Symmetry Reduction in Convex Optimization

with Applications in Combinatorics

DANIEL BROSCH

TILBURG UNIVERSITY

Symmetry Reduction in Convex Optimization
with Applications in Combinatorics

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan Tilburg University op gezag van de rector magnificus, prof. dr. W.B.H.J. van de Donk, in het openbaar te verdedigen ten overstaan van een door het college voor promoties aangewezen commissie in de aula van de Universiteit op

woensdag 19 oktober 2022 om 13.30 uur

door

DANIEL BROSCH

geboren te Leverkusen, Duitsland.

PROMOTORES:

prof. dr. E. de Klerk (Tilburg University)
prof. dr. M. Laurent (Tilburg University)

LEDEN VAN DE PROMOTIECOMMISSIE:

prof. dr. F. Vallentin (University of Cologne)
dr. F.M. de Oliveira Filho (TU Delft)
prof.dr.ir. R. Sotirov (Tilburg University)
prof.dr.ir. E.R. van Dam (Tilburg University)
prof. dr. J.S.H. van Leeuwaarden (Tilburg University)
dr. J.C. Lizcano (Tilburg University)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 764759 (MINOA).

Symmetry reduction in convex optimization with applications in combinatorics

Acknowledgments

This thesis, which was written during the last four years, could not have come to where it is today without the support of all the people in my life. It has been a challenge, especially with COVID forcing most of us to work from home. Despite that, I am truly satisfied with the result.

To start things off, I would like to thank my supervisor, Etienne de Klerk, for providing me with the opportunity to work on my PhD. His advice and guidance helped me whenever I was aimless or stuck on a problem. He encouraged me to pursue promising avenues, and, when appropriate, discouraged me from attempting something hopeless. During the second half of my PhD he gave me the freedom and trust to work on problems at my own discretion, despite the high risks involved. But even outside of work, he was always supportive of me, in personal matters too. He provided me with invaluable feedback for my research, presenting skills, and teaching. I could not think of a better supervisor to have written this thesis under.

I am grateful to my second supervisor Monique Laurent for her unrelenting focus and excellent advice. She was always ready to discuss problems in depth, lending her extensive knowledge. In particular, I want to praise her detailed feedback, patience and the ever present willingness to exchange ideas.

My international colleagues Maciej, Rafael, Tung, Paan, Ming, Joris, Marie, Nora, Zihao, and many others deserve thanks for making me feel included and welcome from the start. Despite not having studied in Tilburg, they immediately invited me to countless lunches, trips, movies, dinners, BBQs, drinks, and more.

In particular, I want to thank my office mates Rafael, Maciej, Nora, and later on Riley, for the fun, support, and interesting discussions. It was never boring with you.

I want to thank my colleagues, especially Riley, Felix, Frank, Pieter, Lorenz, Ernst, Stefan, Pepijn, and, from CWI, Sven and Lucas for all the fruitful and fun discussions.

Furthermore, I am grateful to Frank Vallentin. He was my main source of inspiration during my master's degree, his fantastic lectures shaping the direction of my work.

I want to thank all committee members for the detailed feedback and fruitful discussions on my thesis. They were crucial for the revisions, significantly improving clarity and conciseness. Not only did they impact this work, but also my future papers based on unpublished material in this thesis.

Next, I want to give thanks to the secretaries for helping me organize trips to conferences, dealing with bureaucracy, and much more. Without them my job would have been a lot more tedious. Their attention to detail had everything running smoothly.

The committee and "Mister Manager" of the MINOA grant deserve thanks for organizing the project, various events, and dealing with ridiculous amounts of bureaucracy. They made the last few years incredibly memorable and enjoyable. I also want to give thanks to the other PhDs of the project. I had fun getting to know you all.

My girlfriend Leanne is the best partner one could hope for. She was always there for me, no matter how busy, stressed, tired, focussed on work I was. Spending time with her helped me relax, get my mind of things, and were crucial for me staying sane in these years.

Auch meinen Mathelehrern aus der Schulzeit gebührt Dank, insbesondere Joachim Kuhrig und Daniel Schalow für die tatkräftige Unterstützung und Fördern meines Interesses an der Mathematik. Von der Mathe-AG bis zum Schülerstudium konnte ich mich immer auf sie verlassen. Ohne sie wäre ich nicht da, wo ich heute bin.

Ich möchte mich auch bei meinem Bruder Sebastian bedanken. Er ist der beste Bruder, den man sich wünschen kann, und dank ihm war meine Kindheit immer aufregend.

Zu guter Letzt danke ich meinen Eltern. Sie haben mir eine sorglose, glückliche Kindheit geschenkt, in der es mir nie an etwas fehlte. Dadurch dass sie meine Interessen immer unterstützt und gefördert haben, ohne mich unter Druck zu setzen, konnte ich meinen eigenen Weg finden. Ich widme diese Arbeit meinen Eltern.

Daniel Brosch
Klagenfurt, August 31, 2022.

Contents

List of Figures **ix**

List of Tables **xi**

List of Algorithms **xiii**

Notation **xv**

1 Introduction **1**
- 1.1 Convex (Conic) Optimization . . . 3
- 1.2 Relaxing Non-Convex Problems by Semidefinite Programming . . . 6
  - 1.2.1 Higher Order Relaxations: The Lasserre Hierarchy (for Stable Set) . . . 10
  - 1.2.2 Relaxing Polynomial Optimization Problems: Sums-of-Squares Hierarchies . . . 12
- 1.3 Symmetric Convex Optimization . . . 14
  - 1.3.1 Group Symmetry Reduction . . . 14
- 1.4 Overview of the Thesis . . . 17
- 1.5 Contributions to the Literature . . . 20

2 Jordan Symmetry Reduction over the Doubly Nonnegative Cone **21**
- 2.1 Preliminaries on Jordan Symmetry-Reduction . . . 23
  - 2.1.1 Constraint Set Invariance . . . 23
  - 2.1.2 The Reduction for Jordan-Algebras . . . 26
  - 2.1.3 A Combinatorial Reduction Algorithm . . . 27
- 2.2 Extension to the Doubly Nonnegative Cone . . . 30
- 2.3 Reducing the ϑ'-Function . . . 33
  - 2.3.1 The ϑ'-Function of Erdős-Rényi Graphs . . . 34
- 2.4 The Julia Package . . . 35

2.5 Concluding Remarks . . . 39
2.6 Example Use of the Software Package . . . 39

3 Comparison and Reduction of Relaxations of the Quadratic Assignment Problem 43
3.1 Three Bounds for the QAP . . . 44
3.1.1 Projected Eigenvalue Bound . . . 44
3.1.2 Quadratic Programming Bound . . . 45
3.1.3 SDP Bound . . . 46
3.2 Comparison of the Bounds . . . 47
3.3 Symmetry Reduction for the SDP Bound . . . 52
3.3.1 Results of Reductions of QAPLib Problems . . . 54
3.4 Concluding Remarks . . . 55

4 Energy Minimization on a Toric Grid as a QAP 59
4.1 Energy Minimization on a Toric Grid as a QAP . . . 60
4.1.1 Eigenvalue Bound of Bouman, Draisma and Leeuwaarden . 62
4.1.2 Bound Comparison . . . 64
4.2 Reducing the Relaxation of the Energy Minimization Problem . . . 66
4.2.1 The Jordan Reduction . . . 66
4.2.2 Symmetric Circulant Matrices . . . 68
4.2.3 Admissible Subspaces . . . 69
4.2.4 Block-Diagonalization . . . 73
4.3 Towards the Checkerboard Conjecture . . . 76
4.4 Numerical Results . . . 80
4.5 Concluding Remarks . . . 83

5 Optimizing Hypergraph-Based Polynomials Modeling Job-Occupancy in Queuing with Redundancy Scheduling 93
5.1 Introduction . . . 94
5.2 Preliminaries . . . 98
5.2.1 Motivation . . . 98
5.2.2 Sketch of Proof . . . 101
5.2.3 Preliminaries on the Terwilliger Algebra . . . 103
5.3 Proof of Theorem 5.2 . . . 104
5.3.1 The Case $d = 2$ and $L = 2$ . . . 105
5.3.2 Computing the Hessian Matrix of p_d . . . 106

5.3.3 The General Case $d \geq 2$ and $L \geq 2$ . . . 110
5.4 Investigating the Polynomials f_d . . . 113
5.4.1 Computing the Hessian of f_d . . . 114
5.4.2 The Polynomial f_d in the Case $d = 3$, $L = 2$ . . . 117
5.4.3 Some Numerical Justification for Convexity of f_d . . . 122
5.5 Numerical Results for the Polynomials f_d . . . 123

6 Four Different Views at Flag Algebras 129
6.1 What is a (Non-Induced) Flag Algebra? . . . 130
6.2 Upper Bounds via Flag Sums-of-Squares . . . 132
6.3 General Flag Algebras . . . 136
6.3.1 Razborov's Flag Algebras . . . 137
6.4 Möbius Transformations: Swapping between Induced and Non-Induced Densities . . . 139
6.5 Connection to Polynomial Optimization . . . 143
6.6 Harmonic Flags: The Fourier Expansion of Quantum Flags . . . 147

7 Gatermann-Parrilo Reduction for Polynomial Optimization with S_n Symmetry 155
7.1 Reducing Polynomial Optimization Problems . . . 156
7.1.1 Representation Theory . . . 157
7.1.2 Block-Diagonalization . . . 159
7.2 Representation Theory of the Symmetric Group . . . 163
7.2.1 Young Tableaux, Tabloids and Permutation Modules . . . 163
7.2.2 Irreducible Specht Modules . . . 165
7.2.3 A Symmetry Adapted Basis of Permutation Modules . . . 166
7.2.4 Multiplying and Symmetrizing the Elements of the Symmetry Adapted Basis . . . 168

8 Symmetry Reduced Flag SOS Hierarchies 171
8.1 Warm Up: One Predicate with One Argument. . . . 173
8.2 The (Hyper-) Graph Case . . . 174
8.3 The General Case: Decomposing the Module $M^{F,n}$ . . . 176
8.3.1 The Decomposition of $M^{F,n}$ *Stabilizes* for n Big Enough . . . 178
8.3.2 Decomposing the Modules $M^{F,n}$ Explicitly . . . 179
8.3.3 Multiplying and Symmetrizing the Elements of the Symmetry Adapted Basis . . . 185

8.4 Example and Interpretation: The Reduced Second Level of the Lasserre Hierarchy for Graphs 186
8.5 Applications: Graph Profiles 195
8.6 Flag Algebras for Degenerate Extremal Combinatorics 206
8.7 Software: FlagSOS.jl 211

9 An Alternative Hierarchy Intertwining the Lasserre Hierarchy for Binary Problems with S_n Symmetry 213
9.1 A Vertex Truncation of the Lasserre Hierarchy 215
9.2 Decomposition into Maximal Cliques 218
9.3 Razborov's Hierarchy 222

A Characterizing Nonnegative Projections 229

Index 233

List of Figures

1.1 A convex set C with symmetry σ, and its invariant set $\{x \in C : x = \sigma(x)\}$ in red. 2

4.1 Example of an $n_1 \times n_2 = 8 \times 8$ grid tiling with $m = 4$, and the corresponding toric interpretation of the 8×8 grid. 61
4.2 Optimal arrangements on a 6×6 grid 84
4.3 Optimal arrangements on a 7×7 grid 84
4.4 Optimal arrangements on a 8×8 grid 85
4.5 Optimal arrangements on a 10×10 grid 85
4.6 A small selection of cases where the SDP bound is sharp for larger grids. 86

5.1 The three patterns of pairs of edges in case $(d = 2, L = 2)$ 97
5.2 The eight patterns of triplets of edges in case $(d = 3, L = 2)$ 97

8.1 The edge-triangle graph-profile 199
8.2 Outer approximation of the edge-triangle profile 200
8.3 Piecewise outer approximation of the edge-triangle profile 201
8.4 Graph profiles 202
8.5 Harmonic graph profiles 204
8.6 Outer approximation of the profile of the character flags corresponding to the edge and K_4 205

List of Tables

2.1 Results of the numerical symmetry reduction of the Theta' function of Erdős-Rényi graphs. 36
2.2 The resulting bounds for the stable set number of Erdős-Rényi graphs. 37

3.1 Results of the numerical symmetry reduction of QAPLib problems using Algorithm 3. 56
3.2 Details on solving (3.1) for QAPLib instances. 57

4.1 Number of variables before and after symmetry reduction. 73
4.2 The bounds on a 6x6 grid 87
4.3 The bounds on a 7x7 grid 88
4.4 The bounds on a 8x8 grid 89
4.5 The bounds on a 10x10 grid 91

5.1 Case $d = 3, L = 2$ 123
5.2 Case $d = 4, L = 2$ 124
5.3 Case $d = 5, L = 2$ 125
5.4 Case $d = 6, L = 2$ 128

8.1 Reynolds operators of S_2 on $\mathrm{Hom}(S^\mu, M^{(n-4,2,2)})$ 187
8.2 The symmetry adapted basis of the flag algebra of graphs up to two edges. The vertices have been arranged in order of vertex groups $V_0, \ldots, V_k$ from top to bottom. 195
8.3 The block-sizes of the symmetry reduced Lasserre hierarchy generated by graphs with up to $d = 5$ edges (i.e., graphs with up to $2d$ edges appear in the coefficients). 196
8.4 The block-sizes of the symmetry reduced Lasserre hierarchy generated by graphs with up to $t = 6$ vertices (i.e., graphs with up to $2t$ vertices appear in the coefficients). 196

8.5 The block-sizes of the symmetry reduced Lasserre hierarchy generated by *bipartite* graphs up to $d = 5$ edges resp., $t = 7$ vertices. . . . 197
8.6 Total number of non-zero coefficients in the symmetry reduced Lasserre hierarchy for two different bases, generated by orbits of graphs with at most d edges respectively at most t vertices. 206

9.1 Comparing the block sizes of the Razborov hierarchy before and after additional symmetry reduction 227

List of Algorithms

1 Finding $S_{\min}$ 27
2 Partition algorithm ([Per17]) 29
3 Partition algorithm, randomized ([Per17]) 30
4 Simulated annealing algorithm 81
5 Decompose a generalized tableau T in terms of semistandard tableaux 184

Notation

Sets

$A \cup B$	Union of A and B
$A \sqcup B$	Disjoint union of A and B
$A \cap B$	Intersection of A and B
$\lvert A \rvert$	Cardinality of A
$A \subseteq B$	A is a subset of B
$A \setminus B$	Set difference $\{x \in A : x \notin B\}$
$[k]$	The set of integers 1 trough k
$\mathbb{N}$	The natural numbers
$\aleph_0$	The cardinality of the natural numbers
$\mathrm{Aut}(G)$	The group of automorphisms of G
S_n	The symmetric group over n elements
$\mathbb{R}^{m \times n}$	The space of real $m \times n$ matrices
$\mathbb{R}_+^{m \times n}$	The space of real $m \times n$ matrices with nonnegative entries
$\mathbb{S}^n$	The subspace of symmetric matrices in $\mathbb{R}^{n \times n}$
$\mathbb{S}_+^n$	Cone of $n \times n$ positive semidefinite matrices
$\mathcal{N}^n$	Symmetric matrices with nonnegative entries, $\mathbb{S}^n \cap \mathbb{R}_+^{n \times n}$
$\mathcal{D}^n$	Doubly nonnegative matrices, $\mathbb{S}_+^n \cap \mathcal{N}^n$
$\mathcal{P}_n$	The collection of subsets of $[n]$
$\lvert \alpha \rvert$	$\sum_{i=1}^n \alpha_i$ of an integer sequence $\alpha \in \mathbb{N}^n$
$\mathbb{N}_d^n$	Set of integer sequences α of length n with $\lvert \alpha \rvert = d$
$\mathrm{cl}(A)$	The closure of a set A
$\mathrm{int}(A)$	The interior of a set A
$\mathbb{F}_q$	Finite field with q elements
Δ_m	Standard simplex $\left\{x \in \mathbb{R}^m : x \geq 0, \sum_{e \in E} x_e = 1\right\}$
$\mathbb{B}^n$	Binary hypercube $\{0,1\}^n$

$\emptyset$	The empty set

Conic optimization

$\mathcal{K}$	A (closed, convex, pointed) cone
$\mathcal{K}^*$	The dual cone of $\mathcal{K}$
min	Minimum
inf	Infimum
max	Maximum
sup	Supremum

Semidefinite programming

$\langle A, B \rangle$	Matrix inner product $\sum_{i,j} A_{i,j} B_{i,j}$
$X \succcurlyeq 0$	X is positive semidefinite
$X \succcurlyeq Y$	Shorthand for $X - Y \succcurlyeq 0$
$H(f)(x)$	Hessian of f in x
Las_{2d}	The dth level of the Lasserre hierarchy
ϑ'	The theta prime function

Polynomial optimization

$\mathbb{R}[x]$	(Real) polynomials in x
$\mathbb{R}[x]_t$	(Real) polynomials in x of degree at most t
$M_t(y)$	Moment matrix of order t of y
$\Sigma[x]$	The cone of sums-of-squares
$\langle f_1, \ldots, f_m \rangle$	Ideal generated by $f_1, \ldots, f_m$
$\mathbb{R}[x]/I$	Quotient algebra
$[X]$	Monomial basis vector of $\mathbb{R}[x]$

Linear Algebra

I	Identity matrix (of fitting size)
I_n	Identity matrix of size $n \times n$

J	All-one matrix (of fitting size)
$J_{m,n}$	All-one matrix of size $m \times n$
J_n	Shorthand for $J_{n,n}$
v^T	Transposed vector/matrix
$V^\perp$	The orthogonal complement of a subspace V
$\mathrm{tr}(A)$	Trace of the matrix A
P	A projection operator
P^*	The adjoint projection of P
P_S	Orthogonal projection onto a subspace S
$\mathcal{J}$	Euclidean Jordan algebra
$X \bullet Y$	Commutative matrix product $\frac{1}{2}(XY + YX)$
$X \circ Y$	Hadamard product
$X \otimes Y$	Kronecker product
δ_{ij}	Kronecker-Delta (1 if $i = j$, 0 otherwise)
$V \oplus W$	Direct sum of two vector spaces
$\vec{A}$	Vectorized matrix (gluing the columns together)
P_σ	Permutation matrix of a $\sigma \in S_n$
$\lambda_{\min}$	Minimum eigenvalue
$\mathrm{GL}(\mathcal{V})$	General linear group of $\mathcal{V}$
$v\|_W$	Projection of a vector $v \in V$ to a subspace $W \subseteq V$
$\mathrm{span}(A)$	The (real) span of the elements of A

Representation theory

$\mathcal{R}_G(x)$	Reynolds (averaging) operator of G applied on x
$\mathcal{V}$	(Left) G-module, where G is a group
$\mathcal{V}^G$	Subspace of G-invariant elements of $\mathcal{V}$
$\rho : G \to \mathrm{GL}(\mathcal{V})$	Linear representation of G
$\lambda \vdash n$	A partition of $n = \|\lambda\|$
$\begin{array}{\|c\|c\|c\|c\|}\hline 1&2&3&4\\\hline 5&6\\\cline{1-2}\end{array}$	A Young-tableau of shape $(4,2)$
$\begin{array}{\|cccc\|}\hline 1&2&3&4\\\hline 5&6\\\cline{1-2}\end{array}$	A Young-tabloid of shape $(4,2)$
$\begin{array}{\|c\|c\|c\|c\|}\hline 1&1&1&2\\\hline 2&3\\\cline{1-2}\end{array}$	A generalized tableau of shape $(4,2)$ and content $(3,2,1)$
$t \sim t'$	Row-equivalence between tableaux
$\{t\}$	Row-equivalence class (tabloid) of t
M^λ	Permutation module, the span of tabloids of shape λ

C_t	Column-stabilizer of a tableau t
e_t	Polytabloid of a tableau t
S^λ	Specht module
$\mu \trianglerighteq \lambda$	The partition μ dominates the partition λ
$K_{\mu\lambda}$	Kostka-numbers
$t[T]$	Tabloid of the tableau with entry $t(i,j)$ in its $T(i,j)$th row
$\vartheta_T : M^\mu \to M^\lambda$	Homomorphism constructed from a generalized tableau T

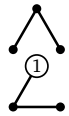
Probability theory

$\mathbf{x}, \mathbf{V}$, etc.	Random objects
$\mathbb{P}[f(\mathbf{x})]$	Probability that the event $f(\mathbf{x})$ is true
$\mathbb{E}[g(\mathbf{x})]$	Expectation of $g(\mathbf{x})$

Graph theory

$G = (V, E)$	(Hyper-) graph
$V(G)$	Set of vertices of a graph
$E(G)$	Set of (hyper-) edges of a graph
$\alpha(G)$	Independence number of a graph G
C_k	Cycle of length k
P_k	Path with k edges
$K_{n,m}$	Complete bipartite graph with bipartition of sizes n,m
$\mathrm{ER}(q)$	Erdős-Rényi graph
$G \simeq H$	Graph isomorphism
$G \subseteq G'$	G is a (non-induced) subgraph of G', i.e. $V(G) \subseteq V(G')$ and $E(G) \subseteq E(G')$
$G\|_W$	Subgraph of G on vertices $W \subseteq V(G)$
$\deg_G(v)$	The degree of vertex v in graph G
$\varnothing$	The empty graph
$\mathrm{Aut}(G)$	Automorphism group of G

Flag algebras

$\mathcal{G} = (G_n)_{n\geq 1}$	Sequence of graphs with $\lvert V(G_n)\rvert = n$
$p(\mathcal{G}, H)$	Density of H as non-induced subgraph in the sequence $\mathcal{G}$
$p_{\text{ind}}(\mathcal{G}, H)$	Density of H as induced subgraph in the sequence $\mathcal{G}$
$\text{ex}_n(G; H)$	Maximum density of G in H-free models (graphs) of size n
$\text{ex}(G; H)$	Maximum density of G in H-free models (graphs) in the limit
	Unlabeled flag with three vertices; the density of P_2 in the limit
	Partially labeled flag
$[\![\cdot]\!]$	Unlabeling operator
P_i	Predicate in the current model theory
$P_{\text{edge}}(i, j)$	Edge indicator predicate
M_{ind}	Induced flag
$M_{\text{non-ind}}$	Non-induced flag
$\mathcal{M}_n$	Models on vertices $[n]$
χ_G	Harmonic flag
$\text{shape}(F)$	The shape of a flag
$M^{F,n}$	The flag module on n vertices
$\text{profile}(G, H)$	The graph profile of G and H
$G_{f(n)}$	A degenerate flag, the density of G at rate f
Raz_T	The Tth level of the Razborov hierarchy

1 Introduction

The overarching topic of this thesis is (convex) optimization under symmetries. The basic idea is simple: A *convex* set $C \subseteq \mathbb{R}^n$ contains all line segments between pairs of elements of C. A *convex optimization problem* p optimizes a linear objective function $f : C \to \mathbb{R}$ over a convex set C:

$$\begin{aligned} p = \min \ & f(x) \\ \text{s.t. } & x \in C. \end{aligned}$$

We call an invertible linear function $\sigma : C \to C$ a *symmetry* of the optimization problem p, if both $\sigma(C) = C$ and $f(\sigma(x)) = f(x)$, i.e., it sends elements of C to C, and does not change the objective value $f(x)$. The set of all symmetries of an optimization problem forms a *group* G, as applying symmetries consecutively always results in another symmetry of the problem. In this thesis the group G is always assumed to be finite.

Then, given an optimal solution $x^\star$ of p, we obtain another optimal solution $\sigma(x^\star)$ by applying a symmetry $\sigma \in G$ to $x^\star$. Since the convex set C is closed under taking convex combinations, we can, by linearity of the objective, average an optimal solution over the whole group G to obtain an *invariant optimal solution*

$$\frac{1}{|G|} \sum_{\sigma \in G} \sigma(x^\star).$$

This tells us that there is always a symmetric optimal solution of p, and allows us to rewrite the optimization problem as

$$\begin{aligned} p = \min\ & f(x) \\ \text{s.t. } & x \in C, \\ & x = \sigma(x) \quad \text{for all } \sigma \in G. \end{aligned}$$

Often, the set of symmetric solutions $\{x \in C : x = \sigma(x) \text{ for all } \sigma \in G\}$ is significantly easier to work with than C. For example, in Figure 1.1 the set C is a convex set, with symmetry given by reflection around its horizontal axis. There the invariant set is simply the red horizontal line, which is both lower dimensional and easier to describe than C.

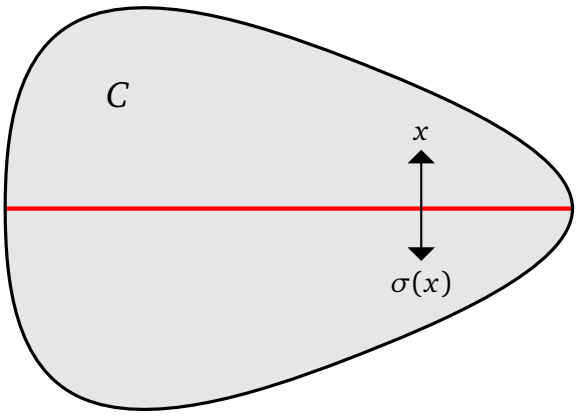


Figure 1.1: A convex set C with symmetry σ, and its invariant set $\{x \in C : x = \sigma(x)\}$ in red.

This thesis investigates different approaches to symmetry reduction, both for detection and exploitation of symmetries, as well various applications coming from physics, queuing theory and (extremal) combinatorics. In this chapter we give a basic introduction to convex (conic) programming, ways to relax problems to fit this framework, even if they are not initially convex, and ways to exploit symmetries.

1.1 Convex (Conic) Optimization

We are interested in linear optimization over slices of (proper) convex cones. A subset $\mathcal{K} \subseteq \mathbb{R}^n$ is called a *convex cone*, if it is closed under non-negative linear combinations. In other words, given any two points $x, y \in \mathcal{K}$ and nonnegative scalars $\alpha, \beta \in \mathbb{R}_+$, the vector

$$\alpha x + \beta y$$

should again be an element of $\mathcal{K}$.

Furthermore, we assume that $\mathcal{K}$ fulfills three basic properties:

- $\mathcal{K}$ is *pointed*: $\mathcal{K} \cap (-\mathcal{K}) = \{0\}$,
- $\mathcal{K}$ is *closed*: $\mathcal{K} = \mathrm{cl}(\mathcal{K})$,
- $\mathcal{K}$ has *non-empty interior*: $\mathrm{int}(\mathcal{K}) \neq \emptyset$.

The first property allows us to define a *partial order* on $\mathbb{R}^n$ by defining $x \succcurlyeq y$ as $x - y \in \mathcal{K}$, while the other two properties are important to obtain strong duality and complexity results. In particular, a closed convex cone satisfies the bipolar theorem assumptions, so that $(\mathcal{K}^\star)^\star = \mathcal{K}$ [Bar02, Chapter IV, Theorem (1.2)], where $\mathcal{K}^\star$ denotes the dual cone defined below (1.2). Moreover, the nonempty interior condition allows for the Slater constraint qualification, which is a sufficient condition in strong duality [Bar02, Chapter IV, page 171, exercise 4]. Finally, in the theory of interior point methods for conic optimization (e.g. [Ren01]), the domain of a self-concordant barrier is usually the interior of $\mathcal{K}$. Convex cones which fulfill all three properties are called *proper cones*.

A *primal conic optimization problem* is of the form

$$\begin{aligned} p^\star = \inf\ & c^T x \\ \text{s.t. } & Ax = b, \\ & x \in \mathcal{K}, \end{aligned} \tag{1.1}$$

where c is a vector in $\mathbb{R}^n$, while $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ give a system of linear equations. Since $Ax = b$ defines an affine subspace of $\mathbb{R}^n$, this is the problem of finding a point as far as possible in direction of c in a slice of the cone $\mathcal{K}$.

Duality. Every conic optimization problem has a corresponding *dual* optimization problem, which optimizes over the *dual cone*

$$\mathcal{K}^\star := \{y \in \mathbb{R}^n : x^T y \geq 0 \quad \text{for all } x \in \mathcal{K}\}. \tag{1.2}$$

The dual conic program of (1.1) is given by

$$\begin{aligned} d^\star = \sup\ & b^T y \\ \text{s.t. } & c - A^T y \in \mathcal{K}^\star, \\ & y \in \mathbb{R}^m. \end{aligned} \tag{1.3}$$

The main idea of many of the bounds considered in this thesis is that every feasible dual solution (i.e., any y with $A^T y - c \in \mathcal{K}^\star$) gives a valid bound for the original, primal optimization problem.

Proposition 1.1 (weak duality)**.** *Let x be feasible for the primal* (1.1) *and y be feasible for the dual* (1.3)*. Then*

$$c^T x \geq p^\star \geq d^\star \geq y^T b.$$

Proof. Since x and y are feasible for their respective programs, we have

$$c^T x - y^T b = c^T x - y^T A x = \underbrace{(c - A^T y)}_{\in \mathcal{K}^\star}^T \underbrace{x}_{\in \mathcal{K}} \geq 0.$$

□

Linear Programming. We are mainly interested in two special cases of conic optimization: Linear, and semidefinite programming. A *linear programming problem* (LP) is a problem of the form (1.1), where we optimize over the *nonnegative orthant*

$$\mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x_i \geq 0 \quad \text{for } i = 1, \ldots, n\}.$$

The nonnegative orthant is a proper cone of dimension n, which is *self-dual*:

$$(\mathbb{R}^n_+)^\star = \mathbb{R}^n_+.$$

Thus, primal and dual linear programs are of the form

$$\begin{aligned} p^\star = \inf\ & c^T x \\ \text{s.t. } & Ax = b, \\ & x \geq 0. \end{aligned} \qquad \geq \qquad \begin{aligned} d^\star = \sup\ & b^T y \\ \text{s.t. } & c - A^T y \geq 0, \end{aligned}$$

Semidefinite Programming. The second class of problems we are interested in optimizes over the *cone of positive semidefinite matrices*, given by

$$\mathbb{S}^n_+ := \{X \in \mathbb{S}^n : v^T X v \geq 0 \quad \text{for all } v \in \mathbb{R}^n\},$$

where $\mathbb{S}^n$ denotes the set of symmetric $n \times n$ matrices over the real numbers, i.e., matrices $X \in \mathbb{R}^{n \times n}$ with $X^T = X$. We write $X \succcurlyeq 0$ for $X \in \mathbb{S}^n_+$, whenever n is clear from context. The inner product, an analogue to $x^T y$ for vectors, is here the *trace inner product*, which we define by

$$\langle X, Y \rangle := \operatorname{tr}(XY) = \sum_{i,j=1,\ldots,n} X_{i,j} Y_{i,j}.$$

Given any vector $u \in \mathbb{R}^n$, the $n \times n$ matrix uu^T is positive semidefinite, since, given any other vector v, we have

$$v^T(uu^T)v = (v^T u)(u^T v) = (v^T u)^2 \geq 0. \tag{1.4}$$

Clearly, nonnegative linear combinations of positive semidefinite matrices are again positive semidefinite, so sums of terms of the form uu^T are positive semidefinite. Indeed, all positive semidefinite matrices are of this form.

Theorem 1.2 (spectral decomposition)**.** *Let X be a symmetric $n \times n$ matrix. Then there exists an orthonormal basis of eigenvectors $u_1, \ldots, u_n \in \mathbb{R}^n$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$, such that*

$$X = \sum_{i=1}^{n} \lambda_i u_i u_i^T.$$

It immediately follows that a symmetric matrix X is positive semidefinite if and only if all of its eigenvalues are nonnegative, since, given one of the eigenvectors u_i of X, we have

$$u_i^T X u_i = \lambda_i \geq 0,$$

if X is positive semidefinite. Moreover, by observation (1.4) above, conic combinations of terms of the form uu^T are positive semidefinite, which shows the other direction.

Corollary 1.3. *The cone of positive semidefinite matrices is self-dual, i.e.,* $(\mathbb{S}^n_+)^\star = \mathbb{S}^n_+$.

Proof. Let X be an element of $(\mathbb{S}^n_+)^\star$, and $v \in \mathbb{R}^n$ be any vector. Then

$$v^T X v = \langle \underbrace{vv^T}_{\succcurlyeq 0}, X \rangle \geq 0,$$

and thus $X \in \mathbb{S}^n_+$.

Let Y now be an element of $\mathbb{S}^n_+$, with spectral decomposition $Y = \sum_{i=1}^n \lambda_i u_i u_i^T$. Then, for any other element $X \in \mathbb{S}^n_+$, we have

$$\langle X, Y \rangle = \sum_{i=1}^n \lambda_i u_i^T X u_i \geq 0,$$

since X is positive semidefinite and $\lambda_i \geq 0$, showing that $Y \in (\mathbb{S}^n_+)^\star$. □

Thus, the general form of a semidefinite programming problem (*SDP*) is

$$\begin{aligned} p^\star = \inf \ & \langle C, X \rangle && \geq \quad d^\star = \sup \ b^T y \\ \text{s.t. } & \langle A_i, X \rangle = b_i \text{ for } i = 1, \ldots, m, && \text{s.t. } C - \sum_{i=1}^m y_i A_i \succcurlyeq 0, \\ & X \succcurlyeq 0. \end{aligned}$$

Here we are now working with data matrices C and A_i instead of a vector c and matrix A before.

Remark 1.4. If we assume that all matrices C and A_i are diagonal matrices, then the SDP simplifies to an LP. Since we can turn any LP into an SDP by turning vectors into matrices by writing the vectors on the diagonal entries, and filling the off-diagonal entries with zeros, linear optimization can be as special case of SDPs.

1.2 Relaxing Non-Convex Problems by Semidefinite Programming

In general, convex conic optimization is NP-hard (see, for example, the recent survey [DR21] on optimization over the copositive cone). But we can solve SDPs and LPs, for a fixed approximation error ϵ, in polynomial time (in the Turing

model) using an *interior point algorithm*, under some weak assumptions [KV16]. More precisely, the runtime is polynomial in the size of the matrices, the number of constraints, the bit size of the problem data, as well as the logarithms of $\frac{R}{r}$ and $\frac{1}{\epsilon}$, where R and r are radii of balls respectively containing and contained in the feasible set. This turns SDPs into a very powerful tool, as we can *relax* various hard problems to SDPs.

Example: The Theta Function and Symmetries. Before we are going to introduce some of the more general, and widely known, relaxation hierarchies, let us first consider a basic example of a semidefinite programming relaxation. We are going to relax the NP-complete problem of determining the size of a *maximum stable set* of a (simple, undirected) graph $G = (V, E)$. For simplicity, we enumerate the vertices as $V = [n] \coloneqq \{1, \dots, n\}$. Thus, the set of edges E is a subset of $\{\{i, j\} : 1 \leq i < j \leq n\}$.

The Independence Number of a Graph. A set of vertices $S \subseteq V$ is called *stable*, if there are no edges between any two distinct vertices in S. We now want to maximize the size of the stable set, which gives us the *independence number* $\alpha(G)$ of the graph:

$$\alpha(G) \coloneqq \max\{|S| : S \text{ is a stable set in } G\}.$$

This problem is well known to be strongly NP-complete [GJ78], and is hard to solve in practice even for relatively small graphs.

Binary Reformulation. We can now translate $\alpha(G)$ into an optimization problem over binary variables by introducing one variable $x_i \in \{0, 1\}$ for each vertex $i \in V$, where $x_i = 1$ denotes that i is part of the chosen stable set S. This vector x is also called the *characteristic vector* of the solution S. One way to enforce that the chosen subset S is stable is by enforcing $x_i x_j = 0$ if $\{i, j\} \in E$:

$$\alpha(G) = \max\left\{x^T x : x_i x_j = 0 \text{ for } \{i, j\} \in E, x \in \{0, 1\}^V\right\},$$

where $x^T x = \sum_{i \in V} x_i = |S|$.

Semidefinite Programming Bound. The basic idea of this semidefinite programming bound is to replace the binary variables x with their scaled outer product $\frac{1}{x^T x} x x^T$, which is a rank one positive semidefinite matrix. Relaxing the rank

one constraint leads to the semidefinite program, first introduced by Lovász in [Lov79], given by

$$\alpha(G) \leq \vartheta(G) := \sup\{\langle J, X\rangle : X \in \mathbb{S}^n_+, \operatorname{tr}(X) = 1, X_{i,j} = 0 \text{ for all } \{i,j\} \in E\}, \quad (1.5)$$

where J denotes the all-one matrix of appropriate size. It is easy to check that $\hat{X} = \frac{1}{x^T x} x x^T$ is a feasible solution of $\vartheta(G)$, if x corresponds to a stable set $|S|$ of G. Indeed, we have

$$\langle J, \hat{X}\rangle = \sum_{i,j \in V} \hat{X}_{i,j} = \frac{1}{x^T x} \sum_{i,j \in V} x_i x_j = x^T x = |S|$$

and

$$\operatorname{tr}(\hat{X}) = \operatorname{tr}\left(\frac{1}{x^T x} x x^T\right) = \frac{x^T x}{x^T x} = 1.$$

We defined $\hat{X}$ to be positive semidefinite, and since x corresponds to a stable set the entries $\hat{X}_{i,j}$ are zero if $\{i,j\}$ is an edge of G.

The Theta Number of the 5-Cycle. Let us take a look at a small example: Let G be the cycle on 5 vertices C_5, ordered such that i connects to $i+1$ and 1 to 5. Solving the SDP (1.5) with an interior point solver returns something very close to the optimal solution

$$\begin{pmatrix} \frac{1}{5} & 0 & \frac{\sqrt{5}-1}{10} & \frac{\sqrt{5}-1}{10} & 0 \\ 0 & \frac{1}{5} & 0 & \frac{\sqrt{5}-1}{10} & \frac{\sqrt{5}-1}{10} \\ \frac{\sqrt{5}-1}{10} & 0 & \frac{1}{5} & 0 & \frac{\sqrt{5}-1}{10} \\ \frac{\sqrt{5}-1}{10} & \frac{\sqrt{5}-1}{10} & 0 & \frac{1}{5} & 0 \\ 0 & \frac{\sqrt{5}-1}{10} & \frac{\sqrt{5}-1}{10} & 0 & \frac{1}{5} \end{pmatrix}$$

with objective value $\sqrt{5} > \alpha(C_5) = 2$. We can see here that the returned solution has only three unique entries: $1/5$ on the diagonal, 0 for entries corresponding to edges, and $(\sqrt{5}-1)/10$ for the non-edges. This is not a coincidence! The pattern of the solution corresponds exactly to the orbits of edges (or more precisely, *pairs of vertices*) under the automorphism group of C_5. This observation was already used by Lovász to compute the Shannon capacity of C_5 [Lov79].

Not all feasible solutions have to be invariant under symmetries, but we will see below that there always exists an invariant optimal solution. Interestingly, an interior point solver will always return such a solution [Kan+01].

Symmetric Optimal Solutions. The main idea of symmetry reduction is that, while not all feasible solutions of convex optimization problems are symmetric, there is always a symmetric optimal solution. Indeed, let us assume that we have an optimal solution $\tilde{X}$ of $\vartheta(G)$, which is not (necessarily) symmetric. Then, for any given automorphism $\sigma : V \to V$ of G, we get another optimal solution by applying σ on both coordinates of $\tilde{X}$ simultaneously:

$$\sigma(\tilde{X}) = (\tilde{X}_{\sigma(i),\sigma(j)})_{i,j\in V}.$$

Since graph automorphisms send edges to edges, $\sigma(\tilde{X})$ is again a feasible solution. Note that the matrix is positive semidefinite again, as $\sigma(\tilde{X})$ simply *permutes rows and columns* of $\tilde{X}$. Since the objective function is invariant under the action of $\mathrm{Aut}(G)$, the objective value of $\sigma(\tilde{X})$ is the same as the one of $\tilde{X}$. This shows that the all automorphisms of G are symmetries of the SDP $\vartheta(G)$ (1.5).

As the semidefinite cone (and thus the feasible set of $\vartheta(G)$) is a convex set, convex combinations of feasible solutions are again feasible. This allows us to *symmetrize* the optimal solution by

$$\mathcal{R}_{\mathrm{Aut}(G)}(\tilde{X}) := \frac{1}{|\mathrm{Aut}(G)|} \sum_{\sigma\in\mathrm{Aut}(G)} \sigma(\tilde{X}),$$

which always gives us an optimal solution which is invariant under the action of $\mathrm{Aut}(G)$. Hence, we can further restrict the feasible set of $\vartheta(G)$ to invariant solutions:

$$\begin{aligned} \vartheta(G) = \sup\ & \langle J, X\rangle \\ \text{s.t. } & X \in \mathbb{S}^n_+, \\ & \mathrm{tr}(X) = 1, \\ & X_{i,j} = 0 \text{ for all } \{i,j\} \in E, \\ & \boxed{\mathcal{R}_{\mathrm{Aut}(G)}(X) = X.} \end{aligned}$$

The constraint $\mathcal{R}_{\mathrm{Aut}(G)}(X) = X$ ensures that $X_{i,j} = X_{\sigma(i),\sigma(j)}$ for each $\sigma \in \mathrm{Aut}(G)$. We can now introduce one variable for each *orbit of pairs* of vertices $i, j \in V$

$$\{(\sigma(i),\sigma(j)) \ : \ \sigma \in \mathrm{Aut}(G)\},$$

and rewrite the variable X in the form

$$X = \sum_{o \text{ orbit of pairs of vertices}} x_o B_o,$$

where

$$(B_o)_{i,j} = \begin{cases} 1, & \text{if } (i,j) \in o, \\ 0, & \text{else.} \end{cases}$$

This already reduces the number of variables in the SDP to the number of orbits of pairs of vertices of G, and can give a significant speed-up for solving the problem. Later (basics in Section 1.3.1, and details for reductions of polynomial optimization problems in Chapter 7) we will further discuss how to *block-diagonalize* the problem to reduce the sizes of the semidefinite matrices appearing in the problem.

1.2.1 Higher Order Relaxations: The Lasserre Hierarchy (for Stable Set)

The Theta-function contains variables corresponding to all products between pairs of variables. The Lasserre hierarchy provides a systematic way to *strengthen* the bounds for binary optimization problems by adding variables for products between bigger tuples of variables.

The hierarchies we work with always come in *levels*, of increasing complexity and accuracy, given by a natural number. Here, we let t denote this level of the hierarchy. For the Theta-function we took the outer product of the characteristic vector of a vertex set with itself. We still want to take the outer product of a vector describing a solution, but we now extend it first to contain *all products between at most t variables*.

$$\begin{aligned} [x]_1 &:= (1, x_1, x_2, \ldots, x_n), \\ [x]_2 &:= (1, x_1, \ldots, x_n, x_1x_2, x_1x_3, \ldots, x_{n-1}x_n), \\ &\vdots \\ [x]_t &:= \left(\prod_{i \in I} x_i : I \subseteq [n], |I| \leq t \right). \end{aligned}$$

The entries of the outer product $[x]_t[x]_t^T$ are generally not unique: We have, for example, that $x_1 \cdot x_3 = x_1x_3 \cdot x_3 = x_1x_3$. Since all variables x_i are binary, the product of two monomials in $[x]_t$ depends only on the union of their indices.

Hence, the outer product is of the form

$$[x]_t[x]_t^T = \left(\prod_{i\in I\cup J} x_i\right)_{\substack{I,J\subseteq[n],\\ |I|,|J|\le t}} \succcurlyeq 0.$$

We can now relax this binary matrix to a general positive semidefinite matrix with the same *pattern* of entries. To do this, we simply introduce a variable y_J for each subset J of $[n]$ with at most $2t$ elements, and define the *moment matrix* of order t of y as

$$M_t(y) := (y_{I\cup J})_{\substack{I,J\subseteq[n],\\ |I|,|J|\le t}}.$$

This leads to the definition of the *tth level of the Lasserre hierarchy* [Las01] (for stable set), given by

$$\mathrm{Las}_{2t}(G) := \sup_y \left\{ \sum_{i\in[n]} y_i : y_\emptyset = 1, y_e = 0 \text{ for all } e\in E, M_t(y)\succcurlyeq 0 \right\}. \tag{1.6}$$

The first level of the Lasserre hierarchy for stable set is thus given by

$$\begin{aligned} \mathrm{Las}_2(G) = \sup_{y,Y}\ & \mathrm{tr}(Y) \\ \text{s.t. } & \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} \succcurlyeq 0, \\ & Y_{i,j} = 0 \quad \text{for all } \{i,j\}\in E \\ & y = \mathrm{diag}(Y), \end{aligned}$$

where $\mathrm{diag}(Y)$ is the vector of diagonal entries of Y. The Lasserre hierarchy can be seen as a generalization and strengthening of the Theta function $\vartheta(G)$, in the sense of the following proposition.

Proposition 1.5 ([LS91, Lemma 2.17])**.** *For any graph G, we have*

$$\vartheta(G) = \mathrm{Las}_2(G).$$

Symmetry in the Lasserre Hierarchy for Stable Set. Let us assume again that the graph G, of which we want to bound the independence number, has a non-trivial automorphism group $\mathrm{Aut}(G)$. In the Theta case we extended this action to the coordinates of the semidefinite programming relaxation, i.e., to *pairs* of vertices. Now, the hierarchy does not just have variables for each pair of variables,

but for each *tuple* of at most $2t$ different variables. We extend the action of $\mathrm{Aut}(G)$ in the obvious way to variables corresponding to tuples $J = (j_1, \ldots, j_k)$ by

$$\sigma(y_J) = y_{\sigma(J)} = \sigma(y_{\sigma(j_1),\ldots,\sigma(j_k)})$$

for any σ in $\mathrm{Aut}(G)$.

With the same *symmetrization* argument we used in the Theta case, we can restrict (1.6) to *invariant* optimal solutions to obtain a reduced, equivalent SDP.

$$\mathrm{Las}_{2t}(G) = \sup_y \left\{ \sum_{i\in[n]} y_i : y_\emptyset = 1, y_e = 0 \forall e \in E, M_t(y) \succcurlyeq 0, \mathcal{R}_{\mathrm{Aut}(G)}(y) = y \right\}.$$

1.2.2 Relaxing Polynomial Optimization Problems: Sums-of-Squares Hierarchies

What if we want to relax problems in variables that are not binary, or problems with more complicated constraints? It turns out that this approach generalizes to *polynomial optimization problems*. Given an objective and constraint polynomials $f, g_1, \ldots, g_m \in \mathbb{R}[x]$, we can define the optimization problem

$$\begin{aligned} &\inf_x \ f(x) \\ &\text{s.t.} \ \ g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m. \end{aligned} \tag{1.7}$$

Here, the approach is a bit different. First, we note that minimizing f is the same as determining the biggest scalar λ such that $f - \lambda$ is *nonnegative* on the set $\{x : g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m\}$. While the nonnegative polynomials form a convex cone, there is no practical way to work with it directly, as it is (NP-)hard to check whether a polynomial lies in it.

Instead, we relax *nonnegativity* to something that is easier to check: we optimize over polynomials that *are sums-of-squares* of polynomials of a fixed degree. Of course, as for SDPs, in computations we can only solve such problems up to an epsilon computationally. Obviously, the square of any polynomial is nonnegative, and the sum of nonnegative functions is again nonnegative. Thus, we obtain the cone of sums-of-squares of polynomials:

$$\Sigma[x] := \left\{ \sum_i f_i^2 : f_i \in \mathbb{R}[x] \right\}.$$

Sums-of-squares of polynomials correspond exactly to semidefinite matrices. To see this, let us first take a look at a single squared polynomial f^2. Let $[x]_d$,

where $d = \deg(f)$, be the vector of all monomials up to degree d. Note that the vector now also contains monomials with exponents higher than one, unlike the binary case we considered before. If we store all coefficients of f in a vector $c = (c_\alpha)_{x^\alpha \in [x]_d}$, we have $f = c^T[x]_d$; thus we can write f^2 as a trace inner product between a monomial matrix and a positive semidefinite matrix:

$$\begin{aligned} f^2 &= \left(\sum_{x^\alpha \in [x]_d} c_\alpha x^\alpha \right)^2 \\ &= \left(c^T[x]_d\right)^2 \\ &= c^T[x]_d[x]_d^T c \\ &= \operatorname{tr}([x]_d[x]_d^T cc^T) \\ &= \left\langle [x]_d[x]_d^T, cc^T \right\rangle. \end{aligned}$$

Since the trace inner product is (bi-)linear, we obtain positive semidefinite matrices from sums-of-squares, and sums-of-squares from positive semidefinite matrices by spectral decomposition.

In the problem (1.7) we also have polynomial constraints of the form

$$g_i(x) \geq 0.$$

We can use these in a straightforward way: We can construct nonnegative polynomials on the set $\{x : g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m\}$ by

$$s_0 + g_1 s_1 + \cdots + g_m s_m, \quad \text{where } s_i \in \Sigma[x].$$

While $g_i s_i$ is generally not nonnegative, it sure is on the set where $g_i \geq 0$. Putting things together, we can relax the problem (1.7) to an (infinite-sized) semidefinite optimization problem by

$$\begin{aligned} &\inf_x \ f(x) && \geq \sup \ \lambda \\ &\text{s.t. } \ g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m. && \text{s.t. } \ f - \lambda = s_0 + g_1 s_1 + \cdots + g_m s_m, \\ & && \quad s_i \in \Sigma[x]. \end{aligned}$$

To turn this into a finite problem (as well as a hierarchy of bounds of increasing quality), we simply bound the degree of each term $g_i s_i$ and s_0. The set of polynomials

$$M(g_1, \ldots, g_m) := \{s_0 + g_1 s_1 + \cdots + g_m s_m : s_i \in \Sigma[x]\} \tag{1.8}$$

is called the *quadratic module* generated by $g_1, \ldots, g_m$.

Theorem 1.6 (Putinar's Positivstellensatz, [Put93]). *Let*

$$K = \{x \in \mathbb{R}^n : g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}$$

and $M = M(g_1, \ldots, g_m)$. *If* M *is* Archimedean*, i.e. there exists an* $N \in \mathbb{N}$ *such that*

$$N - \sum_{i=1}^{n} x_i^2 \in M,$$

then

$$p(x) > 0 \quad \text{for all } x \in K$$

implies that $p \in M$.

1.3 Symmetric Convex Optimization

In most of the problems considered in this thesis we are interested in problems with *symmetries*. We saw a first example of this in Section 1.2, where one of the optimal solutions of the Theta function of the 5-cycle had just three different entries in the matrix. But, of course, we would prefer to know this *before* we solve the optimization problem, especially if the problem size is too big for it to be solved directly. Once we know the symmetry, or some form of subspace we can restrict our problem to, we want to reduce it further to remove the found "redundancies" in the problem.

In this thesis we will make use of a few different techniques to determine and exploit the symmetry of a given problem, each with their own advantages, disadvantages, and potential applications. Here we will describe the basic idea of symmetry reduction for semidefinite programming based on symmetries coming from a group G. Later, in Chapter 7, we will give a more detailed introduction into symmetry reduction for polynomial optimization, with focus on the case $G = S_n$.

1.3.1 Group Symmetry Reduction

Here we consider a semidefinite programming problem given in standard form

$$\begin{aligned} p^\star = \inf \ & \langle C, X \rangle \\ \text{s.t.} \ & \langle A_i, X \rangle = b_i \quad \text{for } i = 1, \ldots, m, \\ & X \in S_+^n, \end{aligned} \tag{1.9}$$

where C and the A_i are symmetric $n \times n$ matrices.

