H.

This implies that these points are of the form $\begin{pmatrix} 1 \\ x \\ t_j \end{pmatrix} \begin{pmatrix} 1 \\ x \\ t_j \end{pmatrix}^T$, where $x_i = x_i^2, \ A_j^T x - t_j = 1,$ and $t_j \in \mathbb{R}_{\geq 0}$ and thus $x_i \in \{0, 1\}, \ A_j^T x = 1 + t_j,$ and $t_j \in \mathbb{R}_{\geq 0}$.

If we consider an optimal solution $\begin{pmatrix} Z_1^* & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & Z_n^* \end{pmatrix} \in \mathrm{LOpt}(CVV)$, we have $Z_j^* \in \tilde{C}_j$ for each j. Observe that $\tilde{C}_j$ is compact. Referring to Lemma 2.26 this implies that we can regard each of these matrices as a convex combination of the extreme points of $\tilde{C}_j$. Observing that the support of x_j^k corresponds to a "k-th" node subset and denoting the set of these finitely many subsets by S_j leads us to

$$Z_j^* = \sum_{k \in S_j} \lambda_k \begin{pmatrix} 1 \\ x_j^k \\ t_j^k \end{pmatrix} \begin{pmatrix} 1 \\ x_j^k \\ t_j^k \end{pmatrix}^T, \ \lambda_k > 0, \sum_{k \in S_j} \lambda_k = 1, \begin{pmatrix} 1 \\ x_j^k \\ t_j^k \end{pmatrix} \begin{pmatrix} 1 \\ x_j^k \\ t_j^k \end{pmatrix}^T \in \mathrm{Ext}(\tilde{C}_j).$$

2. Consider the vector x^*, which is a common component of all

$$Z_j^* = \begin{pmatrix} 1 & (x^*)^T & s_j^* \\ x^* & X^* & S_j^* \\ s_j^* & (S_j^*)^T & \tilde{S}_j^* \end{pmatrix}.$$

Recalling that $\{x \in \mathbb{R}^V : A_j^T x \geq 1\}$ is a halfspace for each j, we obtain:

$$\begin{aligned} x^* \in & \bigcap_{j \in V} \mathrm{conv} \left\{x \in \{0,1\}^V : A_j^T x \geq 1\right\} \\ \subseteq & \bigcap_{j \in V} \left(\mathrm{conv} \left\{x \in \{0,1\}^V\right\} \cap \left\{x \in \mathbb{R}^V : \ A_j^T x \geq 1\right\}\right) \\ \subseteq & \,\mathrm{conv} \left\{x \in \{0,1\}^V : A_j^T x \geq 1 \ \forall j \in V\right\} \end{aligned}$$

It is also possible to show equality of these two sets because

$$\bigcap_{i \in I} \mathrm{conv}\, A_i \supseteq \mathrm{conv} \bigcap_{i \in I} A_i$$

holds by Definition 2.4. This means that we can express x^* as a convex combination, i.e. $x^* = \sum_{l \in S'} \mu_l x^l$, where S' denotes the corresponding index set, $x^l \in \{0,1\}^V$ and $s^l := A^T x^l - \mathbb{1} \geq 0$. We obtain

$$\mathbb{1} + s^* = Ax^* = \sum_{l \in S'} \mu_l A x^l = \sum_{l \in S'} \mu_l (\mathbb{1} + s^l) = \mathbb{1} + \sum_{l \in S'} \mu_l s^l,$$

implying

$$\begin{pmatrix} x^* \\ s^* \end{pmatrix} = \sum_{l \in S'} \mu_l \begin{pmatrix} x^l \\ s^l \end{pmatrix} : \ \sum_{l \in S'} \mu_l = 1, \ \mu_l > 0, \ (x^l, s^l) \in \mathrm{Feas}(P).$$

We now consider the corresponding matrix

$$\begin{pmatrix} W_1^l & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & W_n^l \end{pmatrix} \in \mathrm{LFeas}_3(P), \text{ where } W_i^l = \begin{pmatrix} 1 \\ x^l \\ s_i^l \end{pmatrix} \begin{pmatrix} 1 \\ x^l \\ s_i^l \end{pmatrix}^T.$$

And thus, recalling that the value of L is only affected by x^* as a part of $Z_1^* = \sum_{l \in S'} \mu_l W_1^l$

$$\begin{aligned} \mathrm{opt}(CVV) = L \left(\begin{pmatrix} Z_1^* & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & Z_n^* \end{pmatrix} \right) &= \sum_{l \in S'} \mu_l L \left(\begin{pmatrix} W_1^l & 0 & ... & 0 \\ 0 & ... & ... & 0 \\ 0 & ... & 0 & W_n^l \end{pmatrix} \right) \\ \geq \sum_{l \in S'} \mu_l \min_{Y \in \mathrm{LFeas}_3(P)} L(Y) = \mathrm{opt}(P). \end{aligned}$$

For (ii):
If we consider the solutions $(x^l, s^l) \in \mathrm{Feas}(P)$ from part i), then we know $\mathbb{1}^T x^l \geq \mathrm{opt}(P)$. Further we conclude

$$\mathrm{opt}(P) \overset{(i)}{=} \mathrm{opt}(CVV) = \sum_{k \in S'} \mu_l \mathbb{1}^T x^l, \text{ where } \mathbb{1}^T x^l \geq \mathrm{opt}(P).$$

Thus we have $\mathbb{1}^T x^l = \mathrm{opt}(P)$, which means that (x^l, s^l) are optimal solutions of (P). Finally the equation $(x^*, s^*) = \sum_{l \in S'} \mu_l (x^l, s^l)$ finishes the proof. □

To obtain the corresponding copositive program we have to dualize (CVV).

Theorem 3.14. *The dual formulation for* (CVV) *is:*

$$\begin{aligned}
\sup \; & \sum_{k \in V} (2(X_k)_{1,n+2} - (X_k)_{11} - (X_k)_{n+2,n+2}) && \text{(DVV)} \\
s.t. \; & \sum_{k \in V} (X_k)_{1+i,1+j} + \sum_{\{k,i\} \in E, \{k,j\} \in E} (X_k)_{n+2,n+2} = 0 && \forall i < j \\
& \sum_{k \in V} ((X_k)_{1+i,1+i} + 2(X_k)_{1,1+i}) + \\
& \sum_{\{k,i\} \in E} (2(X_k)_{1,n+2} - (X_k)_{n+2,n+2}) = 1 && \forall i \in V \\
& (X_k)_{n+2,1+i} + (X_k)_{n+2,n+2} = 0 && \forall \{i,k\} \in E \\
& (X_k)_{n+2,1+i} = 0 && \forall \{i,k\} \in \bar{E} \\
& X_k \in \mathcal{C}_{n+2} && \forall k \in V
\end{aligned}$$

Proof. Let us recall the primal formulation of (CVV):

$$\begin{aligned}
\min \; & \mathbb{1}^T x && \text{(CVV)} \\
\text{s.t.} \; & \left(A_i^T, -1\right) \begin{pmatrix} x \\ s_i \end{pmatrix} = 1 && \forall\, i \in V \\
& \langle \begin{pmatrix} A_i \\ -1 \end{pmatrix} \begin{pmatrix} A_i \\ -1 \end{pmatrix}^T, \begin{pmatrix} X & S_i \\ S_i^T & \tilde{S}_i \end{pmatrix} \rangle = 1 && \forall\, i \in V \\
& x_i = X_{ii} && \forall\, i \in V \\
& \begin{pmatrix} 1 & x^T & s_i \\ x & X & S_i \\ s_i & S_i^T & \tilde{S}_i \end{pmatrix} \in \mathcal{C}^*_{n+2} && \forall\, i \in V
\end{aligned}$$

If we use standard notation and denote $Z_k := \begin{pmatrix} z_k & x^T & s_k \\ x & X & S_k \\ s_k & S_k^T & \tilde{S}_k \end{pmatrix}$ (consider constraints

(3.0.9) respectively (3.0.10) below) this is equivalent to:

$$
\begin{aligned}
\min\ & \langle \frac{1}{2}\begin{pmatrix} 0 & \mathbb{1}^T & 0 \\ \mathbb{1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, Z_1\rangle && && \text{(CVV2)} \\
\text{s.t. } & \langle \begin{pmatrix} 0 & A_k^T & -1 \\ A_k & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, Z_k\rangle = \ 2 && \forall\, k \in V && \\
& \langle \begin{pmatrix} 0 & 0^T & 0 \\ 0 & A_k A_k^T & -A_k \\ 0 & -A_k^T & 1 \end{pmatrix}, Z_k\rangle = \ 1 && \forall\, k \in V && \\
& \langle \begin{pmatrix} 0 & e_k^T & 0 \\ e_k & -2e_k e_k^T & 0 \\ 0 & 0^T & 0 \end{pmatrix}, Z_1\rangle = 0 && \forall\, k \in V && \\
& \langle \begin{pmatrix} 1 & 0^T & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, Z_1\rangle = 1 && && (3.0.9) \\
& \langle \begin{pmatrix} e_i e_j^T + e_j e_i^T & 0 \\ 0^T & 0 \end{pmatrix}, Z_1 - Z_k\rangle = 0 && \forall\, i,j,k \in V,\ k \neq 1 && (3.0.10) \\
& Z_k \in \ \mathcal{C}_{n+2}^* && \forall\, k \in V &&
\end{aligned}
$$

By dualizing we obtain:

$$
\begin{aligned}
\max\ & \sum_{k\in V}(2y_k^1 + y_k^2) + y_1^3 \qquad\qquad \text{(DVV1)} \\
\text{s.t. } & \frac{1}{2}\begin{pmatrix} 0 & \mathbb{1}^T & 0 \\ \mathbb{1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} - y_1^1 \begin{pmatrix} 0 & A_1^T & -1 \\ A_1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} - y_1^2 \begin{pmatrix} 0 & 0^T & 0 \\ 0 & A_1 A_1^T & -A_1 \\ 0 & -A_1^T & 1 \end{pmatrix} - y_1^3 \begin{pmatrix} 1 & 0^T & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
& - \sum_{k\in V} y_k^4 \begin{pmatrix} 0 & e_k^T & 0 \\ e_k & -2e_k e_k^T & 0 \\ 0 & 0^T & 0 \end{pmatrix} - \sum_{\substack{i,j,k\in V \\ k\geq 2}} y_{ijk}^5 \begin{pmatrix} e_i e_j^T + e_j e_i^T & 0 \\ 0^T & 0 \end{pmatrix} \in \ \mathcal{C}_{n+2} \\
& \sum_{i,j\in V} y_{ijk}^5 \begin{pmatrix} e_i e_j^T + e_j e_i^T & 0 \\ 0^T & 0 \end{pmatrix} - y_k^1 \begin{pmatrix} 0 & A_k^T & -1 \\ A_k & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \\
& \qquad - y_k^2 \begin{pmatrix} 0 & 0^T & 0 \\ 0 & A_k A_k^T & -A_k \\ 0 & -A_k^T & 1 \end{pmatrix} \in \ \mathcal{C}_{n+2}\ \forall k \in V \setminus \{1\}
\end{aligned}
$$

Let us denote $Y_k^5 = \sum_{i,j\in V} y_{ijk}^5(e_ie_j^T + e_je_i^T)$. Also we denote X_k as follows:

$$
\begin{aligned}
\max\ & \sum_{k\in V}(2y_k^1 + y_k^2) + y_1^3 && \text{(DVV2)}\\
\text{s.t.}\ & \\
& X_1 = \begin{pmatrix} -y_1^3 & \frac{1}{2}\mathbb{1}^T - y_1^1A_1^T - (y^4)^T & y_1^1 \\ \frac{1}{2}\mathbb{1} - y_1^1A_1 - y^4 & -y_1^2A_1A_1^T + 2\operatorname{diag}(y^4) & y_1^2A_1 \\ y_1^1 & y_1^2A_1^T & -y_1^2 \end{pmatrix} \\
& \qquad\qquad - \sum_{k\in V\setminus\{1\}} \begin{pmatrix} Y_k^5 & 0 \\ 0^T & 0 \end{pmatrix} \in \mathcal{C}_{n+2} \\
& X_k = \begin{pmatrix} Y_k^5 & 0 \\ 0^T & 0 \end{pmatrix} + \begin{pmatrix} 0 & -y_k^1A_k^T & y_k^1 \\ -y_k^1A_k & -y_k^2A_kA_k^T & y_k^2A_k \\ y_k^1 & y_k^2A_k^T & -y_k^2 \end{pmatrix} \in \mathcal{C}_{n+2}\ \forall k \in V\setminus\{1\}.
\end{aligned}
$$