Group symmetry reduction requires that one already knows the symmetry of a given problem, in form of a *group* G acting on the variables. More precisely, let $\mathcal{V} = \mathbb{R}^n$ be the n-dimensional real (or, more generally, yet not much different, complex) vector space corresponding to the rows and columns of X. Now we pair up each element $\sigma \in G$ with a linear operator $P_\sigma \in \mathrm{GL}(\mathcal{V})$, such that $\sigma \mapsto P_\sigma$ is a group-homomorphism.

In the example above we had the automorphism group of the 5-cycle C_5 acting on $\mathbb{R}^5$ — corresponding to the five original, binary variables — by permuting indices. So here the operators P_σ simply correspond to permutation matrices, when choosing the canonical basis.

Such an action of a group on a vector space is called a *representation* of G, or, equivalently, turns $\mathcal{V}$ into a (left) G-module by defining $\sigma(v) := P_\sigma v$ for $v \in \mathcal{V}$. We will go into more detail about modules in the background Chapter 7, with focus on the case $G = S_n$. Here, we will just say that this gives us various algebraic tools, to work with and simplify invariant solutions.

As defined in Section 1.1 a matrix X is positive semidefinite, if and only if $v^T X v$ is nonnegative for every $v \in \mathcal{V}$. This gives us the motivation to use operation of G on $\mathcal{V}$ to define an action on the $n \times n$-matrix X itself, which corresponds to applying the same action on both vectors v simultaneously:

$$\sigma(X) = P_\sigma^T X P_\sigma.$$

Here again, if G acts by permuting the canonical basis of $\mathcal{V}$, it acts directly on the indices of X via

$$\sigma(X)_{ij} = X_{\sigma(i)\sigma(j)}.$$

Since $v^T \sigma(X) v = (P_\sigma^T v)^T X (P_\sigma v) \geq 0$, this operation sends positive semidefinite matrices to positive semidefinite matrices.

Remark 1.7. One can also set $\sigma(X)_{ij} = X_{\sigma^{-1}(i)\sigma^{-1}(j)}$ instead, depending on the way one defines the action of products of group elements. We fix that $(\sigma\pi)(x) = \sigma(\pi(x))$ in this thesis, but some authors prefer $x^{\sigma\pi} = (x^\sigma)^\pi$, which changes the order of execution of the two group elements (and thus turns $\mathcal{V}$ instead into a right-G-module). In practice these two conventions do not have much influence on the work in this thesis.

We say that the problem $p^\star$ (1.9) has symmetry G, if, given any feasible solution X and group element $\sigma \in G$, the matrix $\sigma(X)$ is again feasible with the same objective value as X.

Invariant Solutions. If an SDP has symmetry G, then we can restrict it to only optimize over *symmetric* (or *invariant*) solutions. Indeed, given an optimal solution $X^\star$ of (1.9), we can *symmetrize* it by averaging it over all elements of G:

$$\mathcal{R}_G(X^\star) \coloneqq \frac{1}{|G|}\sum_{\sigma\in G}\sigma(X^\star).$$

The resulting matrix is again feasible for the same SDP, with the same objective value as $X^\star$, and is invariant under action of any $\rho \in G$:

$$\rho(\mathcal{R}_G(X^\star)) = \frac{1}{|G|}\sum_{\sigma\in G}\rho(\sigma(X^\star)) = \frac{1}{|G|}\sum_{\sigma'\in G}\sigma'(X^\star) = \mathcal{R}_G(X^\star),$$

where we substituted the unique product $\rho^{-1}\circ\sigma' = \sigma$ in the second step. Thus, the original SDP (1.9) has the same optimal value as the same SDP restricted to invariant solutions:

$$\begin{aligned} p^\star = \inf\ & \langle C, X\rangle \\ \text{s.t. } & \langle A_i, X\rangle = b_i \quad \text{for } i = 1,\dots,m, \\ & X \in S^n_+, \\ & \sigma(X) = X \quad \text{for all } \sigma \in G. \end{aligned} \tag{1.10}$$

Since the action of G on matrices is linear, the additional constraints (1.10) define a subspace of $\mathbb{S}^n_+$. If we determine a basis $\{B_1,\dots,B_k\}$ of this subspace, we can reduce the number of variables in the SDP to obtain the equivalent SDP

$$\begin{aligned} p^\star = \inf\ & \sum_{i=1}^k x_i\langle C, B_i\rangle \\ \text{s.t. } & \sum_{i=1}^k x_i\langle A_i, B_i\rangle = b_i \quad \text{for } i = 1,\dots,m, \\ & \sum_{i=1}^k x_i B_i \in S^n_+, \\ & x_1,\dots,x_k \in \mathbb{R}, \end{aligned}$$

which optimizes over just k variables $x_1,\dots,x_k$. If G acts on the rows and columns of X via permutations, then the B_i are exactly the indicator matrices of *orbits of pairs of indices* (also called *orbitals*), which have a 1 in position (i,j) if (i,j) lies in the corresponding orbit, and 0 otherwise.

Block-Diagonalization. We can now use Artin-Wedderburn theory [Art27; Wed08] to *block-diagonalize* the space of invariant solutions (see also [Gij09] for a full proof). Not only can we reduce the number of variables to the number of pairs of orbits of indices k (resp. the dimension of the space of invariant matrices), but we can also replace the SDP constraints with multiple smaller ones.

The idea is here to transform the SDP according to a "symmetry-adapted basis" of $\mathcal{V}$, which is chosen according to the decomposition of $\mathcal{V}$ into irreducible submodules. This basis transformation block-diagonalizes the SDP; it turns all the matrices B_i into block-matrices with blocks of sizes $m_1, \dots, m_\ell$. Often, but not always, we have $m_1 + \cdots + m_\ell \ll n$, significantly reducing the size of the optimization problem.

1.4 Overview of the Thesis

Chapter 2: Jordan Symmetry Reduction over the Doubly Nonnegative Cone. We generalize the Jordan reduction method for symmetry reduction, which was recently introduced by Parrilo and Permenter in [PP19], to optimization over doubly-nonnegative matrices. These matrices are both positive semidefinite and entry-wise nonnegative, and appear in various strong, but usually hard to compute relaxations of combinatorial problems. While the reduction was originally introduced for symmetric cones, we show that it can be applied to non-symmetric cones as well. In this case, under small additional assumptions, we can show that the optimal reduction can be found using a partition-space based algorithm. We provide an implementation of this algorithm in form of a Julia-package, and use it to compute the ϑ'-function of Erdős-Rényi graphs for larger instances than what could be solved before [Kle+09b].

Chapter 3: Comparison and Reduction of Relaxations of the Quadratic Assignment Problem. Here we focus on three relaxations of the quadratic assignment problem, one based on eigenvalues, one on quadratic programming, and one on optimization over the doubly-nonnegative cone. We prove that the bounds do improve as one considers bounds which are harder to compute in practice. While this was certainly expected, it was surprisingly tricky to prove formally. Finally, we investigate the Jordan reduction method of Chapter 2 for the bound based on doubly-nonnegative matrices, and apply it to benchmark instances contained in QAPLib [BKR97].

Chapter 4: Energy Minimization on a Toric Grid as a QAP. Given m charged particles, how do we arrange them on a grid with $n_1 \times n_2$ possible positions, which we then repeat periodically in all directions? We investigate a strong bound based on optimization over the doubly nonnegative cone, and compare it with previous bounds from the literature. The bound, while growing quickly in the size of the grid, exhibits strong symmetries, which we exploit analytically using the Jordan reduction method of Chapter 2. This reduces the bound to an equivalent bound over $\mathcal{O}(n_1 n_2)$ second order cones, which is significantly easier to compute. We use this bound to prove (numerical) optimality of various periodic solutions, extending and complementing results by Bouman, Draisma and Van Leeuwaarden [BDL13], and make progress towards a construction of a sharp bound for the general 'checkerboard' case where we fill exactly half of the grid points.

Chapter 5: Optimizing Hypergraph-Based Polynomials Modeling Job-Occupancy in Queuing with Redundancy Scheduling. In this chapter we consider a question posed in [CBL21] coming from redundancy scheduling in queuing theory. Redundancy scheduling is based on the idea that sending the same job to multiple distinct servers can be advantageous, if balanced against the risk of wasted capacity. Here one wants to determine the optimal policy of choosing which subset of servers one should send the copies of the job to, and it is conjectured that the uniform probability distribution is optimal. This can be formulated as saying that a certain highly symmetric polynomial attains its minimum at the normalized all-one vector. While we do not manage to prove the general case, we prove a similar result for a simplification of the family of polynomials by exploiting its symmetries, as well as some special cases of the original problem.

Chapter 6: Four Different Views at Flag Algebras. In this chapter we give an introduction to Razborov's theory of flag algebras [Raz07], which can be used to model and approximate problems coming from extremal combinatorics. We show how three different bases of the algebra can be used interchangeably, each with their own advantages and disadvantages, each of which can equivalently be described by limits of polynomials in an increasing number of variables. While Razborov introduced the basis describing densities of induced sub-flags, we show that working with non-induced densities can be significantly easier. We use facts coming from harmonic analysis of the binary hypercube to first introduce a third basis, which we use to determine a recursive formulation of Sidorenko's conjecture, and later use in Chapter 8 for computational advantages.

Chapter 7: Gatermann-Parrilo Reduction for Polynomial Optimization with S_n Symmetry. In this background chapter we describe the symmetry reduction method of Gatermann and Parrilo [GP04] for polynomial optimization. First, the general theory is introduced for arbitrary groups, and then we give details for special case where the symmetry is given by an action of the full symmetric group S_n.

Chapter 8: Symmetry Reduced Flag SOS Hierarchies. In this chapter we fully exploit the symmetries of the Lasserre hierarchy for flag sums-of-squares. The translation of flags into (limits of) polynomials of [Ray+18], explained in Chapter 6, reveals a family of highly symmetric polynomial optimization problems, to which we then apply the reduction method described in Chapter 7. To make it possible to decompose the high dimensional S_n-modules in practice, we needed a novel algorithm to determine the decomposition of generalized permutation modules into irreducible submodules. We investigate the thus obtained reduced hierarchies, which are based on flags where vertices are grouped together instead of explicitly labeled, and use them to compute outer approximations of profiles of graphs and harmonic graphs. Finally, we give an example as to how flag algebras can be generalized to obtain certificates for the leading term of subgraph densities in degenerate cases, where the density of edges approaches zero as the number of vertices grows.

Chapter 9: An Alternative Hierarchy Intertwining the Lasserre Hierarchy for Binary Problems with S_n Symmetry. Here we investigate the hierarchy Razborov introduces for flag algebras in [Raz07]. This hierarchy prioritizes graphs with few vertices, in contrast to the Lasserre hierarchy prioritizing graphs with few edges. We show how one can obtain and generalize this hierarchy as a truncation of a high level of the Lasserre hierarchy, and reduce the size of the hierarchy further from there. Finally, we use facts from the representation theory of S_n to compare the resulting hierarchy to the Lasserre hierarchy in both directions.

1.5 Contributions to the Literature

This thesis is based on the following articles, while Chapters 6, 8 and 9 are based on unpublished work:

[BK22b] Daniel Brosch and Etienne de Klerk. "Minimum energy configurations on a toric lattice as a quadratic assignment problem". In: *Discrete Optimization* 44 (May 2022), p. 100612. DOI: 10.1016/j.disopt.2020.100612

[BLS21] Daniel Brosch, Monique Laurent, and Andries Steenkamp. "Optimizing Hypergraph-Based Polynomials Modeling Job-Occupancy in Queuing with Redundancy Scheduling". In: *SIAM Journal on Optimization* 31.3 (2021), pp. 2227–2254. DOI: 10.1137/20m1369592

[BK22a] Daniel Brosch and Etienne de Klerk. "Jordan symmetry reduction for conic optimization over the doubly nonnegative cone: theory and software". In: *Optimization Methods and Software* (2022), pp. 1–20. DOI: 10.1080/10556788.2021.2022146

These articles are used in the chapters of this thesis as follows:

Chapter 2	Based on [BK22a, Sections 1-3, 5–7]
Chapter 3	Based on [BK22a, Section 4] and [BK22b, Sections 1-2]
Chapter 4	Improves on [BK22b, Sections 3-6] by adding Section 4.3
Chapter 5	Based on [BLS21]
Chapter 6	Unpublished work
Chapter 7	Background material
Chapter 8	Unpublished work
Chapter 9	Unpublished work

2

Jordan Symmetry Reduction over the Doubly Nonnegative Cone

This chapter studies symmetry reduction of semidefinite programs (SDPs) where the matrix variable is also entry-wise nonnegative, i.e. symmetry reduction of conic linear programming over the doubly nonnegative cone. Such problems appear naturally, among others, in the study of convex relaxations of combinatorial problems. In particular, we are interested in such relaxations of the independence number of a graph (in this chapter), and of the quadratic assignment problem (in Chapter 3).

The independence number α of a graph G is the maximum number of nodes of G we can choose, such that there is no edge between any of them. The Theta-Prime function (ϑ'-function)[Sch79] is a semidefinite programming relaxation of α, and as such gives an upper bound to it. Given the adjacency matrix A of G, the function is defined by

$$\begin{aligned} \vartheta'(G) = \sup \ & \langle J, X \rangle \\ \text{s.t.} \ & \operatorname{trace}(X) = 1, \\ & \langle A, X \rangle = 0, \\ & X \geq 0, \end{aligned} \tag{2.1}$$

$$X \succcurlyeq 0.$$

The second family of problems we are interested in, which is investigated in Chapter 3, are quadratic assignment problems, which are of the form

$$QAP(A,B) = \min_{\varphi \in S_n} \sum_{i,j=1}^{n} a_{ij} b_{\varphi(i)\varphi(j)}, \tag{2.2}$$

where $A = (a_{ij})$ and $B = (b_{ij})$ are symmetric $n \times n$ matrices, and S_n denotes the symmetric group (i.e. all permutations) on n elements. Here we are interested in the SDP relaxation of by Zhao et al. [Zha+98], as reformulated by Povh and Rendl [PR09].

Symmetry reduction for SDP was first introduced by Schrijver in 1979 in [Sch79]; see for example the chapter [Bac+11] by Bachoc, Gijswijt, Schrijver and Vallentin for a review of later developments up to 2012. The specific case of SDP relaxations of quadratic assignment problems was investigated by de Klerk and Sotirov in [KS10; KS12].

Parrilo and Permenter [PP19] recently introduced a new — and more general — form of symmetry reduction, called *Jordan reduction*. We will extend their approach to the doubly-nonnegative cone. The advantage of the Jordan reduction approach is that it requires no knowledge of group symmetries in the data, and therefore is ideal for automated pre-processing. A drawback is that the initial problem size must be small enough to perform basic linear algebra operations. This is not always the case, e.g. when computing the ϑ' function of Hamming graphs [Sch79].

Outline and Contributions of this Chapter

In the next section, we recap relevant definitions and results on the Jordan reduction of Parrilo and Permenter [PP19]. In Section 2.2 we subsequently extend this approach — which was formulated for symmetric cones — to the doubly nonnegative cone. This allows us to apply the method to the SDP relaxation of the general QAP due to Zhao et al. [Zha+98] in Section 3.3, and to the ϑ'-function of Erdős-Rényi -graphs in Section 2.3.1. Our extension of the Jordan reduction method of Parrilo and Permenter [PP19] in Section 2.2 should lead to additional applications in SDP relaxations of other combinatorial problems. Finally, in Section 2.4, we describe a Julia software package implementing this method.

This package complements the two existing packages QDimSum [TRR19] and YALMIP [Lof09], of which the first exploits the symmetry of semidefinite programming problems coming from quantum mechanics, and the second allows for symmetry reduction of polynomial optimization problems with sign symmetries.

2.1 Preliminaries on Jordan Symmetry-Reduction

We will study conic optimization problems in the form

$$\left.\begin{array}{lll} \inf & \langle C, X\rangle & = \inf \quad \langle C, X\rangle \\ \text{s.t.} & \langle A_i, X\rangle = b_i \text{ for } i \in [m] & \text{s.t.} \quad X \in X_0 + \mathcal{L} \\ & X \in \mathcal{K} & \quad\;\; X \in \mathcal{K}, \end{array}\right\} \tag{2.3}$$

where $[m] = \{1, \ldots, m\}$, $\mathcal{K} \subseteq \mathcal{V}$ is a closed, convex cone in a real Hilbert space $\mathcal{V}$, $X_0 \in \mathcal{V}$ satisfies $\langle A_i, X_0\rangle = b_i$ for all $i \in [m]$, and $\mathcal{L} \subseteq \mathcal{V}$ is the nullspace of the linear operator A, where $A(X) = (\langle A_i, X\rangle)_{i=1}^m$. The objective function is given using the inner product $\langle \cdot, \cdot \rangle$ of $\mathcal{V}$, with which one defines the *dual cone* as:

$$\mathcal{K}^* := \{Y \in \mathcal{V} \mid \langle X, Y\rangle \geq 0 \;\; \forall X \in \mathcal{K}\}.$$

In this chapter, we will mostly deal with the case where $\mathcal{V}$ is the space $\mathbb{S}^n$ of $n \times n$ symmetric matrices equipped with the Euclidean inner product, and where $\mathcal{K}$ is the cone of doubly nonnegative matrices.

2.1.1 Constraint Set Invariance

Parrilo and Permenter [PP19] introduced a set of three conditions a subspace has to fulfill, such that it is possible to use it for symmetry reduction. Here we revisit some of their results.

Definition 2.1. A *projection* is a linear transformation $P\colon \mathcal{V} \to \mathcal{V}$ which is *idempotent*, i.e., $P^2 = P$.

Definition 2.2 (Definition 2.1. in [PP19])**.** A projection $P\colon \mathcal{V} \to \mathcal{V}$ fulfills the *Constraint Set Invariance Conditions (CSICs)* for $(\mathcal{K}, X_0 + \mathcal{L}, C)$ if

(i) The projection is positive: $P(\mathcal{K}) \subseteq \mathcal{K}$,

(ii) $P(X_0 + \mathcal{L}) \subseteq X_0 + \mathcal{L}$,

(iii) $P^*(C + \mathcal{L}^\perp) \subseteq C + \mathcal{L}^\perp$,

where P^* is the adjoint of P, which satisfies $\langle P(X), Y\rangle = \langle X, P^*(Y)\rangle$ for all $X, Y \in \mathcal{V}$.

Note that this definition is symmetric going from primal to dual, since

$$P(\mathcal{K}) \subseteq \mathcal{K} \quad \Leftrightarrow \quad P^*(\mathcal{K}^*) \subseteq \mathcal{K}^*.$$

While it is clear that such a projection has to send feasible solutions to feasible solutions in both the primal and dual program, it seems less obvious that it does not change the objective value of feasible elements (Proposition 1.4.1 in [Per17]). We give an alternative, more compact proof of this fact. For an $X \in X_0 + \mathcal{L}$ we have $X - P(X) \in \mathcal{L}$ by *(ii)*, and *(iii)* tells us $C - P^*(C) \in \mathcal{L}^\perp$. Thus,

$$\begin{aligned}\langle C, X\rangle - \langle C, P(X)\rangle &= \langle C, X - P(X)\rangle - \langle C, P(X) - P(P(X))\rangle \\ &= \langle C, X - P(X)\rangle - \langle P^*(C), X - P(X)\rangle \\ &= \langle C - P^*(C), X - P(X)\rangle = 0.\end{aligned}$$

These projections send feasible solutions to feasible solutions with the same objective value, as the next result shows. (We include a proof for completeness.)

Proposition 2.3 (Proposition 1.4.1 in [Per17])**.** *If a projection $P\colon \mathcal{V} \to \mathcal{V}$ fulfills the CSICs, then*

- $P((X_0 + \mathcal{L}) \cap \mathcal{K}) \subseteq (X_0 + \mathcal{L}) \cap \mathcal{K}$,
- $P^*((C + \mathcal{L}^\perp) \cap \mathcal{K}^*) \subseteq (C + \mathcal{L}^\perp) \cap \mathcal{K}^*$,
- *For $X \in X_0 + \mathcal{L}$:* $\langle C, X\rangle = \langle C, P(X)\rangle$,
- *For $Y \in C + \mathcal{L}^\perp$:* $\langle X_0, Y\rangle = \langle X_0, P^*(Y)\rangle$.

Proof from [Per17]. The first two properties are direct consequences of the definition, and the remaining two are equivalent by its symmetry. Note that because of *(ii)* the difference $X - P(X)$ lies in $\mathcal{L}$ for any $X \in X_0 + \mathcal{L}$, and similarly $C - P^*(C) \in \mathcal{L}^\perp$, which means that these vectors are orthogonal.

$$\begin{aligned}\langle C, X\rangle &= \langle C - P^*(C) + P^*(C), X - P(X) + P(X)\rangle \\ &= \underbrace{\langle C - P^*(C), X - P(X)\rangle}_{=0} + \langle C - P^*(C), P(X)\rangle \\ &\quad + \langle P^*(C), X\rangle + \underbrace{\langle P^*(C), P(X) - P(X)\rangle}_{=0}\end{aligned}$$

$$
\begin{aligned}
&= \langle P^*(C) - P^*(C), X\rangle + \langle C, P(X)\rangle \\
&= \langle C, P(X)\rangle
\end{aligned}
$$

Here we used $PP = P$ and $\langle P(A), B\rangle = \langle A, P^*(B)\rangle$ in the penultimate step. □

To make things easier, we restrict ourselves to *orthogonal* projections P_S to a subspace $S \subseteq \mathcal{V}$, which are exactly the projections of which the range and kernel are orthogonal, or equivalently the projections which are self-adjoint, i.e., $P_S = P_S^*$. If the projection P_S fulfills the CSICs we call the subspace S *admissible*, following [PP19]. In this case, the CSICs may be rewritten as follows, exploiting the fact that orthogonal projections have the property that $P_A(B) \subseteq B \iff P_B(A) \subseteq A$.

Theorem 2.4 (Theorem 5.2.1 in [Per17])**.** *Consider the conic optimization problem* (2.3) *and let $S \subseteq \mathcal{V}$ be the range of an orthogonal projection $P_S : \mathcal{V} \to \mathcal{V}$. Let $P_{\mathcal{L}}$ denote the orthogonal projection onto $\mathcal{L}$, and define $C_{\mathcal{L}} = P_{\mathcal{L}}(C)$ and $X_{0,\mathcal{L}^\perp} = P_{\mathcal{L}^\perp}(X_0)$. Then S is an admissible subspace if, and only if,*

(a) $C_{\mathcal{L}}, X_{0,\mathcal{L}^\perp} \in S$,

(b) $P_{\mathcal{L}}(S) \subseteq S$,

(c) $P_S(\mathcal{K}) \subseteq \mathcal{K}$.

Restricting the conic program to an admissible subspace S thus results in another, potentially significantly smaller program, with the same optimal value.

$$
\begin{aligned}
&\inf \ \langle P_S(C), X\rangle \\
&\text{s.t. } X \in P_S(X_0) + \mathcal{L} \cap S, \\
&\qquad X \in \mathcal{K} \cap S.
\end{aligned}
\tag{2.4}
$$

Let $\{S_1, \ldots, S_k\}$ be a basis of the admissible subspace S. We can rewrite the reduced problem (2.4) as a conic optimization problem with just k scalar variables $x_1, \ldots, x_k$, by considering the original formulation (2.3).

$$
\begin{aligned}
&\inf_{x \in \mathbb{R}^k} \ \sum_{i=1}^{k} x_i \langle P_S(C), S_i\rangle \\
&\text{s.t. } \sum_{i=1}^{k} x_i \langle A_j, S_i\rangle = b_j \quad \text{for } j = 1, \ldots, m, \\
&\qquad \sum_{i=1}^{k} x_i S_i \in \mathcal{K},
\end{aligned}
\tag{2.5}
$$

We can *aggregate* the linear constraints (2.5), reducing their number. Since $\mathcal{L} \cap S$ is generally lower dimensional than S, many of the original constraints become linearly dependent (in practice often identical) in the reduced optimization problem.

2.1.2 The Reduction for Jordan-Algebras

Next we review some results from [Per17] for the situation where the space $\mathcal{V}$ is a *Euclidean Jordan algebra* $\mathcal{J}$, that is a commutative algebra (with product denoted by '$\bullet$') over $\mathbb{R}$ satisfying the *Jordan identity*

$$(x \bullet y) \bullet x^2 = x \bullet (y \bullet x^2),$$

and an inner product with $\langle x \bullet y, z\rangle = \langle y, x \bullet z\rangle$. For every such algebra we can define $\mathcal{K}$ as the cone of squares of $\mathcal{J}$ given by $\mathcal{K} = \{x \bullet x \mid x \in \mathcal{J}\}$, which always is a symmetric cone, i.e., a self-dual and homogenous convex cone (see for example [FA94]).

The only example relevant for us is the case $\mathcal{J} = \mathbb{S}^n$, the symmetric $n \times n$-matrices with real entries, with product defined by

$$X \bullet Y \coloneqq \frac{1}{2}(XY + YX),$$

and the inner product the Euclidean (trace) inner product $\langle X, Y\rangle = \operatorname{trace}(XY)$. It is easy to see (e.g., from the spectral decomposition) that its cone of squares is exactly the positive semidefinite cone $\mathbb{S}^n_+$.

Since the product of a Jordan algebra is commutative, we have

$$2x \bullet y = x \bullet y + y \bullet x = (x+y)^2 - x^2 - y^2,$$

which means that subspaces are closed under multiplication, if and only if they include all squares. Similarly, isomorphisms between (Euclidean) Jordan algebras are exactly the bijective linear maps satisfying $\phi(x^2) = (\phi(x))^2$.

Definition 2.5. A Jordan algebra $\mathcal{J}$ is called *special*, if it is isomorphic to the algebra one gets from a real associative algebra by equipping the latter with the product $x \bullet y = \frac{1}{2}(xy + yx)$.

There is only a single (up to isomorphisms) simple Jordan algebra which is not special, the algebra of Hermitian 3×3-matrices of Octonions $H_3(\mathbb{O})$. The for us relevant case $\mathcal{J} = \mathbb{S}^n$ is special.

Definition 2.6. A subspace (not necessarily a subalgebra) S of a Jordan algebra is called *unital*, if there is an element $e \in S$ such that $e \bullet a = a \bullet e = a$ for all $a \in S$.

An important fact for us is that every Euclidean Jordan algebra is unital.

One main result of [Per17, Theorem 5.2.3] is an alternative description of the CSICs when the ambient space is a special Euclidean Jordan algebra. In this case the condition $P_S(\mathcal{K}) \subseteq \mathcal{K}$ in Theorem 2.4 — with $\mathcal{K}$ the cone of squares in $\mathcal{J}$ — is equivalent to S being closed under taking squares, i.e. to S being a Jordan subalgebra of $\mathcal{J}$, if S is unital.

This gives an algorithm for finding the minimal admissible subspace, which is defined as follows.

Definition 2.7. The unique minimal admissible subspace is

$$S_{\min} := \bigcap_{S \text{ is admissible}} S.$$

As mentioned before, we may now formulate an algorithm for $S_{\min}$.

Theorem 2.8 (Theorem 3.2 in [PP19])**.** *If $\mathcal{V} = \mathcal{J}$ is a Euclidean, special Jordan algebra, and $\mathcal{K}$ its cone of squares, then $S_{\min}$ is the output of Algorithm 1.*

Algorithm 1: Finding $S_{\min}$

$S \leftarrow \operatorname{span}\{C_{\mathcal{L}}, X_{0,\mathcal{L}^\perp}\}$
repeat
  $S \leftarrow S + P_{\mathcal{L}}(S)$
  $S \leftarrow S + \operatorname{span}\{X^2 \mid X \in S\}$
until *converged;*

2.1.3 A Combinatorial Reduction Algorithm

The fourth step of Algorithm 1 is not linear, and hard to implement. But, conveniently, Permenter does introduce three combinatorial algorithms in his PhD thesis ([Per17], Chapter 7) for the cone $\mathbb{S}^n_+$, which all find orthogonal 0/1-bases for an optimal unital admissible subspace with certain additional properties. A 0/1-basis is a basis where each element has entries solely in $\{0, 1\}$. If the basis is orthogonal, this implies that no two basis elements have nonzero entries in the same position. Here we will only mention one of the algorithms, since the other ones cannot give us better reductions for our special case.

Partition Subspaces. The second combinatorial algorithm by Permenter [Per17] finds an optimal unital *partition subspace*, which is a subspace with 0/1-basis, the elements of which sum to the all-one matrix. We call these basis elements *characteristic matrices* of the partition. We can describe the basis uniquely with a partition of the coordinates of $\mathbb{R}^{n\times n}$, i.e. of $[n]\times[n]$, simply by having one part in the partition for every basis element with ones in the corresponding coordinates. For example the following spaces are partition spaces:

$$P_1 = \begin{pmatrix} a & a & b \\ a & a & b \\ b & b & c \end{pmatrix}, \qquad P_2 = \begin{pmatrix} a & b & b \\ b & a & b \\ b & b & c \end{pmatrix}, \qquad P_3 = P_1 \wedge P_2 = \begin{pmatrix} a & b & c \\ b & a & c \\ c & c & d \end{pmatrix},$$

where $P_1 \wedge P_2$ is the coarsest (i.e., the smallest dimensional) partition space *refining* both P_1 and P_2. A partition space A *refines* the partition space B, if B is a subspace of A.

For our purposes an important special case is a so-called Jordan configuration, defined as follows.

Definition 2.9. A partition P of $A\times A$, where A is a finite set, is called a *Jordan configuration*, if its characteristic matrices $\mathcal{B}_P$ satisfy

- $X = X^T$ for all $X \in \mathcal{B}_P$,
- $XY + YX \in \operatorname{span}\ \mathcal{B}_P$ for all $X, Y \in \mathcal{B}_P$,
- $I \in \operatorname{span}\ \mathcal{B}_P$.

In words, a Jordan configuration is a basis of a unital partition space that is also a Jordan subalgebra of $\mathbb{S}^n$.

A more general example of a partition space, also of interest to us, is a so-called coherent algebra (see e.g., [Cam03]).

Definition 2.10. A partition P of $A\times A$, where A is a finite set, is called a *coherent configuration*, if its characteristic matrices $\mathcal{B}_P$ satisfy

- If $X \in \mathcal{B}_P$ then also $X^T \in \mathcal{B}_P$,
- $XY \in \operatorname{span}\ \mathcal{B}_P$ for all $X, Y \in \mathcal{B}_P$,
- $I \in \operatorname{span}\ \mathcal{B}_P$.

Thus, a coherent configuration gives a 0/1 basis of a partition subspace that is also a matrix $*$-algebra, namely the associated coherent algebra. Note that the symmetric part of a coherent configuration is a Jordan configuration. It was recently shown in [MRK20] that the converse is not true, there are infinite Jordan configurations that are not the symmetric part of a coherent configuration.

To restrict the algorithm 1 to partition subspaces, we need more notation: part(A) is the smallest partition space containing the matrix (or subspace) A, which is simply the partition space given by the unique entries of A.

Algorithm 2: Partition algorithm ([Per17])

$P \leftarrow \text{part}(C_{\mathcal{L}}) \wedge \text{part}(X_{0,\mathcal{L}^\perp})$
repeat
  $P \leftarrow P \wedge \text{part}(P_{\mathcal{L}}(P))$
  $P \leftarrow P \wedge \text{part}(\text{span}\{X^2 \mid X \in P\})$
until *converged*;

There are two basic ways to implement this algorithm: One can use polynomial matrices, or randomization. For the first variant one introduces (commuting) variables t_i for each element of a basis $B_1, \ldots, B_k$ of P, and then refines the partition with $\text{part}(P_{\mathcal{L}}(\sum_{i=1}^k t_i B_i)) = \text{part}(\sum_{i=1}^k t_i P_{\mathcal{L}}(B_i))$ and $\text{part}((\sum_{i=1}^k t_i B_i)^2)$. If we for example take P_2 from the example above, one has

$$\begin{pmatrix} t_a & t_b & t_b \\ t_b & t_a & t_b \\ t_b & t_b & t_c \end{pmatrix}^2 = \begin{pmatrix} t_a^2 + 2t_b^2 & 2t_a t_b + t_b^2 & t_a t_b + t_b^2 + t_b t_c \\ 2t_a t_b + t_b^2 & t_a^2 + 2t_b^2 & t_a t_b + t_b^2 + t_b t_c \\ t_a t_b + t_b^2 + t_b t_c & t_a t_b + t_b^2 + t_b t_c & 2t_b^2 + t_c^2 \end{pmatrix},$$

of which the unique polynomials induce the partition P_3.

The second variant refines the partition with a random element in the partition space after projecting it to $\mathcal{L}$ and after squaring it. While one has to be more careful about rounding errors here, it is both easier to implement and much faster.

Remark 2.11. We note that the first variant of the partition algorithm presented here is very similar to the Weisfeiler-Leman (WL) algorithm [LW68], that finds the coarsest coherent configuration refining a given partition of $[n] \times [n]$. The only difference is that the WL algorithm uses non-commuting variables t_i, as opposed to commuting ones; see [Bab+97] on details of the implementation of the WL algorithm.

Algorithm 3: Partition algorithm, randomized ([Per17])

$P \leftarrow \text{part}(C_{\mathcal{L}}) \wedge \text{part}(X_{0,\mathcal{L}^\perp})$
repeat
  $X \leftarrow$ random element of P
  $P \leftarrow P \wedge \text{part}(P_{\mathcal{L}}(X))$
  $P \leftarrow P \wedge \text{part}(X^2)$
until *converged*;

2.2 Extension to the Doubly Nonnegative Cone

We will now fix the cone $\mathcal{K}$ in (2.3) to be the *doubly nonnegative cone* $\mathcal{D}^n := \mathbb{S}^n_+ \cap \mathbb{R}^{n\times n}_+$. Since we will refer to nonnegative, symmetric matrices frequently, we also introduce the notation $\mathcal{N}^n = \mathbb{S}^n \cap \mathbb{R}^{n\times n}_+$. Even though $\mathcal{D}^n$ is not a cone of squares in a Euclidean Jordan algebra, one may readily adapt some results of the last section to this setting.

We start with an elementary, but important observation.

Proposition 2.12. *Assume that a subspace $S \subset \mathbb{S}^n$ has a basis of nonnegative matrices with pairwise disjoint supports. Then the orthogonal projection P_S onto S satisfies $P_S(\mathcal{D}^n) \subseteq \mathcal{D}^n$ if it satisfies $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$.*

Proof. If S has a basis of nonnegative matrices with disjoint supports, then it has an orthonormal basis with this property, say A_i ($i \in [d]$), and the orthogonal projection is of the form

$$P_S(X) = \sum_{i=1}^{d} \langle A_i, X\rangle A_i.$$

Since the Euclidean inner product of two nonnegative matrices is nonnegative, we have

$$P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n,$$

and, since $\mathcal{D}^n \subset \mathbb{S}^n_+$, and $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ by assumption,

$$P_S(\mathcal{D}^n) \subseteq \mathbb{S}^n_+ \cap \mathcal{N}^n = \mathcal{D}^n.$$

□

If we consider partition subspaces, we may therefore use results on admissible partition subspaces for the case $\mathcal{K} = \mathbb{S}^n_+$, as follows.

Corollary 2.13. *Consider a conic optimization problem of the form* (2.3)*, with* $\mathcal{V} = \mathbb{S}^n$*, and* $\mathcal{K} = \mathbb{S}^n_+$*, and let* S *be an admissible partition subspace for this problem. Then,* S *is also an admissible partition subspace for the related problem where we replace* $\mathcal{K} = \mathbb{S}^n_+$ *by* $\mathcal{K} = \mathcal{D}^n$*.*

The important practical implication is that we may use Algorithm 3 to find an admissible Jordan configuration for conic optimization problems on the cone $\mathcal{D}^n$ (but we do not know if it is optimal in general). In the next section we will do precisely this for an SDP relaxation of the quadratic assignment problem.

It is instructive though to ask how restrictive it is to only consider admissible partition subspaces. In what follows, we show that the partition subspace structure is actually imposed by some relatively weak assumptions.

To this end, we first recall a result on nonnegative projection matrices; recall that a matrix P is a nonnegative projection matrix if $P^2 = P$ and P maps nonnegative vectors to nonnegative vectors. If, in addition, $P = P^T$, then it is called a nonnegative, *orthogonal* projection matrix. The following characterization of nonnegative projection matrices is taken from [Gal04, Theorem 2.38], but originally due to Belitskii and Lyubich (cf. [BL88, p. 108]). Since there is a typo in the proof of the proposition in [BL88], we include a detailed fixed proof in Appendix A.

Proposition 2.14 (Theorem 2.1.11 in [BL88])**.** *The general form of a nonnegative projection matrix is*

$$P = (A+B)C^T \tag{2.6}$$

where $r = rank(P)$*,* $A, B, C \in \mathbb{R}^{n\times r}_+$*,* $A^TA = I$*,* $C^TA = I$*,* $B^TA = 0$ *and* $B^TC = 0$*.*

As a consequence, a nonnegative, orthogonal projection matrix has the following structure.

Corollary 2.15. *Any* $n \times n$ *symmetric nonnegative orthogonal projection matrix* P *with* r*-dimensional range takes the form* $P = CC^T$ *for some* $C \in \mathbb{R}^{n\times r}_+$ *such that* $C^TC = I$*. In particular, the columns of* C *form a nonnegative, orthonormal basis of the range of* P*, and these basis vectors therefore have disjoint supports.*

Proof. With reference to (2.6), one has

$$P = P^T \Longrightarrow PA = P^TA \Longleftrightarrow (A+B)C^TA = C(A^T + B^T)A \Longleftrightarrow A + B = C.$$

Thus, by (2.6) one has $P = CC^T$, and $C^TC = I$. Since nonnegative vectors can only be orthogonal if they have disjoint supports, the columns of C have this property.

Finally, recall that a projection matrix is symmetric if and only if it corresponds to an orthogonal projection. □

One may easily extend this to orthogonal projection operators, as follows.

Proposition 2.16. *Assume that a given orthogonal projection P_S with range $S \subseteq \mathbb{S}^n$ satisfies $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$. Then:*

1. *S has a basis of nonnegative matrices with disjoint supports.*
2. *If, in addition, S contains the all-ones matrix J, then it is a partition subspace.*
3. *If, in addition to the condition in item 2), $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ and S contains the identity matrix, then S is a Jordan configuration.*

Proof. Since P_S is self-adjoint, we may write it as a symmetric matrix, say M_{P_S}, with respect to the standard orthonormal basis of $\mathbb{S}^n$. For an $X \in \mathbb{S}^n$, we define the vector $\mathbf{svec}(X) \in \mathbb{R}^{\frac{1}{2}n(n+1)}$ as

$$\mathbf{svec}(X) = \left(X_{11}, \sqrt{2}X_{21}, \ldots, \sqrt{2}X_{n1}, X_{22}, \sqrt{2}X_{32}, \ldots, \sqrt{2}X_{n2}, \ldots, X_{nn}\right)^T.$$

Thus, $\mathbf{svec}(X)$ gives the coordinates of X in the standard orthonormal basis of $\mathbb{S}^n$. One therefore has

$$\mathbf{svec}(P_S(X)) = M_{P_S} \cdot \mathbf{svec}(X) \qquad \forall X \in \mathbb{S}^n.$$

Choosing $\mathbf{svec}(X)$ as the standard unit vectors in $\mathbb{R}^{\frac{1}{2}n(n+1)}$ makes it clear that $M_{P_S} \in \mathcal{N}^{\frac{1}{2}n(n+1)}$. Thus, the first claim now follows from Corollary 2.15, namely that S has a basis of nonnegative matrices with pairwise disjoint supports. If S contains the all-ones matrix J, then it must hold that these basis matrices are 0/1, proving the second claim.

Finally, to prove the third claim, we recall that S unital and $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ implies that S is a Euclidean Jordan algebra, as mentioned in Section 2.1.2. Since it has a 0/1 basis, it is in fact a Jordan configuration if we also assume $I \in S$.

□

The last proposition shows that partition subspaces are closely related to nonnegative projections.

The question remains if there exists an orthogonal projection $P_S : \mathbb{S}^n \to \mathbb{S}^n$ with range $S \subseteq \mathbb{S}^n$ that satisfies $P_S(\mathcal{D}^n) \subseteq \mathcal{D}^n$, but <u>not</u> $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$.

Proposition 2.17. *Let P_S be an orthogonal projection with range S that satisfies $P_S(\mathcal{D}^n) \subseteq \mathcal{D}^n$. If $I \in S$, then $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$.*

Proof. Let $X \in \mathcal{N}^n$. Since $X \in (\mathcal{D}^n)^* = \mathcal{N}^n + \mathbb{S}^n_+$ and $P_S((\mathcal{D}^n)^*) \subseteq (\mathcal{D}^n)^*$, we know that the diagonal entries of $P_S(X)$ are nonnegative. To see the same for the off-diagonal entries, let r be the spectral radius of X. Then $X + rI \in \mathcal{D}^n$, which implies that $P_S(X+rI) \in \mathcal{D}^n$ has nonnegative off-diagonal entries. Since $P_S(I) = I$ we have $P_S(X) = P_S(X + rI) - rI$, thus showing that $P_S(X)$ has nonnegative off-diagonal entries and therefore $P_S(X) \in \mathcal{N}^n$. □

Hence, all admissible subspaces that contain J and I are automatically Jordan configurations for conic problems over the doubly nonnegative cone, if they are $\mathbb{S}^n_+$-positive, by the last two propositions. It is still an open problem whether $\mathcal{D}^n$-positive implies $\mathbb{S}^n_+$-positive.

Remark 2.18. The additional restriction to 0/1 bases is not w.l.o.g., for example consider the subspace spanned by a single idempotent matrix $A \geq 0$ with multiple different entries. This subspace is closed under squaring $(\lambda A)^2 = \lambda^2 A$, and it has a unit $e = A$. Such an A exists, e.g.

$$A = \frac{1}{2}\begin{pmatrix} 1 - \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 1 + \frac{1}{\sqrt{2}} \end{pmatrix}.$$

So $S = \operatorname{span}\{A\}$ is a unital Jordan algebra, with basis $\{A\}$. Since $A \geq 0$, an $\mathbb{S}^n_+$-positive projection to S is $\mathcal{D}^n$-positive, but S does not have a 0/1-basis. Since coherent algebras are Jordan algebras, this shows that a restriction to the subspace having a 0/1-basis or to the subspace being a coherent algebra is a truly stronger condition than is needed by Theorem 2.4.

2.3 Reducing the ϑ'-Function

The ϑ'-function of a graph $G = (V, E)$, as given in (2.1), is a doubly nonnegative semidefinite program of size $n := |V|$. Here we can say a bit less about admissible subspaces in the general case. As seen in Theorem 2.4, every admissible subspace needs to contain $C_{\mathcal{L}}$ and $X_{0,\mathcal{L}^\perp}$. Here it is straightforward to see that $C_{\mathcal{L}}$ is exactly the adjacency matrix of the complementary graph $\overline{G}$, and $X_{0,\mathcal{L}^\perp} = \frac{1}{n} I_n$. Thus, we know at least that S contains the identity. This implies that every admissible subspace for the ϑ'-function has a basis of nonnegative matrices with disjoint supports, by Propositions 2.16 and 2.17.

But an admissible subspace here does not necessarily need to contain the all-one matrix J_n. One obtains an easy example by $G = ([n], \{\{i,j\} \text{ if } i > m \text{ or } j > m\})$ for $n > m > 0$. It is easy to check that an admissible subspace for this problem is of the form

$$\begin{pmatrix} a & b & b & & & & \\ b & a & b & & & & \\ b & b & a & & & & \\ & & & c & & & \\ & & & & c & & \\ & & & & & c & \\ & & & & & & c \end{pmatrix},$$

here shown for $n = 7$ and $m = 3$. While this case is not too interesting, this shows that the Jordan reduction can be better than a group-symmetry reduction, which would result in the five-dimensional subspace given by

$$\begin{pmatrix} a & b & b & d & d & d & d \\ b & a & b & d & d & d & d \\ b & b & a & d & d & d & d \\ d & d & d & c & e & e & e \\ d & d & d & e & c & e & e \\ d & d & d & e & e & c & e \\ d & d & d & e & e & e & c \end{pmatrix}.$$

Do note though that in this case the three additional variables will be eliminated by the constraints of the SDP soon after.

2.3.1 The ϑ'-Function of Erdős-Rényi Graphs

Let q be an odd prime, and let $V = \mathbb{F}_q^3$ be a three-dimensional vector space over the finite field of order q. The set of one dimensional subspaces, i.e. the projective plane, of V is denoted by $\mathrm{PG}(2,q)$. There are $q^2 + q + 1$ such subspaces, which form the vertices of the Erdős-Rényi graph $\mathrm{ER}(q)$. Two vertices are adjacent if they are distinct and orthogonal, i.e., for two representing vectors x and y we have $x^T y = 0$. The interested reader is referred to the papers [GN08; Kle+09a], and the references therein, for more details on these graphs.

We are interested in the size of a maximum stable set of these graphs, specifically upper bounds for this value.

In [GN08] the authors derive the upper bound

$$\frac{\sqrt{q}+\sqrt{q+4(q+1)\frac{q+\sqrt{q}+1}{q^2+q+1}}}{2\frac{q+\sqrt{q}+1}{q^2+q+1}}, \tag{2.7}$$

which was shown to be at most as good as the ϑ'-function in [Kle+09a]. The bound (2.7) is obtained from a generalization of the Hoffman-Delsarte bound [Del73] for nearly regular graphs, where one adds loops to a graph to make it regular.

The ϑ'-function of ER(q) is a doubly nonnegative semidefinite program of size q^2+q+1. Without further reductions one can practically solve this for up to $q=17$. In [Kle+09a] the authors reduced the problem size enough to solve it for up to $q=31$, and in this chapter we managed to solve it for up to $q=97$.

We applied the reduction algorithm, numerically block-diagonalized (more on that next section) and solved the resulting problems for all primes from $q=3$ to 97, as shown in Tables 2.1 and 2.2. Interestingly, the reduced block sizes always are one block of size 3×3, and $\lceil\frac{q}{2}\rceil$ blocks of size 2×2, i.e., the problem nearly reduces to a second order cone problem. By comparison, the problem was reduced to SDPs of matrix size $2q+11$ in [Kle+09a].

2.4 The Julia Package

We provide a package "SDPSymmetryReduction.jl" as part of the Julia registry, with full source code available at `https://github.com/DanielBrosch/SDPSymmetryReduction.jl`. We provide functions to both find an optimal admissible partition subspace for a given SDP, and to block-diagonalize it after.

To enter a semidefinite program one has to provide (potentially sparse) vectors and matrices $C\in\mathbb{R}^{n^2}, A\in\mathbb{R}^{m\times n^2}$ and $b\in\mathbb{R}^m$ as in (2.3) (vectorizing the variable X). We provide examples for how one can approach this for both the ϑ'-function of a given graph and for the SDP-bound (3.1). Since we return a partition based symmetry reduction, it is not necessary to give entry-wise nonnegativity constraints, as long as one remembers to use nonnegative variables in the final, reduced SDP (see also the example in 2.6).

Determining an Admissible Subspace. The function ‘admPartSubspace‘ determines an optimal admissible partition subspace for the problem, by Algorithm 3.

q	q^2+q+1	Jordan red. (s)	block-diag. (s)	blocks (size × mult.)
3	13	0.001	0.001	$3\times 1, 2\times 2$
5	31	0.002	0.006	$3\times 1, 2\times 3$
7	57	0.007	0.011	$3\times 1, 2\times 4$
11	133	0.067	0.033	$3\times 1, 2\times 6$
13	183	0.092	0.051	$3\times 1, 2\times 7$
17	307	0.241	0.170	$3\times 1, 2\times 9$
19	381	0.421	0.303	$3\times 1, 2\times 10$
23	553	1.019	0.723	$3\times 1, 2\times 12$
29	871	2.455	2.392	$3\times 1, 2\times 15$
31	993	3.398	3.664	$3\times 1, 2\times 16$
37	1407	7.745	9.068	$3\times 1, 2\times 19$
41	1723	13.039	16.358	$3\times 1, 2\times 21$
43	1893	14.533	21.508	$3\times 1, 2\times 22$
47	2257	19.910	36.711	$3\times 1, 2\times 24$
53	2863	33.271	72.052	$3\times 1, 2\times 27$
59	3541	51.463	140.119	$3\times 1, 2\times 30$
61	3783	54.714	166.267	$3\times 1, 2\times 31$
67	4557	78.579	332.438	$3\times 1, 2\times 34$
71	5113	115.303	487.162	$3\times 1, 2\times 36$
73	5403	118.058	545.498	$3\times 1, 2\times 37$
79	6321	179.084	886.065	$3\times 1, 2\times 40$
83	6973	215.983	1336.901	$3\times 1, 2\times 42$
89	8011	293.947	1931.723	$3\times 1, 2\times 45$
97	9507	434.341	2912.840	$3\times 1, 2\times 49$

Table 2.1: Results of the numerical symmetry reduction of the Theta' function of Erdős-Rényi graphs.

This is done using a randomized Jordan-reduction algorithm, and it returns a Jordan algebra. SDPs can be restricted to such a subspace without changing their optimal value.

Given C, A and b, admPartSubspace(C, a, b) returns a Partition P with $P.n$ giving the number of parts of the partition, and $P.P$ returning an integer valued matrix with entries $1, \ldots, n$ defining the partition.

For example, let C, A and b define the ϑ'-function of the cycle graph C_5. If we label the vertices such that its adjacency matrix is

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix},$$

q	solve time (s)	$\vartheta'(\mathrm{ER}(q))$	EV bound (2.7)
3	0.002	5.000	5.560
5	0.003	10.067	10.556
7	0.002	15.743	16.727
11	0.003	31.088	32.051
13	0.003	40.509	41.025
17	0.004	60.221	61.291
19	0.004	71.301	72.493
23	0.004	96.240	96.858
29	0.006	136.978	137.910
31	0.007	151.702	152.707
37	0.007	199.269	200.203
41	0.009	233.390	234.312
43	0.009	250.917	252.063
47	0.011	287.772	288.907
53	0.013	346.626	347.388
59	0.015	408.548	409.534
61	0.015	430.219	431.030
67	0.020	496.438	497.775
71	0.019	543.128	544.095
73	0.021	566.915	567.787
79	0.945	639.644	640.932
83	1.111	690.583	691.375
89	0.115	768.469	769.481
97	0.108	877.075	878.027

Table 2.2: The resulting bounds for the stable set number of Erdős-Rényi graphs.

then calling admPartSubspace(C, a, b) returns the partition P with $P.n = 3$ and

$$P.P = \begin{pmatrix} 1 & 2 & 3 & 3 & 2 \\ 2 & 1 & 2 & 3 & 3 \\ 3 & 2 & 1 & 2 & 3 \\ 3 & 3 & 2 & 1 & 2 \\ 2 & 3 & 3 & 2 & 1 \end{pmatrix},$$

i.e., we can restrict the feasible set to the three-dimensional subspace given by P.

Block-Diagonalizing a Jordan-Algebra. The function 'blockDiagonalize' determines a block-diagonalization of a (Jordan)-algebra given by a partition P using a randomized algorithm. It implements the Algorithm from [Mur+10] (see also [KDP11]). To our knowledge this is the first implementation available to the public.

Remark 2.19. Every matrix $*$-algebra over $\mathbb{C}$ is isomorphic to a direct sum of full matrix $*$-algebras over $\mathbb{C}$. But the situation over the reals has more cases (as

detailed in [MM10]); this is the underlying property of block-diagonalization over the reals. We currently offer two versions of the block-diagonalization function: One to attempt to block-diagonalize it fully over the reals (which does fail in the cases mentioned in [MM10]), and one over the complex numbers, which never fails (at the cost of potentially being non-optimal when working with real valued SDPs).

Remark 2.20. The algorithm only provides an approximate, rational block-diagonalization. This may lead to numerical issues in some cases, potentially requiring very high precision computations. In some cases it may be more stable numerically to only compute the Jordan-algebra, reducing the number of variables, and skipping the block-diagonalization. However, we did not run into such a case in practice in the examples considered here and in Chapter 3.

blockDiagonalize(P) returns (a rational approximation of) a real block-diagonalization $blkd$, if it exists, otherwise 'nothing'.

- $blkd.blkSizes$ returns an integer array of the sizes of the blocks.
- $blkd.blks$ returns an array of length $P.n$ containing arrays of (real) matrices of sizes $blkd.blkSizes$. I.e., $blkd.blks[i]$ is the image of the basis element given by the 0/1-matrix with a one in the positions where $P.P$ is i.

blockDiagonalize($P; complex = true$) returns the same, but with complex valued matrices, and should be used if no real block-diagonalization was found. To use the complex matrices in practice, remember that a Hermitian matrix Y is positive semidefinite if and only if

$$\begin{pmatrix} \text{real}(Y) & -\text{imag}(Y) \\ \text{imag}(Y) & \text{real}(Y) \end{pmatrix}$$

is positive semidefinite.

Continuing the example of reducing $\vartheta'(C_5)$, blockDiagonalize(P) here returns a block-diagonalization $blkd$ with $blkd.blkSizes = [1,1,1]$ (i.e., three blocks of size 1×1), and the image of the basis is

$$\begin{aligned} blkd.blks \approx [&[1,1,1], \\ &[-1.618, 0.618, 2], \\ &[0.618, -1.618, 2]]. \end{aligned}$$

This means that

$$\begin{pmatrix} a & b & c & c & b \\ b & a & b & c & c \\ c & b & a & b & c \\ c & c & b & a & b \\ b & c & c & b & a \end{pmatrix} \succcurlyeq 0 \Leftrightarrow \begin{cases} a - 1.618b + 0.618c \geq 0, \\ a + 0.618b - 1.618 \geq 0, \\ a + 2b + 2c \geq 0, \end{cases},$$

which allows us to rewrite $\vartheta'(C_5)$ as a linear program in the three variables a, b, c.

2.5 Concluding Remarks

We have extended the Jordan symmetry reduction method to the doubly nonnegative cone, and showed that for this cone that the restriction to admissible subspaces that are partition-spaces is not a strong requirement in Section 2.2. In Section 2.3 we have seen that the optimal admissible subspace of the ϑ'-function can always be given by a nonnegative basis with disjoint supports, and applied this reduction to Erdős-Rényi -graph instances. Finally, we describe the Julia package "SDPSymmetryReduction.jl", available at `https://github.com/DanielBrosch/SDPSymmetryReduction.jl`, implementing the described algorithms in Section 2.4. In Appendix 2.6 we give a complete code example as to how one can use this package to calculate $\vartheta'(\mathrm{ER}(q))$.

It may be possible to combine the Jordan symmetry reduction method with facial reduction, similar to what the authors do in [HSW19]. There a significant speed-up in computations was observed when solving the symmetry and facial reduced doubly nonnegative SDPs with a first order method (more specifically, an alternating direction method of multipliers). In future work similar ideas for the Jordan reduction method may be worth exploring.

2.6 Example Use of the Software Package

In this appendix we give a complete example how to compute the ϑ'-function of $\mathrm{ER}(q)$, where $q = 31$, using block-diagonalization, and solving the reduced SDP with JuMP and Mosek.

```
using SDPSymmetryReduction
using LinearAlgebra, SparseArrays
using JuMP, MosekTools
```

```
## Calculating the Theta'-function of Erdos-Renyi graphs
q = 31

# Generating the adjacency matrix of ER(q)
PG2q = vcat([[0, 0, 1]],
            [[0, 1, b] for b = 0:q-1],
            [[1, a, b] for a = 0:q-1 for b = 0:q-1])
Adj = [x' * y % q == 0 for x in PG2q, y in PG2q]
Adj[diagind(Adj)] .= 0

# Theta' SDP
N = length(PG2q) # = q^2+q+1
C = ones(N^2)
A = hcat(vec(Adj), vec(Matrix(I, N, N)))'
b = [0, 1]

# Find the optimal admissible subspace (= Jordan algebra)
P = admPartSubspace(C, A, b, true)

# Block-diagonalize the algebra
blkD = blockDiagonalize(P, true)

# Calculate the coefficients of the new SDP
PMat = hcat([sparse(vec(P.P .== i)) for i = 1:P.n]...)
newA = A * PMat
newB = b
newC = C' * PMat

# Solve with optimizer of choice
m = Model(Mosek.Optimizer)

# Initialize variables corresponding to the parts of
# the partition P. They are nonnegative since the
# original SDP-matrices are entry-wise nonnegative.
x = @variable(m, x[1:P.n] >= 0)

@constraint(m, newA * x .== newB)
@objective(m, Max, newC * x)

# Setup the block-diagonalized PSD-constraints
psdBlocks = sum(blkD.blks[i] .* x[i] for i = 1:P.n)
for blk in psdBlocks
    if size(blk, 1) > 1
        @constraint(m, blk in PSDCone())
    else
```

```
        @constraint(m, blk .>= 0)
    end
end

optimize!(m)

@show termination_status(m)
@show value(newC * x)
```

3

Comparison and Reduction of Relaxations of the Quadratic Assignment Problem

A quadratic assignment problem in Koopmans-Beckmann form is given by three matrices $A = (a_{ij}), B = (b_{ij}), C = (c_{ij}) \in \mathbb{R}^{n\times n}$, and can be written as

$$QAP(A,B,C) = \min_{\varphi\in S_n}\left(\sum_{i,j=1}^{n} a_{ij}b_{\varphi(i)\varphi(j)} + \sum_{i=1}^{n} c_{i\varphi(i)}\right),$$

where S_n is the set of all permutations of n elements. If $C = 0$, then we shorten the notation to $QAP(A,B)$, and if all data matrices are symmetric, we call the quadratic assignment problem symmetric.

This is a quadratic optimization problem, which can be seen if we write the objective using permutation matrices:

$$\min_{X\in\Pi_n} \langle A, XBX^T\rangle + \langle C, X\rangle,$$

where $\langle X, Y\rangle = \operatorname{tr}(X^T Y)$ is the trace inner product, and Π_n the set of $n \times n$ permutation matrices.

Because of the very general form of the problem, it is not surprising that it is NP-complete (see for example §7.1.7 in [BDM12]), which motivates the search

for good approximations and bounds; see, e.g., the survey [Loi+07] and the book [BDM12] for an overview. In Section 3.1 we describe three such bounds, in both increasing complexity and strength. The first is a projected eigenvalue bound, which was first introduced in [HRW92], which, similar to the eigenvalue bound of [FBR87], is based on the eigenvalues of the data matrices. The second bound, a convex quadratic programming bound, then improves this bound by adding a convex quadratic term to the objective, as introduced in [AB01a] (see also [AB01b; Ans+02]). The third bound, which was introduced in [Zha+98] and later reformulated in [PR09], is a semidefinite programming relaxation of the quadratic assignment problem. As it is the most complex computationally, it is natural to expect it to be stronger than the two other bounds, which we prove in our first main result Theorem 3.3.

In Section 3.3 we then apply the Jordan reduction method, which we investigated in Chapter 2, to the strongest of the three bounds. Finally, we apply this reduction algorithm to the benchmark instances of QAPLib in Section 3.3.1.

3.1 Three Bounds for the QAP

In this section we will consider three different bounds for QAPs, of increasing computational complexity. In Section 3.2 we compare the different bounds, and we reduce the strongest of them in Section 3.3. Later, in Chapter 4 we apply the bounds to a problem coming from discrete energy minimization.

3.1.1 Projected Eigenvalue Bound

The first bound relevant for this chapter is the projected eigenvalue bound, which was introduced in [HRW92], a stronger variant of the eigenvalue bound for QAP (see [FBR87]), which is based on projecting the matrices into a space the same dimension as the span of the permutation matrices.

We denote the all-ones vector with e, and the elements of the canonical basis as e_i.

Proposition 3.1 ([HRW90],[HRW92], cf. Prop. 7.23 in [BDM12])**.** *Let V be the $n\times(n-1)$ matrix, of which the columns form an orthonormal basis of the orthogonal complement of the all-ones vector e. Define $\tilde{A} := V^TAV$, $\tilde{B} := V^TBV$, and collect their eigenvalues in the vectors $\lambda_{\tilde{A}}$ and $\mu_{\tilde{B}}$ respectively. Set $D := \frac{2}{n}Aee^TB$. The* projected

eigenvalue bound *for the symmetric* QAP(A,B) *is given by*

$$\mathrm{PB}(A,B) := \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^- + \min_{\varphi \in S_n} \sum_{i=1}^{n} d_{i\varphi(i)} - \frac{(e^T A e)(e^T B e)}{n^2},$$

where $\langle x, y \rangle^- = \min_{\varphi \in S_n} \sum_{i=1}^{n} x_{\varphi(i)} y_i$. *One then has* $PB(A,B) \leq QAP(A,B)$.

One may calculate PB(A,B) by sorting $\lambda_{\tilde{A}}$ and $\mu_{\tilde{B}}$ to compute $\langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^-$ (see Proposition 5.8 in [BDM12]) and solving one linear assignment problem

$$\min_{\varphi \in S_n} \sum_{i=1}^{n} d_{i\varphi(i)}.$$

3.1.2 Quadratic Programming Bound

The second bound we consider is a convex quadratic programming (CQP) bound, introduced in [AB01a], which is based on the same projection as the bound in Proposition 3.1. We will see that it is at least as good as the projected eigenvalue bound. Here we relax $X \in \Pi_n$ to $Xe = X^T e = e$ and $X \geq 0$, i.e., we optimize over doubly stochastic matrices instead of permutation matrices.

In the following I_n and J_n denote the identity and all-ones matrices respectively of size $n \times n$, E_{ij} denotes the $n \times n$ matrix with a single one at position (i,j), and $\otimes$ the Kronecker product.

Hadley, Rendl and Wolkowicz ([HRW90],[HRW92]) observed that every doubly stochastic matrix can be written as

$$X = \frac{1}{n} e e^T + V Y V^T,$$

where V is the $n \times (n-1)$ matrix of which the columns form an orthonormal basis of the orthogonal complement of e, as before. We have $V^T V = I_{n-1}$, $VV^T = I_n - \frac{1}{n} e e^T$ and $Y = V^T X V$. As before, we set $\tilde{A} = V^T A V$ and $\tilde{B} = V^T B V$, and collect their eigenvalues in the vectors $\lambda_{\tilde{A}}$ and $\mu_{\tilde{B}}$.

In Section 3 of [AB01a], Anstreicher and Brixius introduce the following CQP bound for quadratic assignment problems.

Proposition 3.2 ([AB01a])**.** *Let A and B be symmetric matrices of size $n \times n$, and define the pair (S^*, T^*) to be any optimal solution of the problem*

$$\max \left\{ \mathrm{tr}(S+T) \colon \tilde{B} \otimes \tilde{A} - I_n \otimes S - T \otimes I_n \succcurlyeq 0 \right\},$$

so the matrix $\hat{Q} := \tilde{B} \otimes \tilde{A} - I_n \otimes S^* - T^* \otimes I_n \succcurlyeq 0$ *is positive semidefinite, and* $\operatorname{tr}(S^* + T^*) = \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^-$. *Then we get a convex quadratic bound for QAP*(A,B)*, which is at least as good as PB*(A,B)*, by*

$$\begin{aligned} QPB(A,B) := \min \;\; & y^T \hat{Q} y + \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^- + \frac{2}{n} \operatorname{tr}(B J_n A X) - \frac{(e^T A e)(e^T B e)}{n^2}, \\ \text{s.t.} \;\; & X \geq 0 \text{ is doubly stochastic}, \\ & X = \frac{1}{n} e e^T + V Y V^T, \\ & y = \operatorname{vec}(Y). \end{aligned}$$

In other words, one always has $PB(A,B) \leq QPB(A,B)$.

One may compute QPB(A, B) by solving a linear assignment problem to obtain $\hat{Q}$, and then solving a CQP in $\mathcal{O}(n^2)$ variables. For details, see Section 4 in [AB01a].

3.1.3 SDP Bound

The following semidefinite programming relaxation for QAP(A,B,C) was studied by Povh and Rendl [PR09], which is equivalent to an earlier bound by Zhao, Karisch, Rendl and Wolkowicz [Zha+98]:

$$\begin{aligned} SDPQAP(A,B,C) := \min \;\; & \langle B \otimes A + \operatorname{Diag}(\operatorname{vec}(C)), Y \rangle \\ \text{s.t.} \;\; & \langle I_n \otimes E_{jj}, Y \rangle = 1 \text{ for } j = 1, \ldots, n, \\ & \langle E_{jj} \otimes I_n, Y \rangle = 1 \text{ for } j = 1, \ldots, n, \\ & \langle I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n, Y \rangle = 0, \\ & \langle J_{n^2}, Y \rangle = n^2, \\ & Y \in \mathbb{S}^{n^2}_+ \cap \mathbb{R}^{n^2 \times n^2}_{\geq 0}, \end{aligned} \tag{3.1}$$

where $A, B, C \in \mathbb{R}^{n \times n}$ and A and B are symmetric, and $\mathbb{S}^n_+$ denotes the cone of positive semidefinite matrices of size $n \times n$. We write $SDPQAP(A,B)$ if $C = 0$. In general, if the dimension of matrices is clear, we write $X \succcurlyeq 0$ instead of $X \in \mathbb{S}^n_+$ to denote positive semidefinite matrices.

The bound $SDPQAP(A,B,C)$ is expensive to compute, as it involves an SDP with doubly nonnegative matrix variables of order $n^2 \times n^2$.

3.2 Comparison of the Bounds

The three bounds $PB(A,B)$, $QPB(A,B)$ and $SDPQAP(A,B)$ increase in computational complexity, hence, we would expect that the bounds do get better accordingly. As it turns out, we can show the expected order of the bound quality.

Theorem 3.3. *For symmetric matrices A and B we have*

$$PB(A,B) \leq QPB(A,B) \leq SDPQAP(A,B) \leq QAP(A,B).$$

Proving this bound is quite technical. The outermost inequalities are known, hence, we only need to show the inequality in the middle. Instead of doing this directly, we first introduce another SDP-based bound, which lies in between the bounds QPB and $SDPQAP$. This bound, which we will call $SDPPB$, is based on the same projection used for PB and QPB.

Again we start with the observation that for every doubly stochastic $n \times n$-matrix X we can always find an $n-1 \times n-1$-matrix Y with

$$X = \frac{1}{n}ee^T + VYV^T,$$

where $V = (v_1|\ldots|v_{n-1})$ is the $n \times (n-1)$ matrix of which the columns form an orthonormal basis of the orthogonal complement of the all-one vector e. Set $y = \mathrm{vec}(Y)$, the vector we obtain by gluing the columns of Y together. The idea of this bound is now to relax $U = yy^T$ to be a positive semidefinite matrix with certain constraints, and to relax $y = \mathrm{vec}(Y)$ to a variable u. Since $yy^T - yy^T = 0 \succcurlyeq 0$, we add the constraint that $U - uu^T \succcurlyeq 0$.

We can rewrite the objective function of the QAP at $X = \frac{1}{n}ee^T + VYV^T$ as:

$$\begin{aligned}
\mathrm{tr}(AXBX^T) &= \mathrm{tr}\left(A\left(\frac{1}{n}ee^T + VYV^T\right)B\left(\frac{1}{n}ee^T + VYV^T\right)^T\right) \\
&= \mathrm{tr}\left(\frac{1}{n^2}Aee^TBee^T\right) + \mathrm{tr}\left(\frac{1}{n}Aee^TBVY^TV^T\right) \\
&\quad + \mathrm{tr}\left(\frac{1}{n}AVYV^TBee^T\right) + \mathrm{tr}\left(\frac{1}{n}AVYV^TBVY^TV^T\right) \\
&= \mathrm{tr}\left(\tilde{A}Y\tilde{B}Y^T\right) + \frac{2}{n}\mathrm{tr}\left(BJ_nAVYV^T\right) + \frac{1}{n^2}(e^TAe)(e^TBe) \\
&= \langle \tilde{B} \otimes \tilde{A}, yy^T \rangle + \frac{2}{n}\mathrm{vec}\left(V^TBJ_nAV\right)^T y + \frac{1}{n^2}(e^TAe)(e^TBe).
\end{aligned}$$

Thus, we can write it as linear function of yy^T and y, which we relax to U and u respectively. To make this bound at least as good as the convex quadratic bound

$QPB(A,B)$, we have to add more conditions, which follow from $U = yy^T = \mathrm{vec}(V^TXV)\mathrm{vec}(V^TXV)^T$. We have, for all $1 \le i,j \le n-1$, that

$$\begin{aligned}\langle I \otimes E_{ij}, U\rangle &= \mathrm{tr}\left(E_{ij}V^TXVIV^TX^TV\right)\\ &= \mathrm{tr}\left(v_jv_i^TX\left(I-\frac{1}{n}J\right)X^T\right)\\ &= \mathrm{tr}\left(v_jv_i^T\right)-\frac{1}{n}\mathrm{tr}\left(v_jv_i^TJ\right)\\ &= \mathrm{tr}\left(v_i^Tv_j\right)-\frac{1}{n}\mathrm{tr}\left(e^Tv_jv_i^Te\right)=\delta_{ij},\end{aligned}$$

and analogously we can show that $\langle E_{ij}\otimes I, U\rangle = \delta_{ij}$ as well. Here δ_{ij} denotes the Kronecker-Delta, which is one if $i=j$, and zero otherwise. Finally, the property that $X = \frac{1}{n}ee^T + VYV^T$ is nonnegative is equivalent to $(V\otimes V)y \ge -\frac{1}{n}e\otimes e$.

Proposition 3.4. *With $\tilde{A},\tilde{B}$ and V as defined in Proposition 3.1, we obtain a bound for $QAP(A,B)$ by:*

$$\begin{aligned}SDPPB(A,B) := \min\ & \langle \tilde{B}\otimes\tilde{A}, U\rangle + \frac{2}{n}\mathrm{vec}(V^TBJAV)^Tu + \frac{1}{n^2}(e^TAe)(e^TBe) \qquad (3.2)\\ s.t.\ & \begin{pmatrix}1 & u^T\\ u & U\end{pmatrix} \succcurlyeq 0,\\ & \langle E_{ij}\otimes I_{n-1}, U\rangle = \delta_{ij} \quad \forall i,j=1,\dots,n-1,\\ & \langle I_{n-1}\otimes E_{ij}, U\rangle = \delta_{ij} \quad \forall i,j=1,\dots,n-1,\\ & (V\otimes V)u \ge -\frac{1}{n}e\otimes e.\end{aligned}$$

Lemma 3.5.

$$SDPPB(A,B) \ge QPB(A,B).$$

Proof. Let (U,u) be an optimal solution of $SDPPB(A,B)$. Then we construct a feasible solution for $QPB(A,B)$ by setting $y=u$, since $(V\otimes V)u + \frac{1}{n}ee^T \ge 0$, and thus $X = \frac{1}{n}ee^T + VYV^T \ge 0$ for $\mathrm{vec}(Y) = y$. The matrix X is doubly stochastic, since $(e\otimes e_i)^T(V\otimes V)u = (e^TV\otimes e_i^TV)u = 0$ and $(e_i\otimes e)^T(V\otimes V)u = 0$, so adding VYV^T to the doubly stochastic matrix $\frac{1}{n}ee^T$ results in another doubly stochastic matrix.

In the following $\hat{Q}$ is as defined in Proposition 3.2. By the Schur complement theorem we know that $U - uu^T \succcurlyeq 0$, hence, we have that

$$u^T\hat{Q}u \le \langle \hat{Q}, U\rangle$$

$$
\begin{aligned}
&= \langle \tilde{B} \otimes \tilde{A}, U\rangle - \langle I \otimes S^*, U\rangle - \langle T^* \otimes I, U\rangle \\
&= \langle \tilde{B} \otimes \tilde{A}, U\rangle - \sum_{i,j=1}^{n-1} S^*_{ij} \langle I \otimes E_{ij}, U\rangle - \sum_{i,j=1}^{n-1} T^*_{ij} \langle E_{ij} \otimes I, U\rangle \\
&= \langle \tilde{B} \otimes \tilde{A}, U\rangle - \operatorname{tr}(S^*) - \operatorname{tr}(T^*) \\
&= \langle \tilde{B} \otimes \tilde{A}, U\rangle - \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}}\rangle^-.
\end{aligned}
$$

Thus, we can compare the two bounds by

$$
\begin{aligned}
QPB(A,B) &\leq y^T \hat{Q} y + \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}}\rangle + \frac{2}{n}\operatorname{tr}(BJAX) - \frac{1}{n^2}(e^TAe)(e^TBe) \\
&\leq \langle \tilde{B} \otimes \tilde{A}, U\rangle + \frac{2}{n}\operatorname{tr}(BJAX) - \frac{1}{n^2}(e^TAe)(e^TBe) \\
&= \langle \tilde{B} \otimes \tilde{A}, U\rangle + \frac{2}{n}\operatorname{vec}(V^TBJAV)^T u + \frac{1}{n^2}(e^TAe)(e^TBe) \\
&= SDPPB(A,B).
\end{aligned}
$$

□

To prove the other inequality, we make use of a lemma of Povh and Rendl [PR09]. They give an alternative description of the feasible set of *QAPSDP* in terms of blocks of Y. For this we split the $n \times n$-matrix-variable Y of (3.1) into n^2 blocks of size $n \times n$, which we call $Y^{(ij)}$. We write $Y = \left[Y^{(ij)}\right]_{1\leq i,j\leq n}$, and use similar notation for other block-matrices.

Lemma 3.6 (Lemma 6 in [PR09])**.** *A* $Y = \left[Y^{(ij)}\right]_{1\leq i,j\leq n} \in \mathbb{S}^{n^2}_+, Y \geq 0$ *is feasible for* (3.1) *if and only if*

(i) $\langle I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n, Y\rangle = 0$,

(ii) $\operatorname{tr}\left(Y^{(ii)}\right) = 1 \quad \forall i, \qquad \sum_{i=1}^n \operatorname{diag}\left(Y^{(ii)}\right) = e$,

(iii) $Y^{(ij)}e = \operatorname{diag}\left(Y^{(jj)}\right) \quad \forall i,j$,

(iv) $\sum_{i=1}^n Y^{(ij)} = e\ \operatorname{diag}\left(Y^{(jj)}\right)^T$.

Lemma 3.7.

$$SDPPB(A,B) \leq SDPQAP(A,B).$$

Proof. With the properties of Lemma 3.6 we can show that we get a feasible solution for $SDPPB(A,B)$ from a feasible solution Y of $SDPQAP(A,B)$ by setting $U = (V^T \otimes V^T)Y(V \times V)$ and $u = (V^T \otimes V^T)y$, which is the transformation to a

Slater-feasible variant of *SDPQAP* (see e.g., the thesis of Uwe Truetsch [Tru14]). Similarly to Y, we can split U into $(n-1)^2$ blocks of size $(n-1)\times(n-1)$, which we call $U^{(ij)}$. We get an explicit formula for these blocks in terms of the $Y^{(ij)}$, if we see $V\otimes V$ as $n(n-1)$ blocks of size $n\times(n-1)$, since then all block sizes are compatible with multiplication.

$$\begin{aligned}
U &= (V^T\otimes V^T)Y(V\times V),\\
&= (V^T\otimes V^T)\left(\sum_{k=1}^{n} Y^{(ik)}V_{kj}V\right)_{\substack{1\le i\le n\\ 1\le j\le n-1}},\\
&= \left(\sum_{l=1}^{n} V_{li}V^T\sum_{k=1}^{n} Y^{(lk)}V_{kj}V\right)_{1\le i,j\le n-1},
\end{aligned}$$

hence

$$U^{(ij)} = \sum_{l,k=1}^{n} V_{li}V_{kj}V^T Y^{(lk)}V.$$

We can now use $((i))$-$((iv))$ to derive some properties of U. First note that by $((i))$ and $((ii))$ we know that $\operatorname{tr}(Y^{(ij)}) = \delta_{ij}$, and by $((ii))$ and $((iii))$ that $\operatorname{tr}(Y^{(ij)}J) = 1$. Hence, we see that

$$\begin{aligned}
\langle E_{ij}\otimes I_{n-1}, U\rangle &= \operatorname{tr}(U^{(ij)}),\\
&= \operatorname{tr}\left(\sum_{l,k=1}^{n} V_{li}V_{kj}V^T Y^{(lk)}V\right),\\
&= \sum_{l,k=1}^{n} V_{li}V_{kj}\operatorname{tr}\left(Y^{(lk)}\left(I_n-\frac{1}{n}J_n\right)\right),\\
&= \sum_{l=1}^{n} V_{li}V_{lj} - \frac{1}{n}\sum_{l,k=1}^{n} V_{li}V_{kj},\\
&= v_i^T v_j - 0 = \delta_{ij}.
\end{aligned}$$

Similarly, we can use $\sum_{i=1}^{n-1} V_{li}V_{ki} = (VV^T)_{lk} = \delta_{lk} - \frac{1}{n}$, $((i))$, $((ii))$ and $((iv))$ to show that

$$\begin{aligned}
\langle I_{n-1}\otimes E_{ij}, U\rangle &= \left(\sum_{i=1}^{n-1} U^{(ii)}\right)_{ij},\\
&= \left(\sum_{i=1}^{n-1}\sum_{l,k=1}^{n} V_{li}V_{ki}V^T Y^{(lk)}V\right)_{ij},
\end{aligned}$$

$$
\begin{aligned}
&= \left(\sum_{l=1}^{n} V^T Y^{(ll)} V - \frac{1}{n} \sum_{l,k=1}^{n} V^T Y^{(lk)} V \right)_{ij}, \\
&= \left(V^T V - \frac{1}{n} V^T J V \right)_{ij}, \\
&= (I_{n-1} - 0)_{ij} = \delta_{ij}.
\end{aligned}
$$

To construct a feasible u with the objective value we need, we use that we can add $Y - yy^T \succcurlyeq 0$, $y = \operatorname{diag}(Y)$ to (3.1) without changing the optimal value of $SDPQAP$. With an optimal solution (Y, y) we thus set

$$
(U, u) = \left((V^T \otimes V^T) Y (V \otimes V), (V^T \otimes V^T) y \right).
$$

With $((ii))$ we see that

$$
\begin{aligned}
(V \otimes V)u &= (V \otimes V)(V^T \otimes V^T)y, \\
&= (I - \frac{1}{n}J) \otimes (I - \frac{1}{n}J)y, \\
&= y - \frac{1}{n}(J \otimes I) - \frac{1}{n}(I \otimes J) + \frac{1}{n^2}Jy, \\
&= y - \frac{1}{n}e \geq -\frac{1}{n}e.
\end{aligned}
$$

Since Y and $Y - yy^T$ are positive semidefinite, the matrices $(V^T \otimes V^T)Y(V \otimes V) = U$ and $(V^T \otimes V^T)(Y - yy^T)(V \otimes V) = U - uu^T$ are positive semidefinite as well, and thus feasible for $SDPPB(A, B)$. What remains to be seen is that the objective values of the two programs are the same.

$$
\begin{aligned}
\langle \tilde{B} \otimes \tilde{A}, U \rangle &= \operatorname{tr}((V \otimes V)(V^T \otimes V^T)(B \otimes A)(V \otimes V)(V^T \otimes V^T)Y), \\
&= \operatorname{tr}(((I - \frac{1}{n}J) \otimes (I - \frac{1}{n}J))(B \otimes A)((I - \frac{1}{n}J) \otimes (I - \frac{1}{n}J))Y), \\
&= \operatorname{tr}((B \otimes A)(Y - \frac{1}{n}ey^T)((I - \frac{1}{n}J) \otimes (I - \frac{1}{n}J))), \\
&= \operatorname{tr}((B \otimes A)(Y - \frac{1}{n}ey^T - \frac{1}{n}ye^T + \frac{1}{n^2}J)), \\
&= \langle B \otimes A, Y \rangle - \frac{2}{n}e^T(B \otimes A)y + \frac{1}{n^2}(e^T Ae)(e^T Be),
\end{aligned}
$$

and

$$
\frac{2}{n}\operatorname{vec}(V^T BJAV)^T u = \frac{2}{n}\operatorname{vec}(V^T BJAV)^T (V^T \otimes V^T)y,
$$

$$
\begin{aligned}
&= \frac{2}{n}\mathrm{vec}(VV^T BJAVV^T)^T y, \\
&= \frac{2}{n}\mathrm{vec}((I - \frac{1}{n}J)BJA(I - \frac{1}{n}J))^T y, \\
&= \frac{2}{n}\mathrm{vec}(BJA)^T y - \frac{2}{n^2}\mathrm{vec}(BJA)^T (I \otimes J)y \\
&\quad - \frac{2}{n^2}\mathrm{vec}(BJA)^T (J \otimes I)y + \frac{2}{n^3}\mathrm{vec}(BJA)^T Jy, \\
&= \frac{2}{n}\mathrm{vec}(BJA)^T y - \frac{2}{n^2}(e^T Ae)(e^T Be),
\end{aligned}
$$

thus

$$\langle \tilde{B} \otimes \tilde{A}, U \rangle + \frac{2}{n}\mathrm{vec}(V^T BJAV)^T u + \frac{1}{n^2}(e^T Ae)(e^T Be) = \langle B \otimes A, Y \rangle.$$

Here we used properties $((i))$-$((iv))$, that $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$ and that A and B are symmetric. □

Proof of Theorem 3.3. The only inequality we have to show is

$$QPB(A,B) \leq SDPQAP(A,B),$$

since the leftmost inequality was shown in [AB01a]. By Lemma 3.5 and Lemma 3.7 we have

$$QPB(A,B) \leq SDPPB(A,B) \leq SDPQAP(A,B)$$

and the theorem follows. □

Remark 3.8. While it was expected that $SDPQAP(A,B)$, which has $\mathcal{O}(n^4)$ linear inequality constraints, is better than the projected eigenvalue bound $PB(A,B)$, it was less so for the bound $SDPPB(A,B)$ introduced during the proof of Theorem 3.3, since it only has $\mathcal{O}(n^2)$ linear inequality constraints.

3.3 Symmetry Reduction for the SDP Bound

We will now investigate the Jordan reduction method, which was described in Chapter 2, for the SDP bound (3.1) for quadratic assignment problems.

First, we have to transform this program into conic form, as seen on the right side of equation (2.3). We get a feasible solution X_0 by forming the outer product of a vectorized permutation-matrix, for example we can set

$$X_0 = \mathrm{vec}(I_n)\mathrm{vec}(I_n)^T.$$

We get the space $\mathcal{L}$, as seen in Chapter 2, by

$$\mathcal{L} = \{X \in \mathbb{S}^{n^2} \mid \langle A_i, X\rangle = 0 \quad \forall i \in [m]\},$$

where

$$\{A_i\}_{i\in[m]} = \{J_{n^2}, I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n, I_n \otimes E_{jj} \text{ and } E_{jj} \otimes I_n \ (j \in [n])\}$$

are the data-matrices of the constraints of the SDP relaxation (3.1). Accordingly, the orthogonal complement is exactly $\mathcal{L}^\perp = \operatorname{span}\{A_1, \ldots, A_m\}$.

Theorem 3.9. *Consider an admissible subspace, say $S \subset \mathbb{S}^{n^2}$, for the QAP relaxation (3.1) with $n > 2$. Assume that $I \in S$ or $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$. Then:*

1. *S has a basis of nonnegative matrices with disjoint supports.*
2. *If S is a Jordan subalgebra of $\mathbb{S}^{n^2}$, i.e. closed under taking squares, then S is a Jordan configuration.*
3. *If $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, and S is unital, then S is a Jordan configuration.*

Proof. Let S be an admissible subspace for the QAP relaxation (3.1) with $n > 2$. The first claim of the theorem is an immediate consequence of Propositions 2.17 and 2.16.

To show the second claim, note that, by assumption, S contains $X_{0,\mathcal{L}^\perp}$ and its square, which we will now calculate. In [BK22b] the authors show that:

$$X_{0,\mathcal{L}^\perp} = \frac{1}{n^2-n}(J_{n^2} - I_n \otimes J_n - J_n \otimes I_n) + \frac{1}{n-1} I_{n^2}.$$

Straightforward calculation now yields

$$X^2_{0,\mathcal{L}^\perp} = \frac{n^2-2n+2}{n^2(n-1)^2} J_{n^2} - \frac{1}{n(n-1)^2}(I_n \otimes J_n + J_n \otimes I_n) + \frac{1}{(n-1)^2} I,$$
$$X^4_{0,\mathcal{L}^\perp} = \frac{1}{(n-1)^2} X^2_{0,\mathcal{L}^\perp} + \frac{n-2}{n(n-1)^2} J_{n^2}.$$

Thus, S contains the all-ones matrix if $n > 2$, since

$$\frac{n-2}{n(n-1)^2} J_{n^2} = X^4_{0,\mathcal{L}^\perp} - \frac{1}{(n-1)^2} X^2_{0,\mathcal{L}^\perp},$$

and the right-hand-side terms both belong to S. Thus, S must therefore have a 0/1 basis, i.e. it must be a partition subspace, since it has a basis of nonnegative

matrices with disjoint supports. To show that it is in fact a Jordan configuration, we only need to show still that it contains the identity matrix. To this end, it suffices to note that all the diagonal entries of $X^2_{0,\mathcal{L}^\perp}$ are the same, and different from the off-diagonal entries. Since S has a $0/1$ basis, it must therefore contain the identity.

We now prove the third claim in the theorem. If we assume $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, and that S unital, then S is closed under taking squares, i.e. it is a Jordan subalgebra of $\mathbb{S}^n$ [Per17, Lemma 5.2.2]. Now the result follows from part 2 of this theorem. □

The important practical implication of this theorem is that the optimal admissible Jordan configuration S of the QAP relaxation (3.1) may be computed using Algorithm 3. The resulting reduction is at least as good as the known ones from the literature, as we now show. "At least as good" means here that the other algorithms return partition spaces which refine the partition space returned by Algorithm 3. Thus, the better reduction results in an optimization problem in fewer variables.

Corollary 3.10. *This symmetry reduction of the QAP relaxation* (2.2) *via Algorithm 3 is at least as good as both the group symmetry reduction (see [KS10; KS12]) and the reduction to the smallest coherent algebra containing the data matrices of the program (via the Weisfeiler-Leman algorithm [LW68]).*

Proof. The symmetric part of a coherent configuration is a Jordan configuration, and the partition given by the orbitals of a group leaving the program invariant is a coherent configuration. Both the partition spaces given by the coherent configuration and the partition given by orbitals are admissible since they adhere to stricter conditions allowing for symmetry reduction than required for admissible subspaces. Since Algorithm 3 returns the coarsest admissible Jordan configuration, both coherent configurations returned by the other algorithms refine it. □

3.3.1 Results of Reductions of QAPLib Problems

In practice the (partition) Jordan reduction is not much stronger than group symmetry reduction, and reduction to the smallest coherent algebra containing the data matrices. When comparing reductions for data from QAPLib [BKR97], only a single reduction (esc16f), of the ones that were symmetry reduced before, was stronger, the others were exactly the same as reported in [KS10], where the reduction was done using group symmetry. But we managed to reduce some larger instances for the first time. We also do gain a large speed up in determining the

reduction, since we avoid having to determine the automorphism groups of matrices. In Table 3.1 we give the dimension of the smallest admissible partition subspace for each problem (for which we determined a reduction), the original number of variables of the problem, and the time needed for the reduction. In Table 3.2 we show the time needed to block-diagonalize (using the algorithm described in [Mur+10]) and solve these problems afterwards (if the dimension of the admissible subspace was at most 3000), as well as the resulting bounds. The optimal value of most of these is known exactly which we give in the last row, taken from `http://anjos.mgi.polymtl.ca/QAPLib/inst.html`.

3.4 Concluding Remarks

We have shown that indeed the optimal admissible subspace semidefinite programming relaxation of the quadratic assignment problem is a Jordan configuration in Theorem 3.9 of Section 3.3 under some weak requirements, and applied this to the symmetric instances of QAPLib.

QAP	Dim.	Red. dim.	Jordan red. (s)
chr18b	52650	14742	0.358
esc16a	32896	150	0.162
esc16b	32896	155	0.192
esc16c	32896	405	0.194
esc16d	32896	405	0.171
esc16e	32896	135	0.166
esc16f	32896	3	0.100
esc16g	32896	230	0.207
esc16h	32896	90	0.130
esc16i	32896	280	0.254
esc16j	32896	150	0.214
esc32a	524800	2112	3.826
esc32b	524800	96	3.306
esc32c	524800	366	3.228
esc32d	524800	342	3.097
esc32e	524800	120	2.885
esc32g	524800	180	2.858
esc32h	524800	666	3.051
esc64a	8390656	679	57.581
kra32	524800	28752	3.099
nug12	10440	2952	0.077
nug15	25425	7425	0.152
nug16b	32896	4704	0.147
nug20	80200	21000	0.819
nug21	97461	27783	0.474
nug22	117370	29766	0.757
nug24	166176	41760	1.010
nug25	195625	28675	1.132
nug27	266085	75087	1.489
nug28	307720	78792	1.865
scr12	10440	2952	0.057
scr15	25425	13275	0.147
tai64c	8390656	75	55.839
tho30	405450	112950	2.927
tho40	1280800	333600	9.477
wil50	3126250	813750	27.123

Table 3.1: Results of the numerical symmetry reduction of QAPLib problems using Algorithm 3.

QAP	block-diag. (s)	solve (s)	blocks (size × mult.)	optimal value (3.1)	QAP optimum
esc16a	0.884	0.229	6 × 5,3 × 5,1 × 15,	63.285	68
esc16b	0.749	0.285	7 × 5,1 × 15,	289.999	292
esc16c	2.766	0.759	12 × 5,1 × 15,	153.999	160
esc16d	2.714	0.373	12 × 5,1 × 15,	13.000	16
esc16e	0.753	0.159	6 × 5,2 × 5,1 × 15,	26.337	28
esc16f	0.040	0.048	1 × 3,	0.000	0
esc16g	1.109	0.217	9 × 5,1 × 5,	24.740	26
esc16h	0.414	0.125	5 × 5,1 × 15,	976.228	996
esc16i	1.572	0.296	10 × 5,1 × 5,	11.375	14
esc16j	0.690	0.167	7 × 5,1 × 10,	7.794	8
esc32a	482.027	23.958	26 × 6,1 × 6,	103.320	130
esc32b	11.502	0.041	2 × 24,1 × 24,	131.883	168
esc32c	51.146	0.296	10 × 6,1 × 36,	615.178	642
esc32d	56.076	0.213	9 × 6,2 × 12,1 × 36,	190.227	200
esc32e	11.436	0.054	5 × 6,1 × 30,	1.900	2
esc32g	14.941	0.096	7 × 6,1 × 12,	5.833	6
esc32h	114.943	1.135	14 × 6,1 × 36,	424.398	438
esc64a	1 985.917	0.885	13 × 7,2 × 7,1 × 21,	97.750	116
nug12	12.884	80.019	48 × 2,24 × 2,	567.970	578
scr12	12.894	83.330	48 × 2,24 × 2,	31 409.997	31410
tai64c	182.909	0.153	2 × 15,1 × 30,	1 811 366.481	≥1855928

Table 3.2: Details on solving (3.1) for QAPLib instances via block-diagonalization.

4

Energy Minimization on a Toric Grid as a QAP

In this chapter we apply the three bounds described in Chapter 3 to a discrete energy minimization problem. It was first described in [Tai95] as the problem of printing a particular shade of gray, by repeating the same tile of black and white squares in all directions. Other applications from physics are the search for ground states of a two-dimensional repulsive lattice gas at zero temperature ([Wat97]), and more generally the Falicov-Kimball model ([FK69; Ken94]), which is relevant for modelling valence fluctuations in transition metal oxides, binary alloys and high-temperature super-conductors ([Wat97]).

To get a distribution of black and white tiles as equal as possible, it is natural to view the problem of printing a shade of gray as a problem of minimizing the potential energy between repulsive particles on a toric grid. This problem can then be reformulated as a quadratic assignment problem, which allows us to apply the three bounds of Chapter 3 to this problem. We will see in Proposition 4.4 and Proposition 4.5 that both the projected eigenvalue bound, and the convex quadratic programming bound, coincide with an eigenvalue bound for this problem introduced in [BDL13]. In Section 4.2 we describe the technique one may use to calculate the semidefinite programming bound, which involves a symmetry

reduction of the problem to a more manageable size. Our approach is based on the recent Jordan reduction method of Parrilo and Permenter [PP19]. Finally, in Section 4.4 we present numerical results on the bounds for instances on different grid sizes, including the semidefinite programming bound after Jordan reduction, and thus prove optimality of certain grid arrangements. In this way we extend earlier results by Bouman, Draisma and van Leeuwaarden [BDL13].

Energy minimization in the continuous setting, specifically on spheres and in Euclidean space, were studied before by Cohn and Kumar [CK06]. There the authors prove *universal optimality* of certain particle configurations, i.e. optimality for all reasonable potential functions. After symmetry reduction, they reduce their bound to an infinite sized linear program, with variables that correspond to the distances between particles, and how often they appear. The main difference in this work, other than us placing the particles on a discrete (toric) grid, is that we do not consider universal optimality, but instead work with a fixed function for the potential energy of pairs of particles. This allows us to consider bounds which are, after reduction, based on variables that correspond to all vectors between pairs of particles.

4.1 Energy Minimization on a Toric Grid as a QAP

We generalize a problem described by Taillard in [Tai95], which models the problem of printing a certain shade of gray with only black and white squares ("pixels"). An example of these problems is included in the QAPLib dataset [BKR97], namely Tai64c.

The goal is to print a particular shade of gray with a given density m/n (ratio of black to total squares), which is done by repeating a grid of $n = n_1 \times n_2$ square cases, exactly m of which are black. We want the cases to be as regular as possible, so it is natural to model this as an energy minimization problem, with repulsive particles corresponding to the black squares. We define $[k] := \{1, 2, \ldots, k\}$. If we have two particles (i.e. black squares) at locations $(x_1, y_1) \in [n_1] \times [n_2]$ and $(x_2, y_2) \in [n_1] \times [n_2]$, the potential energy between them is inverse to their distance, where the distance is given by the shortest path metric on the toric grid, which also known as *Lee metric*. We use this grid, since we wish to tile the plane with the $n_1 \times n_2$-rectangles, as can be seen in Figure 4.1. The potential energy associated with two repulsive particles at respective positions $(x_1, y_1) \in [n_1] \times [n_2]$

and $(x_2, y_2) \in [n_1] \times [n_2]$ is

$$f_{(x_1,y_1),(x_2,y_2)} = \frac{1}{d_{\text{Lee}}((x_1, y_1), (x_2, y_2))},$$

if the coordinates are different, where

$$d_{\text{Lee}}((x_1, y_1), (x_2, y_2)) = \min(|x_1 - x_2|, n_1 - |x_1 - x_2|) + \min(|y_1 - y_2|, n_2 - |y_1 - y_2|)$$

is the Lee distance, given by the shortest path metric on the toric grid. We set $f_{i,i} = 0$.

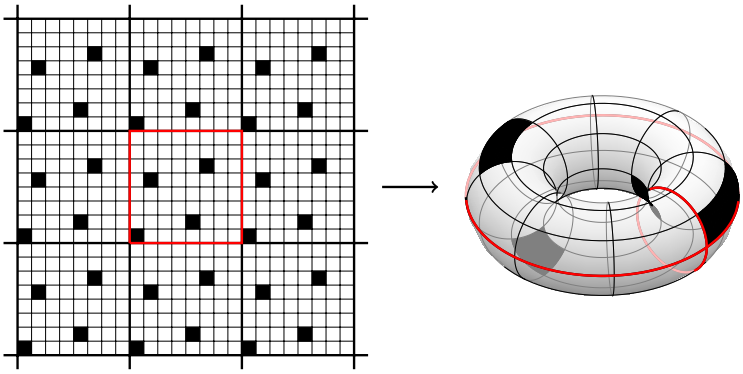


Figure 4.1: Example of an $n_1 \times n_2 = 8 \times 8$ grid tiling with $m = 4$, and the corresponding toric interpretation of the 8×8 grid.

We can then formulate this problem as QAP with matrices $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{n \times n}$, indexed by grid points $i = (x_i, y_i) \in [n_1] \times [n_2]$, $j = (x_j, y_j) \in [n_1] \times [n_2]$, given by

$$a_{ij} = \begin{cases} 1, & \text{if } i, j \leq m \\ 0, & \text{otherwise.} \end{cases}, \qquad b_{ij} = f_{i,j} = f_{(x_i,y_i),(x_j,y_j)}. \tag{4.1}$$

In the definition of A we compared a grid point to an integer m. The ordering of the grid points does not matter for the optimal value or the symmetry reduction, so it is enough to assume that we have any fixed ordering of the indices, i.e. we may associate $[n]$ with $[n_1] \times [n_2]$ and will write $[n] = [n_1] \times [n_2]$ when convenient. We may also assume that the ordering is such that the nonzero elements of the matrix A are given by the $m \times m$-block in the upper left corner, so that

$$\min_{\pi \in S_n} \sum_{i,j=1}^{n} a_{ij} b_{\pi(i)\pi(j)} = \min_{\pi \in S_n} \sum_{i,j \leq m} b_{\pi(i)\pi(j)} = \min_{\substack{T \subseteq [n_1] \times [n_2] \\ |T| = m}} \sum_{a,b \in T} f_{a,b} \tag{4.2}$$

Note that the QAP has dimension $n = n_1 \times n_2$, and its semidefinite relaxation has dimension n^2, which is already 4096 on an 8×8 grid.

To reduce the number of cases one has to look at, we can show a that selecting the complement of an optimal solution leads to another optimal solution.

Proposition 4.1. *Consider a $[n] = [n_1] \times [n_2]$ grid, and a function $f : [n] \times [n]$ with $\sum_{j\in[n]} f(i,j) = \sum_{j\in[n]} f(j,i) = c$ for all $i \in [n]$ and a constant $c \in \mathbb{R}$. Then, if $T \subseteq [n]$ minimizes*

$$\min_{\substack{T\subseteq[n_1]\times[n_2]\\|T|=m}} \sum_{a,b\in T} f_{a,b},$$

then $S = [n] \setminus T$ minimizes

$$\min_{\substack{S\subseteq[n_1]\times[n_2]\\|S|=n-m}} \sum_{a,b\in S} f_{a,b}.$$

Proof. We can rewrite the objective function as

$$\begin{aligned}\sum_{a,b\in T} f_{a,b} &= \sum_{a,b\in[n]} f_{a,b} - \sum_{\substack{a\notin T\\ b\in[n]}} f_{a,b} - \sum_{\substack{a\in[n]\\ b\notin T}} f_{a,b} + \sum_{\substack{a\notin T\\ b\notin t}} f_{a,b}\\ &= cn - 2c(n-m) + \sum_{a,b\in S} f_{a,b}.\end{aligned}$$

Since the term $cn - 2c(n-m)$ is independent of T and S, minimizing $\sum_{a,b\in T} f_{a,b}$ is equivalent to minimizing $\sum_{a,b\in S} f_{a,b}$. □

4.1.1 Eigenvalue Bound of Bouman, Draisma and Leeuwaarden

Problem (4.2) was considered before in [BDL13], specifically for the case of the Lee-metric, and $m = \frac{n}{2}$ (which we can see as special case of our variant). They took a look at a different relaxation of the problem, which they call *fractional total energy*

$$\begin{aligned}\min\ & x^T B x \\ \text{s.t.}\ & x^T x = x^T e = m,\end{aligned} \tag{4.3}$$

where $B = (b_{i,j})$ is the matrix of potential energies between grid points i and j, as defined in 4.1. In the relaxation (4.3), the discrete variables x_i correspond to particle positions, relaxed to continuous values. Thus, if x is the characteristic vector of a subset of m particles, then it is exactly the potential energy of the set of particles. The optimal solution of the relaxation (4.3) may be expressed in terms of the eigenvalues of B as follows.

Proposition 4.2 (Proposition 2.5. in [BDL13]). *Let $\lambda_{\min}$ be the smallest eigenvalue of B, and λ_1 the eigenvalue of B corresponding to e. Then the set of optimal solutions of the minimization problem* (4.3) *consists of all vectors of the form $\frac{m}{n}e + y$, where y belongs to the eigenspace of B with eigenvalue $\lambda_{\min}$, is perpendicular to e, and satisfies $y^T y = m - \frac{m^2}{n}$.*

The optimal value, and thus a lower bound for the minimum potential energy of m particles on a toric grid with n nodes, is given by:

$$\lambda_1 \frac{m^2}{n} + \lambda_{\min}\left(m - \frac{m^2}{n}\right). \tag{4.4}$$

We will from now on refer to the bound (4.4) as BDL-eigenvalue bound, named after the authors.

Remark 4.3. The energy minimization problem on a toric grid is a special case of the so-called *m-cluster* problem, or *minimum weight m-subgraph* problem (see e.g., [MR12]). Indeed, the toric grid defines the graph in question, the edge weights are the energies between grid points, and the minimum weight m-subgraph corresponds to the m grid points where the particles are placed in a minimum energy configuration. An SDP-bound with $n \times n$ matrix variables was proposed for the maximum weight m-subgraph problem in [MR12], which is of the form 4.5:

$$\begin{aligned} \inf\ & \frac{1}{4}e^T B e + \frac{1}{2}e^T B y + \frac{1}{4}\langle B, Y\rangle \\ \text{s.t. } & e^T y = 2m - n, \\ & \sum_i Y_{ij} = (2m-n)y_j \quad \text{for } j = 1,\dots,n, \\ & Y_{ii} = 1 \quad \text{for } i = 1,\dots,n, \\ & \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} \succcurlyeq 0. \end{aligned} \tag{4.5}$$

It is straightforward to check that this bound is at most as good as (4.3) for our problem. Indeed, since the weighted graph given by B is vertex transitive, we obtain a feasible solution (Y, y) for (4.5) (with the same objective value) from a feasible solution x for (4.3) by setting

$$y = \mathrm{sym}(2x - e) = \left(\frac{2m}{n} - 1\right)e \text{ and } Y = \mathrm{R}((2x-e)(2x-e)^T),$$

where sym averages a vector over the orbits of the automorphism group of the graph given by B, and R is the Reynolds operator of the same group (i.e. it averages

matrix entries over the 2-orbits of the group). Thus, this bound is not strong enough to improve the BDL-eigenvalue bound (4.4).