Observe that $(X_k)_{ij}$ are free entries (except for symmetry) for all $i,j \in \{1,...,n+1\}$ and thus we obtain:

$$
\begin{aligned}
\max\ & \sum_{k\in V}(2y_k^1 + y_k^2) + y_1^3 \\
\text{s.t.}\ & \\
& X_1 = \begin{pmatrix} -y_1^3 & \frac{1}{2}\mathbb{1}^T - \sum_{k\in V} y_k^1A_k^T - (y^4)^T & \sum_{k\in V} y_k^1 \\ \frac{1}{2}\mathbb{1} - \sum_{k\in V} y_k^1A_k - y^4 & -\sum_{k\in V} y_k^2A_kA_k^T + 2\operatorname{diag}(y^4) & \sum_{k\in V} y_k^2A_k \\ \sum_{k\in V} y_k^1 & \sum_{k\in V} y_k^2A_k^T & -\sum_{k\in V} y_k^2 \end{pmatrix} \\
& \qquad - \sum_{k\in V\setminus\{1\}} X_k \\
& (X_k)_{n+2,1+i} + (X_k)_{n+2,n+2} = 0 && \forall\{i,k\}\in E,\ k\neq 1 \quad (3.0.11)\\
& (X_k)_{n+2,1+i} = 0 && \forall\{i,k\}\in \bar{E},\ k\neq 1 \quad (3.0.12)\\
& X_k \in \mathcal{C}_{n+2} && \forall k\in V\setminus\{1\} \quad (3.0.13)
\end{aligned}
$$

We observe that except for the central block of X_1, there are always free variables left or we can handle them by using restrictions (3.0.11), (3.0.12) and (3.0.13) for $k = 1$. Hence the central block of X_1, denoted by $\tilde{X}_1$, provides a couple of other restrictions. If we denote $\tilde{X}_k$ analogously we derive:

$$
\begin{aligned}
\tilde{X}_1 &= \sum_{k\in V} -y_k^2A_kA_k^T + 2\operatorname{diag}(y^4) - \sum_{k\in V\setminus\{1\}} \tilde{X}_k \\
&= \sum_{k\in V}(X_k)_{n+2,n+2}A_kA_k^T + 2\operatorname{diag}(y^4) - \sum_{k\in V\setminus\{1\}} \tilde{X}_k.
\end{aligned}
$$

Further we have for y^4:

$$
\begin{aligned}
(y^4)^T &= \frac{1}{2}\mathbb{1}^T - \sum_{k\in V} y_k^1A_k^T - \sum_{k\in V}((X_k)_{1,2},...,(X_k)_{1,n+1}) \\
&= \frac{1}{2}\mathbb{1}^T - \sum_{k\in V}(X_k)_{1,n+2}A_k^T - \sum_{k\in V}((X_k)_{1,2},...,(X_k)_{1,n+1}).
\end{aligned}
$$

And thus we obtain for $\tilde{X}_1$

$$\tilde{X}_1 = \sum_{k\in V}(X_k)_{n+2,n+2}A_kA_k^T + I - 2\sum_{k\in V}(X_k)_{1,n+2}\operatorname{diag}(A_k) - 2\sum_{k\in V}\operatorname{diag}\left(\begin{pmatrix}(X_k)_{1,2}\\ \dots\\ (X_k)_{1,n+1}\end{pmatrix}\right) - \sum_{k\in V\setminus\{1\}}\tilde{X}_k,$$

which implies

$$\sum_{k\in V}\tilde{X}_k - \sum_{k\in V}(X_k)_{n+2,n+2}A_kA_k^T + 2\sum_{k\in V}\left((X_k)_{1,n+2}\operatorname{diag}(A_k) + \operatorname{diag}\left(\begin{pmatrix}(X_k)_{1,2}\\ \dots\\ (X_k)_{1,n+1}\end{pmatrix}\right)\right) = I.$$

This we can split up in two restrictions:

$$\sum_{k\in V}(X_k)_{1+i,1+j} - \sum_{\{i,k\},\{j,k\}\in E}(X_k)_{n+2,n+2} = 0\ \forall\ i<j$$

$$\sum_{k\in V}(X_k)_{1+i,1+i} - \sum_{\{i,k\}\in E}(X_k)_{n+2,n+2} + 2\sum_{\{i,k\}\in E}(X_k)_{1,n+2} + 2\sum_{k\in V}(X_k)_{1,1+i} = 1\ \forall\ i\in V.$$

The resulting program is:

$$\begin{aligned}
\max\ & \sum_{k\in V}(2(X_k)_{1,n+2} - (X_k)_{n+2,n+2} - (X_k)_{11}) & \\
\text{s.t.}\ & \sum_{k\in V}(X_k)_{1+i,1+j} - \sum_{\{i,k\},\{j,k\}\in E}(X_k)_{n+2,n+2} = 0 & \forall\ i<j\\
& \sum_{k\in V}(X_k)_{1+i,1+i} - \sum_{\{i,k\}\in E}(X_k)_{n+2,n+2} + & \\
& 2\sum_{\{i,k\}\in E}(X_k)_{1,n+2} + 2\sum_{k\in V}(X_k)_{1,1+i} = 1 & \forall\ i\in V\\
& (X_k)_{n+2,1+i} + (X_k)_{n+2,n+2} = 0 & \forall\{i,k\}\in E\\
& (X_k)_{n+2,1+i} = 0 & \forall\{i,k\}\in \bar{E}\\
& X_k \in \mathcal{C}_{n+2} & \forall k\in V
\end{aligned} \tag{3.0.14}$$

□

Similarly to (D2V) we need to verify that this dual formulation (DVV) provides the same objective value as (CVV), i.e. that strong duality holds. We can prove this by checking that Theorem 2.27 can be applied to (DVV1) and thus strong duality holds for each equivalent program in the above proof, e.g. for the programs (CVV) and (DVV).

Theorem 3.15. *Let $y^1 = 0$, $y^2 = -\mathbb{1}$, $y^3_1 = -1$, $y^4 = \frac{1}{2}\mathbb{1}$, $y^5_{iik} = \frac{1}{2}$ and $y^5_{ijk} = 0$, $i \neq j$ then the corresponding matrices in* (DVV1) *are interior points of the feasibility set of* (DVV1).

Proof.

$$
\begin{aligned}
&\frac{1}{2}\begin{pmatrix} 0 & \mathbb{1}^T & 0 \\ \mathbb{1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} - y_1^1 \begin{pmatrix} 0 & A_1^T & -1 \\ A_1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} - y_1^2 \begin{pmatrix} 0 & 0^T & 0 \\ 0 & A_1A_1^T & -A_1 \\ 0 & -A_1^T & 1 \end{pmatrix} - y_1^3 \begin{pmatrix} 1 & 0^T & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
&- \sum_{k\in V} y_k^4 \begin{pmatrix} 0 & e_k^T & 0 \\ e_k & -2e_ke_k^T & 0 \\ 0 & 0^T & 0 \end{pmatrix} - \sum_{i,j,k\in V, k\geq 2} y_{ijk}^5 \begin{pmatrix} e_ie_j^T + e_je_i^T & 0 \\ 0^T & 0 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 0 & \mathbb{1}^T & 0 \\ \mathbb{1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ A_1 \\ -1 \end{pmatrix}\begin{pmatrix} 0 \\ A_1 \\ -1 \end{pmatrix}^T + \begin{pmatrix} 1 & 0^T & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & \frac{1}{2}\mathbb{1}^T & 0 \\ \frac{1}{2}\mathbb{1} & -I & 0 \\ 0 & 0^T & 0 \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0^T & 0 \\ 0 & \frac{1}{2}I & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 1 & 0^T & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ A_1 \\ -1 \end{pmatrix}\begin{pmatrix} 0 \\ A_1 \\ -1 \end{pmatrix}^T \in \operatorname{Int}(\mathcal{C}_{n+2})
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{i,j\in V} y_{ijk}^5 \begin{pmatrix} e_ie_j^T + e_je_i^T & 0 \\ 0^T & 0 \end{pmatrix} - y_k^1 \begin{pmatrix} 0 & A_k^T & -1 \\ A_k & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} - y_k^2 \begin{pmatrix} 0 & 0^T & 0 \\ 0 & A_kA_k^T & -A_k \\ 0 & -A_k^T & 1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} 1 & 0^T & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 \\ A_k \\ -1 \end{pmatrix}\begin{pmatrix} 0 \\ A_k \\ -1 \end{pmatrix}^T \in \operatorname{Int}(\mathcal{C}_{n+2}) \ \forall k \in V \setminus \{1\}
\end{aligned}
$$

We can verify the strict copositivity for the last matrices similarly to the end of the proof in Theorem 3.10. □

We have obtained two copositive formulations of DOMINATING SET in standard form. It is possible to efficiently consider every constraint in a solver, such as CSDP, except for the copositivity constraints (3.0.14). Checking these constraints is an NP-hard problem whereas checking positive semidefiniteness of a matrix can be checked in polynomial time (for example using Gaussian elimination). To obtain lower bounds for DOMINATING SET in polynomial time we need to find a semidefinite relaxation for (CVV), i.e. a semidefinite program with a larger set of feasible solutions. This corresponds to a semidefinite program whose feasible set is contained in the feasible set of the dual program (DVV). To find such an approximation or more precisely a hierarchy of such approximations is content of the next chapter.

4 An Approximation Algorithm

With the information given in the chapters before and some additional information we are able to compute an approximate solution of DOMINATING SET. This chapter is organized as follows: First we use a hierarchy of semidefinite programs developed by P. Parrilo [21] to be able to calculate a sequence of lower bounds for (CVV) and (C2V) respectively their dual copositve formulations. Algorithms to solve these programs are given via pseudocode in the appendix (see Sections 6.3 and 6.4), the implementation was done in matlab. They consist of a code for creating the semidefinite program and using the solver CSDP to solve it via an interior point method. After that we will use the fact that (DVV) belongs to a special class of programs, so-called G-invariant programs, where G is a group acting on the underlying matrices. We close with illustrating that the corresponding modified program is applicable to the "football pools" example introduced in Chapter 2 (see Subsection 2.3.1). This provides the opportunity of a modified algorithm with enhanced performance for certain cases, such as "football pools".

4.1 A Semidefinite Test for Copositivity

The following two sections are based on the PHD thesis of P. Parrilo ([21], chapters 4 and 5) and provide an increasingly powerful hierarchy of sufficient conditions for copositivity. Parrilo's approach was obtained through the use of the sum of squares decomposition for multivariate forms. Thus we begin this section by introducing some background information on the theory of sum of squares.

Definition 4.1. A *n-variate polynomial* f in $x_1, ..., x_n$ with coefficients in the real numbers is a finite linear combination of monomials:

$$f = \sum_{\alpha} c_\alpha x^\alpha = \sum_{\alpha} c_\alpha x_1^{\alpha_1} \cdots x_n^{\alpha_n}, \quad c_\alpha \in \mathbb{R},$$

where the sum is over a finite number of n-tuples $\alpha = (\alpha_1, ..., \alpha_n),\ \alpha_i \in \mathbb{N}_0$. We further define the *degree* of f as $\max\limits_{\alpha} \sum_{i=1}^n \alpha_i$. The set of all polynomials in $x_1, ..., x_n$ with coefficients in $\mathbb{R}$ is denoted $\mathbb{R}[x_1, ..., x_n]$.

Definition 4.2. A *form* is a polynomial where all the monomials have the same degree $d := \sum_i \alpha_i$. In this case, the polynomial is homogeneous of d, since it satisfies $f(\lambda x_1, ..., \lambda x_n) = \lambda^d f(x_1, ..., x_n)$.

The following example illustrates that checking the nonnegativity of forms helps us to check whether a matrix is copositive or not.

Example 4.3. Recall that a matrix M is copositive if and only if $x^T M x \geq 0$ for all $x \in \mathbb{R}^n_{\geq 0}$. Furthermore if we denote $(y_i)^2 = x_i,\ \mathbf{y} = (y_1^2, ..., y_n^2)$ we have

$p_M(\mathbf{y}) = \mathbf{y}^T M \mathbf{y} \geq 0$ for all $y_i \in \mathbb{R}$. Now we observe that $p_M(\mathbf{y}) = \sum_{i,j} m_{ij} y_i^2 y_j^2$ is a multivariate form with degree $d = 4$. This means checking whether the matrix M is copositive is equivalent to checking whether the form $p_M(\mathbf{y})$ is non-negative.