4.1.2 Bound Comparison

We now want to compare the BDL-eigenvalue bound with the QAP relaxations described last section, as well as the different QAP relaxations for this specific case.

Proposition 4.4. *The BDL-eigenvalue bound of [BDL13], see* (4.4)*, coincides with the projected eigenvalue bound (see Proposition 3.1) for the QAP problem* (4.2).

Proof. The matrix D in Proposition 3.1 is now given by

$$D = \frac{2}{n} A e e^T B = \lambda_1 \frac{2}{n} (\underbrace{m, \dots, m}_{m \text{ times}}, 0, \dots, 0)^T e^T,$$

and thus has entries $\lambda_1 \frac{2m}{n}$ in the first m rows, and zeros otherwise. As such, the permutation φ does not influence the result, and

$$\min_{\varphi} \sum_{i=1}^{n} d_{i\varphi(i)} = \lambda_1 \frac{2m^2}{n}. \tag{4.6}$$

We know that e is an eigenvector of B, hence, we have

$$\frac{(e^T A e)(e^T B e)}{n^2} = \frac{m^2 \lambda_1 n}{n^2} = \lambda_1 \frac{m^2}{n}.$$

The matrix A has rank one, so $\tilde{A}$ has rank one as well. Since e is an eigenvector of B, $\tilde{B}$ has the same eigenvalues as B, except for λ_1, thus we get

$$\langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^- = \lambda_{\max}(\tilde{A}) \lambda_{\min}(\tilde{B}) = \operatorname{tr}(\tilde{A}) \lambda_{\min}.$$

The eigenvalue of $\tilde{A}$ is exactly

$$\begin{aligned} \operatorname{tr}(\tilde{A}) &= \operatorname{tr}(V^T A V), \\ &= \operatorname{tr}(A) - \frac{1}{n} \operatorname{tr}(AJ), \\ &= m - \frac{m^2}{n}. \end{aligned}$$

Combining these, we see that the projection bound is the same as the eigenvalue bound:

$$\begin{aligned} PB(A,B) &= \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^- + \min_{\varphi} \sum_{i=1}^{n} d_{i\varphi(i)} - \frac{(e^T A e)(e^T B e)}{n^2} \\ &= \lambda_1 \frac{m^2}{n} + \lambda_{\min}\left(m - \frac{m^2}{n} \right). \end{aligned}$$

□

Thus, the BDL-eigenvalue bound (4.4) is the same as the weakest of the QAP bounds we considered, namely the bound $PB(A,B)$. Furthermore, even the convex quadratic bound cannot give us better bounds here, as we show now.

Proposition 4.5. *If A and B are of the form*

$$a_{ij} = \begin{cases} 1, & \text{if } i,j \le m \\ 0, & \text{otherwise.} \end{cases}, \quad b_{i,j} = f_{i,j} = f_{(x_i,y_i),(x_j,y_j)},$$

as defined for the energy minimization problem, then we have that

$$PB(A,B) = QPB(A,B),$$

where

$$PB(A,B) = \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^- + \min_{\varphi} \sum_{i=1}^{n} d_{i\varphi(i)} - \frac{(e^T A e)(e^T B e)}{n^2}$$

$$QPB(A,B) = \min_{\substack{X \ge 0 \text{ doubly stochastic} \\ X = \frac{1}{n} e e^T + V Y V^T \\ y = \mathrm{vec}(Y)}} y^T \hat{Q} y + \langle \lambda_{\tilde{A}}, \mu_{\tilde{B}} \rangle^- + \frac{2}{n} \mathrm{tr}(BJAX) - \frac{(e^T A e)(e^T B e)}{n^2},$$

where $D = (d_{ij}) = \frac{2}{n} AJB$ and $\hat{Q}$ is positive semidefinite, as defined before (see Proposition 3.1, (3.2)*).*

Proof. If we eliminate the terms which appear in both programs, we see that we want to show

$$\min_{\varphi} \sum_{i=1}^{n} d_{i\varphi(i)} = \min_{\substack{X \ge 0 \text{ doubly stochastic} \\ X = \frac{1}{n} e e^T + V Y V^T \\ y = \mathrm{vec}(Y)}} y^T \hat{Q} y + \frac{2}{n} \mathrm{tr}(BJAX).$$

By definition of D the two linear terms are equal, except that on the left we minimize over permutations, and on the right over doubly stochastic matrices. Because the terms are linear, and doubly stochastic matrices are convex combinations of permutations, the minima of the two linear terms are equal. Since $\hat{Q}$ is positive semidefinite, we thus want to find a doubly stochastic $X = \frac{1}{n}J + VYV^T$ with $y^T\hat{Q}y = 0$, which minimizes the linear term. Earlier in (4.6) we have seen that

$$\sum_{i=1}^{n} d_{i\varphi(i)} = \lambda_1 \frac{2m^2}{n} \quad \forall \varphi,$$

and is thus the linear term is constant. Hence, the term is also minimized for the average $X = \frac{1}{n}J$ of all permutations. For this X we have $Y = \frac{1}{n}V^TJV = 0$, and consequently $y^T\hat{Q}y = 0$. Thus, there is a feasible X of $QPB(A,B)$ with objective value $PB(A,B)$, and the Proposition follows since $QPB(A,B) \geq PB(A,B)$. □

Thus, it makes sense to consider the SDP-bound $SDPQAP(A,B)$ for the energy minimization QAP problem (4.1), if one wants to find stronger bounds than the BDL-bound used in [BDL13].

4.2 Reducing the Relaxation of the Energy Minimization Problem

In this section we exploit the symmetry of the SDP-bound SDPQAP in the case of the energy minimization problem. Recall from Section 4.1, and with reference to Figure 4.1, that this is a quadratic assignment problem given by A and B, where $B = (b_{ij})$ is indexed by toric grid points, and b_{ij} $(i \neq j)$ equals the inverse of the Lee distance (the shortest path on the grid) between grid points i and j, and $b_{ii} = 0$ for all i. The matrix A is zero except for a square block of all-ones in the upper left corner, of size equal to the number of particles on the toric grid. There are many approaches to exploit the symmetry of a conic optimization problem. Earlier work on group-theoretical symmetry reductions of SDP bounds for QAP was done in [KS10; KS12]. We chose to apply the Jordan Reduction method of Parrilo and Permenter [PP19] to our problem, which we will quickly summarize.

4.2.1 The Jordan Reduction

Parrilo and Permenter [PP19] introduced a set of three conditions a subspace has to fulfill, such that it is possible to use it for symmetry reduction. Here we revisit

just some of their results. A more detailed introduction, as well as an extension to the doubly nonnegative cone, is in Chapter 2.

Definition 4.6. A *projection* is a linear transformation $P\colon \mathcal{V} \to \mathcal{V}$ defined on a vector space $\mathcal{V}$ which is *idempotent*, i.e., $P^2 = P$. If $\mathcal{V}$ is equipped with an inner product, the projection is called *orthogonal*, if it is self-adjoint with respect to this inner product. We denote the orthogonal projection onto a subspace $\mathcal{L} \subset \mathcal{V}$ by $P_{\mathcal{L}}$.

We assume that the problem to be reduced is in the form

$$\left.\begin{array}{ll} \inf & \langle C, X\rangle \\ \text{s.t.} & X \in X_0 + \mathcal{L} \\ & X \in \mathbb{S}^n_+, \end{array}\right\} \tag{4.7}$$

where $X_0 \in \mathbb{R}^{n\times n}$, and $\mathcal{L}$ is a linear subspace of $\mathbb{R}^{n\times n}$.

Theorem 4.7 (Theorem 5.2.4 and Proposition 1.4.1 in [Per17])**.** *Consider the conic optimization problem* (4.7) *and let $S \subseteq \mathbb{R}^{n\times n}$ be a subspace of $\mathbb{R}^{n\times n}$. Define $C_{\mathcal{L}} = P_{\mathcal{L}}(C)$ and $X_{0,\mathcal{L}^\perp} = P_{\mathcal{L}^\perp}(X_0)$. If S fulfills*

(a) $C_{\mathcal{L}}, X_{0,\mathcal{L}^\perp} \in S$,

(b) $P_{\mathcal{L}}(S) \subseteq S$,

(c) $S \supseteq \{X^2 \ : \ X \in S\}$,

then restricting the feasible set of conic program (4.7) *to S results in another — potentially significantly smaller — program, with the same optimal value:*

$$\begin{array}{ll} \inf & \langle P_S(C), X\rangle \\ \text{s.t.} & X \in P_S(X_0) + \mathcal{L} \cap S, \\ & X \in \mathbb{S}^n_+ \cap S. \end{array}$$

We call such an S admissible for the problem (4.7).

We will use the concept of a Jordan algebra: for our purposes this will be a subspace of symmetric matrices that is closed under the product $X \bullet Y = \frac{1}{2}(XY + YX)$. It follows from condition (c) in the theorem that every admissible subspace is a Jordan algebra. Indeed, a subspace of symmetric matrices is a Jordan algebra if and only if it is closed under taking squares, due to the identity $X \bullet Y = \frac{1}{2}\left((X+Y)^2 - X^2 - Y^2\right)$. We will denote the full Jordan algebra of symmetric $n \times n$ matrices by $\mathbb{S}^n$.

Note that for the SDP-bound SDPQAP one would actually use the cone of entry-wise nonnegative matrices (i.e. doubly nonnegative matrices), as opposed to $\mathbb{S}^n_+$. To deal with the nonnegativity, we will in fact work with admissible subspaces that have 0-1 bases and where the basis matrices have disjoint support (so-called partition subspaces).

4.2.2 Symmetric Circulant Matrices

First, we need some well-known properties of (symmetric) circulant matrices, which will appear later in the construction of the admissible subspaces of the relaxation of the energy minimization problem.

Definition 4.8. An $n \times n$ matrix C is called *circulant*, if each row is rotated one element to the right relative to the row above, i.e., $C_{ij} = c_{j-i \bmod n}$ for all i, j and fitting c_k, $k = 0, \ldots, n-1$.

Proposition 4.9. *A symmetric circulant $n \times n$ matrix C has at most $\lfloor \frac{n}{2} \rfloor + 1 = \lceil \frac{n+1}{2} \rceil$ unique entries, and $c_k = c_{n-k}$.*

Proof. Let $j \geq i$ and $k = j - i$. By definition, we have $c_k = C_{ij} = C_{ji} = c_{n-(j-i)} = c_{n-k}$. Hence, C is given by $c_0, \ldots, c_{\lfloor \frac{n}{2} \rfloor}$. □

This allows us to construct symmetric circulant matrices from a given $c \in \mathbb{R}^{\lceil \frac{n+1}{2} \rceil}$. We call this function $C = \operatorname{circ}_n(c_0, \ldots, c_{\lfloor \frac{n}{2} \rfloor})$.

Proposition 4.10 (E.g., Theorem 7 in [Gra06])**.** *The product of two circulant matrices is a circulant matrix, and the product commutes. The product of symmetric circulant matrices is symmetric.*

We call the Jordan algebra (with product $X \bullet Y = \frac{1}{2}(XY + YX) = XY$) of symmetric circulant $n \times n$ matrices $\mathfrak{C}^n$. We define a 0/1-basis for $\mathfrak{C}^n$ by

$$\left\{ C_i^n = \operatorname{circ}_n(d_i) \ : \ i = 0, \ldots, \lfloor \frac{n}{2} \rfloor \right\}, \tag{4.8}$$

where for $i \notin \{0, \frac{n}{2}\}$ we set $d_i = e_i \in \mathbb{R}^{\lfloor \frac{n}{2} \rfloor + 1}$, the vector with a one in position i, and zero otherwise. For $i = 0$ and $i = \frac{n}{2}$, if n is even, we set $d_i = 2e_i$.

4.2.3 Admissible Subspaces

To reduce the program $SDPQAP(A,B)$ in (3.1) for the energy minimization problem (4.1), one should find an admissible subspace S for every such problem. In this case $\mathcal{L}$ is the subspace given by the $Y \in \mathbb{S}^{n^2}$ with

$$\begin{aligned}\langle I_n \otimes E_{jj}, Y\rangle &= 0 \text{ for } j \in [n],\\ \langle E_{jj} \otimes I_n, Y\rangle &= 0 \text{ for } j \in [n],\\ \langle T, Y\rangle &= 0,\\ \langle J_{n^2}, Y\rangle &= 0,\end{aligned}$$

where $T = I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n$, and X_0 is any symmetric matrix satisfying the linear constraints of the SDP (3.1), e.g., $X_0 = \mathrm{vec}(I_n)\mathrm{vec}(I_n)^T$.

Note here that we are missing the constraint that X is entry-wise nonnegative. But it is easy to check that if we restrict S to be a *partition* subspace (i.e., a subspace with an orthogonal 0/1-basis), then $P_S(X)$ is doubly-nonnegative if X is.

Recall that, for $n = n_1 n_2$, the matrix $B \in \mathbb{R}^{n\times n}$ is defined by $B_{(x_1,y_1),(x_2,y_2)} = 1/d_{\mathrm{Lee}}((x_1,y_1),(x_2,y_2))$, where d_{Lee} is the Lee-distance (length of the shortest path on the toric grid). The ordering of the indices $[n_1]\times[n_2] = [n]$ we left implicit in earlier sections of this chapter, but now we fix it to $(x,y) \mapsto n_2(x-1)+y$. $A \in \mathbb{R}^{n\times n}$ is the matrix with an $m \times m$ all-one block in the top left corner, and otherwise zero.

In this section we will make use of Tensor products of algebras. As a reminder, if $A_1,\ldots,A_{d_1} \in \mathbb{R}^{n_1\times n_1}$ is a basis of a matrix algebra $\mathcal{A}$, and $B_1,\ldots,B_{d_2} \in \mathbb{R}^{n_2\times n_2}$ a basis of a matrix algebra $\mathcal{B}$, then $\mathcal{A}\otimes\mathcal{B}$ is the $n_1n_2 \times n_1n_2$ matrix algebra with basis $A_i \otimes B_j$, for $i \in [d_1], j \in [d_2]$.

We restrict ourselves to a partition subspace, which means that the exact values of the entries of the matrix do not matter to us, only the pattern of unique elements. For the first of the three properties, we take a look at the structure of $\mathbf{C} \coloneqq B \otimes A$.

Lemma 4.11. *$B \in \mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_2}$, i.e., B is a block matrix, with n_1 rows and columns of blocks, which are arranged in a symmetric circulant pattern, and each of these blocks is an $n_2 \times n_2$ symmetric circulant matrix.*

Proof. The Lee-distance between (x_1,y_1) and (x_2,y_2) depends only on $x_2 - x_1 \mod n_1$ and $y_2 - y_1 \mod n_2$, and the order of the arguments do not matter. This

means that both the sub-matrices for fixed x and for fixed y coordinates are symmetric circulant matrices:

$$\left(B_{(i,y_1),(j,y_2)}\right)_{1\le i,j,\le n_1} \in \mathfrak{C}^{n_1}, \qquad \left(B_{(x_1,i),(x_2,j)}\right)_{1\le i,j,\le n_2} \in \mathfrak{C}^{n_2}.$$

The chosen ordering of the indices $(x,y) \mapsto n_2(x-1)+y$ thus results in $B \in \mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_2}$. □

In the case $n_1 = n_2$ we can restrict the algebra further.

Lemma 4.12. *If $n_1 = n_2$, then*

$$B \in \mathfrak{C}^{n_1,n_1} := \left\{X \in \mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_1} \;:\; X_{(x_1,y_1),(x_2,y_2)} = X_{(y_1,x_1),(y_2,x_2)}\right\}, \tag{4.9}$$

and $\mathfrak{C}^{n_1,n_1}$ is a Jordan subalgebra of $\mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_1}$.

Proof. B is has this symmetry by definition of the Lee-distance. $\mathfrak{C}^{n_1,n_1}$ is a subalgebra, because it is the restriction of an algebra to the commutant of $\{P, I\}$, where P is the $n \times n$ permutation matrix switching the indices corresponding to each (x,y) with the one corresponding to (y,x). □

The other relevant Jordan algebra for our problem is described in the following proposition.

Proposition 4.13. *The subspace of $n \times n$ matrices with pattern*

$$\begin{array}{c} \\ m \\ \\ \\ \\ n-m \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{cccc|cccc}
a & b & \cdots & b & c & \cdots & \cdots & c \\
b & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\vdots & \ddots & \ddots & b & \vdots & & & \vdots \\
b & \cdots & b & a & c & \cdots & \cdots & c \\
\hline
c & \cdots & \cdots & c & d & e & \cdots & e \\
\vdots & & & \vdots & e & \ddots & \ddots & \vdots \\
\vdots & & & \vdots & \vdots & \ddots & \ddots & e \\
c & \cdots & \cdots & c & e & \cdots & e & d
\end{array}
\end{array}$$

form a Jordan algebra, say $\mathcal{J}^{n,m}$. We call the $0/1$-basis corresponding to this pattern J_A, J_B, J_C, J_D, J_E.

Proof. A straightforward calculation shows that squaring such a matrix results in another matrix of the same pattern with parameters

$$\begin{aligned}
a' &= a^2 + (m-1)b^2 + (n-m)c^2,\\
b' &= 2ab + (m-2)b^2 + (n-m)c^2,\\
c' &= (a + (m-1)b)c + (d + (n-m-1)e)c,\\
d' &= d^2 + (n-m-1)e^2 + mc^2,\\
e' &= 2de + (n-m-2)e^2 + mc^2.
\end{aligned}$$

□

We now want to show that the space $S \coloneqq \mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_2} \otimes \mathcal{J}^{n,m}$, respectively $S = \mathfrak{C}^{n_1,n_1} \otimes \mathcal{J}^{n,m}$ if $n_1 = n_2$, is admissible. We do this by verifying the three conditions listed in Theorem 4.7.

Theorem 4.14. *The subspace $S \coloneqq \mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_2} \otimes \mathcal{J}^{n,m}$, respectively $S = \mathfrak{C}^{n_1,n_1} \otimes \mathcal{J}^{n,m}$ if $n_1 = n_2$ is admissible for* (3.1)*, where B and A are the matrices corresponding to the problem of minimizing the energy of m particles on an $n_1 \times n_2$ grid.*

For $2 < m < n-2$ the dimension of S is $5\lceil\frac{n_1+1}{2}\rceil\lceil\frac{n_2+1}{2}\rceil$ in the case $n_1 \neq n_2$, and $\frac{5}{2}\lceil\frac{n_1+1}{2}\rceil\left(\lceil\frac{n_2+1}{2}\rceil+1\right)$ in the case $n_1 = n_2$.

Proof. We first show that $P_{\mathcal{L}}(S) \subseteq S$. To this end, note that both $T = I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n$ and J_{n^2} are elements of S, since $J_n = J_{n_1} \otimes J_{n_2}$ and $I_n = I_{n_1} \otimes I_{n_2}$ are both in $\mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_2}$ (and in $\mathfrak{C}^{n_1,n_1}$ if $n_1 = n_2$), as well as in $\mathcal{J}^{n,m}$ because $I_n = J_A + J_D$ and $J_n = I_n + J_B + J_C + J_E$. Thus, T and J_{n^2} can be written as linear combination of Kronecker products of elements of $\mathfrak{C}^{n_1}$, $\mathfrak{C}^{n_2}$ and $\mathcal{J}^{n,m}$, and are as such elements of S.

The other two constraints are given by matrices $I_n \otimes E_{jj}$ and $E_{jj} \otimes I_n$, which only overlap with the two basis elements $C_0^{n_1} \otimes C_0^{n_2} \otimes J_A$ and $C_0^{n_1} \otimes C_0^{n_2} \otimes J_D$. Since $C_0^{n_1} \otimes C_0^{n_2} \otimes J_A = \sum_{j=1}^{m} I_n \otimes E_{jj}$ and $C_0^{n_1} \otimes C_0^{n_2} \otimes J_D = \sum_{j=m+1}^{n} I_n \otimes E_{jj}$, both of these matrices are projected to zero.

Thus, all basis elements of S are sent to elements of S, and $P_{\mathcal{L}}(S) \subseteq S$.

Next, we show $\mathbf{C}_{\mathcal{L}} \in S$. By Lemma 4.11, Lemma 4.12 and the definition of A, we know that $C = B \otimes A \in S$. Since $P_{\mathcal{L}}(S) \subseteq S$, that $\mathbf{C}_{\mathcal{L}} \in S$ as well.

Next, we show that $X_{0,\mathcal{L}^\perp} \in S$. To project $X_0 = \mathrm{vec}(I_n)\mathrm{vec}(I_n)^T$ onto $\mathcal{L}^\perp$, the span of the constraint matrices, we first notice only two of them have nonzero entries outside the diagonal, the all-one matrix J_{n^2}, and the matrix $I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n$, which we will call T from now on. The matrices $I_n \otimes E_{jj}$ for

$j = 1,\dots,n$ sum to the identity matrix I_{n^2}, which means that we can easily find an orthogonal basis of the off diagonal part of $\mathcal{L}^\perp$:

$$B_1 = T,$$
$$B_2 = J_{n^2} - I_{n^2} - T.$$

Since $\langle T, X_0\rangle = 0$ and $\langle J_{n^2}, X_0\rangle = n^2$ we get

$$\langle B_2, X_0\rangle = \langle J_{n^2}, X_0\rangle - \langle I_{n^2}, X_0\rangle = n^2 - n.$$

Hence, the off-diagonal part of $X_{0,\mathcal{L}^\perp}$ is the matrix

$$\frac{\langle B_2, X_0\rangle}{\langle B_2, B_2\rangle} B_2 = \frac{n^2-n}{n^4-n^2-2n(n^2-n)} B_2 = \frac{1}{n^2-n} B_2.$$

The diagonal part of $X_{0,\mathcal{L}^\perp}$ is the matrix $\frac{1}{n} I_{n^2}$, since

$$\langle E_{jj} \otimes I_n, \frac{1}{n} I_{n^2} - X_0\rangle = \langle E_{jj} \otimes I_n, \frac{1}{n} I_{n^2}\rangle - \langle E_{jj} \otimes I_n, X_0\rangle = 1 - 1 = 0,$$

and analogously $\langle I_n \otimes E_{jj}, \frac{1}{n} I_{n^2} - X_0\rangle = 0$. Combining the two parts we see

$$X_{0,\mathcal{L}^\perp} = \frac{1}{n^2-n} B_2 + \frac{1}{n} I_{n^2}.$$

Since J_{n^2}, I_{n^2} and T are elements of S, we get that $X_{0,\mathcal{L}^\perp} \in S$.

Finally, we note that S is a Jordan algebra. This completes the proof that S is admissible. The dimension of S follows from $\mathfrak{C}^n$ having dimension $\lceil \frac{n+1}{2} \rceil$ and $\mathcal{J}^{n,m}$ having dimension 5. In the case $n_1 = n_2$ the dimension is lower, since we can combine the basis elements $C_i^{n_1} \otimes C_j^{n_1}$ and $C_j^{n_1} \otimes C_i^{n_1}$ for each pair $i \neq j$. □

Thus, we have found an admissible subspace S for (3.1), where A and B are the matrices corresponding to the problem of minimizing the energy of m particles on an $n_1 \times n_2$ toric grid. Its dimension is of order $\mathcal{O}(n_1 n_2)$, which is significantly less than the original number of variables $\frac{n_1^4 n_2^4 + n_1^2 n_2^2}{2} = \mathcal{O}(n_1^4 n_2^4)$. The number of variables can be reduced further by fixing the variables corresponding to nonzero entries of $I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n$ to zero. Thus, if $\{B_1,\dots,B_k\}$ is a 0/1-basis of an admissible subspace, then it is enough to optimize over variables in the subspace S_0 with basis

$$\{B_i \ : \ \langle B_i, I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n\rangle = 0\}.$$

This results in $3\lceil\frac{n_1+1}{2}\rceil\lceil\frac{n_2+1}{2}\rceil-1$ variables in the case $n_1 \neq n_2$, and

$$1.5\lceil\frac{n_1+1}{2}\rceil\left(\lceil\frac{n_2+1}{2}\rceil+1\right)-1$$

variables in the case $n_1 = n_2$. A few examples can be seen in Table 4.1. Note that the resulting subspace is generally not a Jordan algebra anymore.

Table 4.1: Number of variables before and after symmetry reduction.

(n_1, n_2)	$\dim(\mathbb{S}^{n_1^2 n_2^2})$	$\dim(S)$	$\dim(S_0)$
$(4,4)$	32896	30	17
$(5,5)$	195625	30	17
$(6,6)$	840456	50	29
$(8,8)$	8390656	75	44
$(10,10)$	50005000	105	62
$(12,12)$	215001216	140	83
$(24,24)$	55037822976	455	272
$(100,100)$	$\approx 5\cdot 10^{15}$	6630	3977
$(1000,1000)$	$\approx 5\cdot 10^{23}$	628755	377252
$(6,5)$	405450	60	35
$(10,5)$	3126250	90	53
$(24,12)$	3439895040	455	272

4.2.4 Block-Diagonalization

We now want to block-diagonalize the admissible subspace $S \coloneqq \mathfrak{C}^{n_1} \otimes \mathfrak{C}^{n_2} \otimes \mathcal{J}^{n,m}$, respectively $S = \mathfrak{C}^{n_1,n_1} \otimes \mathcal{J}^{n,m}$ if $n_1 = n_2$. We do this by making use of the fact that S is a tensor product of algebras, which allows us to block-diagonalize each part on its own.

Lemma 4.15 (See, for example, [Gra06], [KPS08])**.** *The* 0/1*-basis* $\{C_i^n : i = 0,\dots,\lfloor\frac{n}{2}\rfloor\}$ *of* $\mathfrak{C}^n$ *has a common set of eigenvectors, given by the columns of the discrete Fourier transform matrix:*

$$Q_{ij}^n \coloneqq \frac{1}{\sqrt{n}} e^{-2\pi\sqrt{-1}ij/n}, \qquad i,j = 0,\dots,n-1.$$

The eigenvalues are

$$\lambda_m(C_k^n) = 2\cos(2\pi mk/n), \qquad m = 0,\dots,n-1, k = 0,\dots,\lfloor\frac{n}{2}\rfloor,$$

and note that

$$\lambda_m(C_k^n) = \lambda_{n-m}(C_k^n), \qquad m = 1, \ldots, \lfloor \frac{n}{2} \rfloor, k = 0, \ldots, \lfloor \frac{n}{2} \rfloor.$$

Thus, we can block-diagonalize $\mathfrak{C}^n$ by sending C_k^n to the vector

$$\hat{\lambda}(C_k^n) := (\lambda_0(C_k^n), \ldots, \lambda_{\lfloor \frac{n}{2} \rfloor}(C_k^n)).$$

To block-diagonalize $\mathcal{J}^{n,m}$, one may use the Jordan isomorphism $\phi : \mathcal{J}^{n,m} \to \mathbb{R} \oplus \mathbb{R} \oplus \mathbb{S}^2$ given by

$$\phi(J_A) = \begin{pmatrix} n-m & & \\ & \begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix} & \\ & & \end{pmatrix}, \ \phi(J_B) = \begin{pmatrix} -1 & & \\ & \begin{smallmatrix} m-1 & 0 \\ 0 & 0 \end{smallmatrix} & \\ & & \end{pmatrix}, \ \phi(J_C) = \sqrt{m(n-m)} \begin{pmatrix} 0 & & \\ & \begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix} & \\ & & \end{pmatrix},$$

$$\phi(J_D) = \begin{pmatrix} 0 & & \\ & \begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix} & \\ & & \end{pmatrix}, \ \phi(J_E) = \begin{pmatrix} 0 & & \\ & \begin{smallmatrix} -1 & 0 \\ 0 & n-m-1 \end{smallmatrix} & \\ & & \end{pmatrix}.$$

This isomorphism was used implicitly in [KOP11], but may also be verified directly by confirming that $\phi(X^2) = [\phi(X)]^2$ for all $X \in \mathcal{J}^{n,m}$.

We can now combine these block-diagonalizations by noticing that it is enough to block-diagonalize each of the algebras separately; see, for example, Section 7.2. in [KS10]. We obtain the final reduction shown in the next theorem. The proof is omitted since it is straightforward: One just has to calculate the inner products between the basis elements of the algebra and the data matrices, and eliminate variables fixed to zero by $\langle B_i, I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n \rangle = 0$. We then further scaled some variables and matrices to simplify terms further.

In the following we use the constants

$$d_{ij}^{kl} := \cos\left(\frac{2\pi ki}{n_1}\right) \cos\left(\frac{2\pi lj}{n_2}\right),$$

which arise from the diagonalization of the circulant matrices.

Theorem 4.16. *The bound from* (3.1)*, where the matrices A, B correspond to the energy minimization problem with parameters n_1, n_2, $n = n_1 n_2$ and m, equals the optimal value of the following semidefinite program:*

$$\inf \; n \sum_{i+j>0} \frac{y_{ij}^{b\to b}}{i+j} \tag{4.10}$$

$$\text{s.t.} \sum_{i+j>0} \left(y_{ij}^{b\to b} + y_{ij}^{w\to w} + y_{ij}^{b\leftrightarrow w} \right) = n-1, \tag{4.11}$$

$$\textit{for all } 0 \le k \le \left\lfloor \frac{n_1}{2} \right\rfloor, 0 \le l \le \left\lfloor \frac{n_2}{2} \right\rfloor :$$

$$\begin{pmatrix} \frac{m}{n} & 0 \\ 0 & \frac{n-m}{n} \end{pmatrix} + \sum_{i+j>0} d_{ij}^{kl} \begin{pmatrix} y_{ij}^{b\to b} & \frac{1}{2} y_{ij}^{b\leftrightarrow w} \\ \frac{1}{2} y_{ij}^{b\leftrightarrow w} & y_{ij}^{w\to w} \end{pmatrix} \succcurlyeq 0, \tag{4.12}$$

$$\frac{m(m-1)}{n} - \sum_{i+j>0} d_{ij}^{kl} y_{ij}^{b\to b} \ge 0, \tag{4.13}$$

$$\frac{(n-m)(n-m-1)}{n} - \sum_{i+j>0} d_{ij}^{kl} y_{ij}^{w\to w} \ge 0, \tag{4.14}$$

$$y_{ij}^{b\to b}, y_{ij}^{w\to w}, y_{ij}^{b\leftrightarrow w} \ge 0 \;\forall\; i+j>0, \;\; i = 0, \ldots, \left\lfloor \frac{n_1}{2} \right\rfloor, \;\; j = 0, \ldots, \left\lfloor \frac{n_2}{2} \right\rfloor.$$

We can interpret the variables as averaged occurrences of pairs of black points ($b \to b$), pairs of white points ($w \to w$), and pairs of a white and a black point ($b \to w$), at distance (i, j). I.e. if we have a given configuration, we can count how many pairs of points at distance (i, j) are both black, and then set $y_{ij}^{b\to b}$ to this value divided by the total number of pairs, to construct a feasible solution. Variables corresponding to distances bigger than $\left\lfloor \frac{n_1}{2} \right\rfloor$ respectively $\left\lfloor \frac{n_2}{2} \right\rfloor$ are looped around and added onto smaller distance variables, i.e. we may have variable values bigger than one.

The semidefinite program in Theorem 4.16 has block sizes of order at most 2×2, and is therefore a second-order cone program, which can be solved very efficiently; see e.g. [Lob+98]. Thus, we were able to solve the SDP relaxation for toric grids of sizes up to 100×100. Subsequently, we were also able to prove optimality of certain configurations of particles on toric grids, as detailed in Section 4.4.

4.3 Towards the Checkerboard Conjecture

In [BDL13] the authors conjecture the following.

Conjecture 4.17 (Checkerboard conjecture, [BDL13])**.** For any d-dimensional toric grid of size $n_1 \times n_2 \times \cdots \times n_d$, with $n_1,\ldots,n_d$ all even, and for any reasonable function f of the Lee distance, the arrangements of $m = \frac{1}{2}\prod_{i=1}^{d} n_i$ particles that minimize the maximal f-energy experienced by any of the particles are the two checkerboard patterns, and only these.

While the general case of the checkerboard conjecture is still open, the authors prove various cases in [BDL13], the strongest of which assumes that f is strictly completely monotonic, and each n_i is either 2 or a multiple of 4.

In this section we attempt to make some progress towards the general two-dimensional case for the fixed function $f = x^{-1}$, i.e. the energy is given by the inverse of the Lee distance and n_1 and n_2 are even.

We start by taking a look at the dual of SDP (4.10).

Corollary 4.18. *The bound from* (3.1)*, where the matrices A, B correspond to the energy minimization problem with parameters n_1, n_2, $n = n_1 n_2$ and m, equals the optimal value of the following semidefinite program:*

$$
\begin{aligned}
\sup\ & (n-1)z^{\Sigma} - \sum_{k,l}\left(\frac{m}{n}X_{11}^{kl} + \frac{n-m}{n}X_{22}^{kl}\right) \\
& \quad - \frac{m(m-1)}{n}e^t z^b - \frac{(n-m)(n-m-1)}{n}e^t z^w \\
\text{s.t. } & \textit{for all } 0 \le i \le \left\lfloor \frac{n_1}{2} \right\rfloor, 0 \le j \le \left\lfloor \frac{n_2}{2} \right\rfloor, i+j > 0\colon \\
& \quad z^{\Sigma} + \sum_{kl} d_{ij}^{kl}(X_{11}^{kl} - z_{kl}^b) \le \frac{n}{i+j}, \\
& \quad z^{\Sigma} + \sum_{kl} d_{ij}^{kl}(X_{22}^{kl} - z_{kl}^w) \le 0, \\
& \quad z^{\Sigma} + \sum_{kl} d_{ij}^{kl} X_{12}^{kl} \le 0, \\
& z^{\Sigma} \in \mathbb{R}_+, \\
& z^b, z^w \in \mathbb{R}_+^{\{(k,l) \ \mid \ k \in \{0,\ldots,\lfloor \frac{n_1}{2} \rfloor\},\ l \in \{0,\ldots,\lfloor \frac{n_2}{2} \rfloor\}\}}, \\
& X^{kl} \in \mathbb{S}_+^2 \quad \textit{for all } k \in \{0,\ldots,\left\lfloor \frac{n_1}{2} \right\rfloor\},\ l \in \{0,\ldots,\left\lfloor \frac{n_2}{2} \right\rfloor\}.
\end{aligned}
\tag{4.15}
$$

Proof. We dualized the symmetry reduced SDP (4.10). Here z^{Σ}, z^{b}, z^{w} and X^{kl} dualize (4.11), (4.13), (4.14) and (4.12) respectively. □

Here it is less clear how to interpret the variables. Numerically we observed that the optimal solution is not unique, but that there always is a "nice" optimal solution to the program in a sparse subspace. Since every feasible dual solution results in a valid bound, adding further constraints can only result in another, potentially weaker, valid bound. Since we do not have an argument here as to why we can add these constraints without making the bound worse, we see the following linear bound as a weaker version of the dual bound (4.15).

Corollary 4.19. *The following linear program is a lower bound for the total energy of m particles on a toric grid of size $n_1 \times n_2$:*

$$
\begin{aligned}
\sup\ & (m-1)z^{\Sigma} - \frac{m}{n}e^t x \\
\text{s.t. } & \textit{for all } 0 \le i \le \left\lfloor \frac{n_1}{2} \right\rfloor, 0 \le j \le \left\lfloor \frac{n_2}{2} \right\rfloor, i+j>0\text{:} \\
& \frac{n}{m}z^{\Sigma} + \sum_{k+l>0} d_{ij}^{kl} x_{kl} \le \frac{n}{i+j}, \\
& z^{\Sigma} \in \mathbb{R}_+, \\
& x \in \mathbb{R}_+^{\{(k,l) \ \mid \ k+l>0,\ \ k\in\{0,\dots,\lfloor \frac{n_1}{2} \rfloor\},\ \ l\in\{0,\dots,\lfloor \frac{n_2}{2} \rfloor\}\}}.
\end{aligned}
$$

Proof. We eliminate variables of the dual bound (4.15) by setting

$$
\begin{aligned}
z^{b} &= 0, \\
z^{w}_{00} &= \frac{n}{n-m}z^{\Sigma}, \\
z^{w}_{kl} &= 0 \quad \text{if } k+l>0, \\
X^{00} &= z^{\Sigma}\begin{pmatrix} \frac{n-m}{m} & -1 \\ -1 & \frac{m}{n-m} \end{pmatrix}, \\
X^{kl}_{ij} &= 0 \quad \text{if } k+l>0 \text{ and } i+j>0.
\end{aligned}
$$

□

Remark 4.20. Numerically this bound seems to be exactly the same as the original bound. Since we did not find a proof of this, we conjecture that this bound is equivalent to the bound obtained from (3.1).

Now we want to construct an optimal solution for this problem in the case $n_1 = 2k_1, n_2 = 2k_2, m = \frac{n}{2}$ to prove the Checkerboard Conjecture. For this, we first add a redundant constraint $\frac{n}{m}z^\Sigma + e^t x = \frac{n}{m}z^\Sigma + e^t x$, which allows us to rewrite the program.

$$\sup \ (m-1)z^\Sigma - \frac{m}{n}e^t x \tag{4.16}$$

$$\text{s.t.} \ \left(D_{n_1} \otimes D_{n_2}\right)\begin{pmatrix} \frac{n}{m}z^\Sigma \\ x \end{pmatrix} \leq \begin{pmatrix} \frac{n}{m}z^\Sigma + e^t x \\ b \end{pmatrix}, \tag{4.17}$$

$$z^\Sigma, x \geq 0$$

where

$$D_t = \left(\cos\left(\frac{2\pi ij}{t}\right)\right)_{i,j=0,\ldots,\lfloor \frac{t}{2} \rfloor}$$

and

$$b = \left(\frac{n}{0+1}, \ldots, \frac{n}{0+k_2}, \frac{n}{1+0}, \ldots, \frac{n}{1+k_2}, \ldots, \frac{n}{k_1+k_2}\right)^t.$$

Numerically we observed that in the checkerboard case $m = \frac{n}{2}$ we can always obtain an optimal solution if we assume that the inequality constraint (4.17) is sharp. The inverse of D_t is

$$D_t^{-1} = \frac{2}{t}\left((D_t)_{ij} c^t(i) c^t(j)\right)_{ij},$$

where

$$c^t(i) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } i \in \{0, \frac{t}{2}\} \\ \sqrt{2}, & \text{otherwise.} \end{cases}$$

Assuming equality we have

$$\begin{pmatrix} \frac{n}{m}z^\Sigma \\ x \end{pmatrix} = \left(D_{n_1}^{-1} \otimes D_{n_2}^{-1}\right)\begin{pmatrix} \frac{n}{m}z^\Sigma + e^t x \\ b \end{pmatrix}, \tag{4.18}$$

and writing out the first row allows us to eliminate the x variable completely.

$$\begin{aligned} \frac{n}{m}z^\Sigma &= \frac{4}{n}\left(\frac{1}{4}\left(\frac{n}{m}z^\Sigma + e^t x\right) + \frac{1}{2}C^{n_1,n_2}\right), \\ &= \frac{1}{m}z^\Sigma + \frac{1}{n}e^t x + \frac{2}{n}C^{n_1,n_2} \\ \Rightarrow e^t x &= n\left(\frac{n}{m}z^\Sigma - \frac{1}{m}z^\Sigma - \frac{2}{n}C^{n_1,n_2}\right) \end{aligned}$$

$$= \frac{n^2-n}{m} z^\Sigma - 2C^{n_1,n_2}$$
$$= (2n-2)z^\Sigma - 2C^{n_1,n_2}$$

where

$$C^{n_1,n_2} := \sum_{\substack{i=0,\dots,k_1 \\ j=0,\dots,k_2 \\ i+j>0}} c^{n_1}(i)c^{n_2}(j)\frac{n}{i+j}.$$

Note that C^{n_1,n_2} is exactly the potential energy in the case that there is a particle on every spot on the grid. Substituting this for $e^T x$ and assuming equality in (4.18) results in the univariate linear program

$$\begin{aligned} \sup \ & -\frac{n}{2}z^\Sigma + C^{n_1,n_2} \\ \text{s.t.} \ & \left(D_{n_1}^{-1} \otimes D_{n_2}^{-1}\right)\begin{pmatrix} 2nz^\Sigma - 2C^{n_1,n_2} \\ b \end{pmatrix} \geq 0, \\ & z^\Sigma \geq 0. \end{aligned} \tag{4.19}$$

Note that every feasible solution still results in a valid lower bound of the energy minimization problem (in the case $m = n/2$), as we are only making the dual more strict.

Let us take a look now at the constraints in (4.19). The first row is redundant, as we know it is exactly $\frac{n}{m}z^\Sigma = 2z^\Sigma \geq 0$. The other constraints are of the form

$$nz^\Sigma - C^{n_1,n_2} + \sum_{\substack{i=0,\dots,k_1 \\ j=0,\dots,k_2 \\ i+j>0}} c^{n_1}(i)c^{n_2}(j)\frac{n}{i+j}\cos\left(\frac{2\pi ki}{n_1}\right)\cos\left(\frac{2\pi lj}{n_2}\right) \geq 0$$

for $k = 0,\dots,k_1$, $l = 0,\dots,k_2$, $k+l > 0$, if we rescale the constraints to make the coefficient of z^Σ the same in all constraints. The only difference between the different constraints are the cosine terms, and all coefficients in the sum are nonnegative.

We can make this problem more clear by bringing the objective term to one side in each of the constraints to obtain

$$-\frac{n}{2}z^\Sigma + C \leq \frac{1}{2}\left(C^{n_1,n_2} + \sum_{\substack{i=0,\dots,k_1 \\ j=0,\dots,k_2 \\ i+j>0}} c^{n_1}(i)c^{n_2}(j)\frac{n}{i+j}\cos\left(\frac{2\pi ki}{n_1}\right)\cos\left(\frac{2\pi lj}{n_2}\right)\right)$$

$$= \frac{1}{2} \sum_{\substack{i=0,\dots,k_1 \\ j=0,\dots,k_2 \\ i+j>0}} c^{n_1}(i) c^{n_2}(j) \frac{n}{i+j} \left(1 + \cos\left(\frac{2\pi k i}{n_1}\right) \cos\left(\frac{2\pi l j}{n_2}\right)\right).$$

The sum here describes exactly the energy of a "fractional", but periodic set of particles with weight

$$\frac{1}{2} + \frac{1}{2} \cos\left(\frac{2\pi k i}{n_1}\right) \cos\left(\frac{2\pi l j}{n_2}\right) \tag{4.20}$$

in position (i, j). I.e., the minimum energy of these fractional solutions is a lower bound for the minimum energy in the system. Note that while we were optimizing over $\binom{n_1 n_2}{m}$ possible choices of particle positions before, there are now just $k_1 k_2 - 1$ possibilities. While this bound is generally weaker than the SDP bound we started with, this one seems to be sharp in the checkerboard case $m = \frac{n}{2}$.

Indeed, in the case $k = k_1$ and $l = k_2$ this (fractional) set of particles is exactly the checkerboard solution

$$\frac{1}{2} + \frac{1}{2}(-1)^{i+j}.$$

What remains to be shown is that this is really the minimum energy between the different fractional solutions for $k = 0, \dots, k_1$, $l = 0, \dots, k_2$, $k + l > 0$.

While we do not manage to prove that the checkerboard is the optimal configuration of the fractional solutions described by (4.20), this approach reduces the (numerically sharp) bound for the two-dimensional case of Conjecture 4.17 for a fixed potential function to a search over just $\mathcal{O}(n_1 n_2)$ fractional solutions. It seems reasonable to expect that, with some more work, optimality of the checkerboard can be proven this way. It remains to be seen whether this approach, while stronger than the bounds used in [BDL13], is strong enough to prove the conjecture for arbitrary potential functions (as the approach of [BDL13] seems to fail there). As we will see in Section 4.4 our bound is sharp for more general particle configurations other than $m = \frac{n}{2}$, which may lead to a more general conjecture about lattice-like particle arrangements on toric grids.

4.4 Numerical Results

Here we compare the eigenvalue-bound with the SDP-bound for the energy minimization problems. The upper bounds were found using simulated annealing (see Algorithm 4).

Algorithm 4: Simulated annealing algorithm

$iter \leftarrow$ number of iterations to perform
$P \leftarrow$ random configuration of m particles
$val \leftarrow$ energy of P $= E(P)$
$T \leftarrow 1$
$a \leftarrow \sqrt[iter]{\frac{1}{iter}}$
for $it \leftarrow 1$ **to** *iterations* **do**
  $T \leftarrow aT$
  $P' \leftarrow P$
  Move a random particle of P' to a neighboring position
  if $E(P') < val$ *or* $\exp(-(E(P') - val)/T) \geq \text{rand}(0,1)$ **then**
    $val \leftarrow E(P')$
    $P \leftarrow P'$
return val

Calculating the SDP-bound directly is prohibitively slow, which is why we exploited the symmetries of the problems first, as described in Section 4.2. After this reduction we can calculate these bounds very efficiently, solving a case of the problem on a 10×10-grid in 0.2s, 1.3s on a 50×50-grid, and in about 40s on 100×100-grid, using Mosek on a 4-core 3.4GHz Processor. In [BDL13] Bouman, Draisma and van Leeuwaarden prove optimality for the checkerboard arrangement in the cases that $n_1 = n_2$ are even, and $m = \frac{n_1 n_2}{2}$. This can be seen for the grid sizes we checked as well, but we do get some more cases where the bound is sharp.

If one of the bounds is sharp, then we get a proof of optimality for these parameters, if we can construct the exact optimal solution. Furthermore, we can even prove optimality in some cases even if the bound is not completely sharp, as explained in the following Proposition. This allows us to work around small numerical rounding issues in some cases, by rounding to an exact rational solution close enough to the optimum.

Proposition 4.21. *Let $V = \{v_1, \ldots, v_k\}$ be the set of unique entries of the matrix of potentials B. The one-dimensional shortest-vector-problem for these values is*

$$\begin{aligned} r = \min\ & |s - t| \\ \text{s.t. } & s, t \in \left\{ x = 2 \sum_{i=1}^{k} \alpha_i v_i \ \mid\ \alpha_i \in \mathbb{Z} \right\} \\ & s \neq t. \end{aligned}$$

If a feasible solution of the QAP has objective value that differs by less than r from the SDP lower bound, then this feasible solution is optimal.

Proof. Both the matrix A and the optimization variable $X \in \Pi_n$ of this QAP are symmetric 0/1-matrices, and B is symmetric as well, with zeros on the main diagonal, which means that the objective value is of the form $\sum_{i,j,k,l=1}^{n} a_{ik} b_{jl} X_{ij} X_{kl} = \sum_{i,j=1,i<j}^{n} 2\alpha_{ij} b_{ij}$, where $\alpha_{ij} \in \mathbb{Z}$. Hence, different objective values have to at least differ by r. □

For the inverse Lee-distance potential on a 6×6-grid the set V is

$$\left\{0, 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}\right\},$$

and $r = 2\left(\frac{1}{6} + \frac{3}{5} - \frac{3}{4}\right) = \frac{1}{30}$, which is optimal since 30 is the least common denominator of the fractions in $2V$. Similarly, one finds $r = \frac{1}{210}$ for a 7×7-grid, $r = \frac{1}{420}$ for a 8×8-grid and $r = \frac{1}{1260}$ on a 10×10-grid. In the Tables 4.2,4.3, 4.4 and 4.5 we give the bounds for square grids of sizes 6,7,8 and 10 respectively. As proven in Proposition 4.1, we only need to consider $m \leq \frac{n}{2}$. Bold font in the tables signify sharp bounds, in the sense of Proposition 4.21, which we then illustrate in Figures 4.2, 4.3, 4.4 and 4.5.

Note that we could find several new optimal configurations by computing the SDP bound $SDPQAP(A, B)$, for example the case $m = 20$ on a 10×10 grid. We do not include results for the weaker SDP bound $SDPPB(A, B)$ in the tables, since these turned out to equal the projected eigenvalue bound for small instances. We do not know if these bounds coincide in general, though.

In these small cases we noticed that the bound is sharp only in (some) cases where m divides $n_1 n_2$, and are, except in the case $m = 4$ for some choices of n_1 and n_2, given by lattices. In these cases the nonzero variables $y_{ij}^{b \to b}$ do actually hint as to how the lattice can be constructed. For example in the case $n_1 = n_2 = 10$, $m = 20$ the optimal solution has $y_{ij}^{b \to b} = 0$ except for $y_{05}^{b \to b} = y_{55}^{b \to b} = 0.2$ and $y_{12}^{b \to b} = y_{13}^{b \to b} = y_{24}^{b \to b} = y_{34}^{b \to b} = 0.4$, and their symmetric counterparts. Compare these to the drawing in Figure 4.5: These are exactly the distances appearing in the drawing, namely $(0, 5)$, $(5, 5)$, $(1, 2)$, $(1, 3)$, $(2, 4)$, and $(3, 4)$. We can use this knowledge to construct the solution as follows. The size of the orbit of a single point, repeatedly shifting it in direction (i, j), is $\operatorname{lcm}\left(\frac{\operatorname{lcm}(n_1, i)}{i}, \frac{\operatorname{lcm}(n_2, j)}{j}\right)$, where $\operatorname{lcm}(a, b)$ denotes the least common multiple of a and b. We say that two lattice directions are *distinct*, if the corresponding orbits of the same point overlap only in that point. In this example the orbit sizes are 2 for the directions $(0, 5)$ and

$(5,5)$, 5 for $(2,4)$, and 10 for $(1,2)$, $(1,3)$ and $(3,4)$. If we now choose pairwise distinct orbits of orbit sizes factoring m, we can reconstruct the lattice solution by taking repeated orbits. In this case we can construct the solution in Figure 4.5 for $n_1 = n_2 = 10$ and $m = 20$ by choosing the triple of generators $\{(0,5),(5,5),(2,4)\}$ or any of the pairs $\{(1,2),(0,5)\}, \{(1,2),(5,5)\}, \{(1,3),(0,5)\}, \{(3,4),(5,5)\}$. Alternatively we can find a second solution (which is the same, but mirrored), by choosing swapping the two axes of each generator.

This way one can rapidly find cases where the bound is sharp, at least numerically. We do not get formal mathematical proofs for these cases in the sense of Proposition 4.21, as we cannot make the numerical gap small enough. In about 30 minutes we were able to identify 2382 such cases (of the 15083 cases where m divides $n_1 n_2$), from grid sizes 1×1 to 50×50. We list a few such cases in Figure 4.6.

4.5 Concluding Remarks

We have been able to use semidefinite programming bounds to identify several new minimum energy configurations of repulsing particles on a toric grid, as shown in Figure 4.6, for example. The next step would be to use our insight on optimal lattice configurations that we obtained numerically, to prove optimality for the generalized families of corresponding lattice configurations, similar to the way the authors of [BDL13] proved optimality of certain 'checkerboard' configurations. While even the general checkerboard case is still open, we constructed parts of the optimal solution for these cases in 4.3, which is numerically sharp, but did not manage to prove sharpness. Investigating this and similar cases remains a topic for future research.

While we only explicitly considered the potential energy function given by the inverse of the Lee metric, all results in this chapter generalize to a much wider family of functions in a straightforward manner. It may be of interest to attempt to prove *universal optimality* of certain configurations for all (reasonable) potential functions, similar to what was done in [CK06] for points on spheres. Another interesting question is whether one could obtain stronger bounds than the ones considered in [CK06] based on the QAP formulation for both the sphere and Euclidean space, and how the bounds [CK06] fit between the three bounds considered in this chapter. If one considers the limit of the linear program in Corollary 4.19 as the grid size approaches infinity, one would obtain an infinite dimensional

LP similar to [CK06], but with constraints and variables corresponding to each appearing vector between points, not just their distances.