Of course in general this problem remains (for $d \geq 4$) NP-hard. This means there are no established methods to obtain the correct answer for every possible instance in polynomial time. Nevertheless we will show that there are some conditions, guaranteeing global nonnegativity of a form, which can be checked in polynomial time. One of these conditions is the existence of a sum of squares decomposition for the polynomial f, i.e.:

$$f(x) = \sum_i f_i^2(x)$$

We observe that the existence of such a decomposition provides the nonnegativity of f for all values of $x \in \mathbb{R}$. The sum of squares decomposition will become the fundamental method used in the upcoming algorithm. It is mainly based on the first part of the "Gram matrix" method presented in [23], which is a useful tool to decide whether a polynomial can be expressed as a sum of squares or not.
The basic idea in Parrilo's paper is that it is possible to express a polynomial f of degree $2d$ as a quadratic form in the following way. Let z consist of all monomials of degree less than or equal to d. Then we have the representation:

$$f(x) = z^T Q z \tag{4.1.1}$$

where Q is a coefficient matrix for f. A possible positive semidefiniteness of Q would imply $f(x) \geq 0$ for all x. A major problem that arises from this kind of representation is that Q might not be unique due to the fact that the z_i are not independent variables. This causes the possibility that Q may not be positive semidefinite for certain representations of f but for others. Hence it would be useful to restrict the matrices Q to a (smaller) subspace in which the existence of such a matrix Q is equivalent to the existence of a sum of squares decomposition of f (implying nonnegativity). This subspace is obtained by handling the dependence of the z_i with the help of constraints of the form $z_i z_j = z_k z_l$, respectively $z_i^2 = z_k z_l$. It can be shown that these constraints provide a linear subspace of matrices Q that satisfies (4.1.1). If there is a positive semidefinite matrix Q' in this subspace then we can conclude that f can be decomposed as a sum of squares. The reason for this is that given an eigenvalue decomposition $Q' = T\,\mathrm{diag}(d)T^T,\ d_i \geq 0$ this implies $f(x) = \sum_i d_i (Tz)_i^2$. The reverse case would be that if we know that f can be expressed as a sum of squares of polynomials then splitting up the sum of squares to monomials provides the representation (4.1.1). For an example to illustrate the paragraph above for the form $f(x_1, x_2) = (x_1^2 - x_2^2)^2$ see Example 4.5.

Example 4.3 (Continuation)**.** We recall that the degree of the monomials was $d = 4$ and every monomial $y_i^2 y_j^2$ can be expressed as a product of the monomials with degree $d = 2$. If we store all the monomials of degree 2 in a vector $\mathbf{Y}$ this implies that we have a representation of $p_M(\mathbf{y})$ in the way:

$$p_M(\mathbf{y}) = \sum_{i,j} m_{ij} y_i^2 y_j^2 = \mathbf{Y}^T Q \mathbf{Y}$$

Furthermore monomials of degree $d < 2$ cannot provide monomials of degree 4, thus we can neglect these kind of monomials in our vector $\mathbf{Y}$. The dependence of the $\mathbf{Y}_i$ is caused by the identities of the form

$$\begin{aligned}(y_iy_j)^2 &= (y_i^2)(y_j^2)\\ (y_iy_j)(y_iy_k) &= (y_i^2)(y_jy_k)\\ (y_iy_j)(y_ky_l) &= (y_iy_k)(y_jy_l) = (y_iy_l)(y_jy_k).\end{aligned}$$

Observe that only the first kind of identities have coefficients m_{ij} in $p_M(\mathbf{y})$ not equal to zero. We denote the associated multipliers in matrix Q by $q_{ij},\ q_{ijk},\ q_{ijkl}$ and q'_{ijkl}. If we further group the variables in $\mathbf{Y}$ in the way y_i^2 first, y_iy_j second we obtain a matrix Q of the following structure

$$Q = \begin{pmatrix} m_{11} & m_{12}-q_{12} & \dots & m_{1n}-q_{1n} & * & * & \dots & * \\ m_{12}-q_{12} & m_{22} & \dots & m_{2n}-q_{2n} & * & * & \dots & * \\ \dots & \dots & \dots & \dots & * & * & \dots & * \\ m_{1n}-q_{1n} & m_{2n}-q_{2n} & \dots & m_{nn} & * & * & \dots & * \\ * & * & \dots & * & 2q_{12} & * & \dots & * \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ * & * & \dots & * & * & * & \dots & 2q_{(n-1)n} \end{pmatrix},$$

where the asterisks symbolize either zero or linear combinations of $q_{ijk},\ q_{ijkl}$ and q'_{ijkl}. Because of our observation above that the coefficients in $p_M(\mathbf{y})$ associated to the above entries of Q are zero we obtain that the sum of each of the above entries is zero. This implies that we can choose these entries to be zero without changing the value of $\mathbf{Y}^TQ\mathbf{Y}$. This choice leads us to the linear matrix inequality system

$$\begin{pmatrix} m_{11} & m_{12}-q_{12} & \dots & m_{1n}-q_{1n} \\ m_{12}-q_{12} & m_{22} & \dots & m_{2n}-q_{2n} \\ \dots & \dots & \dots & \dots \\ m_{1n}-q_{1n} & m_{2n}-q_{2n} & \dots & m_{nn} \end{pmatrix} \in \mathcal{S}^n_{\succeq 0},\ \ q_{ij} \geq 0.$$

This is equivalent to show the existence of a positive semidefinite matrix P and a componentwise non-negative matrix N such that $M = P + N$, which is a well-known sufficient condition for the copositivity of M. For more details concerning the equivalence of these tests Parrilo suggests the paper of M. D. Choi and T. Y. Lam ([4], especially Lemma 3.5).

4.2 Parrilo's Hierarchy

We have seen so far that the above theory is applicable to design a sufficient test for checking copositivity. The main drawback of the method seen so far is that we need stronger conditions to be able to check copositivity more precisely. This can be done by a hierarchy of increasingly powerful tests, which was developed by P. Parrilo [21]. To do this we consider the $2(r+2)$-forms

$$p_r(\mathbf{y}) = \left(\sum_{i=1}^n y_i^2\right)^r p_M(\mathbf{y}).$$

There are two main facts, we obtain from this representation. First if p_r is a sum of squares $p_{r+1}(\mathbf{y}) = \sum_{i=1}^n y_i^2 p_r(\mathbf{y})$ is a sum of squares but not necessarily the converse proposition holds. Second the inequality $p_r(\mathbf{y}) \geq 0$ implies $p_M(\mathbf{y}) \geq 0$. This means by testing whether $p_r(\mathbf{y})$ is a sum of squares we check a stricter condition implying copositivity of M. Let $A_r := \{M \in \mathcal{S}^n : \ p_r(\mathbf{y}) \text{ is SOS}\}$, e.g. $A_0 = \mathcal{C}$ then the two facts translate to

$$A_r \subseteq A_{r+1} \subseteq \mathcal{C}.$$

Thus it is possible to sharpen the lower bounds for DOMINATING SET with increasing r.
Let us denote $|\alpha| = \sum_{i=1}^n \alpha_i$ and for $m \in \mathbb{N}_0$ denote $\Lambda_m := \{(\alpha_1, ..., \alpha_n) \in \mathbb{N}_0^n : |\alpha| = m\}$. Let us assume that the finitely many, say k, elements of Λ_m are ordered, i.e. $\Lambda_m = \{\beta^1, ..., \beta^k\}$. Following these notations the last part of section 4.1 can be stated for forms with the help of the following theorem, of which a similar form can be found in [23].

Theorem 4.4. *Suppose $f(x) = \sum_{\alpha \in \Lambda_{2m}} a_\alpha x^\alpha$ and $\bar{x} = (x^{\beta_1}, ..., x^{\beta_k})$. Then f is a sum of squares of polynomials in $\mathbb{R}$ if and only if there exists a matrix $Q \in \mathcal{S}^k_{\succeq 0}$ such that*

$$f(x) = \bar{x} Q \bar{x}^T.$$

Proof. If $f = \sum_{i=1}^t h_i^2$ is a sum of squares of polynomials h_i, we have that each monomial is of degree m because otherwise we would have a monomial of degree strictly less than $2m$ in f. Take $Q = HH^T$ with $H = \begin{pmatrix} b^1_{h_1} & ... & b^1_{h_t} \\ ... & ... & .... \\ b^k_{h_1} & ... & b^k_{h_t} \end{pmatrix}$, where $b^j_{h_i}$ is the coefficient corresponding to x^{β_j} in the polynomial h_i. If we assume that there is a matrix $Q \in \mathcal{S}^k_{\succeq 0}$ with $rank(Q) = t$, then there exists an eigenvalue decomposition

$$f(x) = \bar{x} T \operatorname{diag}(d_1, ..., d_t, 0, ..., 0) T^T \bar{x}^T.$$

If we define $h_i := \sqrt{d_i} \sum_{j=1}^k t_{ji} x^{\beta_j}$ we obtain $f = h_1^2 + ... + h_t^2$. □

To illustrate the above construction consider the following example.

Example 4.5. Let $f(x_1, x_2) = x_1^4 - 2x_1^2 x_2^2 + x_2^4$, which is a sum of squares witnessed by the function $h(x_1, x_2) = x_1^2 - x_2^2$. Consider the two representations for $\bar{x} = (x^{(2,0)}, x^{(0,2)}, x^{(1,1)})$

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix} \notin \mathcal{S}^3_{\succeq 0}, \ Q_2 = HH^T = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}^T = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \in \mathcal{S}^3_{\succeq 0},$$

i.e. $f(x_1, x_2) = \bar{x} Q_1 \bar{x}^T$ respectively $f(x_1, x_2) = \bar{x} Q_2 \bar{x}^T$. This means we have two representations $\bar{x} Q \bar{x}^T$ for f, both handling the constraint $x_1^2 x_2^2 = (x_1 x_2)^2$, via $q_{21} + q_{12} + q_{33} = -2$, but only Q_2 satisfies the additional positive semidefiniteness constraint. The reverse conclusion is as follows: Suppose Q_2 is given, then we obtain the eigenvalue decomposition

$$Q_2 = T \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} T^T, \text{ where } T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

If we define $h(x) = \sqrt{2}\left(\frac{1}{\sqrt{2}}x_1^2 - \frac{1}{\sqrt{2}}x_2^2 + 0 \cdot (x_1x_2)\right) = x_1^2 - x_2^2$ we obtain exactly our sum of squares $f(x) = h(x)^2$.

Consider that the form $p_r(\mathbf{y})$, which is homogeneous in $d = 2(r+2)$, can also be expressed with the help of

$$p_r(\mathbf{y}) = \sum_{\alpha \in \Lambda_{2(r+2)}} a_\alpha y^\alpha = \sum_{\alpha \in \Lambda_{2(r+2)}} \sum_{i,j:\ \beta_i + \beta_j = \alpha} y^{\beta_i} Q_{ij} y^{\beta_j}.$$

Comparing coefficients leads to

$$\sum_{i,j:\ \beta_i + \beta_j = \alpha} Q_{ij} = a_\alpha.$$

Similar to Example 4.3 we can reduce the considered part of the matrix Q. First we need to order the β_k.

Definition 4.6. First we define the *1-order* for n-variate forms as

$$(\tilde{\beta}_1^1, ..., \tilde{\beta}_n^1) = (e_1, ..., e_n).$$

Inductively we define an *r-order* of n-variate forms:

$$\tilde{\beta}^{r+1} := (\tilde{\beta}_1^r + e_1, ..., \tilde{\beta}_1^r + e_n, \tilde{\beta}_2^r + e_1, ..., \tilde{\beta}_2^r + e_n, ..., \tilde{\beta}_{n^r}^r + e_1, ..., \tilde{\beta}_{n^r}^r + e_n).$$

Definition 4.7. Consider the n-variate monomials of degree 2 ordered in the way

$$x_1^2, ..., x_n^2, x_1x_2, ..., x_{n-1}x_n.$$

The corresponding degrees are

$$\beta^* = (\beta_1^*, ..., \beta_n^*, \beta_{n+1}^*, ..., \beta_k^*) = (2e_1, ..., 2e_n, e_1 + e_2, ..., e_{n-1} + e_n).$$

We consider $\tilde{\beta}^r$ and add the vector β^* component-by-component to each entry of $\tilde{\beta}^r$. Reordering of the resulting vector defines the vector β^r of $r+2$ degrees by

$$\beta^r := (\tilde{\beta}_1^r + 2e_1, ..., \tilde{\beta}_1^r + 2e_n, \tilde{\beta}_2^r + 2e_1, ..., \tilde{\beta}_2^r + 2e_n, ..., \tilde{\beta}_{n^r}^r + 2e_1, ..., \tilde{\beta}_{n^r}^r + 2e_n, \beta_{end}^r),$$

where β_{end}^r is a vector of degrees in Λ_{r+2}, which haven't been considered in the first n^{r+1} components, ordered in lexicographic ascending order. Let us further define $\alpha^r := 2\beta^r$ and $\alpha_{end}^r := 2\beta_{end}^r$.