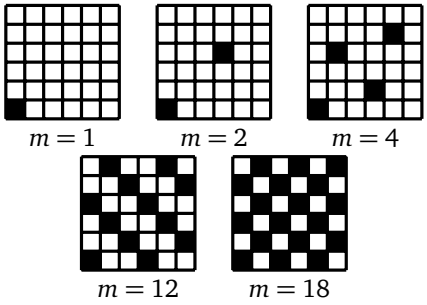


Figure 4.2: Optimal arrangements on a 6×6 grid

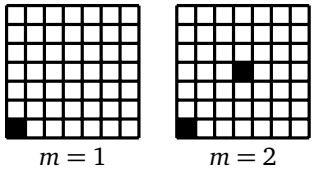


Figure 4.3: Optimal arrangements on a 7×7 grid

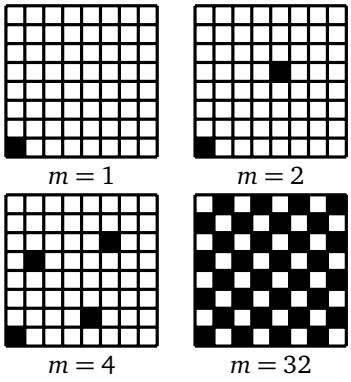


Figure 4.4: Optimal arrangements on a 8×8 grid

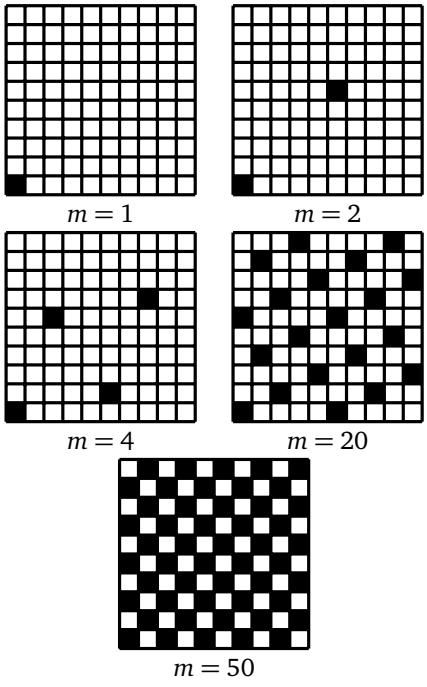


Figure 4.5: Optimal arrangements on a 10×10 grid

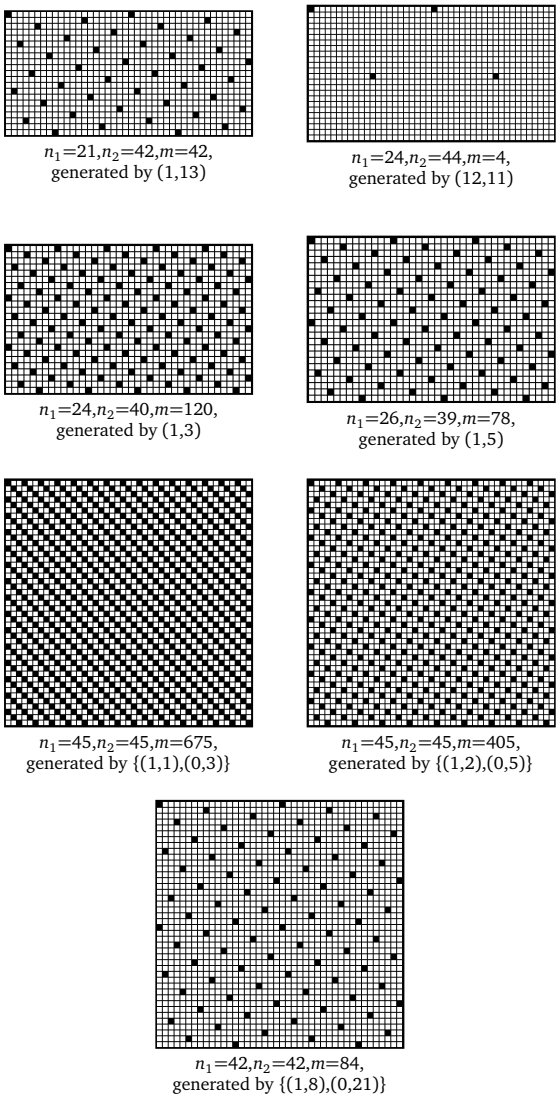


Figure 4.6: A small selection of cases where the SDP bound is sharp for larger grids.

m	$PB(A,B)$	$SDPQAP(A,B)$	Upper bounds from simulated annealing
1	-1.514815	**0.000000**	**0.000000**
2	-2.125926	**0.333319**	**0.333333**
3	-1.833333	1.349939	1.500000
4	-0.637037	**2.999892**	**3.000000**
5	1.462963	5.416640	5.666667
6	4.466667	8.599983	8.666667
7	8.374074	12.622685	13.000000
8	13.185185	17.407305	17.600000
9	18.900000	22.937178	23.466667
10	25.518519	29.212957	29.666667
11	33.040741	36.233780	36.666667
12	41.466667	**44.000000**	**44.000000**
13	50.796296	53.065277	54.366667
14	61.029630	62.959998	64.666667
15	72.166667	73.687450	75.500000
16	84.207407	85.273263	86.666667
17	97.151852	97.718432	98.666667
18	**111.000000**	**111.000000**	**111.000000**

Table 4.2: The bounds on a 6x6 grid

m	$PB(A,B)$	$SDPQAP(A,B)$	Upper bounds from simulated annealing
1	-1.535637	**0.000000**	**0.000000**
2	-2.287844	**0.333330**	**0.333333**
3	-2.256623	1.243763	1.300000
4	-1.441972	2.723982	2.800000
5	0.156109	4.784851	4.866667
6	2.537619	7.533726	7.800000
7	5.702558	10.916369	10.966667
8	9.650926	15.043550	15.500000
9	14.382724	19.814560	20.366667
10	19.897950	25.325560	25.900000
11	26.196607	31.554779	32.166667
12	33.278692	38.455887	39.033333
13	41.144207	46.029212	46.733333
14	49.793151	54.274568	54.933333
15	59.225525	63.260772	64.433333
16	69.441328	73.172931	74.100000
17	80.440560	83.797024	85.200000
18	92.223221	95.162225	96.600000
19	104.789312	107.298752	109.033333
20	118.138832	120.154716	122.000000
21	132.271782	133.732016	134.866667
22	147.188160	148.029982	150.066667
23	162.887969	163.048746	165.700000
24	179.371206	179.371185	182.266667

Table 4.3: The bounds on a 7x7 grid

m	$PB(A,B)$	$SDPQAP(A,B)$	Upper bounds from simulated annealing
1	-1.670238	**0.000000**	**0.000000**
2	-2.654762	**0.249994**	**0.250000**
3	-2.953571	1.014038	1.133333
4	-2.566667	**2.266433**	**2.266667**
5	-1.494048	4.062460	4.233333
6	0.264286	6.435304	6.583333
7	2.708333	9.423375	9.666667
8	5.838095	12.965443	13.000000
9	9.653571	17.078833	17.442857
10	14.154762	21.749475	22.126190
11	19.341667	26.990007	27.628571
12	25.214286	32.848535	33.666667
13	31.772619	39.445606	40.352381
14	39.016667	46.636662	47.350000
15	46.946429	54.421402	54.950000
16	55.561905	62.799758	63.076190
17	64.863095	71.971752	72.921429
18	74.850000	81.768179	83.023810
19	85.522619	92.238774	93.535714
20	96.880952	103.368587	104.300000
21	108.925000	115.126506	116.114286
22	121.654762	127.522322	128.483333
23	135.070238	140.541217	141.719048
24	149.171429	154.193846	155.514286
25	163.958333	168.487184	170.390476
26	179.430952	183.448522	185.711905
27	195.589286	199.055388	201.416667
28	212.433333	215.278915	217.333333
29	229.963095	232.135382	234.083333
30	248.178571	249.661718	251.083333
31	267.079762	267.846258	268.750000
32	286.666667	**286.666665**	**286.666667**

Table 4.4: The bounds on a 8x8 grid

m	$PB(A,B)$	$SDPQAP(A,B)$	Upper bounds from simulated annealing
1	-1.716889	**0.000000**	**0.000000**
2	-2.883429	**0.199988**	**0.200000**
3	-3.499619	0.811542	0.904762
4	-3.565460	**1.807631**	**1.809524**
5	-3.080952	3.233819	3.333333
6	-2.046095	5.131049	5.233333
7	-0.460889	7.479826	7.700000
8	1.674667	10.307204	10.355556
9	4.360571	13.579004	13.800000
10	7.596825	17.298929	17.433333
11	11.383429	21.467188	21.876190
12	15.720381	26.234485	26.679365
13	20.607683	31.472262	32.029365
14	26.045333	37.183990	37.744444
15	32.033333	43.378553	44.096825
16	38.571683	50.037058	50.736508
17	45.660381	57.183049	57.922222
18	53.299429	64.786226	65.342857
19	61.488825	72.867194	73.285714
20	70.228571	**81.428359**	**81.428571**
21	79.518667	90.810097	91.739683
22	89.359111	100.699416	102.068254
23	99.749905	111.093201	112.648413
24	110.691048	121.990693	123.492857
25	122.182540	133.400133	134.938889
26	134.224381	145.304277	146.824603
27	146.816571	157.729721	159.401587
28	159.959111	170.652552	172.295238
29	173.652000	184.080092	185.607937
30	187.895238	198.017798	199.388095
31	202.688825	212.458779	214.014286
32	218.032762	227.417991	228.861111
33	233.927048	242.967654	244.254762
34	250.371683	259.050406	260.466667

m	$PB(A,B)$	$SDPQAP(A,B)$	Upper bounds from simulated annealing
35	267.366667	275.649622	277.788889
36	284.912000	292.759839	295.125397
37	303.007683	310.386054	312.823810
38	321.653714	328.520878	331.160317
39	340.850095	347.242354	349.413492
40	360.596825	366.466726	368.701587
41	380.893905	386.186323	389.195238
42	401.741333	406.463567	410.162698
43	423.139111	427.236004	431.381746
44	445.087238	448.524441	452.888889
45	467.585714	470.378907	474.000000
46	490.634540	492.906204	496.033333
47	514.233714	515.926534	518.650000
48	538.383238	539.531651	541.466667
49	563.083111	563.667331	564.800000
50	**588.333333**	**588.332003**	**588.333333**

Table 4.5: The bounds on a 10x10 grid

5

Optimizing Hypergraph-Based Polynomials Modeling Job-Occupancy in Queuing with Redundancy Scheduling

In this chapter we consider the minimization of two classes of polynomials over the standard simplex. These polynomials have their variables labeled by the edges of a complete uniform hypergraph and their coefficients are defined in terms of some cardinality patterns of unions of edges. They arise naturally within the modeling of job-occupancy in some queuing problems with redundancy scheduling policies [CBL21]. The question is whether these polynomials attain their minimum value at the barycenter of the standard simplex, which corresponds to showing optimality of the uniform distribution for the underlying queuing problem. This chapter is devoted to this question, which is conjectured to be true by [CBL21]. While in all the other chapters we use symmetry reduction to reduce the size of a symmetric optimization problem, we here use it as a tool to show that a symmetric polynomial is convex.

5.1 Introduction

We now introduce the classes of polynomials of interest. Given integers $n, L \geq 2$ we set $V = [n] = \{1, \ldots, n\}$ and $E = \{e \subseteq V : |e| = L\}$, so that (V, E) can be seen as the complete L-uniform hypergraph on n elements. We set $m := |E| = \binom{n}{L}$, where we omit the explicit dependence on n, L to simplify notation, and we let

$$\Delta_m = \left\{ x = (x_e)_{e \in E} \in \mathbb{R}^m : x \geq 0, \sum_{e \in E} x_e = 1 \right\}$$

denote the standard simplex in $\mathbb{R}^m$. The elements of Δ_m correspond to probability vectors on m items and the barycenter $x^* = \frac{1}{m}(1, \ldots, 1)$ of Δ_m corresponds to the uniform probability vector.

Given an integer $d \geq 2$ we consider the following m-variate polynomial in the variables $x = (x_e : e \in E)$, which is a main player in the chapter:

$$f_d(x) = \sum_{(e_1, \ldots, e_d) \in E^d} \prod_{i=1}^{d} \frac{x_{e_i}}{|e_1 \cup \ldots \cup e_i|}. \tag{5.1}$$

So f_d is a homogeneous polynomial with degree d. We are interested in the following optimization problem

$$f_d^* := \min_{x \in \Delta_m} f_d(x),$$

asking to minimize the polynomial f_d over the simplex Δ_m. The main conjecture, which is stated in [CBL21], claims that the minimum is attained at the uniform probability.

Conjecture 5.1. Given integers $n, d, L \geq 2$, is the polynomial $f_d(x)$ in (5.1) attains its minimum over Δ_m at the barycenter x^* of Δ_m.

As explained in [CBL21], the motivation for this conjecture comes from its relevance to a problem in queuing theory, that we will briefly describe in the next section. In this chapter we are only able to give a partial positive answer to this conjecture, namely, in the case $d = 2$ (which follows from Theorem 5.2 below) and in the case $d = 3$ and $L = 2$ (Theorem 5.3 below). As a first step toward understanding the polynomials f_d we investigate a related, easier to analyze, class of polynomials.

Given an integer $d \geq 2$ we consider the following related class of polynomials

$$p_d(x) = \sum_{(e_1, \ldots, e_d) \in E^d} \frac{1}{|e_1 \cup \ldots \cup e_d|} x_{e_1} \cdots x_{e_d}, \tag{5.2}$$

which are also homogeneous with degree d. Note that, for degree $d = 2$, we have $f_2 = \frac{1}{L}p_2$. For degree $d \geq 3$ the polynomials f_d have a related, but more complicated structure than the polynomials p_d (see Section 5.4 for more details on the links between both classes). Here too we may ask whether the minimum of p_d over the standard simplex Δ_m is attained at the uniform probability vector x^*. For the polynomials p_d we are able to give a positive answer in the general case. The following is the first main result of the chapter.

Theorem 5.2. *For any integers $n, L, d \geq 2$, the global minimum of the polynomial p_d from (5.2) over the standard simplex Δ_m is attained at the barycenter $x^* = \frac{1}{m}(1, \ldots, 1)$ of Δ_m.*

As noted above, f_2 and p_2 coincide up to positive scaling, hence, it follows directly that Question 5.1 has a positive answer in the case $d = 2$. As a further partial result we give a positive answer for the case of degree $d = 3$ and edge size $L = 2$. The following is the second main result.

Theorem 5.3. *For $n \geq 2$, $d = 3$ and $L = 2$, the global minimum of the polynomial f_d from (5.1) over the standard simplex Δ_m is attained at the barycenter $x^* = \frac{1}{m}(1, \ldots, 1)$ of Δ_m.*

As we will see, the analysis of the polynomials f_d is technically much more involved than for the polynomials p_d, and we have only partial results so far. In both cases the key ingredient is showing that the polynomials are convex on the simplex, i.e., that they have positive semidefinite Hessians at any vector in Δ_m. It turns out that the Hessian of the polynomial p_d enters some way as a component of the Hessian of the polynomial f_d. So this forms a natural motivation for the study of the polynomials p_d, though they form a natural class of symmetric polynomials that are interesting for their own sake.

Exploiting symmetry plays a central role in our proofs. Indeed, the key idea is to show that the polynomials are convex, which, combined with their symmetry properties, implies that the global minimum is attained at the barycenter of the simplex. For this we show that their Hessian matrices are positive semidefinite at each point of the simplex, which we do through exploiting again their symmetry structure and links to Terwilliger algebras.

Symmetry is a widely used ingredient in optimization, in particular in semidefinite optimization and algebraic questions involving polynomials. We mention a few landmark examples as background information. Symmetry can indeed be used to formulate equivalent, more compact reformulations for semidefinite

programs. The underlying mathematical fact is Artin-Wedderburn theory, which shows that matrix $*$-algebras can be block-diagonalized (see Theorem 5.7 below). An early well-known example is the linear programming reformulation from [Sch79] for the Theta number of Hamming graphs, showing the link to the Delsarte bound and Bose-Mesner algebras of Hamming schemes [Del73; DL98]. Symmetry is used more generally to give tractable reformulations for the semidefinite bounds arising from the next levels of Lasserre's hierarchy in [Sch05] (which gives the explicit block-diagonalization for the Terwilliger algebra of Hamming schemes, see Theorem 5.8 below) and, e.g., in [GST08], [GMS12], [Lau07], [LPS17]. For more examples and a broad exposition about the use of symmetry in semidefinite programming we refer, e.g., to [Bac+11; KPS07] and further references therein. Symmetry is also a crucial ingredient in the study of algebraic questions about polynomials, like representations in terms of sums-of-squares, and in polynomial optimization, as we will explain in detail in Chapter 7. We refer to [GP04] for a broad exposition and, e.g., to [Rie+13] (for compact reformulations of Lasserre relaxations of symmetric polynomial optimization problems), [Rie12] (for methods to reduce the number of variables in programs involving symmetric polynomials), and the recent works [Ray+18; RST18] (which consider symmetric polynomials with variables indexed by the k-subsets hypercube (as in our case) and uncover links with the theory of flag algebras by Razborov [Raz07]).

Example 5.4. As an illustration let us consider the polynomial p_d for edge size $L = 2$. Given a sequence $\underline{e} = (e_1, \dots, e_d) \in E^d$ set $c_{\underline{e}} = 1/|e_1 \cup \dots \cup e_d|$ as a shorthand for the coefficients in the definition (5.2) of the polynomial p_d. So we need to enumerate the possible configurations of d-tuples of edges, i.e., the distinct multigraphs with d edges. Note that their number is given by the OEIS sequence A050535 [OEI99a], which takes the values $1, 3, 8, 23, 66, 212, 686$ for $d = 1, 2, 3, 4, 5, 6, 7$.

For $d = 1$, we have $p_1(x) = \frac{1}{2}\sum_{e\in E} x_e$. For $d = 2$ we have

$$p_2(x) = \frac{1}{2}\sum_{e\in E} x_e^2 + \frac{1}{3}\sum_{\substack{(e_1,e_2)\in E^2:\\ |e_1\cup e_2|=3}} x_{e_1}x_{e_2} + \frac{1}{4}\sum_{\substack{(e_1,e_2)\in E^2:\\ |e_1\cup e_2|=4}} x_{e_1}x_{e_2}.$$

We show in Figure 5.1 the three possible patterns for pairs of edges $\underline{e} = (e_1, e_2)$ and the corresponding coefficients $c_{\underline{e}}$.

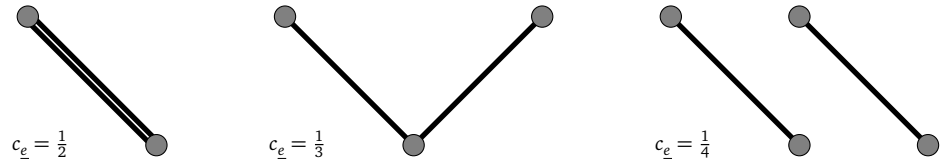


Figure 5.1: The three patterns of pairs of edges in case $(d = 2, L = 2)$

In the same way, for $d \geq 3$, $p_d(x) = \sum_{k=2}^{2d} \frac{1}{k} q_{d,k}(x)$, where the summand $q_{d,k}(x)$ is a summation over all d-tuples of edges with a given pattern, depending on the cardinality of their union:

$$q_{d,k}(x) = \sum_{\substack{(e_1,\ldots,e_d)\in E^d: \\ |e_1\cup\ldots\cup e_d|=k}} x_{e_1} \cdots x_{e_d}.$$

For the case $d = 3$ we need to consider the values $k = 2, 3, 4, 5, 6$; as an illustration we show in Figure 5.2 all the eight possible patterns of triplets of edges $\underline{e} = (e_1, e_2, e_3)$ and the corresponding coefficients $c_{\underline{e}}$ that contribute to the summands $q_{3,k}$.

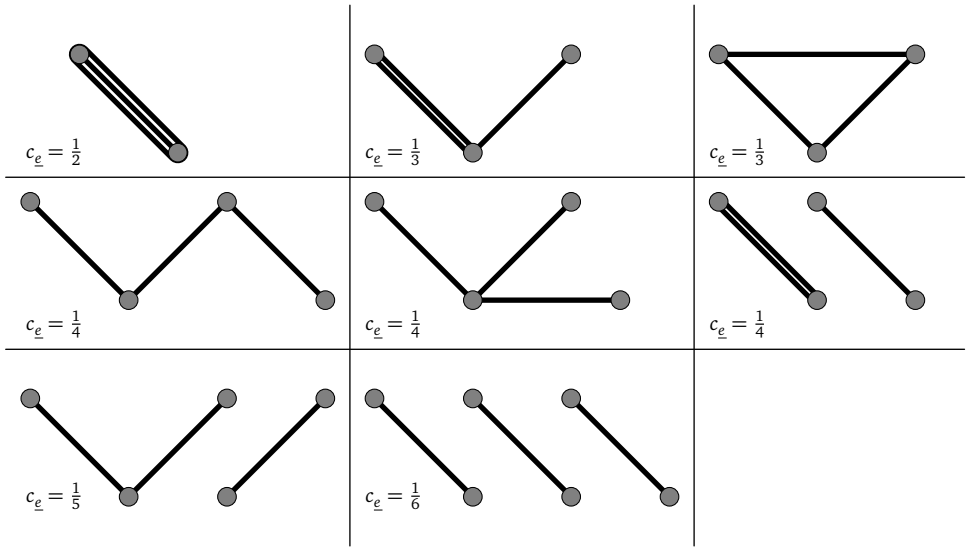


Figure 5.2: The eight patterns of triplets of edges in case $(d = 3, L = 2)$

Organization of the chapter. In the rest of this section we first indicate in Section 5.2.1 how the polynomials f_d naturally arise within a problem of queuing

theory with redundancy scheduling policies. After that we present in Section 5.2.2 the main ideas of the proofs, which highly rely on exploiting symmetry properties of the polynomials. This involves in particular using the Terwilliger algebra of the binary Hamming cube, so we include some preliminaries about these Terwilliger algebras in Section 5.2.3.

In Section 5.3 we give the full proof for Theorem 5.2 showing that the polynomials p_d attain their global minimum at the barycenter of the simplex and, in Section 5.4, we investigate the second class of polynomials f_d. We prove several properties of these polynomials, which we use to show Theorem 5.3. We also present a range of values of (n, d, L) for which the polynomials f_d are indeed convex and thus Conjecture 5.1 is indeed true.

Some notation. Throughout we let I, J denote the identity matrix and the all-ones matrix, whose size should be clear from the context. When we want to specify the size we let I_n (resp. J_n) denote the $n \times n$ identity matrix (resp. all-ones matrix) and, given two integers $n, m \geq 1$, $J_{m,n}$ denotes the $m \times n$ all-ones matrix. For a symmetric matrix A the notation $A \succeq 0$ means that A is positive semidefinite. Given two matrices $A, B \in \mathbb{R}^{n\times n}$ we let $A \circ B \in \mathbb{R}^{n\times n}$ denote their Hadamard product, with entries $(A \circ B)_{ij} = A_{ij}B_{ij}$ for $i, j \in [n]$. It is known that $A \succeq 0$ and $B \succeq 0$ implies $A \circ B \succeq 0$, which follows from the fact that the matrix $A \circ B$ is a principal sub-matrix of the Kronecker product of A and B.

For a sequence $\alpha \in \mathbb{N}^n$ we set $|\alpha| = \sum_{i=1}^n \alpha_i$ and, for an integer $d \in \mathbb{N}$, we set $\mathbb{N}^n_d = \{\alpha \in \mathbb{N}^n : |\alpha| = d\}$. Given a vector $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{N}^n$ we set $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. Throughout we let $u_1, \ldots, u_m$ denote the standard basis of $\mathbb{R}^m$, where all entries of u_i are 0 except its ith entry which is equal to 1. We let S_n denote the set of permutations of the set $V = [n]$.

5.2 Preliminaries

In this section we first explain the relevance of the polynomials f_d and p_d for the problem from queuing theory considered in [CBL21]. Then we present a sketch of proof for our main results, and we conclude with some preliminaries about Terwilliger algebras that we will use in the symmetry reduction.

5.2.1 Motivation

Our motivation for the study of the polynomials p_d and f_d comes from their relevance to a problem in queuing theory. The question whether they attain their

minimum at the uniform probability distribution was posed to us by the authors of [CBL21], who conjecture this to establish a result about the asymptotic behavior of the job occupancy in a parallel-server system with redundancy scheduling in the light-traffic regime (in contrast to the heavy-traffic regime considered in [CBL20]). In what follows we will give only a high level sketch of this connection, and we refer to the paper [CBL21] for a detailed exposition. We also refer to [CBL21; CBL20] for an extended review of the relevant literature.

A crucial mechanism that has been considered to improve the performance of parallel-server systems in queuing theory is redundancy scheduling. The key feature of this policy is that several replicas are created for each arriving job, which are then assigned to distinct servers (and then, as soon as the first of these replicas completes (or enters) service on a server the remaining ones are stopped). The underlying idea is that sending replicas of the same job to several servers will increase the chance of having shorter queuing times. This however must be weighted against the risk of wastage of capacity. An important question is thus to assess the impact of redundancy scheduling policies. While most papers in the literature of redundant scheduling assume that the set of servers to which the replicas are sent is selected uniformly at random, the paper [CBL21] considers the case when the set of servers is selected according to a given probability distribution, and it investigates what is the impact of this probability distribution on the performance of the system. It is shown there that while the impact remains relatively limited in the heavy-traffic regime, the system occupancy is much more sensitive to the selected probability distribution in the light-traffic regime.

We will now only introduce a few elements of the model considered in [CBL21], so that we can make the link to the polynomials studied in this chapter. We keep our presentation high level and refer to [CBL21] for details. The setting is as follows. There are n parallel servers, with average speed μ. Jobs arrive as a Poisson process of rate $n\lambda$ for some $\lambda > 0$. When a job arrives, L replicas of it are created that are sent — with probability x_e — to a subset $\mathbf{e} \subseteq [n]$ of L servers. Here, $L \geq 2$ is an integer and $x = (x_e)_{e \in E}$ is a probability distribution on the set $E = \{e \subseteq [n] : |e| = L\}$ of possible collections of L servers. As noted in [CBL21] this can be seen as selecting an edge $e \in E$ with probability x_e in the uniform hypergraph $(V = [n], E)$ (with edge size L).

An important performance parameter is the system occupancy at time t, which is represented by a vector $(\mathbf{e_1}, ..., \mathbf{e_M}) \in E^M$, where $M = M(t)$ is the total number of jobs present in the system and $\mathbf{e_i} \in E$ is the collection of servers to which

the replicas of the ith longest job in the system have been assigned. We need three modeling assumptions. First one needs to assume suitable stability conditions. Second, all servers should have the same speed μ and, third, the service requirements of the jobs are assumed to be independent and exponentially distributed with unit mean. Under these assumptions, the stationary distribution of the occupancy of the above edge selection is given by

$$\pi(e_1,\dots,e_M) = C\prod_{i=1}^{M}\frac{n\lambda x_{e_i}}{\mu|e_1\cup\ldots\cup e_i|}$$

for some constant $C>0$ ([Gar+16], see equation (1) in [CBL21]). Following [CBL21], let $\mathbf{Q}_\lambda(\mathbf{x})$ be a random variable with the stationary distribution of the system occupancy when the edge selection is given by the probability vector $x=(x_e)_{e\in E}$. It then follows that, for any integer $d\geq 1$, the probability that d jobs are present in the system is given by

$$\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})=d] = \sum_{(e_1,\dots,e_d)\in E^d}\pi(e_1,\dots,e_d).$$

Hence, $\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})=0]=C$ and

$$\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})=d] = \mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})=0]\Big(\frac{n\lambda}{\mu}\Big)^d\sum_{(e_1,\dots,e_d)\in E^d}\prod_{i=1}^{d}\frac{x_{e_i}}{|e_1\cup\ldots\cup e_i|}.$$

(See equation (7) in [CBL21]). Therefore, $\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})=d]$ is the polynomial $f_d(x)$ (up to a scalar multiple). In [CBL21] the light-traffic regime is considered, i.e., when $\lambda\downarrow 0$, in the case $L=2$. By doing a Taylor expansion one can see that

$$\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})=0]=1+o(1),\ \mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})\geq d]=\Big(\frac{n\lambda}{\mu}\Big)^d f_d(x)+o(\lambda^d)$$

(see equation (10) in [CBL21]). Therefore, with $x^*=(1,\dots,1)/|E|$ denoting the uniform probability vector, we have

$$\lim_{\lambda\downarrow 0}\frac{\mathbb{P}[Q_\lambda(x^*)\geq d]}{\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})\geq d]}=\lim_{\lambda\downarrow 0}\frac{f_d(x^*)+o(1)}{f_d(x)+o(1)}.$$

Hence, if the polynomial f_d attains its minimum at the uniform distribution x^*, then one has

$$\lim_{\lambda\downarrow 0}\frac{\mathbb{P}[Q_\lambda(x^*)\geq d]}{\mathbb{P}[\mathbf{Q}_\lambda(\mathbf{x})\geq d]}\leq 1.$$

If this holds for each d, this indicates that in the light-traffic regime the system occupancy is minimized when selecting the assignments to the servers of the job replicas uniformly at random. This thus motivates Conjecture 5.1 stating that the polynomial f_d attains its minimum over the probability simplex at the uniform point x^*.

5.2.2 Sketch of Proof

Here we give a sketch of proof for our main results. We start with indicating the main steps for proving Theorem 5.2, dealing with the (simpler) class of polynomials p_d and after that we briefly indicate how to deal with the polynomials f_d.

A first easy observation is that in order to show that the polynomial p_d attains its minimum at the barycenter of the standard simplex Δ_m it suffices to show that p_d is convex over Δ_m. This follows from a symmetry argument, namely we exploit the fact that the polynomial p_d is invariant under the permutations of the edge set E that are induced by permutations of $[n]$.

Lemma 5.5. *Assume the polynomial p_d is convex on the simplex Δ_m. Then the point $x^* = (1/m)(1,\ldots,1) \in \Delta_m$ is a global minimizer of p_d over Δ_m.*

Proof. The key fact we use is that the polynomial p_d enjoys some symmetry property; namely, for any tuple $(e_1,\ldots,e_d) \in E^d$, the coefficient of the monomial $x_{e_1}\cdots x_{e_d}$ in p_d is $1/|e_1\cup\ldots\cup e_d|$, which depends only on the cardinality of the set $e_1\cup\ldots\cup e_d$. Recall that $E = \{e \subseteq V = [n] : |e| = L\}$. Any permutation $\sigma \in S_n$ of $[n]$ induces a permutation of E (still denoted σ) by setting $\sigma(e) = \{j_{\sigma(1)},\ldots,j_{\sigma(L)}\}$ for $e = \{j_1,\ldots,j_L\} \in E$. In turn, σ acts on Δ_m by setting $\sigma(x) = (x_{\sigma(e)})_{e\in\Delta_m}$ for $x = (x_e)_{e\in E} \in \Delta_m$. We now observe that p_d is invariant under this action of permutations $\sigma \in S_n$. Indeed, for any $\sigma \in S_n$, we have

$$\begin{aligned}\sigma(p_d)(x) = p_d(\sigma(x)) &= \textstyle\sum_{(e_1,\ldots,e_d)\in E^d} \frac{1}{|e_1\cup\ldots\cup e_d|} x_{\sigma(e_1)}\cdots x_{\sigma(e_d)}\\ &= \textstyle\sum_{(f_1,\ldots,f_d)\in E^d} \frac{1}{|\sigma^{-1}(f_1)\cup\ldots\cup\sigma^{-1}(f_d)|} x_{f_1}\cdots x_{f_d}\\ &= \textstyle\sum_{(f_1,\ldots,f_d)\in E^d} \frac{1}{|f_1\cup\ldots\cup f_d|} x_{f_1}\cdots x_{f_d}\\ &= p_d(x).\end{aligned}$$

Let $x \in \Delta_m$ be a global minimizer of p_d. For any permutation $\sigma \in S_n$ the permuted point $\sigma(x)$ belongs to Δ_m and $p_d(x) = p_d(\sigma(x))$ holds. Hence, for the full symmetrization of x,

$$x^* := \frac{1}{n!}\sum_{\sigma\in S_n} \sigma(x),$$

we have $x^* \in \Delta_m$ and all its entries are equal, so that $x^* = \frac{1}{m}(1, \ldots, 1)$. Moreover, as the polynomial p_d is convex over Δ_m, we have:

$$p_d(x^*) \leq \frac{1}{n!} \sum_{\sigma \in S_n} p_d(\sigma(x)) = p_d(x).$$

This shows that x^* is again a global minimizer of p_d in Δ_m. As $x^* = \frac{1}{m}(1, \ldots, 1)$, the proof is complete. □

Therefore, we are left with the task of showing that the polynomial p_d is convex over the simplex Δ_m or, equivalently, that its Hessian matrix

$$H(p_d)(x) = (\partial^2 p_d(x)/\partial x_e \partial x_f)_{e,f \in E}$$

is positive semidefinite over Δ_m. This forms the core technical part of the proof. Here is a rough sketch of our proof technique.

A first step is to express the Hessian matrix of p_d as a matrix polynomial, involving a collection of matrices M_γ, which (up to positive scaling) are the coefficients of the Hessian $H(p_d)$ in the monomial basis; see Lemma 5.11. The next step is to show that each of the matrices M_γ appearing in this decomposition of the Hessian is positive semidefinite. For this, one first reduces to the task of showing that a certain set of well-structured matrices are positive semidefinite, see Lemma 5.12 and Lemma 5.13. After that, this last task is done by showing that these matrices lie in the Terwilliger algebra of the Hamming cube, which enables us to exploit its explicitly known block-diagonalization. The proof is then concluded by using an integral representation argument, see Section 5.3.3.

The treatment for the polynomials f_d has the same starting point: the polynomial f_d is invariant under any permutation of the edge set E induced by permutations of $[n]$, and thus it suffices to show that f_d is convex in order to conclude that it attains its global minimum at the barycenter of the simplex (i.e., the analogue of Lemma 5.5 holds for f_d). After that we again express the Hessian matrix $H(f_d)$ as a matrix polynomial, involving a collection of matrices Q_γ that occur as its coefficients in the monomial basis; see Lemma 5.18. Hence, here too, the task boils down to showing that each of these matrices Q_γ is positive semidefinite. This task turns out to be considerably more difficult than for the matrices M_γ which occurred in the analysis of the polynomial p_d. As a first step toward the analysis of the matrices Q_γ we give a recursive reformulation for them, which also makes apparent how the matrices M_γ enter their definition (namely as a factor

of a Hadamard product definition of Q_γ); see Lemma 5.21. Based on this we can show that the matrices Q_γ are indeed positive semidefinite in the case $d = 3$ and $L = 2$, thus showing Theorem 5.3; see Section 5.4.2.

5.2.3 Preliminaries on the Terwilliger Algebra

As mentioned above we need to exploit the symmetry structure of the polynomial p_d in order to show that its Hessian matrix is positive semidefinite. A crucial ingredient will be that the Hessian matrix can be decomposed into matrices that (after some reduction steps) all lie in the Terwilliger algebra of the binary Hamming cube. We begin with introducing the definition of the Terwilliger algebra $\mathcal{A}_n$ of the binary Hamming cube on n elements.

Definition 5.6 (**Terwilliger algebra of the binary Hamming cube**)**.** Let $\mathcal{P}_n$ denote the collection of all subsets of the set $V = [n]$. For every triple of nonnegative integers i, j, t we define the $2^n \times 2^n$ matrix $D^t_{i,j}$, indexed by $\mathcal{P}_n$, with entries

$$\left(D^t_{i,j}\right)_{S,T} = \begin{cases} 1 & if\ |S| = i, |T| = j, |S \cap T| = t, \\ 0 & \text{else} \end{cases},$$

for sets $S, T \in \mathcal{P}_n$. Then the *Terwilliger algebra of the binary Hamming cube*, denoted by $\mathcal{A}_n$, is defined as the (real) span of all these matrices:

$$\mathcal{A}_n = \Big\{ \sum_{i,j,t \geq 0} x^t_{i,j} D^t_{i,j} : x^t_{i,j} \in \mathbb{R} \Big\}.$$

It is easy to see that $\mathcal{A}_n$ is a *matrix $*$-algebra*, i.e., $\mathcal{A}_n$ is closed under taking linear combinations, matrix multiplications and transposition. One way to see this is by realizing that the matrices $D^t_{i,j}$ are exactly the indicator matrices of the orbits of pairs in $\mathcal{P}_n \times \mathcal{P}_n$ under the element-wise action of the symmetric group S_n.

All matrix $*$-algebras can be block-diagonalized by Artin-Wedderburn theory (see [Wed34], see also [BEK78] for a proof).

Theorem 5.7 (Artin-Wedderburn)**.** *Let $\mathcal{A}$ be a matrix $*$-algebra. Then there exist nonnegative integers d and $m_1, \ldots, m_d$ and a $*$-algebra isomorphism*

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

The important property here is that φ is an algebra isomorphism. Hence, we know that this isomorphism maintains positive semidefiniteness: for any matrix $A \in \mathcal{A}$, we have $A \succeq 0 \iff \varphi(A) \succeq 0$. Moreover, the matrix $\varphi(A)$ is block-diagonal, with d diagonal blocks of sizes $m_1, \ldots, m_d$. This is a crucial property which can be exploited in order to get a more efficient way of encoding positive semidefiniteness of matrices in $\mathcal{A}$.

The explicit block-diagonalization of the Terwilliger algebra $\mathcal{A}_n$ was given by Schrijver [Sch05].

Theorem 5.8 (Schrijver [Sch05])**.** *The Terwilliger algebra $\mathcal{A}_n$ can be block-diagonalized into $\lfloor \frac{n}{2} \rfloor + 1$ blocks, of sizes $m_k = n - 2k + 1$ for $k = 0, \ldots, \lfloor \frac{n}{2} \rfloor$. The algebra isomorphism φ sends the matrix*

$$A = \sum_{i,j,t=0}^{n} x_{i,j}^t D_{i,j}^t$$

to the block-matrix $\varphi(A) = \oplus_{k=0}^{\lceil n/2 \rceil} B_k$, where the matrix $B_k \in \mathbb{R}^{m_k \times m_k}$ is given by

$$B_k := \left(\binom{n-2k}{i-k}^{-\frac{1}{2}} \binom{n-2k}{j-k}^{-\frac{1}{2}} \sum_t \beta_{i,j,k}^t x_{i,j}^t \right)_{i,j=k}^{n-k} \tag{5.3}$$

for $k = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$. Here, for any nonnegative integers i, j, t, k, we set

$$\beta_{i,j,k}^t := \sum_{\ell=0}^{n} (-1)^{\ell - t} \binom{\ell}{t} \binom{n-2k}{n-k-\ell} \binom{n-k-\ell}{i-\ell} \binom{n-k-\ell}{j-\ell}. \tag{5.4}$$

In particular, we have

$$\sum_{i,j,t=0}^{n} x_{i,j}^t D_{i,j}^t \succeq 0 \iff B_k \succeq 0 \text{ for } k = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor. \tag{5.5}$$

5.3 Proof of Theorem 5.2

In this section we give the proof of Theorem 5.2. As a warm-up we start with the special case when the degree is $d = 2$ and the edge size is $L = 2$, where we can easily show that the polynomial p_2 is convex.

After that we proceed to the general case. We follow the steps as sketched earlier. First we express the Hessian matrix of p_d as a matrix polynomial, so that

it suffices to show that a set of matrices are positive semidefinite; namely the matrices M_γ in (5.11) for any $\gamma \in \mathbb{N}^m_{d-2}$, which are (up to scaling) the coefficients of $H(p_d)$ in the monomial basis. After that we indicate some reductions that lead to the task of showing that another set of smaller, well-structured matrices are positive semidefinite; namely the matrices M_p in (5.15) for any integer $p \le L(d-2)$. Finally, we show the positive semidefiniteness of these matrices M_p by exploiting a link to the Terwilliger algebra of the Boolean Hamming cube.

5.3.1 The Case $d = 2$ and $L = 2$

Here we consider the polynomial

$$p_2(x) = \sum_{e,f \in E} \frac{1}{|e \cup f|} x_e x_f,$$

where $E = \{e \subseteq [n] : |e| = 2\}$. We show that the polynomial p_2 is convex over the standard simplex or, equivalently, that its Hessian matrix is positive semidefinite over Δ_m. Here, the Hessian matrix of p_2 is given by $H(p_2) = 2M$, where M is the matrix indexed by E with entries

$$M_{e,f} = \frac{1}{|e \cup f|} \quad \text{for } e, f \in E. \tag{5.6}$$

Consider the matrices A_2, A_3, A_4 indexed by E, with entries

$$(A_s)_{e,f} = 1 \text{ if } |e \cup f| = s, \quad (A_s)_{e,f} = 0 \text{ otherwise}, \quad \text{for } s = 2, 3, 4.$$

Then, we have $A_2 = I$ and $A_2 + A_3 + A_4 = J$. Clearly we can express the matrix M as a linear combination of these matrices:

$$M = \frac{1}{2}I + \frac{1}{3}A_3 + \frac{1}{4}A_4 = \frac{1}{4}I + \frac{1}{12}A_3 + \frac{1}{4}J = \frac{1}{12}I + \frac{1}{4}J + \frac{1}{12}(A_3 + 2I). \tag{5.7}$$

We can now conclude that $M \succeq 0$ (and thus the polynomial p_2 is convex) in view of the next lemma, which claims that $A_3 + 2I \succeq 0$.

Lemma 5.9. *Consider the* $\binom{n}{2} \times n$ *matrix* Γ_n*, with entries* $(\Gamma_n)_{e,i} = |e \cap \{i\}|$ *for* $e \in E$ *and* $i \in [n]$*. Then* $A_3 + 2I = \Gamma_n \Gamma_n^T \succeq 0$.

Proof. Direct verification. □

Note that the matrices $A_2 = I, A_3, A_4$ generate the Bose-Mesner algebra of the Johnson scheme J^n_2, with length n and weight 2, and thus the matrix M belongs

to this Bose-Mesner algebra (see [DL98] for details on the Johnson scheme). For arbitrary degree $d \geq 3$ and edge size $L = 2$ one could proceed to show that the Hessian matrix of p_d is convex by using a similar symmetry reduction based on the Bose-Mesner algebra of the Johnson scheme J_2^p for suitable values of p. However, for general edge size $L \geq 3$ we will need to use a richer algebra, namely the Terwilliger algebra of the Hamming cube. Hence, we will treat in the rest of the section the general case $d \geq 2$ and $L \geq 2$.

5.3.2 Computing the Hessian Matrix of p_d

In this section we indicate how to compute the Hessian matrix of the polynomial

$$p_d(x) = \sum_{(e_{i_1},\ldots,e_{i_d})\in E^d} \frac{1}{|e_{i_1} \cup \ldots \cup e_{i_d}|} x_{e_{i_1}} \cdots x_{e_{i_d}}, \tag{5.8}$$

where, as before, $E = \{e \subseteq V = [n] : |e| = L\}$ with $L \geq 2$. We begin with getting the explicit coefficients of the polynomial p_d expressed in the standard monomial basis. The basic fact we will now use is that the coefficients depend only on the set of distinct edges that are present in the tuple $(e_{i_1}, \ldots, e_{i_d}) \in E^d$ and not on their multiplicities.

To formalize this, recall $m = |E|$ and let us label the edges as $e_1, \ldots, e_m$ so that $E = \{e_1, \ldots, e_m\}$. For a d-tuple $\underline{e} := (e_{i_1}, \ldots, e_{i_d}) \in E^d$ with $i_1, \ldots, i_d \in [m]$, define the sequence $\alpha(\underline{e}) \in \mathbb{N}^m$, where, for $\ell \in [m]$, $\alpha(\underline{e})_\ell$ is the number of indices among $i_1, \ldots, i_d$ that are equal to ℓ. Then we have:

$$x_{e_{i_1}} \cdots x_{e_{i_d}} = x_{e_1}^{\alpha(\underline{e})_1} \cdots x_{e_m}^{\alpha(\underline{e})_m} = x^{\alpha(\underline{e})}$$

and $|\alpha(\underline{e})| = d$ so that $\alpha(\underline{e}) \in \mathbb{N}_d^m$. This justifies the following definition. For $\alpha \in \mathbb{N}_d^m$, consider a d-tuple $\underline{e} = (e_{i_1}, \ldots, e_{i_d}) \in E^d$ such that $\alpha(\underline{e}) = \alpha$ and define

$$c_\alpha := \frac{1}{|e_{i_1} \cup \ldots \cup e_{i_d}|}. \tag{5.9}$$

As an example, for $d = n = m = 3$, if $\alpha = (1, 0, 2)$ then $c_\alpha = \frac{1}{|e_1 \cup e_3|}$ and if $\alpha = (2, 0, 1)$ then we also have $c_\alpha = \frac{1}{|e_1 \cup e_3|}$.

We can now reformulate the polynomial p_d in the (usual) monomial basis.

Lemma 5.10. *The polynomial p_d from (5.8) can be reformulated as follows:*

$$p_d(x) = \sum_{\alpha \in \mathbb{N}_d^m} c_\alpha \frac{d!}{\alpha!} x^\alpha, \tag{5.10}$$

setting $\alpha! = \alpha_1! \cdots \alpha_m!$ and where c_α is as defined in (5.9).

Proof. Using the definition of the coefficients c_α, we can rewrite p_d as

$$p_d(x) = \sum_{\alpha \in \mathbb{N}_d^m} \Big(\sum_{\substack{\underline{e}=(e_{i_1},\ldots,e_{i_d}) \in E^d : \\ \alpha(\underline{e})=\alpha}} \frac{1}{|e_{i_1} \cup \ldots \cup e_{i_d}|} \Big) x^\alpha = \sum_{\alpha \in \mathbb{N}_d^m} \Big(\sum_{\underline{e} \in E^d : \alpha(\underline{e})=\alpha} c_\alpha \Big) x^\alpha,$$

which is equal to $\sum_{\alpha \in \mathbb{N}_d^m} c_\alpha \frac{d!}{\alpha!} x^\alpha$. Here, for this last equality, we use the monomial theorem, which claims the identity

$$\left(\sum_{i=1}^m x_i \right)^d = \sum_{\alpha \in \mathbb{N}_d^m} \frac{d!}{\alpha!} x^\alpha,$$

or, equivalently, that the number of d-tuples $\underline{e} \in E^d$ for which $\alpha(\underline{e}) = \alpha$ is equal to $d!/\alpha!$. □

We now proceed to compute the Hessian matrix of p_d.

Lemma 5.11. *The Hessian of the polynomial p_d is the matrix*

$$H(p_d)(x) = \Big(\frac{\partial^2 p_d(x)}{\partial x_{e_i} \partial x_{e_j}} \Big)_{i,j=1}^m = \sum_{\gamma \in \mathbb{N}_{d-2}^m} \frac{d!}{\gamma!} x^\gamma M_\gamma,$$

where, for any $\gamma \in \mathbb{N}_{d-2}^m$, we set

$$M_\gamma = (c_{\gamma + u_i + u_j})_{i,j=1}^m \tag{5.11}$$

and where the vectors $u_1, \ldots, u_m \in \mathbb{R}^m$ denote the standard basis of $\mathbb{R}^m$.

Proof. The partial derivatives of p_d are

$$\frac{\partial p_d(x)}{\partial x_{e_i}} = \sum_{\alpha \in \mathbb{N}_d^m : \alpha_i \geq 1} \frac{d!}{(\alpha - u_i)!} c_\alpha x^{\alpha - u_i} = \sum_{\beta \in \mathbb{N}_{d-1}^m} \frac{d!}{\beta!} c_{\beta + u_i} x^\beta.$$

Similarly, we see that

$$\frac{\partial^2 p(x)}{\partial x_{e_j} \partial x_{e_i}} = \sum_{\beta \in \mathbb{N}_{d-1}^m : \beta_j \geq 1} \frac{d!}{(\beta - u_j)!} c_{\beta + u_i} x^{\beta - u_j} = \sum_{\gamma \in \mathbb{N}_{d-2}^m} c_{\gamma + u_i + u_j} \frac{d!}{\gamma!} x^\gamma.$$

This concludes the proof. □

Hence, if we can show that the matrices M_γ in (5.11) are all positive semidefinite then it follows directly that the Hessian matrix of p_d is positive semidefinite on the standard simplex. In the rest of this section we indicate two successive simplifications that reduce the task of checking positive semidefiniteness of the matrices M_γ (for $\gamma \in \mathbb{N}^m_{d-2}$) to the same task for a smaller set of simpler matrices: first for the matrices M_W (for $W \subseteq V$), and second for the matrices M_p (for $0 \leq p \leq n$ integer). In Section 5.3.3 we will make a final reduction to show that the matrices M_p are positive semidefinite, exploiting the fact that they belong to Terwilliger algebras.

We begin with the first reduction. For $\gamma \in \mathbb{N}^m$, define its *support* as the set $S_\gamma = \{e \in E : \gamma_e \geq 1\}$ and let

$$W_\gamma = \bigcup_{e \in S_\gamma} e$$

denote the subset of elements of $V = [n]$ that are covered by some edge in the support of γ. Then, for any $i, j \in [m]$, the support of $\gamma + u_i + u_j$ is the set $S_\gamma \cup \{e_i, e_j\}$, and we have

$$(M_\gamma)_{e_i, e_j} = c_{\gamma + u_i + u_j} = \frac{1}{|W_\gamma \cup e_i \cup e_j|}.$$

Hence, the matrix M_γ depends only on the set W_γ (and not on the specific choice of the sequence γ). This justifies defining the matrices

$$M_W = \Big(\frac{1}{|W \cup e \cup f|}\Big)_{e,f \in E} \tag{5.12}$$

for any set $W \subseteq V = [n]$. Hence, for any $\gamma \in \mathbb{N}^m_{d-2}$, we have:

$$M_\gamma = M_{W_\gamma}. \tag{5.13}$$

Summarizing, we have shown:

Lemma 5.12. *Assume that the matrices M_W from (5.12) are positive semidefinite for all $W \subseteq V$ with $|W| \geq L$ (if $d \geq 3$) and $|W| \leq L(d-2)$. Then the polynomial p_d is convex over the standard simplex.*

If $d = 2$ then there is only one matrix to check, namely the matrix $M_\emptyset$ (for $W = \emptyset$). Note that the matrix $M_\emptyset$ coincides with the matrix in (5.6), so we already know that it is positive semidefinite when $L = 2$. However, if $d \geq 3$, then one needs to check all the matrices of the form M_W in (5.12).