Remark 4.8. The order within α_{end}^r and β_{end}^r does not affect the following theorems.

Definition 4.9. Let $i_1, ..., i_k \in \{1, ..., n\}$ be a set of indices. Define the k-dimensional *multiindex number K_n* inductively:

$$K_n(i_1) = i_1,\ K_n(i_1, ..., i_k) = n(K_n(i_1, ..., i_{k-1}) - 1) + i_k.$$

Remark 4.10. The r-dimensional multiindex number $K_n(i_1, ..., i_r)$ helps us to denote the degrees $\tilde{\beta}_{K_n(i_1,...,i_r)}^r = e_{i_1} + ... + e_{i_r}$ and further $\beta_{K_n(i_1,...,i_{r+1})}^r = e_{i_1} + ... + e_{i_r} + 2e_{i_{r+1}}$.

Similar to Parrilo's (see [21]) theorem for the case $r = 1$ we deduce the basic theorem for the upcoming algorithms.

Theorem 4.11. *Let us denote* $I_\alpha = \{(i,j,k) \in [n] \times [n] \times [n^r] : \beta^r_{n(k-1)+i} + \beta^r_{n(k-1)+j} = \alpha\}$. *Consider the system of linear matrix inequalities given by:*

$$M - \Lambda^k \succeq 0 \qquad \forall k \in \{1, ..., n^r\}$$

$$\sum_{(i,j,k)\in I_\alpha} \Lambda^k_{ij} = 0 \qquad \forall \alpha \in \alpha^r \setminus \alpha^r_{end} \tag{4.2.1}$$

$$\sum_{(i,j,k)\in I_\alpha} \Lambda^k_{ij} \geq 0 \qquad \forall \alpha \in \alpha^r_{end} \tag{4.2.2}$$

with n^r *matrices* $\Lambda^k \in \mathcal{S}^n$. *If there exists a feasible solution, then* M *is copositive. Furthermore, this test is at least as powerful as the same test for* $r-1$.

Proof. Let us denote $\mathbf{j}$ the $r-1$-tuple $(j_3, ..., j_{r+1})$. First we show the nonnegativity of $p_r(\mathbf{y})$. We denote w.l.o.g $k = K_n(\mathbf{j}, j_{r+2})$. Consider

$$\sum_{j_1,j_2\in[n]} m_{j_1,j_2} y^2_{j_1} y^2_{j_2} \geq \sum_{j_1,j_2\in[n]} \Lambda^{K_n(\mathbf{j},j_{r+2})}_{j_1,j_2} y^2_{j_1} y^2_{j_2} \ \forall j_3, ..., j_{r+2} \in \{1, ..., n\},$$

which implies

$$\begin{aligned}
p_r(\mathbf{y}) &= \sum_{j_1,...,j_{r+2}\in[n]} m_{j_1,j_2} y^2_{j_1} y^2_{j_2} \cdot ... \cdot y^2_{j_{r+2}} \\
&\geq \sum_{j_1,...,j_{r+2}\in[n]} \Lambda^{K_n(\mathbf{j},j_{r+2})}_{j_1,j_2} y^2_{j_1} y^2_{j_2} \cdot ... \cdot y^2_{j_{r+2}} \\
&= \sum_{j_1,...,j_{r+2}\in[n]} \Lambda^{K_n(\mathbf{j},j_{r+2})}_{j_1,j_2} y^{2e_{j_1}+\sum_{l=3}^{r+2} e_{j_l}} y^{2e_{j_2}+\sum_{l=3}^{r+2} e_{j_l}} \\
&= \sum_{j_1,...,j_{r+2}\in[n]} \Lambda^{K_n(\mathbf{j},j_{r+2})}_{j_1,j_2} y^{\beta^r_{K_n(\mathbf{j},j_{r+2},j_1)}} y^{\beta^r_{K_n(\mathbf{j},j_{r+2},j_2)}}.
\end{aligned}$$

Thus each $\alpha \in \alpha^r$ can be expressed in the form

$$\alpha = \beta^r_{n(k-1)+i} + \beta^r_{n(k-1)+j},$$

where $\beta^r_{n(k-1)+i}$ and $\beta^r_{n(k-1)+j}$ are no entries in β^r_{end}. The sum above equals

$$\sum_{(i,j,k)\in I_\alpha:\ \alpha\in\alpha^r\setminus\alpha^r_{end}} \Lambda^k_{ij} y^\alpha + \sum_{(i,j,k)\in I_\alpha:\ \alpha\in\alpha^r_{end}} \Lambda^k_{ij} y^\alpha = \sum_{(i,j,k)\in I_\alpha:\ \alpha\in\alpha^r_{end}} \Lambda^k_{ij} y^\alpha \geq 0,$$

due to restrictions (4.2.1) and (4.2.2), implying the copositivity of M.
Second we show that if the system in Theorem 4.11 has a feasible solution $\tilde{\Lambda}$ for $r-1$ then this implies that the system has a feasible solution for r. Denoting $\alpha(i) := \alpha - 2e_i$, we set

$$\Lambda^{K_n(\mathbf{j},j_{r+2})}_{j_1,j_2} = \tilde{\Lambda}^{K_n(\mathbf{j})}_{j_1,j_2} \text{ for } j_1 \neq j_2$$

and

$$\Lambda_{i,i}^{K_n(\mathbf{j},j_{r+2})} = \tilde{\Lambda}_{i,i}^{K_n(\mathbf{j})} - h_i(\mathbf{j}, j_{r+2}, \tilde{\Lambda}),$$

where for $\alpha = 2\beta^r_{n(K_n(\mathbf{j},j_{r+2})-1)+i}$

$$h_i(\mathbf{j}, j_{r+2}, \tilde{\Lambda}) = \frac{1}{|(i,i,K_n(\mathbf{j}, j_{r+2})) \in I_\alpha|} \sum_{(i,j,K_n(\mathbf{j}))\in I_{\alpha(j_{r+2})}} \tilde{\Lambda}_{i,j}^{K_n(\mathbf{j})}.$$

Observe that

$$\alpha(j_{r+2}) = \alpha - 2e_{j_{r+2}} = 2\beta^r_{n(K_n(\mathbf{j},j_{r+2})-1)+i} - 2e_{j_{r+2}} = 2\beta^{r-1}_{n(K_n(\mathbf{j})-1)+i} \in \alpha^{r-1}$$

and thus implies

$$\sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_{\alpha(j_{r+2})}} \tilde{\Lambda}_{i,j}^{K_n(\mathbf{j})} \geq 0.$$

Hence $h_i \geq 0$ and thus

$$M - \Lambda^k = (M - \tilde{\Lambda}^k) + \operatorname{diag}(h),$$

where both matrices $M - \tilde{\Lambda}^k$ and $\operatorname{diag}(h)$ are positive semidefinite, implying the positive semidefiniteness of the r-system. Furthermore let $\alpha \in \alpha^r \setminus \alpha^r_{end}$ we obtain

$$\begin{aligned}
\sum_{(i,j,k)\in I_\alpha} \Lambda_{ij}^k &= \sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \Lambda_{ij}^{K_n(\mathbf{j},j_{r+2})} \\
&= \sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_\alpha:\ i\neq j} \Lambda_{ij}^{K_n(\mathbf{j},j_{r+2})} + \sum_{(i,i,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \Lambda_{ii}^{K_n(\mathbf{j},j_{r+2})} \\
&= \sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \tilde{\Lambda}_{ij}^{K_n(\mathbf{j})} - \sum_{(i,i,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \frac{\sum_{(i,j,K_n(\mathbf{j}))\in I_{\alpha(j_{r+2})}} \tilde{\Lambda}_{ij}^{K_n(\mathbf{j})}}{|(i,i,K_n(\mathbf{j},j_{r+2}))\in I_\alpha|} \\
&= \sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \tilde{\Lambda}_{ij}^{K_n(\mathbf{j})} - \sum_{(i,j,K_n(\mathbf{j}))\in I_{\alpha(j_{r+2})}} \tilde{\Lambda}_{ij}^{K_n(\mathbf{j})} = 0
\end{aligned}$$

and for the lower restriction, i.e. for $\alpha \in \alpha^r_{end}$ implying $i \neq j$

$$\begin{aligned}
\sum_{(i,j,k)\in I_\alpha} \Lambda_{ij}^k &= \sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \Lambda_{ij}^{K_n(\mathbf{j},j_{r+2})} \\
&= \sum_{(i,j,K_n(\mathbf{j},j_{r+2}))\in I_\alpha} \tilde{\Lambda}_{ij}^{K_n(\mathbf{j})} \\
&= \sum_{(i,j,K_n(\mathbf{j}))\in I_{\alpha(j_{r+2})}} \tilde{\Lambda}_{ij}^{K_n(\mathbf{j})} \geq 0.
\end{aligned}$$

Thus a solution $\tilde{\Lambda}$ for the $r-1$-program provides a solution for the r-program, which makes the r-test more powerful. □

We have shown that this family of systems is less strict with growing parameter r and thus provides a series of increasing lower bounds for DOMINATING SET. For the question of conservativeness for Parrilo's hierarchy we consider the famous theorem of Pólya:

Theorem 4.12 ([21], Theorem 5.3)**.** *Given a form $F(x_1, ..., x_n)$, strictly positive for $x_i \geq 0$, $\sum_{i=1}^{n} x_i > 0$, then F can be expressed as*

$$F = \frac{G}{H},$$

where G and H are forms with positive coefficients. In particular, we can choose

$$H = (x_1 + x_2 + ... + x_n)^r$$

for a suitable r.

We can apply Pólya's theorem to any interior point of (DVV), which is a strictly copositive matrix M, providing for a suitable r:

$$p_r(\mathbf{y}) = p_M(\mathbf{y})(y_1^2 + ... + y_n^2)^r = F(\mathbf{y})H(\mathbf{y}) \text{ is a sum of squares.}$$

Thus Pólya's theorem states that there is a finite r for which the copositive program (DVV) and the more restricted semidefinite program corresponding to $p_r(\mathbf{y})$ have the same interior. Due to the fact that the algorithm will be based on an interior point method to solve the resulting semidefinite relaxation, we obtain the same approximate solution for the r-relaxation of (DVV) as for (DVV) itself. However so far no such r has been found to express r as a polynomial in n, meaning that finding an exact solution for (DVV) remains a very hard problem. Examples of effective bounds for r have been presented by J. de Loera and F. Santos in [6] and [7].
If we combine the copositive formulation (DVV) with Theorem 4.11, we obtain a semidefinite program, whose feasible solutions are contained in the feasible solutions of (DVV):

$$\begin{aligned}
\sup \; & \sum_{k \in V} (2(X_k)_{1,n+2} - (X_k)_{11} - (X_k)_{n+2,n+2}) && \text{(RVV)} \\
\text{s.t.} \; & \sum_{k \in V} (X_k)_{1+i,1+j} + \sum_{\{k,i\} \in E, \{k,j\} \in E} (X_k)_{n+2,n+2} = 0 && \forall i < j \\
& \sum_{k \in V} ((X_k)_{1+i,1+i} + 2(X_k)_{1,1+i}) + \\
& \sum_{\{k,i\} \in E} (2(X_k)_{1,n+2} - (X_k)_{n+2,n+2}) = 1 && \forall i \in V \\
& (X_k)_{n+2,1+i} + (X_k)_{n+2,n+2} = 0 && \forall \{i,k\} \in E \\
& (X_k)_{n+2,1+i} = 0 && \forall \{i,k\} \in \bar{E} \\
& \sum_{(i,j,l) \in I_\alpha} \Lambda_{ij}^{kl} = 0 && \forall \alpha \in \alpha^r \setminus \alpha_{end}^r, \; \forall k \in V \\
& \sum_{(i,j,l) \in I_\alpha} \Lambda_{ij}^{kl} \geq 0 && \forall \alpha \in \alpha_{end}^r, \; \forall k \in V \\
& X_k - \Lambda^{kl} \succeq 0 && \forall l \in \{1, ..., (n+2)^r\}, \; \forall k \in V
\end{aligned}$$

This program can be solved like every semidefinite program with an interior point method in polynomial time. An algorithm to solve (RVV) by using CSDP is given in the appendix (see Section 6.4). Similarly we can combine (D2V) with Theorem 4.11 and obtain:

$$
\begin{aligned}
\sup\ & \mathbb{1}^T \begin{pmatrix} 2X_{1,n+2} \\ \dots \\ 2X_{1,2n+1} \end{pmatrix} - \mathbb{1}^T \begin{pmatrix} X_{n+2,n+2} \\ \dots \\ X_{2n+1,2n+1} \end{pmatrix} - X_{11} && \text{(R2V)} \\
\text{s.t. } & X_{1+i,1+i} + 2 \sum_{j:\{i,j\}\in E} X_{1,n+1+j} + 2X_{1,1+i} - \sum_{j:\{i,j\}\in E} X_{n+1+j,n+1+j} = 1 && \forall i \in V \\
& X_{1+i,1+j} - \sum_{k:\{i,k\}\in E, \{j,k\}\in E} X_{n+1+k,n+1+k} = 0 && \forall i,j \in V,\ i<j \\
& X_{1+i,n+1+j} + X_{n+1+j,n+1+j} = 0 && \forall \{i,j\} \in E \\
& X_{1+i,n+1+j} = 0 && \forall \{i,j\} \notin E \\
& X_{n+1+i,n+1+j} = 0 && \forall i,j \in V,\ i<j \\
& \sum_{(i,j,k)\in I_\alpha} \Lambda_{ij}^k = 0 && \forall \alpha \in \alpha^r \setminus \alpha_{end}^r \\
& \sum_{(i,j,k)\in I_\alpha} \Lambda_{ij}^k \geq 0 && \forall \alpha \in \alpha_{end}^r \\
& X - \Lambda^k \succeq 0 && \forall k \in \{1, \dots, (2n+1)^r\}
\end{aligned}
$$

Again an algorithm to solve (R2V) is given in the appendix (see Section 6.3). To check how precise our approximations are we consider the following table for the case $r = 1$ and the average of the approximation ratios $\frac{\text{opt}(R2V)}{\gamma(G)}$ respectively $\frac{\text{opt}(RVV)}{\gamma(G)}$ for different input instances, i.e. randomized graphs with $|V| = n$.

Qual. for	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$	$n=12$	$n=13$	$n=14$
$\frac{\text{opt}(R2V)}{\gamma(G)}$	0.81	0.97	0.97	0.87	0.98	0.85	0.94	0.87
$\frac{\text{opt}(RVV)}{\gamma(G)}$	0.81	0.97	0.97	0.87	0.98	0.85	0.94	0.87

Table 4.1: Approximation quality of R2V and RVV

The following table illustrates the run time of the two algorithms, where we use a naive exponential algorithm calculating the exact domination number as a comparison. Moreover we use the quotient of the run times of the approximation algorithms and the naive algorithm to illustrate the behaviour of the run time over the different sizes of the input problems.
However this does not seem to be a very efficient way of solving these problems, because it is necessary to use a large number of semidefinite matrices. Thus we run out of memory very quickly (e.g. using a normal desktop computer for $n = 16$ for (RVV)). Nevertheless for some instances of DOMINATING SET it is possible to reduce the memory space needed immensely just as the run time (as shown in Section 4.3). One of the applications allowing this are so-called "covering codes", which are used for example in the field of the European betting pools, such as "football pools".

Time for	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$	$n=12$	$n=13$	$n=14$
App. D2V	11.9	23.5	46.9	95.1	174.1	326.5	562.2	909.3
App. DVV	13.1	30.2	72.6	184.8	428.6	981.1	2122.1	5070.1
Naive	0.005	0.009	0.013	0.027	0.04	0.07	0.127	0.257
D2V/Naive	2185	2513	3709	3473	4311	4670	4421	3532
DVV/Naive	2416	3228	5747	6748	10612	14031	16687	19694

Table 4.2: Behaviour of run times in seconds for CSDP solving programs R2V and RVV

4.3 A Simplification for Symmetric Programs

To be able to reduce the run time and memory needed to solve certain instances of (RVV), we need some further theory, developed by C. Bachoc, D. Gijswijt, A. Schrijver and F. Vallentin [1] on semidefinite programs like:

$$\begin{aligned} \inf\ & \langle C, X\rangle \\ \text{s.t. } & \langle A_i, X\rangle = b_i \ \forall i \in \{1, ..., l\} \\ & X \succeq 0 \\ & X \in \mathbb{R}^{n\times n} \end{aligned} \tag{Psem}$$

Let G be a finite group and $O(n)$ denote the group of orthogonal matrices, i.e. matrices $Q \in \mathbb{R}^{n\times n}$ with property $QQ^T = I_n$. Let $\pi : G \longrightarrow O(n)$ be an *orthogonal representation* of G, i.e. a group homomorphism from G to $O(n)$. We can denote an inner product between these two groups with

$$\langle g, Q\rangle \mapsto \pi(g)Q\pi(g).$$

An *action* of a group G on a set A is a map

$$G \times A \to A,\ \langle g, a\rangle \mapsto ga,$$

satisfying: Let e denote the neutral element of G, then $ea = a$ for all $a \in A$ and $(g_1g_2)(a) = g_1(g_2a)$ for all $g_1, g_2 \in G$ and all $a \in A$.

Definition 4.13. Consider a finite group G and an action $G \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$. A matrix M is called *G-invariant* if $M = gM$ for all $g \in G$. We denote the set of these matrices by $(\mathbb{R}^{n\times n})^G$. Likewise we call a semidefinite program (Psem) *G-invariant* if for each feasible solution X and $g \in G$ the matrix gX is also feasible for (Psem) and gX provides the same objective value, i.e. $\langle C, X\rangle = \langle C, gX\rangle$.

Let G be a finite group for which (Psem) is G-invariant. Then convexity of (Psem) provides that if X is an optimal solution of (Psem) we have that the *group average*

$\frac{1}{|G|}\sum_{g\in G} gX$ is an optimal solution of:

$$\begin{aligned}\inf\ & \langle C, X\rangle \\ \text{s.t. } & \langle A_i, X\rangle = b_i\ \forall i \in \{1,...,l\} \\ & X \succeq 0 \\ & X \in (\mathbb{R}^{n\times n})^G\end{aligned} \tag{Pres}$$

Remark 4.14. The group average $\frac{1}{|G|}\sum_{g\in G} gX$ of a matrix is G-invariant because

$$\tilde{g}\frac{1}{|G|}\sum_{g\in G} gX = \frac{1}{|G|}\sum_{g\in G} \tilde{g}gX = \frac{1}{|G|}\sum_{g\in G} gX \text{ for all } \tilde{g} \in G.$$

Example 4.15. Consider a semidefinite program of the form (Psem) and the permutation group on the index set $[n] = \{1,..,n\}$. Let Σ denote a finite subgroup of permutations, also acting on $[n]$, for which the set of feasible solutions X for (Psem) is invariant under the corresponding simultaneous permutations of rows and columns. These permutations of rows and columns can be expressed with the help of transformation matrices T_σ defined by

$$(T_\sigma)_{ij} = \begin{cases} 1, & \text{if } i = \sigma(j), \\ 0, & \text{otherwise} \end{cases},$$

where $\pi(\sigma) = T_\sigma$. The action on the feasible solutions X is

$$\sigma(X) = T_\sigma X T_\sigma^T, \text{ where } \sigma(X)_{ij} = X_{\sigma^{-1}(i),\sigma^{-1}(j)}.$$

Thus we can consider the following equivalent program:

$$\begin{aligned}\inf\ & \langle C, X\rangle \\ \text{s.t. } & \langle A_i, X\rangle = b_i\ \forall i \in \{1,...,l\} \\ & X \succeq 0 \\ & X \in (\mathbb{R}^{n\times n})^\Sigma\end{aligned}$$

Consider the matrices in the intersection $(\mathbb{R}^{n\times n})^G \cap \mathcal{S}^n_{\succeq 0}$, which form a vector space. Let $B_1, ..., B_N$ be a basis of this space. We use this basis to simplify (Pres) by expressing it with the following equivalent program:

$$\begin{aligned}\inf\ & \langle C, X\rangle \\ \text{s.t. } & \langle A_i, X\rangle = b_i\ \forall i \in \{1,...,l\} \\ & X = \lambda_1 B_1 + ... + \lambda_N B_N \succeq 0 \\ & \lambda_1, ..., \lambda_N \in \mathbb{R}\end{aligned} \tag{Pbas}$$

Example 4.16. The permutation action mentioned above (see Example 4.15) provides a set of Σ-invariant matrices. We will determine a canonical basis $C_1, ..., C_M$ of this space by looking at the orbits of the group action on pairs. Furthermore this basis can be used to determine a basis $B_1, ..., B_N$ of the intersection $V = (\mathbb{R}^{n\times n})^\Sigma \cap \mathcal{S}^n_{\succeq 0}$. The *orbit* of the pair $(i, j) \in [n] \times [n]$ under the group Σ is defined by

$$O(i, j) = \{(\sigma(i), \sigma(j)) : \ \sigma \in \Sigma\}.$$

Decomposing the set $[n] \times [n]$ into the orbits $O_1, ..., O_M$ under the action of Σ leads to $M \leq n^2$ because some pairs may provide the same orbit. For every $k \in \{1, ..., M\}$ we define the matrix $C_k \in \{0,1\}^{n \times n}$ by

$$(C_k)_{ij} = \begin{cases} 1 & \text{if } (i,j) \in O_k \\ 0 & \text{otherwise} \end{cases}.$$

For obtaining the basis $B_1, ..., B_N$ we need to consider the orbits of unordered pairs and define

$$B_{\{i,j\}} = \begin{cases} C_k & \text{if } (i,j), (j,i) \in O_k \\ C_{k1} + C_{k2} & \text{if } (i,j) \in O_{k1},\ (j,i) \in O_{k2},\ k1 \neq k2 \end{cases}.$$

Due to the fact that $X_{ij} = \lambda_{\{i,j\}}$ because $(i,j) = (id(i), id(j))$ and $(\sigma(i), \sigma(j))$ are in the same orbit we have

$$X_{ij} = X_{\sigma(i),\sigma(j)} = X_{\sigma(j),\sigma(i)} = X_{ji}$$

for each $\sigma \in \Sigma$.

For the next simplification of our program we use that $(\mathbb{R}^{n \times n})^G$ has the structure of a matrix $*$-algebra. A *matrix $*$-algebra* is a set of complex matrices, which is closed under addition, scalar multiplication, matrix multiplication and taking the conjugate transpose. Linearity of each $g \in G$ provides the first two properties, whereas

$$g(AB) = \pi(g)AB\pi(g)^T = (\pi(g)A\pi(g)^T)(\pi(g)B\pi(g)^T) = g(A)g(B)$$

and

$$g(A^*) = \pi(g)A^*\pi(g)^T \overset{A \in \mathbb{R}^{n \times n}}{=} \pi(g)A\pi(g)^T = g(A),$$

provide the other properties.

Theorem 4.17. *[1] Let $\mathcal{A} \subseteq \mathbb{C}^{n \times n}$ be a matrix $*$-algebra. There are numbers $m_1, ..., m_d$ such that there is a $*$-isomorphism between $\mathcal{A}$ and a direct sum of complete matrix algebras*

$$\varphi:\ \mathcal{A} \to \bigoplus_{k=1}^{d} \mathbb{C}^{m_k \times m_k}.$$

A detailed proof of this theorem can be found e.g. in [1]. We apply a $*$-isomorphism φ to $\mathcal{A} = (\mathbb{R}^{n \times n})^G$:

$$\varphi:\ (\mathbb{R}^{n \times n})^G \to \bigoplus_{k=1}^{d} \mathbb{C}^{m_k \times m_k}.$$

Since φ is a $*$-isomorphism between matrix algebras with unity, this leads to the fact that φ preserves eigenvalues and hence positive semidefiniteness. In fact if X is G-invariant we have that $X - \lambda I$ is G-invariant because

$$g(X - \lambda I) = g(X) - \lambda g(I) = X - \lambda\pi(g)I\pi(g)^T = X - \lambda I.$$

Furthermore $X-\lambda I$ has an inverse if and only if $\varphi(X)-\lambda I$ has an inverse. Denoting $\psi^{-1}(X-\lambda I)=I$ as the first inverse leads to the fact that $\psi^{-1}\circ\varphi^{-1}$ is an inverse for $\varphi(X)-\lambda I$ because

$$(\psi^{-1}\circ\varphi^{-1})(\varphi(X)-\lambda I)=\psi^{-1}(\varphi^{-1}(\varphi(X-\lambda I)))=I$$

and denoting $\chi^{-1}(\varphi(X)-\lambda I)=I$ as the second inverse leads to

$$(\chi^{-1}\circ\varphi)(X-\lambda I)=\chi^{-1}(\varphi(X)-\lambda I)=I.$$

For positive semidefiniteness this means that for an eigenvalue λ we have that $\det(X-\lambda I)=0$ is equivalent to $\det(\varphi(X)-\lambda I)=0$ and thus X is positive semidefinite if and only if $\varphi(X)$ is positive semidefinite. Hence we can simplify (Pbas) and obtain the G-invariant program (Psem)

$$\begin{aligned}\inf\ & \langle C,X\rangle && \text{(PS2)}\\ \text{s.t. }& \langle A_i,X\rangle=b_i && \forall i\in\{1,...,l\}\\ & \lambda_1\varphi(B_1)+...+\lambda_N\varphi(B_N)\succeq 0\\ & X=\lambda_1B_1+...+\lambda_NB_N,\end{aligned}$$

where $\varphi(B_i)$ are block diagonal matrices with blocks of size $m_1,...,m_d$. Hence instead of a semidefinite program of dimension n^2 we only need to solve a semidefinite program of dimension $m_1^2+...+m_d^2$.