Now comes the second reduction, which will be useful to link these matrices M_W to the Terwilliger algebra. We observe that in the matrix M_W there are identical rows and columns and the reduction consists simply in removing duplicate rows and columns in M_W and keeping just one copy. For this set $p := |W|$ and $U := V \setminus W$, so that $|U| = n - p$. In addition, set

$$F := \{e \setminus W : e \in E\} = \{e \subseteq U : L - p \leq |e| \leq L\}, \tag{5.14}$$

which consists of the intersections with U of the edges in E. Then, $F = E$ when $p = 0$ and the condition $|e| \geq L - p$ is redundant when $p \geq L$. Now we consider the following matrix M_p, which is indexed by F, with entries

$$(M_p)_{e,f} = \frac{1}{p + |e \cup f|} \quad \text{for } e, f \in F. \tag{5.15}$$

Note that for $p = 0$ the matrix M_0 coincides with the matrix $M_\emptyset$ in (5.12) (and with the matrix in (5.6)). The next lemma links the matrices M_W and M_p and relies on showing that M_p is obtained from M_W by deleting duplicate rows and columns.

Lemma 5.13. *Let $L \geq 2$ and $d \geq 2$. Consider the matrices M_W in (5.12) and M_p in (5.15). The following assertions are equivalent:*

(i) $M_W \succeq 0$ *for all* $W = e_1 \cup \ldots \cup e_{d-2}$ *with* $e_1, \ldots, e_{d-2} \in E$.

(ii) $M_p \succeq 0$ *for all* $p \leq L(d-2)$ *such that* $p \geq L$ *if* $d \geq 3$.

Proof. If $d = 2$ then the result holds since $M_0 = M_\emptyset$ as observed above. So assume now $d \geq 3$. Let $W = e_1 \cup \ldots \cup e_{d-2}$, where $e_1, \ldots, e_{d-2} \in E$, and set $p = |W|$. Consider the partition of the set E into $E = \cup_{i=0}^{L} E_i$, where $E_i = \{e \in E : |e \setminus W| = i\}$. With respect to this partition of its index set, the matrix M_W has the following block-form:

$$M_W = \left(\begin{array}{c|c|c|c} M_W^{0,0} & M_W^{0,1} & \cdots & M_W^{0,L} \\ \hline M_W^{1,0} & M_W^{1,1} & \cdots & M_W^{1,L} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline M_W^{L,0} & M_W^{L,1} & \cdots & M_W^{L,L} \end{array}\right),$$

where the block $M_W^{i,j}$ has its rows indexed by E_i and its columns by E_j. Note that, if two edges $e, e' \in E$ satisfy $e \setminus W = e' \setminus W$, then the two rows of M_W indexed by e and e' coincide: for any $f \in E$ we have

$$(M_W^{i,j})_{e,f} = \frac{1}{|W| + |(e \cup f) \setminus W|} = \frac{1}{|W| + |(e' \cup f) \setminus W|} = (M_W^{i,j})_{e',f}.$$

In fact, after removing these duplicate rows (and columns) and keeping only one copy for each subset of $U = V \setminus W$, we obtain the matrix

$$\left(\begin{array}{c|c|c|c} M_p^{0,0} & M_p^{0,1} & \cdots & M_p^{0,L} \\ \hline M_p^{1,0} & M_p^{1,1} & \cdots & M_p^{1,L} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline M_p^{L,0} & M_p^{L,1} & \cdots & M_p^{L,L} \end{array}\right),$$

which coincides with the matrix M_p in (5.15). Indeed, the above matrix is indexed by the set F in (5.14) and its block-form is with respect to the partition $F = \cup_{i=0}^{L} F_i$, where $F_i = \{e \subseteq U : |e| = i\}$. So the block $M_p^{i,j}$ has its rows indexed by F_i, its columns indexed by F_j, and its entries are

$$(M_p^{i,j})_{e,f} = \frac{1}{p + |e \cup f|} = \frac{1}{p + i + j - |e \cap f|} \quad \text{for } e \in F_i, f \in F_j. \tag{5.16}$$

As the matrices M_p arise from M_W by removing its duplicate rows and columns it is clear that the matrices M_W are positive semidefinite if and only if the same holds for the matrices M_p. This concludes the proof. □

In the next section we show that the matrices M_p are positive semidefinite for all $0 \leq p \leq n$ by exploiting their link to Terwilliger algebras.

5.3.3 The General Case $d \geq 2$ and $L \geq 2$

In Section 5.2.3 we gave preliminary results on the Terwilliger algebra, which we will now use to prove that the matrices M_p in (5.15) are positive semidefinite. Fix an integer $0 \leq p \leq n$ and consider the matrix M_p in (5.15), which has a block-form with blocks as in (5.16). We start with observing that M_p belongs to the Terwilliger algebra $\mathcal{A}_{n-p}$. This is clear since relation (5.16) provides the explicit correspondence between the blocks $M_p^{i,j}$ of M_p and the generating matrices $D_{i,j}^t$ of the algebra $\mathcal{A}_{n-p}$:

$$M_p = \sum_{i=0}^{L} \sum_{j=0}^{L} \sum_{t=0}^{\min\{i,j\}} \frac{1}{p + i + j - t} D_{i,j}^t = \sum_{i=0}^{L} \sum_{j=0}^{L} \sum_{t=0}^{\min\{i,j\}} x_{i,j}^t D_{i,j}^t,$$

after setting

$$x_{i,j}^t = \frac{1}{p + i + j - t}. \tag{5.17}$$

Let B_k be the corresponding matrices from (5.3) (replacing n by $n-p$). Then, in view of Theorem 5.8, we know that $M_p \succeq 0$ if and only if $B_k \succeq 0$ for all $0 \leq k \leq \lfloor (n-p)/2 \rfloor$.

In what follows p, k are fixed integers. We now proceed to show that $B_k \succeq 0$. To simplify the notation we introduce the following parameters

$$a(i) := \binom{n-p-2k}{i-k}^{-\frac{1}{2}}, \quad b(\ell, i) := \binom{n-p-k-\ell}{i-\ell}, \quad c(\ell) := \binom{n-p-2k}{n-p-k-\ell}$$

for any integers i, ℓ. Note that we may omit the obvious bounding conditions on i and ℓ since the corresponding parameters are zero if these conditions are not satisfied; for instance, $a(i) = 0$ if $i < k$ and $b(\ell, i) = 0$ if $\ell > i$. Then we have

$$B_k = \left(a(i)a(j) \sum_{t=0}^{min\{i,j\}} \beta_{i,j,k}^t x_{i,j}^t \right)_{i,j=k}^{n-p-k} \tag{5.18}$$

and

$$\beta_{i,j,k}^t := \sum_{\ell=0}^{n-p} (-1)^{\ell-t} \binom{\ell}{t} c(\ell) b(\ell, i) b(\ell, j). \tag{5.19}$$

We now give an integral reformulation for the entries of the matrix B_k from (5.18). It is based on the fact that

$$\frac{1}{i} = \int_0^1 z^{i-1} dz \quad \text{for any integer } i \geq 1, \tag{5.20}$$

which permits to give an integral reformulation for the scalars $x_{i,j}^t$ in (5.17). This simple but powerful fact will be very useful to show $B_k \succeq 0$. Note that this is similar to the classical argument used by Hilbert [Hil94] to show that the Hilbert matrix $(\frac{1}{i+j-1})_{i,j=1}^n$ is positive semidefinite for any $n \in \mathbb{N}$.

Lemma 5.14. *We have*

$$\sum_{t=0}^{\min\{i,j\}} \beta_{i,j,k}^t x_{i,j}^t = \sum_{\ell=0}^{\min\{i,j\}} c(\ell) b(\ell, i) b(\ell, j) \int_0^1 g(\ell, z) z^{i+j} dz,$$

where we define the function $g(\ell, z) = z^{p-1} (\frac{1-z}{z})^\ell$ *for* $z \in (0, 1]$.

Proof. First we use the expressions of $\beta^t_{i,j,k}$ in (5.19) and of $x^t_{i,j}$ in (5.17) and we exchange the summations in t and ℓ to obtain

$$\sum_{t=0}^{\min\{i,j\}} \beta^t_{i,j,k} x^t_{i,j} = \sum_{\ell=0}^{\min\{i,j\}} \left(\sum_{t=0}^{\ell} \frac{1}{p+i+j-t} (-1)^{\ell-t} \binom{\ell}{t} \right) c(\ell) b(\ell,i) b(\ell,j). \quad (5.21)$$

Now we use (5.20), which gives the following integral representation

$$\frac{1}{p+i+j-t} = \int_0^1 z^{p+i+j-t-1} dz.$$

Using this integral representation (and the binomial theorem for the equality marked (*) below) we can reformulate the inner summation appearing in (5.21) as follows:

$$\begin{aligned}
\sum_{t=0}^{\ell} \frac{1}{p+i+j-t} (-1)^{\ell-t} \binom{\ell}{t} &= \sum_{t=0}^{\ell} (-1)^{\ell-t} \binom{\ell}{t} \int_0^1 z^{p+i+j-t-1} dz \\
&= \int_0^1 z^{p+i+j-1} (-1)^{\ell} \left(\sum_{t=0}^{\ell} \left(-\frac{1}{z} \right)^t \binom{\ell}{t} \right) dz \\
&\overset{(*)}{=} \int_0^1 z^{p+i+j-1} (-1)^{\ell} \left(1 - \frac{1}{z} \right)^{\ell} dz \\
&= \int_0^1 z^{p+i+j-1} (-1)^{\ell} \left(\frac{z-1}{z} \right)^{\ell} dz \\
&= \int_0^1 z^{p-1} \left(\frac{1-z}{z} \right)^{\ell} z^{i+j} dz.
\end{aligned}$$

This concludes the proof. □

We can now proceed to show that the matrices B_k in (5.18) are positive semidefinite.

Lemma 5.15. *We have $B_k \succeq 0$.*

Proof. We use Lemma 5.14 to reformulate the matrix B_k. First, note that in the result of Lemma 5.14, since $b(\ell,i)b(\ell,j) = 0$ if $\ell > \min\{i,j\}$, we may replace the summation on ℓ from $0 \le \ell \le \min\{i,j\}$ to $0 \le \ell \le n-p$. This implies:

$$B_k = \left(a(i)a(j) \sum_{t=0}^{n-p} \beta^t_{i,j,k} x^t_{i,j} \right)_{i,j=k}^{n-p-k}$$

$$
\begin{aligned}
&= \int_0^1 \Big(\sum_{\ell=0}^{n-p} g(\ell,z)c(\ell) \underbrace{(z^i a(i)b(\ell,i))}_{=:h(\ell,z,i)} \underbrace{(z^j a(j)b(\ell,j))}_{=:h(\ell,z,j)} \Big)_{i,j=k}^{n-p-k} dz \\
&= \sum_{\ell=0}^{n-p} \int_0^1 g(\ell,z)c(\ell) \underbrace{\Big(h(\ell,z,i)h(\ell,z,j) \Big)_{i,j=k}^{n-p-k}}_{=:H(\ell,z,k)} dz \\
&= \sum_{\ell\geq 0} \int_0^1 \underbrace{g(\ell,z)c(\ell)}_{\geq 0} \underbrace{H(\ell,z,k)}_{\succeq 0} dz \succeq 0.
\end{aligned}
$$

Here we used the fact that, for any $\ell \in [0, n-p]$, the function $g(\ell,z)$ is nonnegative on $(0,1]$ and that the matrix $H(\ell,z,k)$ is positive semidefinite for any $z \in [0,1]$ since it is the outer product of the vector $(h(\ell,z,i))_{i=k}^{n-p-k}$ with itself. □

Therefore, we have shown that the matrices B_k are positive semidefinite and thus the following result.

Corollary 5.16. *The matrices M_p from (5.15) are positive semidefinite for all $0 \leq p \leq n$.*

In view of Lemmas 5.12 and 5.13 we can conclude that the polynomial p_d is convex on Δ_m, which concludes the proof of Theorem 5.2.

5.4 Investigating the Polynomials f_d

Here we consider the second class of polynomials f_d from (5.1), namely

$$
f_d(x) = \sum_{(e_1,\ldots,e_d)\in E^d} \prod_{i=1}^{d} \frac{x_{e_i}}{|e_1 \cup \ldots \cup e_i|}.
$$

We address Conjecture 5.1, which states that f_d attains its minimum value on the simplex Δ_m at the barycenter of Δ_m. Here too, the conjecture follows if one can show that f_d is convex over Δ_m, as the analogue of Lemma 5.5 extends easily for the polynomial f_d. We conjecture that convexity holds in general, which is a stronger conjecture than Conjecture 5.1, and is not necessary to prove Conjecture 5.1.

Conjecture 5.17. For any integers $n, L, d \geq 2$ the polynomial f_d is convex over the simplex Δ_m.

For degree $d = 2$, we have $f_2 = \frac{1}{L} p_2$, and thus we know from Theorem 5.2 that f_2 is convex. We will prove in Section 5.4.2 that Conjecture 5.17 holds for degree $d = 3$ and edge size $L = 2$ and, in Section 5.4.3 and Appendix A, we will give a range of values for (n, L, d) that were numerically tested and support Conjecture 5.17.

In what follows we begin in Section 5.4.1 with giving a polynomial matrix decomposition for the Hessian of f_d. Hence, convexity of f_d over Δ_m follows if we can show that a family of well-structured matrices Q_γ, arising as the coefficients of $H(f_d)$ in the monomial basis, are positive semidefinite (see Lemmas 5.18). Then we give a recursive reformulation for the matrices Q_γ, which makes apparent some links to the matrices M_γ arising in the Hessian of p_d (see Lemma 5.21). Using this reformulation we can show positive semidefiniteness of the matrices Q_γ in the case $d = 3$ and $L = 2$ (see Section 5.4.2). However, understanding the general case is technically involved and would require developing new tools for exploiting the symmetry structure present in the matrices Q_γ (which is now not captured by the Terwilliger algebra). This goes beyond the scope of this chapter, and we leave it for further research. In very recent work Polak [Pol22] has carried out this symmetry reduction, which enables him to show that Conjecture 5.17 holds in the case when $d \leq 8$ and $L = 2$.

5.4.1 Computing the Hessian of f_d

We begin with expressing the polynomial f_d in the standard monomial basis:

$$f_d(x) = \sum_{\alpha \in \mathbb{N}_d^m} x^\alpha \sum_{\substack{\underline{e}=(e_1,\ldots,e_d)\in E^d \\ \alpha(\underline{e})=\alpha}} \prod_{i=1}^{d} \frac{1}{|e_1 \cup \ldots \cup e_i|} = \sum_{\alpha \in \mathbb{N}_d^m} b_\alpha x^\alpha, \tag{5.22}$$

where we set

$$b_\alpha = \sum_{\substack{\underline{e}=(e_1,\ldots,e_d)\in E^d \\ \alpha(\underline{e})=\alpha}} \prod_{i=1}^{d} \frac{1}{|e_1 \cup \ldots \cup e_i|}. \tag{5.23}$$

Next we compute the Hessian of f_d and we give a matrix polynomial reformulation for it.

Lemma 5.18. *The Hessian of the polynomial f_d is given by*

$$\frac{\partial^2 f(x)}{\partial x_{e_i} \partial x_{e_j}} = \begin{cases} \sum_{\gamma \in \mathbb{N}_{d-2}^m} (\gamma_i + 1)(\gamma_j + 1) x^\gamma b_{\gamma + u_i + u_j} & \text{if } i \neq j, \\ \sum_{\gamma \in \mathbb{N}_{d-2}^m} (\gamma_i + 1)(\gamma_i + 2) x^\gamma b_{\gamma + 2u_i} & \text{if } i = j, \end{cases},$$

where, as before, $u_1, \dots, u_m$ denote the standard basis of $\mathbb{R}^m$. In other words,

$$H(f_d)(x) = \sum_{\gamma \in \mathbb{N}^m_{d-2}} x^\gamma Q_\gamma,$$

where, for $\gamma \in \mathbb{N}^m_{d-2}$, we define the symmetric $m \times m$ matrix Q_γ with entries

$$(Q_\gamma)_{ij} = (\gamma_i + 1)(\gamma_j + 1) b_{\gamma + u_i + u_j} \text{ if } i \neq j, \quad (Q_\gamma)_{ii} = (\gamma_i + 1)(\gamma_i + 2) b_{\gamma + 2u_i} \tag{5.24}$$

for $i, j \in [m]$. Hence, $H(f_d)(x) \succeq 0$ for all $x \in \Delta_m$ if $Q_\gamma \succeq 0$ for all $\gamma \in \mathbb{N}^m_{d-2}$.

Proof. Direct verification. □

We now give a recursive reformulation for the coefficients of the polynomial f_d and for its Hessian matrix, that may be helpful for a proof by induction. Recall the definition of the coefficients b_α of f_d in (5.23). Fix $\alpha \in \mathbb{N}^m_d$. There are $\frac{d!}{\alpha!}$ distinct tuples $\underline{e}$ such that $\alpha(\underline{e}) = \alpha$. For any such sequence $\underline{e} = (e_{i_1}, \dots, e_{i_d})$ with $i_1, \dots, i_d \in [m]$, $\alpha = \alpha(\underline{e})$ means that, for any $\ell \in [m]$, α_ℓ is the number of occurrences of ℓ within the multiset $\{i_1, \dots, i_d\}$; so $\alpha_\ell \geq 1$ if $\ell \in \{i_1, \dots, i_d\}$ and $\alpha_\ell = 0$ if $\ell \notin \{i_1, \dots, i_d\}$. For instance, for $\underline{e} = (e_1, e_2, e_3, e_2, e_1)$, $d = 5$, $m = 4$, we have $(i_1, \dots, i_5) = (1, 2, 3, 2, 1)$ and $\alpha(\underline{e}) = (2, 2, 1, 0)$.

To reformulate b_α we exploit the fact that b_α enjoys some invariance property under permutations of $[d]$, namely

$$\begin{aligned}
b_\alpha &= \sum_{\substack{\underline{e} = (e_{i_1}, \dots, e_{i_d}) \in E^d : \\ \alpha(\underline{e}) = \alpha}} \prod_{k=1}^{d} \frac{1}{|e_{i_1} \cup \ldots \cup e_{i_k}|} \\
&= \frac{1}{d!} \sum_{\sigma \in S_d} \sum_{\substack{\underline{e} = (e_{i_1}, \dots, e_{i_d}) \in E^d : \\ \alpha(\underline{e}) = \alpha}} \prod_{k=1}^{d} \frac{1}{|e_{i_{\sigma(1)}} \cup \ldots \cup e_{i_{\sigma(k)}}|} \\
&= \frac{1}{d!} \sum_{\substack{\underline{e} = (e_{i_1}, \dots, e_{i_d}) \in E^d \\ \alpha(\underline{e}) = \alpha}} \underbrace{\sum_{\sigma \in S_d} \prod_{k=1}^{d} \frac{1}{|e_{i_{\sigma(1)}} \cup \ldots \cup e_{i_{\sigma(k)}}|}}_{=:S}.
\end{aligned} \tag{5.25}$$

Observe that the inner summation S in (5.25) does not depend on the choice of the sequence $\underline{e}$ such that $\alpha(\underline{e}) = \alpha$; thus we may consider it fixed, denoted as $(e_{i_1}, \dots, e_{i_d})$. Since there are $\frac{d!}{\alpha!}$ possible choices for selecting this sequence, using relation (5.25) we can reformulate b_α as follows:

$$b_\alpha = \frac{1}{d!} \frac{d!}{\alpha!} \sum_{\sigma \in S_d} \prod_{k=1}^{d} \frac{1}{|e_{i_{\sigma(1)}} \cup \ldots \cup e_{i_{\sigma(k)}}|} = \frac{1}{\alpha!} \sum_{\sigma \in S_d} \prod_{k=1}^{d} \frac{1}{|e_{i_{\sigma(1)}} \cup \ldots \cup e_{i_{\sigma(k)}}|}.$$

Next we pull out the factor $\frac{1}{|e_{i_1}\cup\ldots\cup e_{i_d}|} = c_\alpha$ which occurs for $k = d$ and get

$$\begin{aligned} b_\alpha &= \frac{c_\alpha}{\alpha!}\sum_{r=1}^{d}\sum_{\sigma\in S_d:\sigma(d)=r}\prod_{k=1}^{d-1}\frac{1}{|e_{i_{\sigma(1)}}\cup\ldots\cup e_{i_{\sigma(k)}}|} \\ &= \frac{c_\alpha}{\alpha!}\sum_{r=1}^{d} b_{\alpha-u_{i_r}}(\alpha-u_{i_r})! \\ &= c_\alpha\sum_{r=1}^{d}\frac{b_{\alpha-u_{i_r}}}{\alpha_{i_r}} \\ &\overset{(*)}{=} c_\alpha\sum_{k\in[m]:\alpha_k\geq 1} b_{\alpha-u_k}. \end{aligned}$$

Here, in the last equality marked (*), we use the fact that α_k of the elements in the multiset $\{i_1,\ldots,i_d\}$ are equal to k. Summarizing we have shown:

Lemma 5.19. *For any $\alpha\in\mathbb{N}_d^m$ we have*

$$b_\alpha = c_\alpha\sum_{k\in[m]:\alpha_k\geq 1} b_{\alpha-u_k}.$$

We now proceed to give a recursive reformulation for the matrices Q_γ in (5.24). First we reformulate them using the scaled parameters

$$\widehat{b}_\alpha := \alpha!\; b_\alpha, \tag{5.26}$$

which satisfy the recursive relation:

$$\widehat{b}_\alpha = c_\alpha\sum_{k:\alpha_k\geq 1}\alpha_k\widehat{b}_{\alpha-u_k}. \tag{5.27}$$

Indeed, by Lemma 5.19 we have

$$\widehat{b}_\alpha = \alpha!\; b_\alpha = \alpha!\; c_\alpha\sum_{k:\alpha_k\geq 1} b_{\alpha-u_k} = \alpha!\; c_\alpha\sum_{k:\alpha_k\geq 1}\frac{\widehat{b}_{\alpha-u_k}}{\alpha-u_k!} = c_\alpha\sum_{k:\alpha_k\geq 1}\alpha_k\widehat{b}_{\alpha-u_k}.$$

Lemma 5.20. *For any $\gamma\in\mathbb{N}_{d-2}^m$ we have $Q_\gamma = \frac{1}{\gamma!}(\widehat{b}_{\gamma+u_i+u_j})_{i,j=1}^m$.*

Proof. Direct verification: for $i\neq j$ we have $(Q_\gamma)_{ij} = (\gamma_i+1)(\gamma_j+1)b_{\gamma+u_i+u_j} = \widehat{b}_{\gamma+u_i+u_j}(\gamma_i+1)(\gamma_j+1)/(\gamma+u_i+u_j)! = \widehat{b}_{\gamma+u_i+u_j}/\gamma!$ and, for $i=j$, we have $(Q_\gamma)_{ii} = (\gamma_i+1)(\gamma_i+2)b_{\gamma+2u_i} = \widehat{b}_{\gamma+2u_i}(\gamma_i+1)(\gamma_i+2)/(\gamma+2u_i)! = \widehat{b}_{\gamma+2u_i}/\gamma!$. □

Lemma 5.21. *For $d \geq 3$ and $\gamma \in \mathbb{N}_{d-2}^m$ we have*

$$Q_\gamma = \underbrace{(c_{\gamma+u_i+u_j})_{i,j=1}^m}_{M_\gamma} \circ \Big(\sum_{k\in[m]:\gamma_k\geq 1} Q_{\gamma-u_k} + \underbrace{\frac{1}{\gamma!}(\widehat{b}_{\gamma+u_i} + \widehat{b}_{\gamma+u_j})_{i,j=1}^m}_{=:R_\gamma} \Big)$$

$$= M_\gamma \circ \Big(\sum_{k\in[m]:\gamma_k\geq 1} Q_{\gamma-u_k} + R_\gamma \Big),$$

where the matrices M_γ were introduced in (5.11).

Proof. Combining Lemmas 5.19 and 5.21 we obtain

$$\begin{aligned}
(Q_\gamma)_{ij} &= \frac{1}{\gamma!}\widehat{b}_{\gamma+u_i+u_j} = \frac{1}{\gamma!}c_{\gamma+u_i+u_j} \sum_{k:(\gamma+u_i+u_j)_k\geq 1} \widehat{b}_{\gamma+u_i+u_j-u_k}(\gamma+u_i+u_j)_k \\
&= \frac{1}{\gamma!}c_{\gamma+u_i+u_j}\Big(\sum_{k\neq i,j:\gamma_k\geq 1} \widehat{b}_{\gamma+u_i+u_j-u_k}\gamma_k + \widehat{b}_{\gamma+u_j}(\gamma_i+1) + \widehat{b}_{\gamma+u_j}(\gamma_i+1)\Big) \\
&= \frac{1}{\gamma!}c_{\gamma+u_i+u_j}\Big(\sum_{k:\gamma_k\geq 1} \widehat{b}_{\gamma-u_k+u_i+u_j}\gamma_k + \widehat{b}_{\gamma+u_i} + \widehat{b}_{\gamma+u_j}\Big) \\
&= c_{\gamma+u_i+u_j}\Big(\sum_{k:\gamma_k\geq 1} \frac{\widehat{b}_{\gamma-u_k+u_i+u_j}}{(\gamma-u_k)!} + \frac{1}{\gamma!}(\widehat{b}_{\gamma+u_i} + \widehat{b}_{\gamma+u_j})\Big) \\
&= c_{\gamma+u_i+u_j}\Big(\sum_{k:\gamma_k\geq 1} (Q_{\gamma-u_k})_{ij} + \frac{1}{\gamma!}(\widehat{b}_{\gamma+u_i} + \widehat{b}_{\gamma+u_j})\Big),
\end{aligned}$$

which shows the claim. □

5.4.2 The Polynomial f_d in the Case $d = 3$, $L = 2$

Here we show that the polynomial f_d is convex in the case $d = 3$ and $L = 2$. In view of Lemma 5.18 it suffices to show that the matrix Q_γ is positive semidefinite for any $\gamma \in \mathbb{N}_1^m$. Up to symmetry it suffices to show that $Q_\gamma \succeq 0$ for $\gamma = u_1$. In view of Lemma 5.21 we have

$$Q_{u_1} = \underbrace{(c_{u_1+u_i+u_j})_{i,j=1}^m}_{=M_{u_1}} \circ (Q_0 + \underbrace{(\widehat{b}_{u_1+u_i} + \widehat{b}_{u_1+u_j})_{i,j=1}^m}_{=R_{u_1}}).$$

We have shown earlier that the matrix M_{u_1} is positive semidefinite, which follows from the fact that the matrix M_2 is positive semidefinite (in view of relation (5.13), Lemma 5.13 and Corollary 5.16). Hence, if we can show that $Q_0 + R_{u_1} \succeq 0$ then

this will imply that $Q_{u_1} \succeq 0$ and conclude the proof of Theorem 5.3. In the rest of this section we show that $Q_0 + R_{u_1} \succeq 0$.

We begin with describing the entries of the matrix $Q_0 + R_{u_1}$. By definition, the entries of Q_0 (case $\gamma = 0$) are

$$(Q_0)_{ii} = 2b_{2u_i} = \frac{2}{L}, \ \ (Q_0)_{ij} = b_{u_i+u_j} = \frac{2}{|e_i \cup e_j|} \ \text{ for } i \neq j \in [m].$$

Moreover, $\widehat{b}_{2u_1} = 2b_{2u_1} = \frac{2}{L}$ and $\widehat{b}_{u_1+u_i} = b_{u_1+u_i} = \frac{2}{|e_1 \cup e_i|}$ for $i \geq 2$. Using this we obtain that

$$Q_0 + R_{u_1} = 2 \cdot \Big(\frac{1}{|e_1 \cup e_j|} + \frac{1}{|e_i \cup e_j|} + \frac{1}{|e_1 \cup e_i|} \Big)_{i,j=1}^{m} =: 2B,$$

where we define the matrix B as

$$B := \Big(\frac{1}{|e \cup f|} + \frac{1}{|e_1 \cup e|} + \frac{1}{|e_1 \cup f|} \Big)_{e,f \in E}. \tag{5.28}$$

The main result of this section is the next lemma, which shows that the matrix B (and thus $Q_0 + R_{u_1}$) is positive semidefinite. As observed above, this implies that the polynomial f_3 is convex for $L = 2$ and thus settles Conjecture 5.17 for the case $d = 3, L = 2$.

Proposition 5.22. *Assume $L = 2$. The matrix B in (5.28) is positive semidefinite.*

Before proceeding to the proof, let us make a few observations. Note that the matrix B from (5.28) can be decomposed as

$$B = \underbrace{\Big(\frac{1}{|e \cup f|} \Big)_{e,f \in E}}_{=M_0} + \underbrace{\Big(\frac{1}{|e_1 \cup e|} + \frac{1}{|e_1 \cup f|} \Big)_{e,f \in E}}_{=:R}.$$

So, $B = M_0 + R$, where $M_0 = M_\emptyset$ has been shown earlier to be positive semidefinite (recall Corollary 5.16, or note that M_0 is the matrix M from (5.6) as we are in the case $L = 2$). On the other hand, the matrix R is not positive semidefinite. In fact, R has rank 2, and it has a negative eigenvalue. One can infer from the results in Section 5.3.1 that $\lambda_{\min}(M_0) = 1/12$, while one can check that $\lambda_{\min}(R) < -1/12 \ = -0.0833...$ when $n \geq 6$ (see Table 5.1 below). Hence, in general one cannot argue that $B \succeq 0$ by simply looking at the smallest eigenvalues of its summands M_0 and R. On a very high level, we will show positive semidefiniteness of the matrix B by observing that it has a simple block structure, which

we can exploit by taking several successive Schur complements; in this way we obtain well-structured matrices that can be directly shown to be positive semidefinite. The exact details are not difficult, but a bit technically involved.

In the rest of the section we prove Proposition 5.22. To fix ideas we let e_1 be the edge $e_1 = \{1,2\}$ and to simplify notation we set $p = n-2$ and $q = \binom{n-2}{2}$. Then the index set of B can be partitioned into $\{e_1\} \cup I_1 \cup I_2 \cup I_0$, setting $I_k = \{\{k,i\} : 3 \le i \le n\}$ for $k = 1,2$, and $I_0 = \{\{i,j\} : 3 \le i < j \le n\}$. So $|I_1| = |I_2| = p$ and $|I_0| = q$. With respect to this partition one can verify that the matrix B has the following block-form:

$$B = \begin{array}{c} \\ e_1 \\ I_1 \\ I_2 \\ I_0 \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & I_1 & I_2 & I_0 \end{array} \\ \left(\begin{array}{c|c|c|c} \frac{3}{2} & \frac{7}{6}J_{1,p} & \frac{7}{6}J_{1,p} & J_{1,q} \\ \hline \frac{7}{6}J_{p,1} & J_p + \frac{1}{6}I_p & \frac{11}{12}J_p + \frac{1}{12}I_p & \frac{5}{6}J_{p,q} + \frac{1}{12}\Gamma^T \\ \hline \frac{7}{6}J_{p,1} & \frac{11}{12}J_p + \frac{1}{12}I_p & J_p + \frac{1}{6}I_p & \frac{5}{6}J_{p,q} + \frac{1}{12}\Gamma^T \\ \hline J_{q,1} & \frac{5}{6}J_{q,p} + \frac{1}{12}\Gamma & \frac{5}{6}J_{q,p} + \frac{1}{12}\Gamma & M + \frac{1}{2}J_q \end{array}\right) \end{array}.$$

Here M is the matrix from (5.6) (replacing n by $p = n-2$). We have shown in Section 5.3.1 (see relation (5.7) and Lemma 5.9) that M can be decomposed as

$$M = \frac{1}{12}I_q + \frac{1}{4}J_q + \frac{1}{12}\Gamma\Gamma^T,$$

where $\Gamma = \Gamma_p$ is the $\binom{p}{2} \times p$ matrix whose (f,i)th entry is $|\{i\} \cap f|$.

We now proceed to show that the matrix B is positive semidefinite. Note that its lower right diagonal block indexed by the set I_0 is positive semidefinite (since $M \succeq 0$). Our strategy is now to 'eliminate' the three borders indexed by the sets $\{e_1\}$, I_1 and I_2 successively, one by one, by taking Schur complements, until reaching a final matrix (indexed by I_0) whose positive semidefiniteness can be seen directly. To do the Schur complement operations we will need to invert matrices of the form $aI + bJ$. The next lemma indicates how to do that, its proof is straightforward and thus omitted.

Lemma 5.23. *For $a, b \in \mathbb{R}$ such that $a + pb \neq 0$, the matrix $aI_p + bJ_p$ is non-singular with inverse*

$$(aI_p + bJ_p)^{-1} = \frac{1}{a}\Big(I_p - \frac{b}{pb+a}\Big)J_p.$$

Now come three steps where we successively eliminate the three borders of B indexed by $\{e_1\}$, I_1 and I_2, by taking successive Schur complements with respect to the upper left corner.

Step 1: We take a first Schur complement with respect to the upper left corner of B (indexed by e_1). We call $\widetilde{B}_1$ the resulting matrix, which reads

$$\left(\begin{array}{c|c|c} J_p + \frac{1}{6}I_p & \frac{11}{12}J_p + \frac{1}{12}I_p & \frac{5}{6}J_{p,q} + \frac{1}{12}\Gamma^T \\ \hline \frac{11}{12}J_p + \frac{1}{12}I_p & J_p + \frac{1}{6}I_p & \frac{5}{6}J_{p,q} + \frac{1}{12}\Gamma^T \\ \hline \frac{5}{6}J_{q,p} + \frac{1}{12}\Gamma & \frac{5}{6}J_{q,p} + \frac{1}{12}\Gamma & \frac{1}{12}I_q + \frac{1}{12}\Gamma\Gamma^T + \frac{3}{4}J_q \end{array}\right)$$

$$-\frac{2}{3}\begin{pmatrix} \frac{7}{6}J_{p,1} \\ \frac{7}{6}J_{p,1} \\ J_{q,1} \end{pmatrix}\begin{pmatrix} \frac{7}{6}J_{1,p} & \frac{7}{6}J_{1,p} & J_{1,q} \end{pmatrix}$$

$$= \left(\begin{array}{c|c|c} \frac{5}{54}J_p + \frac{1}{6}I_p & \frac{1}{108}J_p + \frac{1}{12}I_p & \frac{1}{18}J_{p,q} + \frac{1}{12}\Gamma^T \\ \hline \frac{1}{108}J_p + \frac{1}{12}I_p & \frac{5}{54}J_p + \frac{1}{6}I_p & \frac{1}{18}J_{p,q} + \frac{1}{12}\Gamma^T \\ \hline \frac{1}{18}J_{q,p} + \frac{1}{12}\Gamma & \frac{1}{18}J_{q,p} + \frac{1}{12}\Gamma & \frac{1}{12}I_q + \frac{1}{12}\Gamma\Gamma^T + \frac{1}{12}J_q \end{array}\right).$$

Setting $B_1 = 6\widetilde{B}_1$, we obtain $B \succeq 0 \iff \widetilde{B}_1 \succeq 0 \iff B_1 \succeq 0$, where

$$B_1 = \left(\begin{array}{c|c|c} \frac{5}{9}J_p + I_p & \frac{1}{18}J_p + \frac{1}{2}I_p & \frac{1}{3}J_{p,q} + \frac{1}{2}\Gamma^T \\ \hline \frac{1}{18}J_p + \frac{1}{2}I_p & \frac{5}{9}J_p + I_p & \frac{1}{3}J_{p,q} + \frac{1}{2}\Gamma^T \\ \hline \frac{1}{3}J_{q,p} + \frac{1}{2}\Gamma & \frac{1}{3}J_{q,p} + \frac{1}{2}\Gamma & \frac{1}{2}I_q + \frac{1}{2}\Gamma\Gamma^T + \frac{1}{2}J_q \end{array}\right).$$

Step 2: We now take the Schur complement with respect to the upper left corner of B_1 (indexed by I_1), where we use Lemma 5.23 to invert it:

$$(I_p + 5/9J_p)^{-1} = I_p - 5/(5p+9)J_p.$$

After taking this Schur complement the resulting matrix $\widetilde{B}_2$ reads:

$$\widetilde{B}_2 = \left(\begin{array}{c|c} \frac{5}{9}J_p + I_q & \frac{1}{3}J_{p,q} + \frac{1}{2}\Gamma^T \\ \hline \frac{1}{3}J_{q,p} + \frac{1}{2}\Gamma & \frac{1}{2}I_q + \frac{1}{2}\Gamma\Gamma^T + \frac{1}{2}J_q \end{array}\right)$$

$$-\begin{pmatrix} \frac{1}{18}J_p + \frac{1}{2}I_p \\ \frac{1}{3}J_{q,p} + \frac{1}{2}\Gamma \end{pmatrix}\Big(I_p - \frac{5}{(5p+9)}J_p\Big)\begin{pmatrix} \frac{1}{18}J_p + \frac{1}{2}I_p & \frac{1}{3}J_{p,q} + \frac{1}{2}\Gamma^T \end{pmatrix}$$

$$= \left(\begin{array}{c|c} \frac{3}{4}I_p + \frac{11p+23}{4(5p+9)}J_p & \frac{1}{4}\Gamma^T + \frac{3p+7}{2(5p+9)}J_{p,q} \\ \hline \frac{1}{4}\Gamma + \frac{3p+7}{2(5p+9)}J_{q,p} & \frac{1}{2}I_q + \frac{1}{4}\Gamma\Gamma^T + \frac{3p+7}{2(5p+9)}J_q \end{array}\right).$$

Setting $B_2 = 4\widetilde{B}_2$ we obtain $B \succeq 0 \iff B_1 \succeq 0 \iff B_2 \succeq 0$, where

$$B_2 = \left(\begin{array}{c|c} 3I_p + \frac{11p+23}{5p+9}J_p & \Gamma^T + \frac{2(3p+7)}{5p+9}J_{p,q} \\ \hline \Gamma + \frac{2(3p+7)}{5p+9}J_{q,p} & 2I_q + \Gamma\Gamma^T + \frac{2(3p+7)}{5p+9}J_q \end{array}\right).$$

Step 3: Inverting the top left block of B_2 via Lemma 5.23 gives

$$\Big(3I_p + \frac{11p+23}{5p+9}J_p\Big)^{-1} = \frac{1}{3}I_p - \frac{(11p+23)}{3(11p^2+38p+27)}J_p.$$

Taking the third and final Schur complement with respect to this block in B_2 we get the matrix

$$B_3 := 2I_q + \Gamma\Gamma^T + \frac{2(3p+7)}{5p+9}J_q$$

$$-\Big(\Gamma^T + \frac{2(3p+7)}{5p+9}J_{q,p}\Big)\Big(\frac{1}{3}I_p - \frac{(11p+23)}{3(11p^2+38p+27)}J_p\Big)\Big(\Gamma^T + \frac{2(3p+7)}{5p+9}J_{p,q}\Big)$$

$$= 2I_q + \frac{2}{3}\Gamma\Gamma^T + \frac{2(9p+25)}{3(11p+27)}J_q.$$

It is now clear that $B_3 \succeq 0$. In turn, this implies that $B_2 \succeq 0$ and thus $B \succeq 0$, which concludes the proof of Proposition 5.22.

We conclude with an indication why the above proof seems difficult to extend to the general case $L \geq 3$. The biggest hurdle lies in the richness of the possible intersections between edges of large size. More concretely, recall that the (e,f)th entry of the matrix B in (5.28) depends on $|e \cup f|$, $|e \cup e_1|$ and $|f \cup e_1|$. So one has to take into account how the two edges e, f pairwise interact within e_1 and outside it, which becomes technically involved when $|e_1| = L$ is large. So the matrix B has an increasingly involved block structure when L grows. In addition, some blocks in B may have a form that requires an additional symmetry reduction to become amenable.

5.4.3 Some Numerical Justification for Convexity of f_d

We have carried out some numerical experiments for a range of values of d, L, n and verified that the matrices Q_γ are positive semidefinite for all $\gamma \in \mathbb{N}^n_{d-2}$ in these cases. Hence, for these values the polynomial f_d is convex and Conjecture 5.17 holds. Recall from Lemma 5.21 that the matrix Q_γ can be decomposed as

$$Q_\gamma = M_\gamma \circ \Big(\underbrace{\sum_{k\in[m]:\gamma_k\geq 1} Q_{\gamma-u_k}}_{=:B_\gamma} + R_\gamma \Big) = M_\gamma \circ (B_\gamma + R_\gamma).$$

By the results in Section 5.3 we already know that the matrix M_γ is positive semidefinite. Hence, it now suffices to show that the matrix $B_\gamma + R_\gamma$ is positive semidefinite for each $\gamma \in \mathbb{N}^n_{d-2}$. We did this in the previous section for the case $d = 3$ (and $L = 2$). We have computed the minimum eigenvalues of the matrices Q_γ, B_γ and R_γ for different values of n, d and L and give this information for the case $L = 2$ in Table 5.1 below (for $d = 3$) and in Tables 5.2-5.4 in Appendix A (for $d \geq 4$). (Further numerical results for $L \geq 3$ can be found in the arXiv version of this chapter). In each case we consider the possible different cases for selecting $\gamma \in \mathbb{N}^n_{d-2}$ up to symmetry; the different instances of γ are indicated in the column labeled γ. For instance, for $d = 3, L = 2$ there is only one possibility, say $\gamma = u_1$ corresponding to edge $e_1 = \{1, 2\}$ (see Table 5.1). For $d = 4, L = 2$ there are three possibilities: $\gamma = 2e_1$ with $e_1 = \{1, 2\}$, $\gamma = u_1 + u_2$ with $e_1 = \{1, 2\}$ and $e_2 = \{1, 3\}$, and $\gamma = u_1 + u_2$ with $e_1 = \{1, 2\}$ and $e_2 = \{3, 4\}$ (see Table 5.2).

In all cases we find that Q_γ is positive semidefinite (in fact, positive definite). As already mentioned in Section 5.4.2 for the case $d = 3$, we see that in general this cannot be deduced by considering its summands separately, since R_γ has a negative smallest eigenvalue and $\lambda_{min}(B_\gamma) + \lambda_{\min}(R_\gamma) < 0$ from a certain n (which depends on d and L). In addition, we observe that $\lambda_{\min}(B_\gamma)$ stays constant from a certain n while $\lambda_{\min}(R_\gamma)$ keeps decreasing. It remains an open problem to show that the property $Q_\gamma \succeq 0$ holds in general.

For recent progress on this problem we refer to Polak [Pol22], who proved that all the matrices Q_γ are positive semidefinite in the case $d \leq 8$ and $L = 2$. One of the difficulties is that one needs to enumerate the distinct patterns for $\gamma \in \mathbb{N}^m_{d-2}$, i.e., the number of multigraphs with $d-2$ edges. As mentioned earlier in Example 5.4 this number is given by the OEIS sequence A050535 [OEI99a], and it grows quickly with d.

5.5 Numerical Results for the Polynomials f_d

We group here Table 5.2 through Table 5.4, which show the eigenvalues of the matrices Q_γ, B_γ and R_γ for $L = 2$ and small values of n, d. To see that these are indeed exhaustive, we again refer to the OEIS sequence A050535 [OEI99a] giving the number of multigraphs with up to four edges. Note that some multigraphs can only appear if the number n of vertices is large enough.

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
3	2	3	[[1, 2]]	0.0556	0.1667	-0.0236
3	2	4	[[1, 2]]	0.0347	0.0833	-0.0478
3	2	5	[[1, 2]]	0.0347	0.0833	-0.0729
3	2	6	[[1, 2]]	0.0347	0.0833	-0.0987
3	2	7	[[1, 2]]	0.0347	0.0833	-0.1249
3	2	8	[[1, 2]]	0.0347	0.0833	-0.1514

Table 5.1: Case $d = 3, L = 2$

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
4	2	3	[[1, 2], [1, 2]]	0.0185	0.0556	-0.0415
4	2	4	[[1, 2], [1, 2]]	0.0133	0.0347	-0.0805
4	2	5	[[1, 2], [1, 2]]	0.0133	0.0347	-0.1189
4	2	6	[[1, 2], [1, 2]]	0.0133	0.0347	-0.1572
4	2	3	[[1, 3], [1, 2]]	0.0593	0.1778	-0.0028
4	2	4	[[1, 3], [1, 2]]	0.0238	0.0802	-0.0478
4	2	5	[[1, 3], [1, 2]]	0.0214	0.0743	-0.092
4	2	6	[[1, 3], [1, 2]]	0.0214	0.0741	-0.1359
4	2	7	[[1, 3], [1, 2]]	0.0214	0.074	-0.1798
4	2	4	[[3, 4], [1, 2]]	0.0174	0.0694	-0.0012
4	2	5	[[3, 4], [1, 2]]	0.0174	0.0694	-0.029
4	2	6	[[3, 4], [1, 2]]	0.0174	0.0694	-0.0565

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
4	2	7	[[3, 4], [1, 2]]	0.0174	0.0694	-0.084
4	2	8	[[3, 4], [1, 2]]	0.0174	0.0694	-0.1115
4	2	9	[[3, 4], [1, 2]]	0.0174	0.0694	-0.139

Table 5.2: Case $d = 4, L = 2$

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
5	2	3	[[1, 2], [1, 2], [1, 2]]	0.0062	0.0185	-0.0425
5	2	4	[[1, 2], [1, 2], [1, 2]]	0.0049	0.0133	-0.0804
5	2	5	[[1, 2], [1, 2], [1, 2]]	0.0049	0.0133	-0.1163
5	2	3	[[1, 3], [1, 2], [1, 2]]	0.0298	0.0894	-0.0062
5	2	4	[[1, 3], [1, 2], [1, 2]]	0.0111	0.0396	-0.0605
5	2	5	[[1, 3], [1, 2], [1, 2]]	0.0098	0.0358	-0.112
5	2	6	[[1, 3], [1, 2], [1, 2]]	0.0098	0.0358	-0.162
5	2	4	[[3, 4], [1, 2], [1, 2]]	0.0077	0.0307	-0.0085
5	2	5	[[3, 4], [1, 2], [1, 2]]	0.0072	0.0307	-0.038
5	2	6	[[3, 4], [1, 2], [1, 2]]	0.0067	0.0307	-0.0667
5	2	7	[[3, 4], [1, 2], [1, 2]]	0.0067	0.0307	-0.0948
5	2	4	[[1, 4], [1, 3], [1, 2]]	0.0263	0.1052	-0.009
5	2	5	[[1, 4], [1, 3], [1, 2]]	0.0162	0.0716	-0.0681
5	2	6	[[1, 4], [1, 3], [1, 2]]	0.0151	0.0676	-0.1255
5	2	7	[[1, 4], [1, 3], [1, 2]]	0.015	0.0675	-0.1819
5	2	8	[[1, 4], [1, 3], [1, 2]]	0.015	0.0675	-0.2374
5	2	4	[[2, 4], [1, 3], [1, 2]]	0.0188	0.0753	-0.0063
5	2	5	[[2, 4], [1, 3], [1, 2]]	0.0151	0.0678	-0.0613
5	2	6	[[2, 4], [1, 3], [1, 2]]	0.0139	0.0635	-0.1147
5	2	7	[[2, 4], [1, 3], [1, 2]]	0.0139	0.0635	-0.167
5	2	8	[[2, 4], [1, 3], [1, 2]]	0.0139	0.0635	-0.2186
5	2	5	[[2, 3], [1, 5], [1, 4]]	0.0114	0.0571	-0.0053
5	2	6	[[2, 3], [1, 5], [1, 4]]	0.0113	0.0569	-0.0395

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
5	2	7	[[2, 3], [1, 5], [1, 4]]	0.0107	0.0569	-0.0731
5	2	8	[[2, 3], [1, 5], [1, 4]]	0.0107	0.0569	-0.1062
5	2	9	[[2, 3], [1, 5], [1, 4]]	0.0107	0.0569	-0.1391
5	2	3	[[2, 3], [1, 3], [1, 2]]	0.0926	0.2778	-0.0
5	2	4	[[2, 3], [1, 3], [1, 2]]	0.0237	0.085	-0.0967
5	2	5	[[2, 3], [1, 3], [1, 2]]	0.0212	0.0764	-0.1882
5	2	6	[[2, 3], [1, 3], [1, 2]]	0.0212	0.0764	-0.2766
5	2	6	[[5, 6], [3, 4], [1, 2]]	0.0087	0.0521	-0.0011
5	2	7	[[5, 6], [3, 4], [1, 2]]	0.0087	0.0521	-0.0233
5	2	8	[[5, 6], [3, 4], [1, 2]]	0.0087	0.0521	-0.0452
5	2	9	[[5, 6], [3, 4], [1, 2]]	0.0087	0.0521	-0.067
5	2	10	[[5, 6], [3, 4], [1, 2]]	0.0087	0.0521	-0.0885
5	2	11	[[5, 6], [3, 4], [1, 2]]	0.0087	0.0521	-0.11

Table 5.3: Case $d = 5, L = 2$

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
6	2	3	[[1, 2], [1, 2], [1, 2], [1, 2]]	0.0021	0.0062	-0.0349
6	2	4	[[1, 2], [1, 2], [1, 2], [1, 2]]	0.0017	0.0049	-0.0652
6	2	5	[[1, 2], [1, 2], [1, 2], [1, 2]]	0.0017	0.0049	-0.0931
6	2	3	[[1, 3], [1, 2], [1, 2], [1, 2]]	0.0124	0.0371	-0.0094
6	2	4	[[1, 3], [1, 2], [1, 2], [1, 2]]	0.0044	0.0165	-0.0579
6	2	5	[[1, 3], [1, 2], [1, 2], [1, 2]]	0.004	0.0148	-0.1029
6	2	6	[[1, 3], [1, 2], [1, 2], [1, 2]]	0.004	0.0148	-0.1457
6	2	3	[[1, 3], [1, 3], [1, 2], [1, 2]]	0.0261	0.0785	-0.0016
6	2	4	[[1, 3], [1, 3], [1, 2], [1, 2]]	0.0064	0.0237	-0.0626
6	2	5	[[1, 3], [1, 3], [1, 2], [1, 2]]	0.0057	0.0211	-0.1193
6	2	6	[[1, 3], [1, 3], [1, 2], [1, 2]]	0.0057	0.0211	-0.1732
6	2	4	[[3, 4], [1, 2], [1, 2], [1, 2]]	0.0031	0.0125	-0.0141
6	2	5	[[3, 4], [1, 2], [1, 2], [1, 2]]	0.0026	0.0124	-0.0384