Remark 4.18. If $G=\Sigma$ is a subset of the permutation group we have that the sum $m_1^2+...+m_d^2=M$, where M is the number of distinct orbits.

A problem that arises when using the simplification (PS2) instead of (Psem) is that we need to find a $*$-isomorphism φ. In the case of permutation action C. Bachoc, D. Gijswijt, A. Schrijver and F. Vallentin ([1]) proposed the following construction for the $*$-isomorphism. The main drawback of this construction is that it does not guarantee to provide the maximum possible simplification but it can be computed only by knowing the orbit structure of the permutation group action $\sigma\in\Sigma$ (see Example 4.16). Consider the canonical basis $C_1,...,C_M$ in the case of permutation action, computed in Example 4.16. We define the *multiplication parameters* or *structural parameters* m_{rs}^t by

$$m_{rs}^t=|\{k\in[n]:\ (i,k)\in O_r,\ (k,j)\in O_s\}|,$$

where $(i,j)\in O_t$. Notice that m_{rs}^t does not depend on the choice of i and j. By using the Frobenius norm $||C_r||=\sqrt{\langle C_r,C_r\rangle}$, which equals the size of the corresponding orbits O_r we can define the matrices $D(C_r)\in\mathbb{R}^{M\times M}$ by

$$(D(C_r))_{ij}=\frac{\langle C_rC_j,C_i\rangle}{||C_j||||C_i||}=\frac{||C_i||}{||C_j||}m_{rj}^i.$$

We can now find a $*$-isomorphism ϕ as stated in the following theorem.

Theorem 4.19. *[1] Let $\mathcal{D}$ denote the algebra generated by $D(C_1),...,D(C_M)$. Then the homomorphism*

$$\phi:\ (\mathbb{R}^{n\times n})^\Sigma\longrightarrow\mathcal{D},\quad \phi(C_r)=D(C_r),\quad r=1,...,M,$$

is a $*$-isomorphism.

The above theorem leads to a simplification for (Psem), similar to (PS2):

$$\begin{aligned} &\inf \ \langle C, X\rangle && \text{(Pfin)}\\ &\quad \text{s.t. } \langle A_i, X\rangle = b_i && \forall i \in \{1, ..., l\}\\ &\qquad \lambda_1\phi(B_1) + ... + \lambda_N\phi(B_N) \succeq 0 &&\\ &\qquad X = \lambda_1 B_1 + ... + \lambda_N B_N, && \end{aligned}$$

where the dimension of the matrix $\lambda_1\phi(B_1) + ... + \lambda_N\phi(B_N)$ is M instead of n. For certain instances of the problem $M \leq n$ holds and thus (Pfin) is not as hard to solve in terms of computational run time as (Psem). Certain instances of football pools belong to these instances.

4.4 Application to Football Pools

As mentioned before our application will refer to the european pendant of the famous British "football pools", the so-called "continental pools" or "toto" competitions. See Subsection 2.3.1 for a more detailed explanation. Basically it is an instance of DOMINATING SET for a graph $G = (V, E)$ where $V = \{1, 2, 0\}^m$ denotes the set of vertices and $E = \{\{u, v\} \in V \times V : |i : u_i \neq v_i| \leq R\}$ the edge set for fixed numbers $m, R \in \mathbb{N}_0$.
For this instance we are able to apply simplification (Pfin) to our approximative program (RVV) we only need a permutation group Σ for which (RVV) is Σ-invariant. We order the 3^m nodes of V in lexicographical ascendent order, i.e. $v^1 = (0, ..., 0, 0), v^2 = (0, ..., 0, 1), ..., v^{3^m} = (2, ..., 2, 2)$. Consider the adjacency matrix for football pools of dimension $m \geq 2$, $A_G^{(m)} \in \mathbb{R}^{3^m \times 3^m}$:

$$A_G^{(m)} = \begin{pmatrix} A_G^{(m-1)} & I_{3^{m-1}} & I_{3^{m-1}} \\ I_{3^{m-1}} & A_G^{(m-1)} & I_{3^{m-1}} \\ I_{3^{m-1}} & I_{3^{m-1}} & A_G^{(m-1)} \end{pmatrix}, \ A_G^{(2)} = \begin{pmatrix} \mathbb{1}\mathbb{1}^T & I_3 & I_3 \\ I_3 & \mathbb{1}\mathbb{1}^T & I_3 \\ I_3 & I_3 & \mathbb{1}\mathbb{1}^T \end{pmatrix}.$$

These matrices remain the same under permutations of the entries of the nodes, i.e. if we consider a permutation

$$\sigma : \ V \longrightarrow V, \ \sigma(v) = (v_{\sigma(1)}, ..., v_{\sigma(m)})$$

then the corresponding adjacency matrix to the corresponding rearranged node set $V_\sigma = (\sigma(v^1), ..., \sigma(v^{3^m}))$ remains $A_G^{(m)}$. To prove this we denote the set of these permutations Σ_m.

Lemma 4.20. *$A_G^{(m)}$ is Σ_m-invariant for any $m \geq 2$.*

Proof. Due to the fact that σ is only acting on the entries of any $v \in V$, it is independent of the number of the node. This implies that

$$v_k^i = v_k^j \Leftrightarrow \sigma(v^i)_{\sigma(k)} = \sigma(v^j)_{\sigma(k)}.$$

For any pair of nodes v^i, v^j using the above result leads to

$$\begin{aligned} \{v^i, v^j\} \in E &\Leftrightarrow |k : v_k^i \neq v_k^j| \leq R \\ &\Leftrightarrow |k : \sigma(v^i)_{\sigma(k)} \neq \sigma(v^j)_{\sigma(k)}| \leq R \\ &\Leftrightarrow \{\sigma(v^i), \sigma(v^j)\} \in E. \end{aligned}$$

This means for each rearranged node set V_σ the entry of the adjacency matrix $(A_G)_{ij}^{(m)}$ remains the same. $\square$

The entry permutation σ corresponds to a row and column permutation $\tilde{\sigma}$ because instead of changing the entries in a certain node we can also permute the node with its counterpart, where the entries are already permuted. To illustrate this consider the following example.

Example 4.21. For $m = 2$, Σ_m consists of the identity and $\tau((v_1, v_2)) = (v_2, v_1)$, this leads to

$$\begin{aligned}
\tau(v^1) &= \tau((0,0)) = (0,0) = v^1,\\
\tau(v^2) &= \tau((0,1)) = (1,0) = v^4,\\
\tau(v^3) &= \tau((0,2)) = (2,0) = v^7,\\
\tau(v^4) &= \tau((1,0)) = (0,1) = v^2,\\
\tau(v^5) &= \tau((1,1)) = (1,1) = v^5,\\
\tau(v^6) &= \tau((1,2)) = (2,1) = v^8,\\
\tau(v^7) &= \tau((2,0)) = (0,2) = v^3,\\
\tau(v^8) &= \tau((2,1)) = (1,2) = v^6,\\
\tau(v^9) &= \tau((2,2)) = (2,2) = v^9.
\end{aligned}$$

and thus to a simultaneous row and column permutation matrix $T_\tau = e_1 e_1^T + e_5 e_5^T + e_9 e_9^T + e_2 e_4^T + e_4 e_2^T + e_3 e_7^T + e_7 e_3^T + e_6 e_8^T + e_8 e_6^T$. It is easy to verify that

$$\tau(A_G^{(2)}) = T_\tau A_G^{(2)} T_\tau^T = A_G^{(2)}.$$

This leads to the needed property of (RVV).

Theorem 4.22. *For a graph with adjacency matrix $A_G^{(m)}$ the semidefinite program* (RVV) *is Σ_m-invariant.*

Proof. Recall formulation (RVV):

$$\begin{aligned}
\sup\ & \sum_{k\in V} (2(X_k)_{1,n+2} - (X_k)_{11} - (X_k)_{n+2,n+2}) && \text{(RVV)}\\
\text{s.t. } & \sum_{k\in V} (X_k)_{1+i,1+j} + \sum_{\{k,i\}\in E, \{k,j\}\in E} (X_k)_{n+2,n+2} = 0 && \forall i < j\\
& \sum_{k\in V} ((X_k)_{1+i,1+i} + 2(X_k)_{1,1+i}) + \\
& \sum_{\{k,i\}\in E} (2(X_k)_{1,n+2} - (X_k)_{n+2,n+2}) = 1 && \forall i \in V\\
& (X_k)_{n+2,1+i} + (X_k)_{n+2,n+2} = 0 && \forall \{i,k\} \in E\\
& (X_k)_{n+2,1+i} = 0 && \forall \{i,k\} \in \bar{E}\\
& \sum_{(i,j,l)\in I_\alpha} \Lambda_{ij}^{kl} = 0 && \forall \alpha \in \alpha^r \setminus \alpha^r_{end},\ \forall k \in V\\
& \sum_{(i,j,l)\in I_\alpha} \Lambda_{ij}^{kl} \geq 0 && \forall \alpha \in \alpha^r_{end},\ \forall k \in V\\
& X_k - \Lambda^{kl} \succeq 0 && \forall l \in \{1, ..., (n+2)^r\},\ \forall k \in V
\end{aligned}$$

Denote the number of nodes $|V| = n = 3^m$. Consider the positive semidefinite restriction $X_k - \Lambda^{kl} \succeq 0 \ \forall l \in \{1, ..., (n+2)^r\},\ k \in V$ in (RVV). An equivalent formulation is to express this restriction with the help of a diagonalmatrix

$$Z = \begin{pmatrix} Y_1 & 0 & 0 \\ 0 & ... & 0 \\ 0 & 0 & Y_n \end{pmatrix} \succeq 0, \text{ where } Y_k = \begin{pmatrix} X_k - \Lambda^{k,1} & 0 & 0 \\ 0 & ... & 0 \\ 0 & 0 & X_k - \Lambda^{k,(n+2)^r} \end{pmatrix}.$$

A permutation $\sigma \in \Sigma_m$ of entries in a node $v \in V$ corresponds to a permutation of nodes $\tilde{\sigma}$ with $V_\sigma := \tilde{\sigma}(V)$, where the permuted nodes $\tilde{\sigma}(v) = \sigma(v)$. The corresponding solution is

$$\tilde{\sigma}(Z) = \begin{pmatrix} \tilde{Y}_{\sigma(1)} & 0 & 0 \\ 0 & ... & 0 \\ 0 & 0 & \tilde{Y}_{\sigma(n)} \end{pmatrix}, \text{where } \tilde{Y}_{\sigma(k)} = \begin{pmatrix} \tilde{X}_{\sigma(k)} - \Lambda_\sigma^{\sigma(k),1} & 0 & 0 \\ 0 & ... & 0 \\ 0 & 0 & \tilde{X}_{\sigma(k)} - \Lambda_\sigma^{\sigma(k),(n+2)^r} \end{pmatrix}$$

and $(\tilde{X}_{\sigma(k)})_{1+i,1+j} = (X_{\sigma(k)})_{1+\sigma(i),1+\sigma(j)}, (\Lambda_\sigma)_{1+i,1+j} = \Lambda_{1+\sigma(i),1+\sigma(j)}$. We can verify that this is a feasible solution for (RVV) by using the invariance of the adjacency matrix A_G from Lemma 4.20. In particular this invariance implies that $(i,j) \in E$ is equivalent to $(\sigma(i), \sigma(j)) \in E$ and thus the first four constraints in (RVV) are not affected by the permutation. For a fixed node k respectively $\sigma(k)$ the two sums of entries in Λ are not affected by a permutation of rows and columns of Λ. Eigenvalues of a matrix are not affected by a simultaneous permutation of rows and columns and thus semidefiniteness holds as well. Finally to sum up over the nodes $k \in V$ does not change under permutations and thus the objective value remains the same. □

Thus we can apply the simplification (Pfin) to (RVV). If $M \leq |V| = 3^m$ holds, this allows us to use an algorithm, needing fewer memory and run time but still giving the same lower bounds for the domination number $\gamma(G)$. Unfortunately the number of orbits will turn out to be $M = \binom{m+8}{8}$, as we can see in Lemma 4.24. To prove this, we use the following Lemma 4.23. Although Burnside himself attributed the lemma to F. Frobenius it became famous as "Burnside's Lemma". For a general version of this lemma see e.g. J. Rotman [25].

Lemma 4.23 (Burnside's Lemma)**.** *Let Σ be a finite permutation group acting on a finite set $[n] \times [n]$. Then the number of orbits M can be expressed as*

$$M = \frac{1}{|\Sigma|} \sum_{\sigma \in \Sigma} F(\sigma),$$

where, for $\sigma \in \Sigma$, $F(\sigma)$ is the number of $(i,j) \in [n] \times [n]$ fixed by σ, where $\sigma((i,j)) := (\sigma'(i), \sigma'(j))$ for the corresponding permutation group Σ' acting on $[n]$.

Lemma 4.24. *For an m-dimensional permutation group Σ_m the number of orbits M_m can be expressed as*

$$M_m = \binom{m+8}{8}.$$

Proof. We can use Burnside's Lemma to obtain

$$M_m = \frac{1}{|\Sigma_m|}\sum_{\sigma\in\Sigma_m} F(\sigma) = \frac{1}{|\Sigma'_m|}\sum_{\sigma'\in\Sigma'_m} F(\sigma')^2$$

for a permutation Σ'_m acting on $[n] = |V| = 3^m$. We prove $\frac{1}{|\Sigma'_m|}\sum_{\sigma'\in\Sigma'_m} F(\sigma')^2 = \binom{m+8}{8}$ via Induction on m. If $m = 2$, recall τ from Example 4.21, which implies

$$M_2 = \frac{1}{2!}\left(F(id)^2 + F(\tau)^2\right) = \frac{1}{2}\left(9^2 + 3^2\right) = 45 = \frac{10\cdot 9}{2} = \frac{10!}{8!\ 2!} = \binom{2+8}{8}.$$

Hence for the m-dimensional permutation group Σ_m we have the induction hypothesis:

$$M_m = \frac{1}{|\Sigma'_m|}\sum_{\sigma'\in\Sigma'_m} F(\sigma')^2 = \binom{m+8}{8}.$$

For the inductive step we observe that any $\sigma' \in \Sigma'_{m+1}$ can be expressed as a composition of a permutation $\sigma' \in \Sigma'_m$ on the last m entries of the nodes $v \in \{1,0,2\}^{m+1}$ and a transposition $\tilde{\sigma}$ with the first entry, as follows:

$$M_{m+1} = \frac{1}{|\Sigma'_{m+1}|}\sum_{\sigma'\in\Sigma'_{m+1}} F(\sigma')^2 = \frac{1}{(m+1)!}\sum_{\tilde{\sigma}_k:\ \tilde{\sigma}_k(1)=k,\ k\in[m+1]}\ \sum_{\sigma'\in\Sigma'_m} F(\tilde{\sigma}_k\circ\sigma')^2.$$

Splitting up the sums leads to

$$\frac{1}{(m+1)!}\left(\sum_{\sigma'\in\Sigma'_m} F(\tilde{\sigma}_1\circ\sigma')^2 + \sum_{\tilde{\sigma}_k:\ \tilde{\sigma}_k(1)=k,\ k\in\{2,\dots,m+1\}}\ \sum_{\sigma'\in\Sigma'_m} F(\tilde{\sigma}_k\circ\sigma')^2\right).$$

To count the fixed points of $\tilde{\sigma}_k\circ\sigma'$ we need the following observation:

$$\begin{aligned}
\tilde{\sigma}_k\circ\sigma'(v) = v &\Leftrightarrow \sigma'(v) = \tilde{\sigma}_k^{-1}(v) = (v_k, v_2, \dots, v_{k-1}, v_1, v_{k+1}, \dots, v_{m+1})\\
&\Leftrightarrow (v_1, v_{\sigma'(2)}, \dots, v_{\sigma'(m+1)}) = (v_k, v_2, \dots, v_{k-1}, v_1, v_{k+1}, \dots, v_{m+1})\\
&\Leftrightarrow v_1 = v_k,\ v_{\sigma'(k)} = v_1,\ v_{\sigma'(j)} = v_j\ \forall j \neq k\\
&\Leftrightarrow v_1 = v_k,\ \sigma'(v_2, \dots, v_{m+1}) = (v_2, \dots, v_{m+1}).
\end{aligned}$$

Hence if $k \neq 1$ each fixed point v of $\tilde{\sigma}\circ\sigma'$ leads to exactly one fixed point of σ' implying $F(\tilde{\sigma}_k\circ\sigma') = F(\sigma')$. In the remaining case v_1 can be chosen arbitrarily out of $\{1,0,2\}$, i.e. $F(\tilde{\sigma}_1\circ\sigma') = 3F(\sigma')$. This means we can express M as follows:

$$\frac{1}{m+1}\left(\frac{1}{m!}\sum_{\sigma'\in\Sigma'_m}(3F(\sigma'))^2 + \frac{1}{m!}\sum_{\tilde{\sigma}_k:\ \tilde{\sigma}_k(1)=k,\ k\in\{2,\dots,m+1\}}\ \sum_{\sigma'\in\Sigma'_m} F(\sigma')^2\right).$$

Finally applying the induction hypothesis leads to

$$\frac{1}{m+1}\left(3^2\binom{m+8}{8} + m\binom{m+8}{8}\right) = \frac{8+1+m}{m+1}\binom{m+8}{8} = \binom{m+1+8}{8}.$$

□

This means the number of orbits grows polynomially in the number of games m. Due to the fact that the run time of any algorithm using this reduction method grows in M instead of 3^m we have a quite powerful tool to reduce the run time in the long run. Unfortunately this affects only the case $m \geq 10$, i.e. for programs of size $|V| = 3^{10}$. Semidefinite programs of this size are still out of reach for present computers and thus the method is not applicable for our purpose yet.

5 Outlook

The central question motivating this thesis can be formulated as follows:
If we have a certain combinatorial problem, which seems not to be solvable with a polynomial time algorithm, can we at least approximate it quite precisely?
Because of the variety of combinatorial problems that are known so far, we consider only one of these problems, namely DOMINATING SET. S. Burer's paper provides the opportunity to apply the theory of copositive optimization on combinatorial problems. Using this approach we achieved two copositive formulations of DOMINATING SET.
Due to the fact that copositive programs remain NP-hard, we needed an approximation hierarchy, where the single steps of this hierarchy are computable in polynomial time. This hierarchy was developed by P. Parrilo and has provided semidefinite hierarchies for each of the two copositive formulations. The corresponding polynomial solving algorithms compute approximations for the domination number $\gamma(G)$ with different behaviour in run time.
It turns out that both algorithms have a large (though polynomial) run time for computably small instances. However, it seems that at least for the first approximation step the algorithms will have a smaller run time for graphs of estimated 50 nodes than the naive brute force approach. For the given instances of nodes up to $n = 14$, computing the formulation (R2V) provides lower bounds of the domination number $\gamma(G)$ with equal quality but in a significantly smaller run time than computing (RVV). This implies that we can approximate larger instances, such as $n = 20$. On the other hand the copositive formulation (RVV) uses smaller matrices, which should enhance the performance in the long run significantly.
Due to the large run time of the presented polynomial algorithms, the use of copositive optimization to calculate lower bounds for DOMINATING SET suffers from a lack of computational capacity. The developed algorithms should disproportionately benefit from faster computers because, if the given instances are large enough, the polynomial run time should pay off. Especially approximation algorithms for instances such as "football pools", for which the simplification in Chapter 4 can be used, should gain significant performance enhancements.
Further research could be based on the questions, whether there are other copositive formulations for DOMINATING SET that enhance the performance or provide better bounds for a comparable performance of the resulting algorithms. For example relaxing the quadratic restriction in the completely positive program (C) could be a convenient approach. It might be another approach to sharpen the bounds for the hierarchy steps needed to obtain an exact or at least approximate solution. Generalizing the results of this thesis to graphs with infinite node sets V could improve our knowledge of NP-hard problems in general. The related problem of computing the stability number of a graph is currently under investigation by C. Dobre, M. Dür, L. Frerick and F. Vallentin [8].

6 Appendix

6.1 Algorithm for Checking Copositivity

To prove that

$$M := \begin{pmatrix} 1 & -1 & -1 & \frac{3}{4} & \frac{3}{4} \\ -1 & 1 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ -1 & 1 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ \frac{3}{4} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{3}{4} & -\frac{1}{2} & -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix} \in \mathcal{C}$$

we need to check whether there are principal submatrices with negative eigenvalues and corresponding positive eigenvectors. This can be verified by the following algorithm.

Algorithm 1 Checking Copositivity of M

Require: M:
Ensure: *iscopositive*: boolean variable indicating whether M is copositive

```
indexset ← {1, 2, 3, 4, 5}
powerset ← 2^indexset
iscopositive ← TRUE
for element ∈ powerset do
    eig ← eigenvectors of M_element
    for k ∈ {1, ..., length of eig} do
        if eig(k) is componentwise strictly negative or strictly positive then
            if eigenvalue corresponding to eig(k) is negative then
                iscopositive ← FALSE
            end if
        end if
    end for
end for
```

6.2 Preliminaries for Solving Algorithms

For the solving algorithms for the semidefinite programs (R2V) and (RVV) for each step r in Parrilo's hierarchy we need to compute the degrees β^r and α^r. Due to the fact that $\alpha^r = 2\beta^r$ we introduce three procedures to compute β^r and neglect the corresponding α^r. First we compute the degrees $\beta^0 := \beta^*$ of an n-variate form with the procedure *degrees0*. Observe that we use different for loops to achieve the wanted order within the degrees β^*. Furthermore we add an additional column of zeros to the variable *beta0matrix*, which becomes important for the next procedure.

Algorithm 2 Procedure *degrees0*

Require: n:
Ensure: $beta0matrix$: matrix with β^* components as rows and zeros in the last column

$nsum \leftarrow n(n+1)/2$
$counter \leftarrow 0$
for $i \in \{1, ..., n\}$ **do**
  $counter \leftarrow \text{counter} + 1$
  $beta0matrix[counter, :] \leftarrow 2(e_i^T, 0)$
end for
for $i < j$ **do**
  $counter \leftarrow \text{counter} + 1$
  $beta0matrix[counter, :] \leftarrow (e_i^T + e_j^T, 0)$
end for

Second we use the procedure *degrees* to compute the degrees β^r inductively by adding $(e_1, ..., e_n)$ to each component of β^{r-1}. In line 9 we add the index of the original β_i^0 to the matrix *degb*. This means if $\beta^r_{n(k-1)+i}$ is stored in row l of *degb*, i.e. $\beta^r_{n(k-1)+i} = degb[l, 1 : end - 1]$ then $i = degb[l, end]$. Thus for any step in the hierarchy the information of i is kept, providing information about the entries i, j for degrees $\alpha^r = \beta^r_{n(k-1)+i} + \beta^r_{n(k-1)+j}$ (later referred to as: β_i^r and β_j^r). as we will see in the solving algorithms below.

Algorithm 3 Procedure *degrees*

Require: b: matrix with entries of β^{r-1}
Ensure: $degb$: matrix with $\beta^r \setminus \beta^r_{end}$ components as rows and the information β_k^* added

1: $n \leftarrow$ (length of rows of b) $- 1$
2: $counter \leftarrow 0$
3: **for** $j \in \{1, ..., n\}$ **do**
4:   **for** $i \in \{1, ..., n^{r+1}\}$ **do**
5:     $counter \leftarrow \text{counter} + 1$
6:     $degb[counter, :] \leftarrow b_i + (e_j^T, 0)$
7:     ▷ Add information about β_k^* added (0 if $k > n$):
8:     **if** $b[i, end] = 0$ **then**
9:       $degb[counter, end] \leftarrow i$
10:     **end if**
11:   **end for**
12: **end for**

The last procedure *minimumevendegrees* provides for a matrix of degrees α^r of size $2(r+2)$ the corresponding even degrees without double entries and additional information.