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
6	2	6	[[3, 4], [1, 2], [1, 2], [1, 2]]	0.0024	0.0115	-0.0616
6	2	4	[[3, 4], [3, 4], [1, 2], [1, 2]]	0.0038	0.0153	-0.0007
6	2	5	[[3, 4], [3, 4], [1, 2], [1, 2]]	0.0036	0.0153	-0.0255
6	2	6	[[3, 4], [3, 4], [1, 2], [1, 2]]	0.0033	0.0153	-0.0492
6	2	7	[[3, 4], [3, 4], [1, 2], [1, 2]]	0.0033	0.0153	-0.0721
6	2	4	[[1, 4], [1, 3], [1, 2], [1, 2]]	0.0147	0.0589	-0.0142
6	2	5	[[1, 4], [1, 3], [1, 2], [1, 2]]	0.0084	0.0386	-0.0771
6	2	6	[[1, 4], [1, 3], [1, 2], [1, 2]]	0.0078	0.036	-0.1372
6	2	7	[[1, 4], [1, 3], [1, 2], [1, 2]]	0.0078	0.036	-0.1952
6	2	4	[[2, 4], [1, 3], [1, 2], [1, 2]]	0.0129	0.0514	-0.0151
6	2	5	[[2, 4], [1, 3], [1, 2], [1, 2]]	0.0079	0.037	-0.0766
6	2	6	[[2, 4], [1, 3], [1, 2], [1, 2]]	0.0073	0.0344	-0.1352
6	2	7	[[2, 4], [1, 3], [1, 2], [1, 2]]	0.0073	0.0344	-0.1919
6	2	4	[[2, 4], [1, 3], [1, 3], [1, 2]]	0.0102	0.0407	-0.0089
6	2	5	[[2, 4], [1, 3], [1, 3], [1, 2]]	0.0074	0.0343	-0.064
6	2	6	[[2, 4], [1, 3], [1, 3], [1, 2]]	0.0068	0.0318	-0.1167
6	2	7	[[2, 4], [1, 3], [1, 3], [1, 2]]	0.0068	0.0318	-0.1675
6	2	5	[[2, 3], [1, 5], [1, 4], [1, 4]]	0.0059	0.0294	-0.0148
6	2	6	[[2, 3], [1, 5], [1, 4], [1, 4]]	0.0052	0.0293	-0.0485
6	2	7	[[2, 3], [1, 5], [1, 4], [1, 4]]	0.0049	0.0278	-0.0813
6	2	8	[[2, 3], [1, 5], [1, 4], [1, 4]]	0.0049	0.0278	-0.1132
6	2	5	[[2, 3], [2, 3], [1, 5], [1, 4]]	0.0053	0.0266	-0.0055
6	2	6	[[2, 3], [2, 3], [1, 5], [1, 4]]	0.0047	0.0259	-0.0344
6	2	7	[[2, 3], [2, 3], [1, 5], [1, 4]]	0.0044	0.0249	-0.0623
6	2	8	[[2, 3], [2, 3], [1, 5], [1, 4]]	0.0044	0.0249	-0.0897
6	2	3	[[2, 3], [1, 3], [1, 2], [1, 2]]	0.0525	0.1574	-0.0017
6	2	4	[[2, 3], [1, 3], [1, 2], [1, 2]]	0.0125	0.0467	-0.1176
6	2	5	[[2, 3], [1, 3], [1, 2], [1, 2]]	0.0112	0.0416	-0.2252
6	2	6	[[2, 3], [1, 3], [1, 2], [1, 2]]	0.0112	0.0416	-0.3274
6	2	6	[[5, 6], [3, 4], [1, 2], [1, 2]]	0.0038	0.023	-0.0086
6	2	7	[[5, 6], [3, 4], [1, 2], [1, 2]]	0.0035	0.0229	-0.0278
6	2	8	[[5, 6], [3, 4], [1, 2], [1, 2]]	0.0033	0.022	-0.0466

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
6	2	9	[[5, 6], [3, 4], [1, 2], [1, 2]]	0.0033	0.022	-0.065
6	2	5	[[1, 5], [1, 4], [1, 3], [1, 2]]	0.0208	0.104	-0.0167
6	2	6	[[1, 5], [1, 4], [1, 3], [1, 2]]	0.0121	0.066	-0.0852
6	2	7	[[1, 5], [1, 4], [1, 3], [1, 2]]	0.0115	0.063	-0.1515
6	2	8	[[1, 5], [1, 4], [1, 3], [1, 2]]	0.0115	0.0629	-0.2164
6	2	5	[[2, 3], [1, 5], [1, 4], [1, 2]]	0.0141	0.0706	-0.0132
6	2	6	[[2, 3], [1, 5], [1, 4], [1, 2]]	0.0107	0.0592	-0.0743
6	2	7	[[2, 3], [1, 5], [1, 4], [1, 2]]	0.0101	0.0562	-0.1336
6	2	8	[[2, 3], [1, 5], [1, 4], [1, 2]]	0.0101	0.0562	-0.1915
6	2	6	[[2, 3], [1, 6], [1, 5], [1, 4]]	0.0084	0.0505	-0.0119
6	2	7	[[2, 3], [1, 6], [1, 5], [1, 4]]	0.0079	0.0503	-0.0503
6	2	8	[[2, 3], [1, 6], [1, 5], [1, 4]]	0.0075	0.049	-0.0879
6	2	9	[[2, 3], [1, 6], [1, 5], [1, 4]]	0.0075	0.0489	-0.1248
6	2	4	[[2, 3], [1, 4], [1, 3], [1, 2]]	0.0246	0.0985	-0.0122
6	2	5	[[2, 3], [1, 4], [1, 3], [1, 2]]	0.0162	0.0746	-0.1185
6	2	6	[[2, 3], [1, 4], [1, 3], [1, 2]]	0.0151	0.0695	-0.2198
6	2	7	[[2, 3], [1, 4], [1, 3], [1, 2]]	0.0151	0.0695	-0.3177
6	2	4	[[3, 4], [2, 4], [1, 3], [1, 2]]	0.0204	0.0815	-0.0003
6	2	5	[[3, 4], [2, 4], [1, 3], [1, 2]]	0.014	0.0644	-0.0946
6	2	6	[[3, 4], [2, 4], [1, 3], [1, 2]]	0.0129	0.0594	-0.1846
6	2	7	[[3, 4], [2, 4], [1, 3], [1, 2]]	0.0129	0.0594	-0.2716
6	2	6	[[2, 6], [2, 3], [1, 5], [1, 4]]	0.0077	0.046	-0.005
6	2	7	[[2, 6], [2, 3], [1, 5], [1, 4]]	0.0074	0.0456	-0.0392
6	2	8	[[2, 6], [2, 3], [1, 5], [1, 4]]	0.0071	0.0456	-0.0727
6	2	9	[[2, 6], [2, 3], [1, 5], [1, 4]]	0.0071	0.0456	-0.1055
6	2	5	[[2, 5], [2, 3], [1, 4], [1, 3]]	0.0121	0.0603	-0.0084
6	2	6	[[2, 5], [2, 3], [1, 4], [1, 3]]	0.01	0.055	-0.0643
6	2	7	[[2, 5], [2, 3], [1, 4], [1, 3]]	0.0094	0.0523	-0.1184
6	2	8	[[2, 5], [2, 3], [1, 4], [1, 3]]	0.0094	0.0523	-0.1712
6	2	6	[[5, 6], [2, 4], [1, 3], [1, 2]]	0.0077	0.0461	-0.0096
6	2	7	[[5, 6], [2, 4], [1, 3], [1, 2]]	0.0073	0.0461	-0.0451
6	2	8	[[5, 6], [2, 4], [1, 3], [1, 2]]	0.007	0.0457	-0.0799

d	L	n	γ	$\lambda_{min}(Q_\gamma)$	$\lambda_{min}(B_\gamma)$	$\lambda_{min}(R_\gamma)$
6	2	9	[[5, 6], [2, 4], [1, 3], [1, 2]]	0.007	0.0457	-0.114
6	2	7	[[4, 5], [2, 3], [1, 7], [1, 6]]	0.0057	0.0399	-0.0047
6	2	8	[[4, 5], [2, 3], [1, 7], [1, 6]]	0.0056	0.0399	-0.0276
6	2	9	[[4, 5], [2, 3], [1, 7], [1, 6]]	0.0053	0.0398	-0.0501
6	2	10	[[4, 5], [2, 3], [1, 7], [1, 6]]	0.0053	0.0398	-0.0723
6	2	5	[[4, 5], [2, 3], [1, 3], [1, 2]]	0.0121	0.0606	-0.0192
6	2	6	[[4, 5], [2, 3], [1, 3], [1, 2]]	0.0112	0.0606	-0.0784
6	2	7	[[4, 5], [2, 3], [1, 3], [1, 2]]	0.0106	0.0594	-0.1359
6	2	8	[[4, 5], [2, 3], [1, 3], [1, 2]]	0.0106	0.0594	-0.1919
6	2	8	[[7, 8], [5, 6], [3, 4], [1, 2]]	0.0043	0.0347	-0.0008
6	2	9	[[7, 8], [5, 6], [3, 4], [1, 2]]	0.0043	0.0347	-0.0161
6	2	10	[[7, 8], [5, 6], [3, 4], [1, 2]]	0.0043	0.0347	-0.0312
6	2	11	[[7, 8], [5, 6], [3, 4], [1, 2]]	0.0043	0.0347	-0.0462

Table 5.4: Case $d = 6, L = 2$

6

Four Different Views at Flag Algebras

Flag algebras, first introduced by Alexander Razborov in 2007 [Raz07], unify and generalize previous ideas of numerous authors, and remain one of the most powerful tools in extremal combinatorics to date. While initially focused on extremal (hyper-)graph theory (see, for example, Razborov's own survey [Raz13] for some early results), it has since found applications for various other extremal settings. For example, the theory has been applied to limits of permutations (also known as Permutons) [SS18], of (point) order types [Goa+18], and of oriented graphs [Gil+19].

Recently, a fascinating connection to *polynomial optimization* in the case of (hyper-)graphs was found by Raymond, Saunderson, Singh and Thomas [Ray+18]. The authors showed that one can recover the optimization hierarchies used to compute flag sums-of-squares certificates by considering a *sequence of symmetric polynomial optimization problems*, and exploiting its symmetries partially. Chapters 8 and 9 investigate these connections further: We generalize the approach to arbitrary flag algebras, we fully exploit the symmetry of two different optimization hierarchies, and we generalize the way these hierarchies can converge in the limit. In the process we generalize notions coming from the representation theory of the symmetric group in Chapter 8, which includes a generalized notion of *Young-tableau* for different types of flags, as well as com-

binatorial algorithms to work with these computationally. Finally, we provide an extensive Julia software package implementing these findings, and provide some examples.

This chapter focuses on flag algebras themselves: We explain the basics of flag algebras in the *non-induced setting*, to our knowledge making it work for the first time in the literature for arbitrary flag algebras. We compare the non-induced and the more commonly used induced setting (which we will call *Razborov setting*), provide a general translation of induced and non-induced flags to (limits of) polynomials, and provide a novel "Character" basis for flag algebras.

6.1 What is a (Non-Induced) Flag Algebra?

We will here introduce some relevant details concerning flag algebras. In contrast to Razborov [Raz07], we will prioritize the setting of *non-induced* sub-flag densities (to be defined in a moment). This simplifies many proofs, formulas and theorems of Razborov considerably, and was, for example, already used in the setting of graphs by the authors of [Ray+18]. At the same time, it is only a Möbius transformation away from Razborov's setting, so translating between the two settings is straightforward, as we will see in Section 6.4.

Triangle Free Graphs. We start with probably the most common example, and one of first milestones in extremal graph theory, known as Mantel's theorem [Man07], which answers the question: What is the maximum edge-density in a triangle free graph, as the number of vertices approaches infinity?

Here we are working with an *increasing sequence* of graphs $\mathcal{G} = (G_n)_{n=1}^{\infty}$, where each graph G_n has more vertices than the graph G_{n-1} before. To formalize the problem, we first have to define the *subgraph density* of a given graph H in this sequence. We choose to define the density of $H = (V(H), E(H))$, where $V(H) = [k] = \{1, \dots, k\}$, $E(H) \subseteq \binom{[k]}{2}$ in $\mathcal{G}$ as

$$\begin{aligned} p(\mathcal{G}, H) &:= \lim_{n\to\infty} \mathbb{P}[\{\boldsymbol{\sigma_n}(i), \boldsymbol{\sigma_n}(j)\} \in E(G_n) \ \ \forall \{i,j\} \in E(H)] \\ &= \lim_{n\to\infty} \mathbb{P}[\boldsymbol{\sigma_n}(H) \text{ is a subgraph of } G_n] \\ &= \lim_{n\to\infty} \mathbb{P}[\boldsymbol{\sigma_n}(H) \subseteq G_n] \in [0,1], \end{aligned}$$

where $\boldsymbol{\sigma_n}$ is a permutation of vertices of G_n, chosen uniformly at random. We use the notation $G \subseteq G'$ for denoting that G is a (non-induced) subgraph of G', i.e.

$V(G) \subseteq V(G')$ and $E(G) \subseteq E(G')$. By definition, this is the limit of the probabilities that, given a random way to embed H in G_n, this results in a (non-induced) subgraph of G_n. Non-induced simply means that we may remove edges from G_n to obtain H this way. While generally the limit $p(\mathcal{G}, H)$ does not exist, we can restrict ourselves to *convergent sequences*, in the sense of the following proposition.

Proposition 6.1. *Let $\mathcal{G} = (G_n)_{n=1}^{\infty}$ be an increasing sequence of graphs. Then there exists an increasing subsequence $\hat{\mathcal{G}}$ of $\mathcal{G}$ such that the limit $p(\hat{\mathcal{G}}, H)$ exists for every graph H.*

Proof. Consider the sequence $(d_n)_{n=1}^{\infty}$, where each d_n is the (countably-infinite dimensional) vector

$$d_n = (\mathbb{P}[\boldsymbol{\sigma_n}(H) \text{ is a subgraph of } G_n])_{H \text{ graph}} \in [0,1]^{\aleph_0}.$$

By Tychonoff's theorem [Tyc35] products of compact spaces are compact, and the countably infinite product of the interval (also known as *Hilbert Cube* [SS78]) is metrizable, implying that $[0,1]^{\aleph_0}$ is sequentially compact. Thus, there exists a convergent subsequence of $(d_n)_{n=1}^{\infty}$, and the corresponding subsequence of graphs $\hat{\mathcal{G}}$ has the property we want to show. □

From now on, we are always working with increasing sequences of graphs, which converge in the sense of Proposition 6.1, whenever we mention graph sequences.

This kind of density is also called (see [Ray+18]) the limit of the *injective density* of H as a (not necessarily induced) subgraph in $\mathcal{G}$. Injective means here that the random embedding is injective (one to one), and no two vertices of H end up on the same vertex of the G_n.

We can now formalize the problem of maximizing the edge-density in triangle free graphs (in the limit) as

$$\text{ex}(\overset{\bullet}{\bullet}\overset{\bullet}{\bullet}; \triangle) := \max_{\mathcal{G}} \{p(\mathcal{G}, \overset{\bullet}{\bullet}\overset{\bullet}{\bullet}) : p(\mathcal{G}, \triangle) = 0\},$$

where the maximum is taken over all (increasing and convergent) sequences $\mathcal{G} = (G_n)_{n=1}^{\infty}$ of graphs.

An Easy Lower Bound. One can easily find a good lower bound for this problem: We split the vertices into two sets with an (almost) equal number of vertices, and fully connect the two subsets, to construct the complete bipartite graph $G_n = K_{\lceil \frac{n}{2} \rceil, \lfloor \frac{n}{2} \rfloor}$. This sequence of graphs is obviously triangle free, and its edge density approaches

$$\lim_{n\to\infty} \frac{\lceil \frac{n}{2} \rceil \lfloor \frac{n}{2} \rfloor}{\binom{n}{2}} = \frac{1}{2},$$

thus ex( ; ) is at least $\frac{1}{2}$.

6.2 Upper Bounds via Flag Sums-of-Squares

But how can one prove that this bound, and thus this construction, is optimal? Here *Flag algebras* come into play. We first note that the product of two densities is equal to the density of the graph obtained by *gluing* the two graphs together on distinct vertices.

Proposition 6.2. *Let H_1, H_2 be graphs, and $\mathcal{G}$ be an (increasing, convergent) graph sequence. Then*

$$p(\mathcal{G}, H_1)p(\mathcal{G}, H_2) = p(\mathcal{G}, H_1 \cup H_2),$$

where $H_1 \cup H_2$ denotes the disjoint union *of the graphs H_1 and H_2.*

Proof. By definition, we have

$$\begin{aligned} p(\mathcal{G}, H_1)p(\mathcal{G}, H_2) &= \lim_{n\to\infty} \mathbb{P}[\boldsymbol{\sigma_n}(H_1) \subseteq G_n] \cdot \mathbb{P}[\boldsymbol{\sigma'_n}(H_2) \subseteq G_n] \\ &= \lim_{n\to\infty} \mathbb{P}[\boldsymbol{\sigma_n}(H_1) \subseteq G_n \wedge \boldsymbol{\sigma'_n}(H_2) \subseteq G_n]. \end{aligned}$$

As n increases, the probability that the two graphs H_1 and H_2 are sent to distinct sets of vertices approaches one:

$$\lim_{n\to\infty} \mathbb{P}[V(\boldsymbol{\sigma_n}(H_1)) \cap V(\boldsymbol{\sigma'_n}(H_2)) = \emptyset] = 1.$$

By the law of total probability, this implies that

$$p(\mathcal{G}, H_1)p(\mathcal{G}, H_2) = \lim_{n\to\infty} \mathbb{P}[\boldsymbol{\sigma_n}(H_1 \cup H_2) \subseteq G_n] = p(\mathcal{G}, H_1 \cup H_2).$$

□

For example, we have

$$p(\mathcal{G},\ )p(\mathcal{G},\ ) = p(\mathcal{G},\ ), \tag{6.1}$$

the product of the edge and triangle density is the density of the graph on five vertices with a disjoint edge and triangle.

We can extend the definitions of density and product to *partially labeled* (or *flagged*) graphs, which we call *flags*. The idea is to fix some entries of the random permutation $\boldsymbol{\sigma_n}$, calling these fixed vertices the *labeled vertices* of a flag. For instance, the density $p\left(\mathcal{G},\ \right)$, where $\ $ is a flag with one labeled vertex, is

$$\begin{aligned} p\left(\mathcal{G},\ \right) &= \lim_{n\to\infty} \mathbb{P}\left[\boldsymbol{\sigma_n}\left(\ \right) \text{ is a subgraph of } G_n \mid \boldsymbol{\sigma_n}(1) = 1\right] \\ &= \lim_{n\to\infty} \frac{\deg_{G_n}(1)}{n-1}, \end{aligned}$$

the limit of the (relative) degree of the vertex 1 in the sequence of graphs.

Formally, we define a *flag* $F = (H, f)$ to be a graph H together with an injective function $f : W \to \mathbb{N}$, which assigns labels to a subset of vertices $W \subseteq V(H)$. The density of a flag F in a sequence of graphs is then defined analogously as

$$p(\mathcal{G}, F) \coloneqq \lim_{n\to\infty} \mathbb{P}[\boldsymbol{\sigma_n}(H) \subseteq G_n \mid \boldsymbol{\sigma_n}(w) = f(w) \forall w \in W] \in [0,1],$$

where we can assume convergence as before in the unlabeled case. If $W = \emptyset$, we identify $F = (H, f) = H$.

Accordingly, the product of densities of two flags glues the vertices with the same label on top of each other, and unlabeled vertices to distinct vertices.

Proposition 6.3. *Let $F_1 = (H_1, f_1 : W_1 \to \mathbb{N})$, $F_2 = (H_2, f_2 : W_1 \to \mathbb{N})$ be flags, and $\mathcal{G}$ be an (increasing, convergent) graph sequence. Then*

$$p(\mathcal{G}, F_1)p(\mathcal{G}, F_2) = p(\mathcal{G}, F_1 \cdot F_2),$$

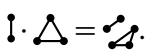
where $F_1 \cdot F_2$ denotes the gluing operator. *The flag $F_1 \cdot F_2$ is obtained by first taking the disjoint union $H_1 \cup H_2$ of the graphs, adopting both of their labels, and then merging vertices $v_1 \in V(H_1)$, $v_2 \in V(H_2)$ if they have the same labels $f_1(v_1) = f_2(v_2)$.*

Proof. Analogous to the proof of Proposition 6.2. □

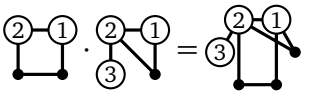
The gluing rules in Propositions 6.2 and 6.3 are independent of the sequences $\mathcal{G}$. This allows us to drop the sequence $\mathcal{G}$ and p in the notation, using the flags themselves to denote the functions which assigns graph sequences the densities of the corresponding flags. I.e. the flag F from now on denotes the function

$$F(\mathcal{G}) \coloneqq p(\mathcal{G}, F).$$

Indeed, the product rules such as equation (6.1) are independent of the graph sequence $\mathcal{G}$, so we can write (6.1) as



An example of Proposition 6.3, gluing (multiplying) two flags, is



Note that double edges are merged, as we are working with simple graphs.

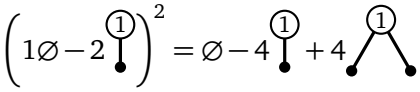
A useful observation here is that the density of the graph consisting of a single isolated vertex is identical to one in any graph, so adding isolated vertices to a graph does not change its density in the limit:

$$G \cdot \bullet = G.$$

This allows us to eliminate appearing graphs with isolated vertices by replacing them with the corresponding graphs without these vertices.

We can now consider real linear combinations of flags, sometimes also called *quantum graphs* [Lov12]. The set of all quantum graphs forms a real vector space of functions assigning graph sequences a real number, with a basis given by all flags without isolated vertices. Together with the gluing operation this vector space forms an algebra, called the *flag algebra of graphs* $\mathcal{A}_{\text{graph}}$. We say that a quantum graph $F \in \mathcal{A}_{\text{graph}}$ is *nonnegative*, denoted $F \geq 0$, if $F(\mathcal{G}) \geq 0$ for all (increasing, convergent) graph sequences $\mathcal{G}$.

Since quantum graphs are functions assigning graph sequences real numbers, squaring a quantum graph always results in a nonnegative quantum graph. For example, we have



where $\varnothing$ denotes the density of the empty graph, which is always 1, as a consequence of the definition of subgraph densities.

The last operation we still need is the *unlabeling operator*. This operator, denoted by $[\![\cdot]\!]$, *averages* a quantum graph over all possible choices of labels, effectively removing all labels from vertices. Given a flag $F = (H, f)$, we define the unlabeling operator as

$$[\![F]\!] := \mathbb{E}[(H, \pi \circ f)] = H,$$

where π is a random permutation in S_∞. We extend the operator linearly to quantum graphs.

Since averaging over nonnegative numbers results in a nonnegative number, the unlabeling operator preserves nonnegativity. Thus, we have

$$\left[\!\!\left[\left(1\varnothing - 2 \right)^2 \right]\!\!\right] = \left[\!\!\left[\varnothing - 4 + 4 \right]\!\!\right] = \varnothing - 4 + 4 \geq 0, \tag{6.2}$$

and, analogously,

$$\left[\!\!\left[\left(- + - \right)^2 \right]\!\!\right] = - 2 + \geq 0. \tag{6.3}$$

Combining the squares (6.2) and (6.3) into one sum-of-squares term we prove

$$\frac{1}{2} - + = \left[\!\!\left[\frac{1}{2} \left(1\varnothing - 2 \right)^2 \right]\!\!\right] + \left[\!\!\left[\left(- + - \right)^2 \right]\!\!\right] \geq 0.$$

Thus, we obtain $\frac{1}{2} - \geq 0$ (as we are in the case of triangle free graphs). This shows that the edge density is at most $\frac{1}{2}$ in triangle free graphs in the limit, and that the lower bound construction coming from complete bipartite graphs is sharp (in the limit).

As in the polynomial optimization case (6.9), we can restrict the appearing flags to obtain semidefinite programming hierarchies to solve flag SOS computationally. There are two naive approaches to truncate the problems for flag algebras: We can either restrict the *number of edges* in the appearing products $H \cdot H'$ between flags, which we call the *Lasserre hierarchy for flag SOS*, and investigate

in Chapter 8. Alternatively, we can also restrict the *number of vertices* in the products $H \cdot H'$, which is more commonly used for flag SOS [FV13], and which we call *Razborov's hierarchy for flag SOS*. This second hierarchy we investigate in Chapter 9.

6.3 General Flag Algebras

Razborov defines flag algebras in a much more general setting, in the language of finite model theory (see, for example, [HS18] for an introduction). To quote, he assumes we are working with a "universal first-order theory T with equality in a language L containing only predicate symbols; we assume T has infinite models."

Basically, this says that our finite models M (for example a graph) are defined on a base set (vertices) $V(M)$. We have a set of predicates $P_i^M : V(M)^{m_i} \to \{0,1\}$, taking m_i arguments each, and thus defining *relations* between vertices. In the theory of graphs, for example, we have one predicate $P_{\text{edge}}(i,j)$ indicating which edges appear in the graph. The set of all true relations in a model M we call $E(M)$, as it generalizes the edge set of graphs. Having no constant symbols implies that each vertex essentially has the same role (which is important, as this implies a certain S_n symmetry), and equality allows us to differentiate between them (as well as enforcing injectivity of homomorphisms). No function symbols and the existence of infinite models imply existence of models of every cardinality (of $V(M)$), and that each subset of vertices induces a sub-model.

Submodels are defined as the restriction of $V(M)$ to a subset $W \subseteq V(M)$ of vertices, while keeping all values of predicates (i.e., relations) intact, as long as all arguments come from within W. Two models M and N are *isomorphic*, if they have the same number of vertices, and there exists a bijection $\sigma : V(M) \to V(N)$ such that $P_i^M(x_1,\dots,x_{m_i}) = P_i^N(\sigma(x_1),\dots,\sigma(x_{m_i}))$ for each choice of vertices $x_j \in V(M)$.

For example, in the case of graphs we have one predicate P_{edge} taking two arguments i,j, and $P_{\text{edge}}(i,j) = 1$ if $\{i,j\}$ is an edge of our graph. The axioms here are $P_{\text{edge}}(i,j) = P_{\text{edge}}(j,i)$ (undirected graphs) and $P_{\text{edge}}(i,i) = 0$ (no loops).

Given a sequence of models $\mathcal{M} = (M_n)_{n=1}^{\infty}$, where $|V(M_n)| = n$, Razborov defines limits of submodel densities as

$$p_{\text{ind}}(\mathcal{M}, N) \coloneqq \lim_{n\to\infty} \mathbb{P}[N \simeq M_n|_{\mathbf{V_n}}]$$

where, as before in the case of graphs, $\mathbf{V_n}$ is a random subset of $V(M)$ with $|V(N)|$ elements. Note here again that Razborov chooses to count the density of N as

induced submodel, and the induced submodel $M_n|_{\mathbf{V_n}}$ only has to be *isomorphic*, not equal to the model N.

We, instead, choose to work with non-induced submodel densities, defined analogously to the graph case before as

$$p(\mathcal{M}, N) \coloneqq \lim_{n\to\infty} \mathbb{P}[P_i^N(x) \Rightarrow P_i^{M_n}(\boldsymbol{\sigma_n}(x)) \forall x \in V(N)^{m_i}, i] \in [0,1],$$

where $\boldsymbol{\sigma_n}$ is an injective embedding $V(N) \to V(M_n)$, chosen uniformly at random.

As before, gluing together two models glues them together on distinct vertices, setting relations involving vertices from both models to zero. In the induced setting the product of two models results in a linear combination containing every possible combination of relations involving both model's vertices, while we here simply obtain one new model. We extend the gluing operation to partially labeled models, called *Flags*, in the way as we did before for graphs. Unlabeling non-induced flags is again simply done by removing the labels from all labeled vertices.

We may still assume that a number of (potentially infinite) axioms hold in our theory. As we chose to work in the non-induced setting, and axioms are (usually) formulated for induced subgraphs, we will need a bit of notation not appearing in the literature: If a flag fulfills all axioms, we call it *valid*, otherwise *incomplete*. If a flag is *incomplete*, and does not have a *valid completion* (as defined in Section 6.4), we call it *invalid*.

6.3.1 Razborov's Flag Algebras

Razborov's flag algebras [Raz07] work slightly different from what we have introduced here. The main difference is that he chooses to work with *induced, injective* submodel densities, defined as

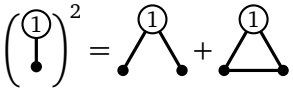
$$p_{\text{ind}}(\mathcal{G}, H) \coloneqq \lim_{n\to\infty} \mathbb{P}[H \simeq G_n|_{\mathbf{V_n}}],$$

where $\mathbf{V_n}$ is a random subset of $|V(H)| = k$ vertices of G_n, $A|_B$ denotes the submodel of A with vertices B, and $\simeq$ denotes model-isomorphism. This is the probability that given a subset of vertices of G_n, the induced submodel is isomorphic to H.

Both approaches are equivalent in the sense that knowing all subgraph densities in one setting, one can calculate the densities in the other using a *Möbius transformation*, as we will see in Section 6.4. Being able to swap between the

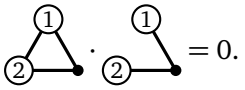
Razborov (i.e. induced) and non-induced settings will be quite useful later to translate problems formulated in induced densities, as well as simplify one of the hierarchies considered in Chapter 9, which is why we will introduce the basic operations later in Section 6.4.

The gluing operation defined by Razborov [Raz07], requires the glued together flags to have the same set of labels. As he is working with induced densities, this results in a more complicated gluing action, as we need to consider all possible ways that edges can be added between vertices of the two graphs, as well as scalars that depend on the automorphisms of the flags. For example, we obtain the product



in Razborov's algebra.

The main advantage of this definition is that the product between flags that differ on the labeled parts is always zero, for example



This splits flag algebras into many subalgebras, one for each fully labeled graph, which Razborov calls *types*. In Chapter 9 we investigate a setting closer to his and exploit the symmetry there.

The unlabeling operator $[\![F]\!]$ is more complicated in Razborov's setting as well, as one has to introduce additional factors due to changing sizes of automorphism groups. In the non-induced setting this operator is as simple as deleting the labels from all appearing graphs.

There are two more technical differences in the definitions of Razborov [Raz07]: He excludes a forbidden family of graphs $\mathcal{H}$ from the start, and, instead of optimizing over sequences of graphs, he works with limit functionals. A function ϕ, which assigns every $\mathcal{H}$-free finite graph a real number in $[0,1]$ is called a *limit functional*, if there is an (increasing, convergent) sequence of $\mathcal{H}$-free graphs $\mathcal{G}$ such that

$$\phi(F) = F(\mathcal{G})$$

for every $\mathcal{H}$-free graph F. The product rules can then be defined in terms of limit functionals $\phi(F_1)\phi(F_2) = \phi(F_1 \cdot F_2)$. In practice these conventions do not change much when working with flag algebras, as limit functionals are equivalent to (limits of) graph sequences, and graphs can be excluded later on in a straightforward manner.

6.4 Möbius Transformations: Swapping between Induced and Non-Induced Densities

Möbius transformations have proven their usefulness for transforming various problems in combinatorics and combinatorial optimization. See, for instance, Appendix 1 in [Lov12] for the general theory, Definition 3.14 in [Ray+18] for the case of graph densities, and [Lau03] for an application to the Lasserre hierarchy of relaxations of problems in binary variables.

The results from [Ray+18] easily generalize to the case of flags of arbitrary theories. We can define linear transformations between non-induced and induced densities. Since a missing "edge" in the non-induced setting allows the edge to be there in the randomly chosen subgraph, we need to consider all induced models we can obtain by adding edges to the model. For example, the non-induced density of paths  is the sum of the (appropriately scaled) induced densities of the path  and the triangle  .

Thus, to swap from non-induced to induced densities, we sum over all possible combinations to add additional relations to the flag (6.4). Since Razborov defines the density of a flag as "sub-flag is *isomorphic*" instead of "sub-flag is *identical*", we have to additionally rescale induced densities with the number of ways to draw the flags, i.e., $\frac{|S_{|V(M)|}|}{|\mathrm{Aut}(M)|} = \frac{|V(M)|!}{|\mathrm{Aut}(M)|}$. In the rare cases that we are working with both induced and non-induced flags, we add a subscript M_{ind} resp., $M_{\mathrm{non-ind}}$ specifying in which setting the densities lie. We have the following relations:

$$M_{\mathrm{non-ind}} = \sum_{\substack{\tilde{M}\,:\,V(M)=V(\tilde{M}),\\ E(M)\subseteq E(\tilde{M})}} \frac{|\mathrm{Aut}(\tilde{M})|}{|V(\tilde{M})|!}\tilde{M}_{\mathrm{ind}}. \tag{6.4}$$

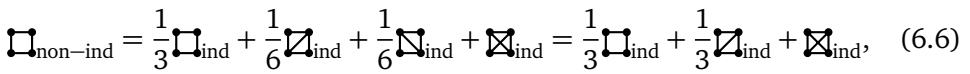
The inverse of this operation is given by

$$M_{\mathrm{ind}} = \frac{|V(M)|!}{|\mathrm{Aut}(M)|} \sum_{\substack{\tilde{M}\,:\,V(M)=V(\tilde{M}),\\ E(M)\subseteq E(\tilde{M})}} (-1)^{|E(\tilde{M})|-|E(M)|}\tilde{M}_{\mathrm{non-ind}}. \tag{6.5}$$

The linear transformation (6.4) is the operation of the *Zeta matrix* of the lattice of all (not necessarily valid) flags of size $|V(M)|$, and its inverse (6.5) is known as the *Möbius transformation* of the same lattice. This way we can see the sets of induced and non-induced flags as different bases of the same flag algebra.

For example, to express the non-induced C_4 density in terms of induced den-

sities, we determine



and, similarly, to express the induced C_4 density in non-induced terms

The first equation (6.6) should be understood as

$$p(\mathcal{G}, \square) = \frac{1}{3} p_{\text{ind}}(\mathcal{G}, \square) + \frac{1}{3} p_{\text{ind}}(\mathcal{G}, \boxslash) + p_{\text{ind}}(\mathcal{G}, \boxtimes)$$

and the second (6.7) as

$$p(\mathcal{G}, \square)_{\text{ind}} = 3p(\mathcal{G}, \square) - 6p(\mathcal{G}, \boxslash) + 3p(\mathcal{G}, \boxtimes),$$

where $\mathcal{G}$ is a sequence of graphs of increasing size.

Note that in these examples we assume P_{edge} takes an unordered pair as argument. If one instead sees $P_{\text{edge}}(i, j)$ and $P_{\text{edge}}(j, i)$ as two different values of relations, one still obtains the same results after *eliminating incomplete flags*, as explained in the next subsection.

Problems when Working with Non-Induced Densities. Some flag algebra's axioms cannot be easily formulated in the non-induced setting. Let us consider, for example, the theory of linear (or total) orderings. Here we have one predicate "<", taking two arguments. The axioms here are simple:

- Transitive: $(x < y) \wedge (y < z) \Rightarrow (x < z)$,
- Linear: $(x < y) \vee (y < x)$,
- Irreflexive: $\neg(x < x)$,
- Antisymmetric: $\neg\big((x < y) \wedge (y < x)\big)$,

for all distinct vertices x, y, z.

But in the non-induced setting, a *false* relation in a flag means that it *may or may not be true*, as it corresponds to the density of sub-flags, which have *at least the same true relations*. For example, the flag F given by relations $\{x < y, y < z\}$ is not transitive, or even antisymmetric, as it leaves $x < z$ or $z < x$ unknown, and is thus *incomplete*. And yet, by definition, it describes the density of sub-flags

on three vertices $\{x_1, x_2, x_3\}$ of a sequence of linear orderings, such that at least $x_1 < x_2$ and $x_2 < x_3$ are true. By transitivity of each element of the sequence of the linear orderings, this tells us that the density of the flag F is exactly the same as the density of the transitive, and hence, *valid*, flag given by $\{x < y, y < z, x < z\}$. The incomplete flag $\{x < y, y < x\}$ on the other hand is *invalid*, as it cannot be completed to a valid flag. Invalid flags have thus always density zero in a sequence of valid models of increasing size.

In contrast, in the Razborov setting, incomplete and invalid flags are the same. Incomplete induced flags, i.e., flags not fulfilling all of their axioms, have always density zero, and can hence, be eliminated in the Zeta transformation (6.5):

$$M_{\text{non-ind}} = \sum_{\substack{\tilde{M}\,:\,V(M)=V(\tilde{M}),\\ E(M)\subseteq E(\tilde{M}),\\ \textbf{M̃ is valid}}} \frac{|\text{Aut}(\tilde{M})|}{|V(\tilde{M})|!}\tilde{M}_{\text{ind}}. \tag{6.8}$$

This tells us that we can always write densities of *incomplete* flags as sum of *valid* flags.

Proposition 6.4. *Let F be an incomplete (non-induced) flag. Then F is equivalent to a linear combination of valid non-induced flags.*

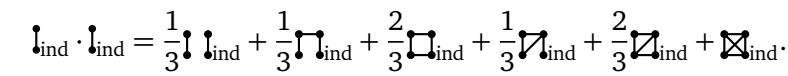
Proof. We obtain the valid quantum flag by calculating Moebius(Zeta(F)), eliminating invalid flags after applying Zeta, and repeating recursively with potentially incomplete flags in the result. If F is incomplete, the induced equivalent to F in the sum is eliminated after applying Zeta. Since all other appearing flags have more true relations, and there is only a finite maximum number of them in total, this has to eventually conclude in a quantum flag only involving valid non-induced flags. □

For example, the incomplete linear ordering $\{a < b, c < d\}$ is the same as the sum of the linear orderings determined by $a < b < c < d$, $a < c < b < d$, $a < c < d < b$, $c < a < b < d$, $c < a < d < b$ and $c < d < a < b$. All six linear orderings are isomorphic (in fact, there is just one linear ordering up to isomorphism for each number of vertices), so the density of $\{a < b, c < d\}$ is six times the density of the linear ordering $a < b < c < d$.

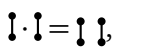
In general, this allows us to always reformulate our problems (and optimization hierarchies) to only involve valid flags, by substituting incomplete flags with valid equivalent quantum flags.

Advantages of Working with Non-Induced Flags. As mentioned before, various factors appearing in Razborov's work [Raz07] simplify to be exactly one in this setting, such as the factors appearing after unlabeling a partially labeled flag. We saw that we can eliminate incomplete flags, reducing the number of considered flags to be identical to the Razborov setting, assuming that we eliminate isolated vertices in both settings and include all flags up to a fixed number of vertices.

Multiplication of non-induced flags is much simpler than multiplying induced flags: In the Razborov setting we need to consider each possible combination of missing edges between unlabeled vertices coming from different flags. For example, in the Razborov setting, the squared edge density is a quantum graph involving six different graphs:



This results in an exponential increase in coefficients in the final SDP in the Razborov setting. Meanwhile, in the non-induced setting, we simply have



corresponding to just one coefficient in the final optimization problem.

This allows us to work with larger, but sparser flags. If our problem is formulated with big sparse graphs, the Razborov setting quickly reaches its limits. For example, if in our problem the edge density to the power five appears, we may want to include products such as

in our computations. In the Razborov setting this results in coefficients corresponding to most of the 12005168 graphs on ten vertices up to isomorphism (see, for example, sequence A000088 in OEIS [OEI99b]). Meanwhile, when working with non-induced densities, we may restrict ourselves to graphs with *up to five edges* instead, of which there are just 114.

But there are also some advantages to working with induced flags, such as easy orthogonal relations between partially labeled flags that differ on the labeled part. We will investigate these further, combining their advantages with those of non-induced flags, by defining *partially induced flags* later in Chapter 9.

6.5 Connection to Polynomial Optimization

We can reformulate problems formulated with flags as sequences of polynomial optimization problems. In the case of (hyper-)graphs, this approach was investigated before in [Ray+18]. We generalize this approach, and take a deeper look at the problem's symmetries in Chapter 8.

Let us first take a look at the setting of graphs as well: We introduce binary variables x_{ij} for every (unordered) pair ij, which correspond to edges. The problems we consider are of the form

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. }\ & g_i(x) \geq 0 \quad \text{for } i = 1, \dots, m, \\ & x = (x_{ij})_{1 \leq i < j \leq n} \in \{0,1\}^{\binom{[n]}{2}}, \end{aligned}$$

where all polynomials f, g_i are fully *symmetric* with respect to the action of the symmetric group S_n given by

$$\sigma(x_{ij}) = x_{\sigma(i)\sigma(j)}$$

for σ in S_n. Note that we need to sort the indices here (or we fix $x_{ij} = x_{ji}$). Throughout this and the following chapters it will always be implied that we are working in the quotient algebra $\mathbb{R}[x]/\langle x_{ij} - x_{ij}^2 : ij \rangle$, even when we abuse the notation and use $\mathbb{R}[x]$ instead.

We write $\mathbb{R}[x]^{S_n}$ for the set of symmetric polynomials (in binary variables), i.e., polynomials p with $\sigma(p) = p$ for all $\sigma \in S_n$. In Chapter 7 we will show how to exploit the symmetries of a polynomial optimization problem with symmetries.

Throughout this chapter we will keep coming back to the basic example of maximizing the edge density in a graph while avoiding triangles. We first use the notation $\mathrm{ex}_n(G, H)$ here for the maximum density of subgraphs G in H-free graphs with n vertices.

Example 6.5. For all $n > 0$ we define

$$\begin{aligned} \mathrm{ex}_n(\graph{1}, \graph{3}) := \max\ & \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} x_{ij} \\ \text{s.t. }\ & \frac{1}{\binom{n}{3}} \sum_{1 \leq i < j < k \leq n} x_{ij} x_{ik} x_{jk} = 0, \\ & x_{ij} \in \{0,1\} \quad \text{for all } i < j. \end{aligned}$$

This problem is clearly fully symmetric: The objective is the (scaled) sum of all variables, which here model edges in a graph, and can be interpreted as the *density* of edges in the graph. The constraint is exactly the sum of monomials lying in the *orbit* of monomials representing triangles in the graph. Since the variables are nonnegative, this implies that the graph is triangle free.

If we take the limit as n goes towards infinity, this results exactly in the definition we discussed earlier for flag algebras:

$$\text{ex}(\edge, \triangle) = \sup\left\{\edge : \triangle = 0\right\} = \limsup_{n\to\infty} \text{ex}_n(\edge, \triangle).$$

These problems may look slightly different to the attentive reader. Indeed, the left side says that the corresponding sequence of graphs $\mathcal{G} = (G_n)_{n=1}^{\infty}$ has a triangle density that *converges to zero*, while the right side implies that *each graph G_n is triangle free*. But it can be shown ([Lov12]) that the problems have the same optimal value.

It was shown in [Ray+18] that, under the right scaling, after a partial symmetry reduction, the problems ex_n indeed converge to the well known *flag SOS* hierarchies introduced by Razborov [Raz07], up to the coefficients in the data matrices, if we restrict the maximum number of vertices appearing in the flags. The reason for deeper look at flag algebras from the polynomial optimization perspective is that the authors there chose to do a partial symmetry reduction, which implies that there is a stronger reduction, and thus a more efficient way to compute flag sums-of-squares. We will investigate the stronger, full symmetry reduction in Chapter 8.

The Lasserre Hierarchy. Before we can exploit the symmetries of the problem, we have to relax and reformulate the problem to make it *convex*. The usual way to make polynomial optimization problems tractable is to restrict solutions to *sums-of-squares* (*SOS*). To do this, we first rewrite the problem as an equivalent problem over nonnegative polynomials. Indeed, the infimum of f over the semi-algebraic set

$$\{x \in \{0,1\}^{\binom{[n]}{2}} : g_i(x) \geq 0 \ \ \forall i = 1, \ldots, m\},$$

where the g_i are polynomials modelling the constraints of our problems, is the same as maximizing a scalar λ such that $f - \lambda$ is nonnegative on the same semi-algebraic set.

As introduced in Section 1.2.2 we can replace the constraint $f - \lambda \geq 0$ with a stronger condition, which is (appropriately truncated) an SDP, by requiring that $f - \lambda \geq 0$ lies in the quadratic module defined by the g_i:

$$f - \lambda = s_0 + \sum_{i=1}^{m} g_i s_i \geq 0 \tag{6.9}$$

where

$$s_j \in \Sigma[x] := \left\{ h : h(x) = \sum_{i=1}^{\ell} p_i(x)^2 \text{ for some } \ell \geq 0, p_i \in \mathbb{R}[x] \right\}.$$

Note that we are still working with binary variables, i.e., we can assume that all appearing polynomials are square-free, by substituting $x^2 = x$. The equality (6.9) thus denotes the equivalence of $f - \lambda$ and $s_0 + \sum_{i=1}^{m} g_i s_i$ modulo the ideal $\langle x_{ij} - x_{ij}^2 : ij \rangle$.

The set $\Sigma[x]$ is a convex cone, and can be modelled using semidefinite programming, if we restrict the degree of the appearing polynomials. For this, let $[x]_d$ denote the vector of monomials up to degree d, and $\mathbb{R}[x]_{\leq d}$ the set of polynomials up to degree d. Then we can easily see by the spectral decomposition of positive semidefinite matrices, that

$$\begin{aligned} \Sigma[x]_{\leq d} :=& \{ s : s(x) = \sum_{i=1}^{\ell} p_i(x)^2 \text{ for some } \ell \geq 0, p_i \in \mathbb{R}[x]_{\leq d} \} \\ =& \{ s : s(x) = [x]_d^T M [x]_d \text{ for some } M \succcurlyeq 0 \}. \end{aligned} \tag{6.10}$$

The bound we obtain from (6.9) by restricting all appearing monomials to at most degree $2d$ we call the *dth level of the Lasserre hierarchy* of the problem, denoted by Las_{2d}, as we defined in Section 6.9. We do this by restricting $s_0 \in \Sigma[x]_{\leq d}$ and $s_i \in \Sigma[x]_{\leq \lfloor \frac{2d - \deg(g_i)}{2} \rfloor}$, making sure that $\deg(s_0) \leq 2d$ and $\deg(g_i s_i) \leq 2d$.

General Flags-Algebras as Limits of Polynomials. In general, we may have more complex predicates in flag-models than the edge predicate for graphs, but they can always be summarized as *the set of all valid relations* in the model. In the case of graphs, we have one predicate determining which edges appear in the graph. In the graph setting we introduced one variable for every possible edge, and we do the same thing in the general case: We introduce one binary variable

for every possible relation in the flag-model, potentially up to given equivalence classes such as $x_{ij} = x_{ji}$.

Thus, we have for every predicate P_j in the model, taking m_j arguments for $j = 1, \dots, k$, a set of binary variables of the form

$$x^{P_j}_{i_1 \dots i_{m_j}} = x^{P_j}_{\bar{i}} \in \{0, 1\},$$

where $\bar{i} \in [n]^{m_j}$. Here we may in general allow duplicate indices, and the order of indices may differentiate variables as well.

This results in a natural action on $\mathbb{R}[x] = \mathbb{R}[x^{P_j}_{\bar{i}} : \bar{i} \in [n]^{m_j} \text{ for } j = 1, \dots, k]$, where $\{P_1, \dots, P_k\}$ is the set of predicates of the current theory, given by

$$\sigma(x^{P_j}_{\bar{i}}) = \sigma(x^{P_j}_{i_1 \dots i_{m_j}}) = x^{P_j}_{\sigma(i_1) \dots \sigma(i_{m_j})} = x^{P_j}_{\sigma(\bar{i})} \quad \text{for } \bar{i} = (i_1, \dots, i_{m_j}) \in [n]^{m_j},$$

for a given $\sigma \in S_n$. This action is then extended to polynomials in the obvious way. We define $\mathbb{R}[x]^{S_n}$ as the ring of polynomials invariant under this action as in the case of graphs.

We can now easily relate non-induced and induced flags to (limits of) polynomials, as we did in the example above for triangle free graphs.

Lemma 6.6. *Let F be a given flag on m vertices, and let $\mathcal{R}_j(F) \subseteq [m]^{m_j}$ be the sets of valid relations for predicate P_j in F. Then*

$$F(x) = \lim_{n \mapsto \infty} \prod_{j=1}^{k} \frac{1}{n!} \sum_{\sigma \in S_n} \prod_{\bar{i} \in \mathcal{R}_j(F)} x^{P_j}_{\sigma(\bar{i})}$$

and

$$F_{\text{ind}}(x) = \frac{|V(F)|!}{|\text{Aut}(F)|} \lim_{n \mapsto \infty} \prod_{j=1}^{k} \frac{1}{n!} \sum_{\sigma \in S_n} \prod_{\bar{i} \in \mathcal{R}_j(F)} x^{P_j}_{\sigma(\bar{i})} \prod_{\bar{i} \in [m]^{m_j} \setminus \mathcal{R}_j(F)} (1 - x^{P_j}_{\sigma(\bar{i})}),$$

Here x is the vector of variables corresponding to the relations in M_n for each n, for the sequence of convergent models $\mathcal{M}$.

Proof. Straightforward translations of their definitions. □

Example 6.7. Let us take a look at a simple example: The density of a path with two edges $P_2 = (\{1, 2, 3\}, \{12, 23\})$. There is only one predicate P_{edge} for graphs

indicating edges in a graph, and the path P_2 has two valid relations $\mathcal{R}_{\text{edge}}(P_2) = \{12, 23\}$. Thus, the (limit of the) non-induced density of the path is exactly

$$\wedge_{\text{non-ind}} = \lim_{n\to\infty} \frac{1}{3}\frac{1}{\binom{n}{3}} \sum_{1\le i<j<k\le n} (x_{ij}x_{ik} + x_{ij}x_{jk} + x_{ik}x_{jk})$$

evaluated for each n at the binary x corresponding to the edges of the nth graph of the sequence $\mathcal{G}$. In contrast, the induced density of the same graph is

$$\wedge_{\text{ind}} = \lim_{n\to\infty} \frac{1}{\binom{n}{3}} \sum_{1\le i<j<k\le n} (x_{ij}x_{ik}(1-x_{jk}) + x_{ij}(1-x_{ik})x_{jk} + (1-x_{ij})x_{ik}x_{jk})$$

evaluated on the same x.

6.6 Harmonic Flags: The Fourier Expansion of Quantum Flags

We show that there is yet another equivalent way to define flags in the limit: harmonic (or zonal spherical) flags. This is, to our knowledge, a novel idea. We will not go into much depth here in this thesis, but this approach opens up possibilities of Fourier analysis in extremal combinatorics in the future. In recent years, Fourier analysis has been a crucial component in the development of major results in coding theory, energy minimization, and packing problems. Bochner's theorem [Boc32] is both the basis of Cohn and Kumar's proofs of universal optimality of points on spheres and in Euclidian space [CK06], as well as the Cohn-Elkies sphere packing bound [CE03]. This in turn lead to Viazovska solving the sphere packing problem in dimension 8 [Via17], and was soon followed up, together with Cohn, Kumar, Miller and Radchenko, by the solution of the problem in dimension 24 [Coh+17]. We generalize Bochner's theorem for flags in terms of harmonic flags. Later on, in Chapter 8 we will investigate *profiles* of pairs of harmonic flags, and see that optimization over this basis has computational advantages.

For the setting of optimization over the boolean hypercube, we refer the reader to [SL21] for details.

Let $\mathcal{M}_n = \{\text{models on the vertices } [n]\}$. We define an inner product between quantum flags by

$$\langle F, G\rangle := \lim_{n\to\infty} \frac{1}{|\mathcal{M}_n|} \sum_{M\in\mathcal{M}_n} F(M)G(M),$$

where $F(M)$ (resp. $G(M)$) denotes the density of F (resp. G) in M. Note how this is the analogue to the inner product of the uniform probability measure on polynomials over the binary hypercube $\mathbb{B}^n = \{0,1\}^n = (\mathbb{Z}/2\mathbb{Z})^n$, which is given by

$$\langle p, q\rangle = \frac{1}{2^n}\sum_{x\in\mathbb{B}^n} p(x)q(x).$$

To apply techniques from Fourier analysis to (limits of) graphs, we first need to turn the set of graph sequences into an abelian group. Let $\mathcal{G}$, $\mathcal{H}$ be graph sequences, where we assume, without loss of generality, that $|V(G_n)| = |V(H_n)|$ for each n, and the graphs in the sequence are constructed by adding vertices and edges to the preceding graph. We define

$$\mathcal{G} + \mathcal{H} := (G_n + H_n)_{n=1}^{\infty} := ((V(G_n), E(G_n)\Delta E(H_n)))_{n=1}^{\infty},$$

where Δ denotes the symmetric difference of sets.