Algorithm 4 Procedure *minimumevendegrees*

Require: *dega*: matrix with all entries of $\alpha^r \setminus \alpha^r_{end}$ and two columns of additional information
Ensure: *mindega*: matrix with even $\alpha^r \setminus \alpha^r_{end}$ components as rows

```
dega ← dega[, end − 2]
vector help ← number of odd entries in rows of dega
l ← number of entries >0 in help
                              ▷ Copy only even degrees in smaller matrix hmindega
counter ← 0
for i ∈ {1, ..., length of help} do
    if help[i] = 0 then
        counter ← counter + 1
        hmindega[counter, ] ← dega[i, ]
    end if
end for
                              ▷ Eliminate double entries in hmindega
for i ∈ {1, ..., l} do
    ▷ Check whether hmindega[i,] is in mindega. If not append it to mindega.
end for
```

6.3 Solving Algorithm for R2V

We declare K as a cone variable with two components, where $K.s$ is the set of $n^r + 2$ positive semidefinite cones with the components

$$X^+ \succeq 0,\ X^- \succeq 0 \text{ and } W_k := X^+ - X^- - \Lambda^k \succeq 0.$$

and $K.l$ is the non-negative orthant for the slack variables for the inequalities given in (R2V). Let further $G = (V, E)$ with adjacency matrix A_G and denote $|V| = V_G$. For any $\alpha \in \alpha^r$ we have a corresponding set of matrices A_α to define the degree constraints in (R2V). Similarly we compute the constraints A_{lin} and b in line 39.

Algorithm 5 Algorithm to solve R2V

Require: A_G: adjacency matrix of the graph G,
Require: r: step of Parrilo's hierarchy
Ensure: *res*: result of CSDP inner point solution for (R2V)

```
1: m ← 2V_G + 1
2: Compute degb:
3: degb ← degrees0(m)
4: for i ∈ {1, ..., r} do
5:     degb ← degrees(degb)
6: end for
7: degb ← degb[1 : m^(r+1), ]
8: Compute matrix dega consisting of the relevant parts of α^r, i.e. β_i^r + β_j^r, where
   β_i^r and β_j^r are in the same (k-th) block. These degrees correspond to the entries
   (W_k)_ij.
9: mindega ← minimumevendegree(dega)                         ▷ Delete double entries
```

Set C as given in (R2V).
Compute size of $K.l$:
for $\alpha \in mindega$ **do**
    if $\alpha \notin 2 \cdot degb$ **then**
        $slack \leftarrow slack + 1$
    end if
end for
$K.l \leftarrow slack$
$C \leftarrow -C$ ▷ Needed for applying CSDP
Compute degree constraints for each $\alpha \in \alpha^r$:
for $\alpha \in mindega$ **do**
    if $\alpha \in 2 \cdot degb$ **then** ▷ $\alpha \notin \alpha^r_{end}$?
        for $l \in$ length of $dega$ **do**
            if $\alpha = dega[l, 1 : end - 2]$ **then** ▷ Is β^r_i resp. $\beta^r_j \in \beta^r_{end}$?
                if $dega[l, end - 1] \leq m^{r+1}$ & $dega[l, end] \leq m^{r+1}$ **then**
                    Compute equality constraint and set A_α.
                end if
            end if
        end for
    else ▷ $\alpha \in \alpha^r_{end}$?
        for $l \in$ length of $dega$ **do**
            if $\alpha = dega[l, 1 : end - 2]$ **then** ▷ Is β^r_i resp. $\beta^r_j \in \beta^r_{end}$?
                if $dega[l, end - 1] \leq m^{r+1}$ & $dega[l, end] \leq m^{r+1}$ **then**
                    Compute inequality constraint and set A_α and slack.
                end if
            end if
        end for
    end if
end for
Compute other constraints. ▷ i.e. A_{lin}, b
$Asp \leftarrow \text{sparse}(A)$
$bsp \leftarrow \text{sparse}(b)$
$Csp \leftarrow \text{sparse}(C)$
$[x,y,z,info] \leftarrow \text{csdp}(Asp, bsp, Csp, K)$
$res \leftarrow -\langle C, X \rangle$

6.4 Solving Algorithm for RVV

We declare K as a cone variable with two components, where $K.s$ is the set of $2n + n \cdot (n+2)^r$ positive semidefinite cones with the components

$$X_k^+ \succeq 0,\ X_k^- \succeq 0 \text{ and } W_{kl} := X_k^+ - X_k^- - \Lambda^{kl} \succeq 0.$$

and $K.l$ is the non-negative orthant for the slack variables for the inequalities given in (RVV). Let further $G = (V, E)$ with adjacency matrix A_G and denote $|V| = V_G$. For any $\alpha \in \alpha^r$ we compute a corresponding set of matrices A_α to define the degree constraints in (RVV). Similarly we compute the constraints A_{lin} and b in line 41.

Algorithm 6 Algorithm to solve RVV

Require: A_G: adjacency matrix of the graph G,
Require: r: step of Parrilo's hierarchy
Ensure: res: result of CSDP inner point solution for (RVV)
1: $m \leftarrow V_G + 2$
2: Compute degb:
3: $degb \leftarrow degrees0(m)$
4: **for** $i \in \{1, ..., r\}$ **do**
5: $\quad degb \leftarrow degrees(degb)$
6: **end for**
7: $degb \leftarrow degb[1 : m^{r+1},]$
8: Compute matrix $dega$ consisting of the relevant parts of α^r, i.e. $\beta_i^r + \beta_j^r$, where β_i^r and β_j^r are in the same (k-th) block. These degrees correspond to the entries $(W_{kl})_{ij}$.
9: $mindega \leftarrow minimumevendegree(dega)$ ▷ Delete double entries
10: **for** $\alpha \in mindega$ **do**
11: $\quad$ Compute size of $K.l$:
12: $\quad$ **if** $\alpha \notin 2 \cdot degb$ **then**
13: $\quad\quad slack \leftarrow slack + 1$
14: $\quad$ **end if**
15: **end for**
16: $K.l \leftarrow V_G \cdot slack$
17: Set C as given in (RVV).
18: $C \leftarrow -C$ ▷ Needed for applying CSDP

Compute degree constraints for each $\alpha \in \alpha^r$:

for $k \in V$ **do**

  for $\alpha \in mindega$ **do**

    if $\alpha \in 2 \cdot degb$ **then** ▷ $\alpha \notin \alpha^r_{end}$?

      for $l \in$ length of $dega$ **do**

        if $\alpha = dega[l, end-2]$ **then** ▷ Is β^r_i resp. $\beta^r_j \in \beta^r_{end}$?

          if $dega[l, end-1] \leq m^{r+1}$ & $dega[l, end] \leq m^{r+1}$ **then**

            Compute equality constraint and set A_α.

          end if

        end if

      end for

    else ▷ $\alpha \in \alpha^r_{end}$?

      for $l \in$ length of $dega$ **do**

        if $\alpha = dega[l, end-2]$ **then** ▷ Is β^r_i resp. $\beta^r_j \in \beta^r_{end}$?

          if $dega[l, end-1] \leq m^{r+1}$ & $dega[l, end] \leq m^{r+1}$ **then**

            Compute inequality constraint and set A_α and slack.

          end if

        end if

      end for

    end if

  end for

end for

Compute other constraints. ▷ i.e. A_{lin}, b

$Asp \leftarrow$ sparse(A)

$bsp \leftarrow$ sparse(b)

$Csp \leftarrow$ sparse(C)

$[x,y,z,info] \leftarrow$ csdp(Asp, bsp, Csp, K)

$res \leftarrow -\langle C, X \rangle$

Bibliography

[1] C. Bachoc, D. Gijswijt, A. Schrijver, and F. Vallentin. *Invariant semidefinite programs*, volume 166 of *Internat. Ser. Oper. Res. Management Sci.*, pages 219–269. Springer, New York, 2012.

[2] M. R. Blanton, H. Lin, R. H. Lupton, F. Miller Maley, N. Young, I. Zehavi, and J. Loveday. An efficient targeting strategy for multiobject spectrograph surveys: The Sloan Digital Sky Survey "Tiling" Algorithm. *The Astronomical Journal*, 125(4):2276–2286, 2003.

[3] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program.*, 120(2):479–495, 2009.

[4] M. D. Choi and T. Y. Lam. An old question of Hilbert. In *Conference on Quadratic Forms–1976*, number 46 in Queen's Papers in Pure and Appl. Math., pages 385–405. Queen's Univ., Kingston, Ont., 1977.

[5] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[6] J. de Loera and F. Santos. An effective version of Pólya's theorem on positive definite forms. *J. Pure Appl. Algebra*, 108(3):231–240, 1996.

[7] J. de Loera and F. Santos. Erratum to: "An effective version of Pólya's theorem on positive definite forms". *J. Pure Appl. Algebra*, 155(2–3):309–310, 2001.

[8] C. Dobre, M. Dür, L. Frerick, and F. Vallentin. A copositive formulation for the stability number of infinite graphs. Preprint, Available at: `http://arxiv.org/abs/1305.1819`, 2014.

[9] M. Dür and G. Still. Interior points of the completely positive cone. *Electron. J. Linear Algebra 17*, pages 48–53, 2008.

[10] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[11] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs*. Marcel Dekker Inc., New York, 1998.

[12] Clay Mathematics Institute. Millenium prize problems. Website. `http://www.claymath.org/millennium-problems`;.

[13] David S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci*, 9:256–278, 1974.

[14] M. Hall Jr. and M. Newman. Copositive and completely positive quadratic forms. *Proc. Cambridge Philos. Soc.*, 59:329–339, 1963.

[15] W. Kaplan. A test for copositive matrices. *Linear Algebra Appl.*, 313(1–3):203–206, 2000.

[16] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), pages 85–103. Plenum, New York, 1972.

[17] S.T. Hedetniemi; R.C. Laskar. Bibliography on domination in graphs and some basic definitions of domination parameters. *Discrete Math.*, 86(1–3):257–277, 1990.

[18] M. Laurent and F. Vallentin. Semidefinite optimization. Scriptum of Semidefinite Optimization, Mastermath Spring 2012, Available at: `https://sites.google.com/site/nichtlinopt/material`, 2012.

[19] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. Assoc. Comput. Mach.*, 41(5):960–981, 1994.

[20] J. G. Mauldon. Covering theorems for groups. *Quart. J. Math. Oxford Ser. (2)*, 1:284–287, 1950.

[21] P.A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, Pasadena, California, 2000. Available at: `http://www.mit.edu/~parrilo/pubs/files/thesis.pdf`.

[22] A. Paz and S. Moran. Non-deterministic polynomial optimization problems and their approximation. In *Automata, languages and programming (Fourth Colloq., Univ. Turku, Turku, 1977)*, volume 52 of *Lecture Notes in Comput. Sci.*, pages 370–379. Springer, Berlin, 1977.

[23] V. Powers and T. Wörmann. An algorithm for sums of squares of real polynomials. *J. Pure Appl. Algebra*, 127(1):99–104, 1998.

[24] J. Renegar. *A mathematical view of interior-point methods in convex optimization*. MPS/SIAM Series on Optimization. SIAM, 2001.

[25] J. Rotman. *An introduction to the theory of groups*, volume 148 of *Grad. Texts in Math.* Springer-Verlag, New York, 1995.

List of Symbols

$[n]$ $:= \{1, ..., n\}$

$\operatorname{diag}(v) := \begin{cases} \operatorname{diag}(v)_{ij} = v_i & \text{for } i = j \\ \operatorname{diag}(v)_{ij} = 0 & \text{for } i \neq j \end{cases}$

$\mathcal{S}^n$ $:= \{M \in \mathbb{R}^{n \times n} : M^T = M\}$

e_i i-th canonical unit vector

I_n n-dimensional identity matrix

Index

Statement of Originality

I hereby declare under oath that this master's thesis is the product of my own independent work. All content and ideas drawn directly or indirectly from external sources are indicated as such. The thesis has not been submitted to any other examining body and has not been published.

(Place and date) (Jan Hendrik Rolfes)

Danksagung

Abschließend möchte ich mich bei allen Menschen bedanken, die mich bei dieser Arbeit unterstützt haben. Ein besonderer Dank geht an meinen Betreuer Herrn Prof. Dr. Vallentin für das interessante Thema und seine weiteren Anregungen. Weiterhin danke ich Herrn Dr. Frederik von Heymann dafür, dass er jederzeit für mich ansprechbar war und damit erheblich zum Gelingen dieser Arbeit beigetragen hat.
Ebenso bedanke ich mich bei allen Anderen, die mich bei dieser Arbeit unterstützt haben, insbesondere Kristina Hoffmann und Johannes Wirtz. Außerdem gebührt großer Dank meinen Eltern und deren Lebensgefährten, die mir während des gesamten Studiums helfend zur Seite standen.