This way we identify the set of limits of graph sequences with a subset of the (countably) infinite binary hypercube $(\mathbb{Z}/2\mathbb{Z})^{\aleph_0}$, containing the vectors corresponding to convergent graph sequences.

Remark 6.8. If both sequences $\mathcal{G}$ and $\mathcal{H}$ converge (i.e. all subgraph densities converge as n approaches infinity), then $\mathcal{G} + \mathcal{H}$ is not necessarily a convergent sequence. Consider, for example, the graph sequences constructed this way: For every n, we add a single new vertex to the previous graph, and, by uniformly random choice, connect it to either all even or all odd vertices coming before. This sequence clearly converges, and the edge density in this sequence of graphs approaches $\frac{1}{2}$. If we now construct a second graph analogously, but now choosing odd and even corresponding to the random choices in the first sequence, we obtain another randomly constructed convergent graph sequence, since the first sequence was constructed randomly. This way, we can "steer" the edge density in the sum $\mathcal{G} + \mathcal{H}$ to follow a curve moving arbitrarily close to 0 and 1 in endless repetition.

It remains to be seen in further research whether there is a more natural way to turn the set of convergent graph sequences into a group, without making the set bigger.

We obtain an orthonormal basis of *characters* (see, for example, [SL21; Val08]) with respect to this inner product by defining

$$\chi_F(\mathcal{M}) := \lim_{n\to\infty} (-1)^{E(F)\cap E(M_n)}$$

for each fully labeled flag F and increasing sequence of flags $\mathcal{M}$. As we assumed that the flags in the sequence build upon each other, this function is well-defined. The characters are *multiplicative* for the defined group action, since

$$\begin{aligned}\chi_F(\mathcal{M}+\mathcal{N}) &= \lim_{n\to\infty}(-1)^{E(F)\cap(E(M_n)\Delta E(N_n))}\\ &= \lim_{n\to\infty}(-1)^{(E(F)\cap E(M_n))\Delta(E(F)\cap E(N_n))}\\ &= \lim_{n\to\infty}(-1)^{(E(F)\cap E(M_n))}(-1)^{(E(F)\cap E(N_n))}\\ &= \chi_F(\mathcal{M})\chi_F(\mathcal{N}).\end{aligned}$$

The character $\chi_\varnothing$ corresponding to the empty graph $\varnothing$ is the *trivial character* of the group, i.e. $\chi_\varnothing \equiv 0$. For every *non-trivial* character χ_F we have $\langle \chi_F, \chi_\varnothing\rangle = 0$, as there exists an N with $\chi_F(N) \neq 1$ and

$$\begin{aligned}\chi_F(N)\lim_{n\to\infty}\sum_{M\in\mathcal{M}_n}\chi_F(M) &= \lim_{n\to\infty}\sum_{M\in\mathcal{M}_n}\chi_F(N)\chi_F(M)\\ &= \lim_{n\to\infty}\sum_{M\in\mathcal{M}_n}\chi_F(N+M)\\ &= \lim_{n\to\infty}\sum_{M\in\mathcal{M}_n}\chi_F(M).\end{aligned}$$

This implies that the characters are orthonormal, since

$$\langle \chi_F, \chi_F\rangle = \lim_{n\to\infty}\frac{1}{|\mathcal{M}_n|}\sum_{M\in\mathcal{M}_n}\chi_F(M)\chi_F(M) = \lim_{n\to\infty}\frac{1}{|\mathcal{M}_n|}\sum_{M\in\mathcal{M}_n}1 = 1,$$

and, for different F, F',

$$\begin{aligned}\langle \chi_F, \chi'_F\rangle &= \lim_{n\to\infty}\frac{1}{|\mathcal{M}_n|}\sum_{M\in\mathcal{M}_n}\chi_F(M)\chi_{F'}(M)\\ &= \lim_{n\to\infty}\frac{1}{|\mathcal{M}_n|}\sum_{M\in\mathcal{M}_n}\chi_{F+F'}(M)\\ &= 0,\end{aligned}$$

since $F+F' \neq \varnothing$ and thus $\chi_{F+F'}$ is non-trivial. Here the sum of two (fully labeled) flags with potentially different labels results in the graph with vertex set given by the union of the vertices of F and F', and edges given by the symmetric difference of their edge sets.

As with induced and non-induced flags, χ_F corresponds to a (limit of) polynomials in binary variables corresponding to edges in an increasing sequence of models $\mathcal{M}$:

$$\chi_F(x) = \lim_{n\to\infty} \prod_{e\in E(F)} (1-2x_e), \tag{6.11}$$

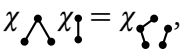
where, for every n, the variable x_e corresponds to the edge e in M_n.

Similarly to Section 6.5, we can symmetrize this basis over the unlabeled vertices to obtain *harmonic (or character) flags*, given by

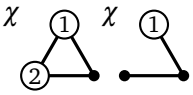
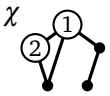
$$\chi_F := \lim_{n\to\infty} \frac{1}{|S_{[n]\setminus T}|} \sum_{\sigma\in S_{[n]\setminus T}} \chi_{\sigma(F)},$$

where T is the set of labeled vertices of F. If the flag F is fully unlabeled, this generalizes the zonal spherical functions (with pole $\varnothing$) of the binary hypercube by splitting them further into the orbits of flags.

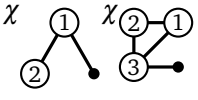
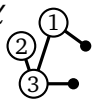
What happens if we multiply two harmonic flags? Since the variables x_e are binary, we know that $(1-2x_e)$ is either 1 or -1, and thus $(1-2x_e)^2 = 1$ for every "edge" e. This results in a multiplication similar to the non-induced case, but now fully labeled edges disappear in the product, if they appear in both factors. This corresponds to the "+" operator defined above, i.e. for fully labeled flags we have $\chi_F\chi_{F'} = \chi_{F+F'}$. For example, we have

χ  $= \chi$,

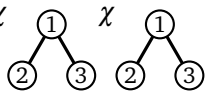
and

χ  $= \chi$ 

as with non-induced flags, but we also have

χ  $= \chi$ 

and

χ  $= \chi_\varnothing = 1.$

Unlabeling a harmonic flag is just as easy as in the non-induced case: Since the normalization was chosen in a way that the constant term in (6.11) is exactly one

in any harmonic flag, unlabeling simply removes the labels here as well. Later on in Chapter 8 we will see that, if one starts with the basis of harmonic flags instead of the monomial basis, we obtain an equivalent hierarchy with computational advantages.

Fourier Decomposition of Flags. As non-induced flags (as well as induced flags) form a basis of the flag algebra, we can rewrite harmonic flags in terms of non-induced flags.

Proposition 6.9. *Let χ_F be an unlabeled harmonic flag. Then χ_F is exactly the quantum flag given by*

$$\chi_F = \sum_{G:E(G)\subseteq E(F)} (-2)^{|E(G)|} G. \tag{6.12}$$

Proof. We expand (6.11) in terms of monomials for F where we label all vertices, and then unlabel both sides of the equation. □

The inverse of this operation, expressing a quantum flag F in the harmonic basis, is the Fourier expansion of F, and if F consists only of unlabeled flags, this results in the Fourier expansion of F in terms of harmonic flags χ_G.

Proposition 6.10. *Let F be an unlabeled non-induced flag. The Fourier expansion of F is*

$$F = \frac{1}{2^{|E(F)|}} \sum_{G:E(G)\subseteq E(F)} (-1)^{|E(G)|} \chi_G. \tag{6.13}$$

Proof. We proof this equation via induction over $|E(F)|$. If $|E(F)| = 0$, then F is the empty flag $\varnothing \equiv 1$, and we have

$$\varnothing = \chi_\varnothing$$

by definition.

Next, assume that the proposition holds for all G with $|E(G)| < |E(F)|$. Note that F itself appears in χ_F with coefficient $(-2)^{|E(F)|}$ by Proposition 6.9. Then, by induction and Proposition 6.9 we have

$$\begin{aligned} F &= \frac{1}{(-2)^{|E(F)|}} \chi_F - \left(\frac{1}{(-2)^{|E(F)|}} \chi_F - F \right) \\ &= \frac{1}{(-2)^{|E(F)|}} \left(\chi_F - \sum_{G:E(G)\subset E(F)} (-2)^{|E(G)|} G \right) \end{aligned}$$

$$= \frac{1}{(-2)^{|E(F)|}} \left(\chi_F - \sum_{G:E(G)\subset E(F)} (-2)^{|E(G)|} \frac{1}{2^{|E(G)|}} \sum_{H:E(H)\subseteq E(G)} (-1)^{|E(H)|} \chi_H \right)$$

$$= \frac{1}{(-2)^{|E(F)|}} \left(\chi_F - \sum_{G,H:E(H)\subseteq E(G)\subset E(F)} (-1)^{|E(G)|+|E(H)|} \chi_H \right)$$

$$= \frac{1}{(-2)^{|E(F)|}} \left(\chi_F - \sum_{H:E(H)\subset E(F)} (-1)^{|E(H)|} \chi_H \sum_{G:E(H)\subseteq E(G)\subset E(F)} (-1)^{|E(G)|} \right) \quad (6.14)$$

We can simplify the final sum to

$$\begin{aligned} \sum_{G:E(H)\subseteq E(G)\subset E(F)} (-1)^{|E(G)|} &= -(-1)^{|E(F)|} + \sum_{G:E(H)\subseteq E(G)\subseteq E(F)} (-1)^{|E(G)|} \\ &= -(-1)^{|E(F)|} + (-1)^{|E(H)|} \sum_{i=0}^{|E(F)|-|E(H)|} \binom{|E(F)|-|E(H)|}{i} (-1)^i \\ &= -(-1)^{|E(F)|}. \end{aligned} \quad (6.15)$$

The last step follows from

$$\sum_{i=0}^{n} \binom{n}{i} (-1)^i = \langle \chi_0, \chi_1 \rangle = 0,$$

where χ_0 and χ_1 denote the characters of the boolean hypercube of the all zero and all-one vector respectively.

Finally, combining equations (6.14) and (6.15) proves the proposition. □

Example 6.11. In the case of graphs, we have

$$\chi_{\triangle} = \varnothing - 2 \cdot 3\,\vdots + 2^2 \cdot 3\,\wedge - 2^3\,\triangle$$

and

$$\triangle = \frac{1}{2^3}\left(\varnothing - 3\chi_{\vdots} + 3\chi_{\wedge} - \chi_{\triangle} \right)$$

as examples of (6.11) and (6.12) respectively.

We can now introduce and characterize *positive semidefinite* quantum flags (or quantum flags *of positive type*), using a special case of Bochner's theorem [Boc32].

Theorem 6.12 (Bochner's theorem for flags)**.** *Let F be a quantum flag. We call F* positive semidefinite*, if for every set S of convergent graph sequences, such that $\mathcal{G} + \mathcal{H}$ converges for all $\mathcal{G}, \mathcal{H} \in S$, we have*

$$(F(\mathcal{G} + \mathcal{H}))_{\mathcal{G},\mathcal{H}\in S} \succcurlyeq 0.$$

F is positive semidefinite if and only if all coefficients in its Fourier expansion, as given in Proposition 6.10, are nonnegative.

Proof. The first direction follows from the multiplicativity of characters: characters are positive definite, as for any fully labeled flag F the matrix

$$(\chi_F(\mathcal{G}+\mathcal{H}))_{\mathcal{G},\mathcal{H}\in S} = (\chi_F(\mathcal{G})\chi_F(\mathcal{H}))_{\mathcal{G},\mathcal{H}\in S}$$

is the outer product of the vector $(\chi_F(\mathcal{G}))_{\mathcal{G}\in S}$.

The other direction follows immediately from the orthonormality of the characters. □

A Natural Recursion of Sidorenko's Conjecture. The Fourier expansion allows for a very natural recursive formulation of some open problems, such as Sidorenko's conjecture [Sid93]. The advantage here is that the Fourier expansion results in a linear combination only containing flags with the same set or subset of edges. (Rewriting a problem in induced flags instead does the opposite, including flags with a superset of edges.) Sidorenko's conjecture states that

$$H - \overset{\bullet}{\underset{\bullet}{\square}}^{|E(H)|} \geq 0 \tag{6.16}$$

for any bipartite graph H, if we work in the non-induced setting. As this inequality is clearly sharp for random graphs, where every edge is chosen independently with the same probability, this conjecture states that these random graphs minimize the density of every bipartite graph, as the number of vertices of the random graph approaches infinity. Sidorenko's conjecture is proven for various families of graphs, including some graphs coming from recursive constructions based on trees in [KLL15]. But the general conjecture remains open, despite attempts by various groups of authors (see [CL18] for the most recent progress). Recently, it was proven by the authors of [Ble+20a] that the cases where H are paths with an odd number of edges cannot be proven purely by flag SOS.

Here we provide a novel (nearly) recursive formulation for the general conjecture based on harmonic flags. The recursion's usefulness still has to be proven.

Combining the two Propositions 6.10 and 6.9, we can rewrite a flag F as sum of the harmonic flag χ_F and sub-flags of F.

Proposition 6.13. *For an unlabeled, non-induced flag F we have*

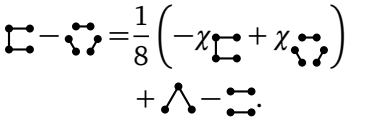
$$F = \frac{1}{(-2)^{|E(F)|}}\chi_F + \sum_{G:E(G)\subset E(F)} 2^{|E(G)|-|E(F)|}(-1)^{|E(G)|+|E(F)|}G$$

Proof. We first apply Proposition 6.10, then Proposition 6.9, and simplify the resulting terms analogously to the proof of Proposition 6.10. □

Since both graphs appearing in Sidorenko's conjecture have the same number of edges, we can pair up graphs in both of their decompositions after applying this transformation, giving us a (nearly) recursive formulation of Sidorenko's conjecture. We can rewrite $H - \mathbb{I}^{|E(H)|}$ as a linear combination of terms of the same form coming from graphs with fewer edges, if we add two harmonic flags to the problem.

$$H - \mathbb{I}^{|E(H)|} = \frac{1}{(-2)^{|E(H)|}} \left(\chi_H - \chi_{\mathbb{I}^{|E(H)|}} \right) + \sum_{G : E(G) \subset E(H)} 2^{|E(G)|-|E(H)|} (-1)^{|E(G)|+|E(H)|} \left(G - \mathbb{I}^{|E(G)|} \right).$$

For example, we have



Note here that all subgraphs G of H are again bipartite, so the appearing terms in the parentheses are indeed again cases of Sidorenko's conjecture, even if some of them appear with a negative sign.

While this recursive formulation of Sidorenko's conjecture does cover all cases, it is not clear at this point whether this will actually help to prove it.

7

Gatermann-Parrilo Reduction for Polynomial Optimization with S_n Symmetry

The basic idea of symmetry reduction is that for a symmetric convex problem, we can *symmetrize* a feasible solution by averaging over the action of the symmetry group to get a *symmetric* feasible solution, which has the same objective value. In this chapter we are first going to give the basics for the case of polynomial optimization as explained in detail in [GP04] for a general group G, and then we will focus on the specific case of $G = S_n$ in Section 7.2. In examples, we will follow the well-understood case of polynomials in variables x_i with a single index, where S_n acts on variables by $\sigma(x_i) = x_{\sigma(i)}$. For this case there exists a great deal of additional theory involving primary and secondary invariants (see, for example, Theorem 6.2 and Section 8.2 in [GP04]). Later on, in Chapter 8, we work with a different, more complicated action of S_n. Applying Theorem 6.2 of [GP04] in this setting seems prohibitively complicated in practice, which is why we restrict ourselves to the more straightforward basic idea of symmetry reduction for polynomials as described by Theorem 4.1 in [GP04].

In this chapter we will follow a simple running example: When is a polynomial

in three binary variables $x_1, x_2, x_3 \in \{0,1\}$ nonnegative, if it is of the form

$$p = a + b(x_1 + x_2 + x_3) + c(x_1x_2 + x_1x_3 + x_2x_3) + dx_1x_2x_3.$$

This polynomial is invariant under the action of S_3 which acts on the indices of the variables.

7.1 Reducing Polynomial Optimization Problems

We consider a polynomial optimization problem

$$\min_x \{f(x) : g_i(x) \geq 0 \quad \text{for } i = 1, \ldots, m\},$$

where f and all g_i are *invariant under the action of a group* G. This group G should act "nicely" on the ring of polynomials $\mathbb{R}[x]$ (formalized in Definition 7.3), and invariant here means that f and the g_i lie in

$$\mathbb{R}[x]^G := \{f \in \mathbb{R}[x] : \sigma(f) = f \text{ for all } \sigma \in G\}.$$

Given such a problem with symmetries, we can show that its semidefinite relaxation (modeling sums-of-squares) has a *symmetric solution*. Sadly, this does not imply that all polynomials p_i appearing in the sums-of-squares decomposition of $f - \lambda$ are symmetric. But what we can say is that the sums-of-squares decomposition itself is symmetric, in the sense that if p_i^2 appears in it, then so does $\sigma(p_i)^2$. More formally, if we have a sum-of-squares decomposition

$$f - \lambda = s_0 + \sum_{i=1}^{m} g_i s_i,$$

then we get a symmetric sum-of-squares decomposition by

$$f - \lambda = \mathcal{R}(f - \lambda) = \mathcal{R}(s_0) + \sum_{i=1}^{m} g_i \mathcal{R}(s_i),$$

since $\mathcal{R}(f) = f$ and $\mathcal{R}(g_i) = g_i$.

Here $\mathcal{R} : \mathbb{R}[x] \mapsto \mathbb{R}[x]^G$ denotes the *Reynolds operator* of the action of G on $\mathbb{R}[x]$, which is the linear operator that symmetrizes polynomials by *averaging* them over the group G by

$$\mathcal{R}(f) := \frac{1}{|G|} \sum_{\sigma \in G} \sigma(f).$$

If we pull this operator further into the sums-of-squares definition, we see that indeed

$$\mathcal{R}(s_i) = \mathcal{R}\left(\sum_{j=1}^{\ell}(p_j(x))^2\right) = \sum_{j=1}^{\ell}\mathcal{R}(p_j(x)^2) = \sum_{j=1}^{\ell}\frac{1}{|G|}\sum_{\sigma\in G}\sigma(p_j(x))^2.$$

Remark 7.1. While we generally do not have that $\mathcal{R}(p_j^2) = \mathcal{R}(p_j)^2$, note that this step implies that the positive semidefinite matrix M in (6.10) for the SOS term s_j is fully symmetric according to the action of G on the monomial vector $[X]$. Indeed, given an SOS certificate $[X]^T M[X]$, where M is positive semidefinite, we can construct an invariant certificate with $\sigma(M) = M$ for each $\sigma \in G$, where the action of G on M is determined by the action of G on $[X]$. One could now work on the SDP side (6.10) by *block-diagonalizing* the algebra of invariant matrices, but in the case of polynomial optimization, while mostly equivalent, there are some additional tools available.

Example 7.2. We can see that *sums-of-squares of symmetric polynomials* are not enough already in the case of

$$p = x_1 + x_2 + x_3 - x_1x_2 - x_1x_3 - x_2x_3 = \frac{3}{2}\mathcal{R}((x_1 - x_2)^2) \geq 0,$$

where $x_i \in \{0, 1\}$, which is an S_3-invariant polynomial of the form

$$a + b(x_1 + x_2 + x_3) + c(x_1x_2 + x_1x_3 + x_2x_3) + dx_1x_2x_3.$$

But p cannot be written as sum of squared symmetric polynomials, as the only way to get a negative coefficient before $x_1x_2 + x_1x_3 + x_2x_3$ would be from a term of the form

$$(a - b(x_1x_2 + x_1x_3 + x_2x_3))^2 = a^2 + (b^2 - 2ab)(x_1x_2 + x_1x_3 + x_2x_3) + 6b^2x_1x_2x_3$$

where $ab < 0$. But this results in a positive constant term a^2 which we cannot eliminate anymore.

7.1.1 Representation Theory

The idea is now to find a *symmetry adapted basis* of $\mathbb{R}[x]$, which is a basis that "behaves well" under the Reynolds operator. To define this formally, we need some basics of representation theory. We refer the interested reader to [Ser77] for additional details.

For everything to work, we need the action of G on $\mathbb{R}[x]$ to turn the ring of polynomials into a G-module.

Definition 7.3. Let $\mathcal{V}$ be a vector space over a ring R, and G a group acting on $\mathcal{V}$ via a group homomorphism $\rho : G \to \mathrm{GL}(\mathcal{V})$, by

$$gv := \rho(g)v,$$

for $g \in G$ and $v \in \mathcal{V}$, where $\mathrm{GL}(\mathcal{V})$ is the *general linear group* of $\mathcal{V}$ consisting of invertible linear functions $\mathcal{V} \to \mathcal{V}$. Then $\mathcal{V}$ is called a *G-module* (over R). (Note that ρ is exactly a linear representation of G.)

In what follows we always assume $\mathcal{V} = \mathbb{R}[x]_{\leq d}$, the space of polynomials up to degree d. We define an inner product on $\mathcal{V}$ by $\langle p, q\rangle := c_p^T c_q$, where c_p and c_q are the coefficient vectors corresponding to p and q in the monomial basis of $\mathbb{R}[x]_{\leq d}$. This way we can further assume that the representation ρ is orthogonal, i.e., $\rho(g)^{-1} = \rho(g)^\star$ for every $g \in G$, where the adjoint $\rho(g)^\star$ is the unique operator in $\mathrm{GL}(\mathcal{V})$ with $\langle \rho(g)p, q\rangle = \langle p, \rho(g)^\star q\rangle$ for all $p, q \in \mathbb{R}[x]_{\leq d}$. This is indeed the case for the family of problems we consider in Chapter 8, as there S_n acts on the variables and monomials by acting on the indices of the variables, which permutes the elements of the monomial basis. For example, in the case of graphs, we are working with variables x_{ij} with two indices (which correspond to edges in a graph), and S_n acts on these by $\sigma(x_{ij}) = x_{\sigma(i)\sigma(j)}$. This action is then extended to monomials and polynomials, sending monomials to monomials, and thus acts via permutations on the monomial basis of $\mathbb{R}[x]_{\leq d}$.

Example 7.4. We again consider polynomials with S_3 symmetry of the form

$$p = a + b(x_1 + x_2 + x_3) + c(x_1x_2 + x_1x_3 + x_2x_3) + dx_1x_2x_3$$

in binary variables. We can turn $\mathbb{R}[x_1, x_2, x_3]/\langle x_i - x_i^2 : i = 1, \ldots, 3\rangle$ into an S_3-module by letting S_3 act on the indices of the variables, and extending the action to polynomials. For example, we have

$$(1\ \ 2)(x_1x_3 + x_2) = x_2x_3 + x_1.$$

The polynomials p are clearly the S_3-invariant polynomials under this action.

We define submodules as subspaces of a module $\mathcal{V}$ that are modules themselves, and call the subspaces $\{0\}$ and $\mathcal{V}$ trivial submodules. Modules that do not have any non-trivial submodules are called *irreducible*.

One of the main tools of algebraic symmetry reduction is Schur's lemma, which we will state here, as we will make use of it explicitly in various places in this chapter and Chapter 8.

Lemma 7.5 (Schur's Lemma)**.** *Let M, N be two irreducible G-modules over a ring R. If M, N are not isomorphic, there are no non-trivial homomorphisms between M and N. If M and N are isomorphic, and R is an algebraically closed field, then any homomorphism between M and N is a multiple of the identity.*

This Lemma allows us to decompose modules into irreducible submodules.

Theorem 7.6 (Maschke's Theorem)**.** *Let G be a finite group, and $\mathcal{V}$ a non-zero G-module. Then $\mathcal{V}$ decomposes into a direct sum of irreducible submodules*

$$V = W_1 \oplus \ldots \oplus W_k.$$

We can sort the submodules by equivalency, where we call two modules *equivalent* if there is a G-homomorphism between them.

$$\begin{aligned}\mathcal{V} &= (V_{1,1} \oplus \cdots \oplus V_{1,m_1}) \oplus \cdots \oplus (V_{h,1} \oplus \cdots \oplus V_{h,m_h}) \\ &= V_1 \oplus \ldots \oplus V_h \\ &\simeq m_1 T_1 \oplus \ldots \oplus m_h T_h,\end{aligned}$$

where $m_i \in \mathbb{N}_0$ and T_i are pairwise inequivalent irreducible G-modules. We call the submodules

$$V_i = V_{i,1} \oplus \cdots \oplus V_{i,m_i} \simeq m_i T_i$$

the *isotypic components* of $\mathcal{V}$. The decomposition into isotypic components is unique, while the decomposition into irreducible submodules is not. One usually sets $V_1 = V^G = \{v \in V : \sigma(v) = v \quad \forall \sigma \in G\}$, the submodule of *invariant* vectors of $\mathcal{V}$, and one calls the elements of the other V_i *semi-invariants*.

We set n_i to be the dimension of T_i, and see

$$\dim \mathcal{V} = \sum_{i=1}^{h} m_i n_i.$$

In the next step we will see that we can choose a special basis of $\mathcal{V}$, which block-diagonalizes the optimization problem to blocks of sizes $m_1, \ldots, m_h$, where we often have

$$m_1 + \cdots + m_h \ll \dim \mathcal{V}.$$

7.1.2 Block-Diagonalization

Let $\varphi_i : G \to O(n_i)$ now be orthogonal representations corresponding to the irreducible modules T_i.

Definition 7.7. Let $\mathbb{B} = \left\{p^i_{j,k} : i \in [h], j \in [n_i], k \in [m_i]\right\}$ be a basis of $\mathcal{V}$, where the elements for each fixed i span the isotypic components

$$V_i = \operatorname{span}\left\{p^i_{j,k} : j \in [n_i], k \in [m_i]\right\},$$

and

$$V_{i,k} = \operatorname{span}\left\{p^i_{j,k} : j \in [n_i]\right\} \simeq T_i$$

are the copies of the irreducible modules T_i. We call $\mathbb{B}$ a *symmetry adapted basis* for the G-module $\mathcal{V}$, if each basis of each copy $V_{i,k}$ of the irreducible module T_i transforms according to the same earlier chosen representation φ_i of T_i. Formally, we want the vector of polynomials $(p^i_{1,k}, \ldots, p^i_{n_i,k})$ to be *ρ-φ_i-equivariant* for all i, k, i.e.

$$\begin{pmatrix} \rho(g)p^i_{1,k} \\ \rho(g)p^i_{2,k} \\ \vdots \\ \rho(g)p^i_{n_i,k} \end{pmatrix} = \varphi_i(g) \begin{pmatrix} p^i_{1,k} \\ p^i_{2,k} \\ \vdots \\ p^i_{n_i,k} \end{pmatrix} \quad \text{for all } g \in G.$$

As the original representation ρ is assumed to be orthogonal, we can always work with an orthonormal symmetry adapted basis (i.e., the coefficient vectors of the polynomials in the basis are pairwise orthogonal and normalized).

Rewriting a symmetric polynomial optimization problem in such a basis *block-diagonalizes* the problem, as described in Theorem 7.8. Since [GP04] does not give a detailed proof, and for completeness, we choose to give our own proof of the theorem, based on the proof for semidefinite programming in [Pol19].

Theorem 7.8 (Gatermann-Parrilo symmetry reduction, [GP04, Section 5])**.** *Let $f \in \mathbb{R}[x]^G$ be a G-invariant polynomial that can be written as a sum-of-squares where each term has at most degree d. If $\{p^i_{j,k} \in \mathbb{R}[x] : i \in [h], j \in [n_i], k \in [m_i]\}$ is an orthonormal symmetry adapted basis of $\mathbb{R}[x]_{\leq d}$ seen as G-module, then f is of the form*

$$f = \sum_{i=1}^{h} \sum_{t=1}^{\ell_i} \mathcal{R}\left(\left(\sum_{k=1}^{m_i} \alpha_{i,t,k} p^i_{1,k}\right)^2\right),$$

for some $\ell \in \mathbb{N}^h$. Equivalently, we can write

$$f = \sum_{i=1}^{h} \langle M_i, P_i \rangle,$$

where $M_i \in S_+^{m_i}$ are positive semidefinite matrices, and

$$P_i = \left(\mathcal{R}(p_{1,k}^i p_{1,l}^i)\right)_{k,l=1,\dots,m_i} \in \mathbb{R}[x]^{m_i \times m_i}. \tag{7.1}$$

None of the (polynomial) entries of the P_i are equal to the zero polynomial.

Proof. This proof is based on the assumption that the element $p_{j,k}^i$ of the symmetry adapted basis lies in the kth copy of the irreducible module T_i, which was given by

$$V_{i,k} = \operatorname{span}\{p_{j,k}^i : j \in [n_i]\} \simeq T_i,$$

and repeatedly applying Schur's Lemma 7.5 to homomorphisms between irreducible modules.

In this proof we interchangeably interpret polynomials either as polynomials or as their coefficient vectors in the standard monomial basis, where the usage should be clear from context.

For each monomial m we define its orbit under G as

$$o = G(m) \coloneqq \{\sigma(m) : \sigma \in G\} \subset \mathbb{R}[x].$$

We can define the indicator matrix A_o of each orbit o, which is indexed by monomials m_1, m_2 of up to degree d, by

$$(A_o)_{m_1,m_2} = \begin{cases} 1 & \text{if } m_1 m_2 \in o, \\ 0 & \text{else.} \end{cases}$$

Then clearly

$$\mathcal{R}(pq) = \sum_{o \text{ orbit}} \left(p^T A_o q \sum_{m \in o} m \right),$$

i.e., $p^T A_o q$ is the coefficient corresponding to the orbit o in the symmetrized product $\mathcal{R}(pq)$ of any two polynomials $p, q \in \mathbb{R}[x]_{\leq d}$. As the symmetry adapted basis is orthonormal (as vectors), we can rewrite

$$(p_{j,k}^i)^T A_o p_{s,t}^r = (p_{j,k}^i)^T \left(A_o p_{s,t}^r\right)\Big|_{V_{i,k}}, \tag{7.2}$$

where we project onto the kth copy $V_{i,k}$ of the ith irreducible submodule. The function

$$V_{r,t} \to V_{i,k} : q \mapsto (A_o q)|_{V_{i,k}}$$

is a homomorphism between submodules of G, as $\mathcal{R}(p\sigma(q)) = \mathcal{R}(\sigma^{-1}(p)q)$ implies $(A_o \sigma(q))|_{V_{i,k}} = \sigma\left((A_o q)|_{V_{i,k}}\right)$ for $\sigma \in G$.

Since $V_{r,t}$ and $V_{i,k}$ are irreducible, Schur's Lemma 7.5 implies that this homomorphism is identical to zero if it is between non-equivalent submodules, so $\mathcal{R}(p^i_{\star,\star}p^j_{\star,\star}) = 0$ for all $i \neq j$ (where the $\star$'s stand for arbitrary indices). Furthermore, if the modules are equivalent (i.e., $i = j$), the homomorphisms in (7.2) have to be multiples of the identity. The definition of symmetry adapted basis fixes this identity to be exactly the function which sends $p^i_{j,k}$ to $p^i_{s,k}$, because the identity is unique between equivalent irreducible modules. So, if $j \neq s \in [n_i]$ we see that

$$(p^i_{j,k})^T \left(A_o p^i_{s,t}\right)\Big|_{V_{i,k}} = \beta (p^i_{j,k})^T p^i_{s,k} = 0 \quad \text{for some } \beta \in \mathbb{R},$$

which is zero since the basis is orthonormal. Thus, $\mathcal{R}(p^i_{j,\star}p^i_{s,\star}) = 0$ if $j \neq s$.

At this point we have shown that the swap of basis block-diagonalizes the algebra into the blocks

$$\left(\mathcal{R}(p^i_{j,k}p^i_{j,l})\right)_{k,l\in[m_i]}.$$

What remains to be seen is that the remaining blocks are exact copies of each other, if they come from different basis elements of the same irreducible submodule, i.e., $\mathcal{R}(p^i_{1,k}p^i_{1,l}) = \mathcal{R}(p^i_{j,k}p^i_{j,l})$ for all j. By definition of symmetry adapted basis, the identity between $V_{i,l}$ and $V_{i,k}$ sends $p^i_{j,l}$ to $p^i_{j,k}$, so there is a constant scalar α_o independent of j such that

$$\left(A_o p^i_{j,l}\right)\Big|_{V_{i,k}} = \alpha_o p^i_{j,k}.$$

Consequently,

$$(p^i_{j,k})^T A_o p^i_{j,l} = \alpha_o (p^i_{j,k})^T p^i_{j,k} = \alpha_o,$$

for each j, as the basis is orthonormal. As this holds for every orbit o, we find that

$$\mathcal{R}(p^i_{j,k}p^i_{j,l}) = \sum_{o \text{ orbit}} \left(\alpha_o \sum_{m\in o} m\right)$$

is independent of the choice of $j \in [n_i]$. Hence, we get multiple copies of the same blocks, and can restrict ourselves the one where $j = 1$.

None of the entries $\mathcal{R}(p^i_{1,k}p^i_{1,l})$ can be zero, since Maschke's theorem 7.6 implies that $\dim(\mathrm{End}_G(\mathbb{R}[x]_{\leq d})) = \sum_{i=1}^h m_i^2$, the total number of coefficients of the P_i. As the block-diagonalization is done via a swap of basis, we cannot reduce the number of nonzero coefficients further. □

This symmetry reduced SDP can be significantly easier to solve, as often $\sum_i m_i \ll \dim(\mathbb{R}[x]_{\leq d})$. Interesting to note is that one does not need to work with

orthogonal representations φ_i, as only the first elements $p^i_{1,k}$ of each irreducible submodule appear in the final decomposition, and we can always extend from those elements to a full orthogonal basis. Normalization just adds a positive factor to each block, not changing the validity of the theorem. In general, a real symmetry adapted basis does not always exist (for some groups we may need polynomials with complex or quaternion coefficients), but for our applications there will always exist a real symmetry adapted basis. For the more complicated cases where there exists no real symmetry adapted basis, we refer the reader to [GP04] [Section 4]. If one instead works with polynomials in complex coefficients and variables, optimizing over *Hermitian sums-of-squares* [DAn02], [Definition IV.5.1], one can always find a complex symmetry adapted basis. For a detailed example of such a block-diagonalization, applied to the problem of packing convex bodies with tetrahedral symmetries, we refer the reader to [Dos+17].

Later on, in Chapter 9, we show that in some cases we can obtain a significantly better symmetry reduction by *breaking Schur's Lemma*, i.e., exploiting the fact that we are optimizing over binary variables to find equivalent irreducible submodules which are orthogonal in the sense that $p^i_{1,k}p^i_{1,l} = \mathcal{R}(p^i_{1,k}p^i_{1,l}) = 0$.

7.2 Representation Theory of the Symmetric Group

The symmetry of flag algebras is given by the (limit of the) symmetric group S_n. Thus, if we want to exploit it, we need to understand the irreducible S_n-modules. These are well studied, and we will give the main details here. We refer to [Sag13] for additional details, and mostly follow the way things are defined there. The notation, especially the one used in Section 7.2.3, is in part new and makes the definition of the homomorphisms ϑ_T easier.

7.2.1 Young Tableaux, Tabloids and Permutation Modules

There are as many irreducible G-modules as there are conjugacy classes of G. For permutations, these are exactly the *cycle types* of the permutations, which correspond to *partitions* of n. We call $\lambda = (\lambda_1, \dots, \lambda_k) \in \mathbb{N}^k$ a *partition* if $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$. We use the notations $|\lambda| := \sum_i \lambda_i$ and write $\lambda \vdash n$ if $|\lambda| = n$.

Each partition defines a *shape*, a set of coordinates, or boxes arranged on a grid

$$\text{shape}(\lambda) := \{(i, j) : i = 1, \dots, k; j = 1, \dots, \lambda_i\}.$$

If we fill these boxes bijectively with the integers 1 trough $|\lambda|$, we call them a *Young-tableau* of shape λ. For example, let $\lambda = (4,2,2)$, then

$$t = \begin{array}{|c|c|c|c|}\hline 2 & 3 & 6 & 5 \\ \hline 1 & 8 \\ \cline{1-2} 4 & 7 \\ \cline{1-2}\end{array}$$

is a Young-tableau of shape λ.

Two tableaux t_1 and t_2 of the same shape are called *row-equivalent*, written $t_1 \sim t_2$, if, for every $i \in [k]$, the set of λ_i elements in the ith row of t_1 contains the same elements as the set of elements in the corresponding ith row of t_2. We call the equivalence classes λ*-tabloids*, and define

$$\{t\} := \{\hat{t} : \hat{t} \sim t\}.$$

To differentiate tableaux from tabloids, we write tabloids using horizontal lines only instead of boxes, e.g., for the tableau above we have

$$\{t\} = \begin{array}{\underline{c c c c}} 2 & 3 & 5 & 6 \\ 1 & 8 \\ \cline{1-2} 4 & 7 \\ \cline{1-2}\end{array}.$$

The symmetric group S_n acts on Young-tableaux by permuting its entries, which extends to an action on tabloids. For each partition λ the vector space spanned by the λ-tabloids forms an S_n-module called *permutation module*,

$$M^\lambda := \mathbb{R}\{\{t_1\}, \ldots, \{t_h\}\}$$

where the $\{t_i\}$ form a complete list of λ-tabloids.

Example 7.9. Consider again the case of polynomials

$$p = a + b(x_1 + x_2 + x_3) + c(x_1x_2 + x_1x_3 + x_2x_3) + dx_1x_2x_3$$

in binary variables. The ring $V = \mathbb{R}[x_1, x_2, x_3]/\langle x_i - x_i^2 : i = 1, \ldots, 3\rangle$, seen as S_3-module, decomposes into a direct sum of 4 permutation modules:

$$V = 2M^{(3)} \oplus 2M^{(2,1)}.$$

Indeed, it is easy to see that the *spans of orbits of monomials* are each isomorphic to a permutation module:

$$\begin{aligned}\operatorname{span}(1) &\simeq M^{(3)},\\ \operatorname{span}(x_1, x_2, x_3) &\simeq M^{(2,1)},\\ \operatorname{span}(x_1x_2, x_1x_3, x_2x_3) &\simeq M^{(2,1)},\\ \operatorname{span}(x_1x_2x_3) &\simeq M^{(3)},\end{aligned}$$

by simply putting all appearing indices in a monomial into one row of a tabloid, and all indices that do not appear in the other. For example, the monomial x_2x_3 corresponds to the tabloid

$$\begin{array}{|c|c|}\hline 2 & 3\\ \hline\end{array}\\ \begin{array}{|c|}\hline 1\\ \hline\end{array} \in M^{(2,1)}.$$

7.2.2 Irreducible Specht Modules

While permutation modules are reducible, there is a natural way to define an irreducible submodule of each M^λ called *Specht module*, which are pairwise inequivalent (and thus are a full set of all irreducible submodules of S_n).

To define these, we need to define the *column stabilizer* C_t of a tableau t as the group

$$C_t := S_{C_1} \times \cdots \times S_{C_\ell},$$

where the C_i are the columns of t.

Definition 7.10. Let λ be a partition of n. The *Specht module* corresponding to λ is

$$S^\lambda := \operatorname{span}\{e_t : t \text{ is a tableau of shape } \lambda\} \subseteq M^\lambda,$$

where e_t denotes the *polytabloid*

$$e_t := \sum_{\sigma \in C_t} \operatorname{sgn}(\sigma)\{\sigma t\}.$$

Example 7.11. Let $\lambda = (4,2)$ and $t = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4\\ \hline 5 & 6\\ \cline{1-2}\end{array}$. Then $C_t = S_{\{1,5\}} \times S_{\{2,6\}} \times S_{\{3\}} \times S_{\{4\}}$ and

$$e_t = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4\\ \hline 5 & 6\\ \cline{1-2}\end{array} - \begin{array}{|c|c|c|c|}\hline 5 & 2 & 3 & 4\\ \hline 1 & 6\\ \cline{1-2}\end{array} - \begin{array}{|c|c|c|c|}\hline 1 & 6 & 3 & 4\\ \hline 5 & 2\\ \cline{1-2}\end{array} + \begin{array}{|c|c|c|c|}\hline 5 & 6 & 3 & 4\\ \hline 1 & 2\\ \cline{1-2}\end{array} \in M^\lambda.$$

7.2.3 A Symmetry Adapted Basis of Permutation Modules

The main symmetry reduction later in this paper will first (try to) decompose the set of polynomials as direct sum of permutation modules, since their full decomposition into Specht modules is well-understood. The decomposition is based on *semistandard generalized Young-tableaux*, which we now define.

We call a tableau t of shape $\mu \vdash n$ where we allow repeated entries *generalized*, and we call such a tableau *semistandard* if its columns are strictly increasing and its rows non-decreasing. We call the vector

$$\lambda = (\text{number of 1s in } t, \text{number of 2s in } t, \dots) \vdash n$$

the *content* of t. For example,

1	1	2	3
3	3		
4			

is a semistandard generalized tableau of shape $\mu = (4,2,1)$ and content $\lambda = (2,1,3,1)$.

If the content μ is a partition itself (i.e., ordered), then it is easy to see that such a tableau can only exist, if μ *dominates* λ, written $\mu \trianglerighteq \lambda$ i.e.

$$\mu_1 + \cdots + \mu_i \geq \lambda_1 + \cdots + \lambda_i \quad \text{for all } i \geq 1,$$

where λ_i, respectively μ_i, are zero if i is outside their index range.

Theorem 7.12. *A permutation module of shape λ decomposes into the (irreducible) Specht modules*

$$M^\lambda \simeq \bigoplus_{\mu \trianglerighteq \lambda} K_{\mu\lambda} S^\mu.$$

Here the multiplicities are given by the Kostka-numbers $K_{\mu\lambda}$*, which are defined as the number of semistandard generalized Young-tableaux of shape μ and content λ.*

Every semistandard generalized Young-tableaux T (of shape μ and content λ) defines an isomorphism between M^μ and a submodule of M^λ (and with that also of $S^\mu \subseteq M^\mu$ to its image). One can define it like this: First, given T and a tableau t of shape μ, we can send it to a tabloid of shape λ by moving entry $t(i,j)$ of t to the $T(i,j)$th row. Then, if we sum these up over the tabloid $\{T\}$, this function is constant over all elements of $\{t\}$ and defines an isomorphism between M^μ and a submodule of M^λ:

$$\vartheta_T : M^\mu \to M^\lambda, \{t\} \mapsto \sum_{T' \in \{T\}} \left\{\text{tableau with entry } t(i,j) \text{ in its } T'(i,j)\text{th row}\right\}.$$

To shorten the notation, we define for a generalized tableau T of shape μ and content λ and a tableau t of shape μ the tabloid

$$t[T] := \{\text{tableau with entry } t(i,j) \text{ in its } T(i,j)\text{th row}\},$$

i.e., we have $\vartheta_T(\{t\}) = \sum_{T' \in \{T\}} t[T']$.

Example 7.13. Let $\mu = (3,2) \trianglerighteq \lambda = (2,2,1)$. We fix a semistandard tableau of shape μ and content λ

$$T = \begin{array}{|c|c|c|}\hline 1 & 1 & 2 \\ \hline 2 & 3 \\ \cline{1-2}\end{array}.$$

Then the row-equivalency class $\{T\}$ contains six generalized tableaux, and

$$\vartheta_T\left(\begin{array}{|c|c|c|}\hline a & b & c \\ \hline d & e \\ \cline{1-2}\end{array}\right) = \begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline e \\ \cline{1-1}\end{array} + \begin{array}{|c|c|}\hline a & c \\ \hline b & d \\ \hline e \\ \cline{1-1}\end{array} + \begin{array}{|c|c|}\hline b & c \\ \hline a & d \\ \hline e \\ \cline{1-1}\end{array} + \begin{array}{|c|c|}\hline a & b \\ \hline c & e \\ \hline d \\ \cline{1-1}\end{array} + \begin{array}{|c|c|}\hline a & c \\ \hline b & e \\ \hline d \\ \cline{1-1}\end{array} + \begin{array}{|c|c|}\hline b & c \\ \hline a & e \\ \hline d \\ \cline{1-1}\end{array}.$$

The images of the Specht module S^μ under these homomorphisms for all semistandard tableaux decompose M^λ fully into irreducible submodules. I.e., the homomorphisms given by semistandard tableaux form a basis of the *multiplicity space* $\mathrm{Hom}(S^\mu, M^\lambda)$.

To now obtain a symmetry adapted basis, one needs to choose the same ordered orthogonal basis for all copies of the same Specht module. As mentioned in Section 7.1, we only need the first basis element of each copy, which means that we can work with the images of one fixed polytabloid for each μ.

Example 7.14. We can now fully exploit the symmetry of polynomials of the form

$$p = a + b(x_1 + x_2 + x_3) + c(x_1x_2 + x_1x_3 + x_2x_3) + dx_1x_2x_3$$

in binary variables. We saw in Example 7.9 that the corresponding polynomial ring decomposes into the 4 permutation modules

$$V = 2M^{(3)} \oplus 2M^{(2,1)}.$$

The two permutation modules can now be decomposed further:

$$M^{(3)} \simeq S^{(3)},$$
$$M^{(2,1)} \simeq S^{(3)} \oplus S^{(2,1)},$$

as the corresponding semistandard tableaux are

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \end{array}, \begin{array}{|c|c|c|}\hline 1 & 1 & 2 \\ \hline \end{array}, \text{and } \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 \\ \cline{1-1} \end{array}.$$

This means that we can block-diagonalize the SOS-SDP into a block of size 4 and a block of size 2, corresponding to $S^{(3)}$ and $S^{(2,1)}$ respectively. The symmetry adapted bases of the two blocks are given by

$$B_1 = \{1, x_1 + x_2 + x_3, x_1x_2 + x_1x_3 + x_2x_3, x_1x_2x_3\},$$
$$B_2 = \{x_2 - x_1, x_1x_3 - x_2x_3\}.$$

As the third level of the Lasserre hierarchy is exact for problems in three binary variables, this then tells us that p is nonnegative if and only if it is of the form

$$p = \langle M_1, P_1 \rangle + \langle M_2, P_2 \rangle$$

for positive semidefinite matrices $M_1 \in S^4_+$ and $M_2 \in S^2_+$, where

$$P_i = \mathcal{R}(B_i B_i^T).$$

7.2.4 Multiplying and Symmetrizing the Elements of the Symmetry Adapted Basis

To obtain the symmetry reduced SDP, we now need to multiply and symmetrize the elements of the symmetry adapted basis. One can do this efficiently using an algorithm by Dion Gijswijt described in [Gij09]. There is another alternative algorithm by Litjens, Polak, and Schrijver, see Appendix 3.4.1 in Sven Polak's thesis [Pol19]. There the algorithm is defined for a different specific application to error correcting codes, which we here generalize.

While we generally do not have a product defined between tabloids, we need to calculate the matrices P_i (7.1), with entries given by $\mathcal{R}(p^i_{j,k} p^i_{j,l})$, to obtain the symmetry reduced SDP. Interestingly, while calculating $p^i_{j,k} p^i_{j,l}$ seems hard (due to exponentially many terms in the polytabloid), there are polynomial time (in n, for fixed maximum number of parts of λ_1 and λ_2) algorithms for calculating $\mathcal{R}(p^i_{j,k} p^i_{j,l})$, if one can apply $\mathcal{R}$ efficiently.

We assume now that we have two isomorphisms ϕ_1 and ϕ_2 from M^{λ^1} and M^{λ^2} respectively to submodules of an S_n-module. We may have $\lambda^1 = \lambda^2$. Let $\mu \trianglerighteq \lambda^1, \lambda^2$, and consider two semistandard generalized tableaux T_1 and T_2 both with content μ and of shape λ^1 and λ^2 respectively. Fix a tableau t of shape μ. Then the coefficients of P_i (7.1) are of the form

$$p = \mathcal{R}(\phi_1(\phi_{T_1}(e_t))\phi_2(\phi_{T_2}(e_t))).$$

By linearity, we can decompose this in the form

$$p = \sum_{\substack{\{t_1\} \text{ tabloid of shape } \lambda^1, \\ \{t_2\} \text{ tabloid of shape } \lambda^2}} \hat{c}_{t_1,t_2} \mathcal{R}(\phi_1(\{t_1\})\phi_2(\{t_2\})),$$

and since $\mathcal{R}$ symmetrizes over all of S_n, we can reduce this to a sum over a fixed tabloid $\{t_1\}$:

$$p = \sum_{\{t_2\}\text{tabloid of shape } \lambda^2} c_{\{t_2\}} \mathcal{R}(\phi_1(\{t_1\})\phi_2(\{t_2\})).$$

With the same argument we can reduce this further to a sum over the equivalence classes of *ordered pairs of tabloids*, where two pairs $(\{t_1\}, \{t_2\})$ and $(\{t_3\}, \{t_4\})$ are equivalent, if there is a $\sigma \in S_n$ such that $(\sigma\{t_1\}, \sigma\{t_2\}) = (\{t_3\}, \{t_4\})$. It is easy to see that these equivalence classes are exactly determined by the way the two tabloids *overlap*, i.e., how many elements of each row of one appear in each row of the other.

To formalize this, the overlap of two tabloids $\{t_1\}$, $\{t_2\}$ is the matrix

$$\text{overlap}(\{t_1\}, \{t_2\}) := (|\text{row}(\{t_1\}, i) \cap \text{row}(\{t_2\}, j)|)_{i,j=1}^{m},$$

where m is the maximum number of parts of λ^1 and λ^2, and $\text{row}(\{t\}, i) \subseteq [n]$ is the set of elements in the ith row of the tabloid $\{t\}$.

We can equivalently describe these overlaps using monomials by

$$\text{overlap}(\{t_1\}, \{t_2\}) \leftrightarrow \prod_{i,j=1}^{m} x_{ij}^{\text{overlap}(\{t_1\},\{t_2\})_{ij}},$$

we can apply one of the two algorithms described in [Gij09; Pol19] to calculate a polynomial that describes the overlaps and multiplicities needed to calculate p.

These algorithms are based on polynomial addition, multiplication and differentiation, and can be done in polynomial time (for fixed m). Indeed, the sum defining p over pairs of tabloids up to S_n is completely described by the polynomial

$$\left(\prod_{j=1}^{m-1}\prod_{s=j+1}^{m}\frac{(d_{s\to j})^{r(s,j)}(d^{\star}_{j\to s})^{u(s,j)}}{r(s,j)!u(s,j)!}\right)\cdot P_{\mu}(X),$$

where

$$\begin{aligned}
P_{\mu}(X) &:= \prod_{k=1}^{m}(k!\det((x_{ij})_{i,j=1}^{k}))^{\mu_k-\mu_{k+1}},\\
d_{s\to j} &:= \sum_{i=1}^{k}x_{si}\frac{\partial}{\partial x_{ji}},\\
d^{\star}_{j\to s} &:= \sum_{i=1}^{k}x_{is}\frac{\partial}{\partial x_{ij}},\\
r(s,j) &:= \text{number of } s\text{'s in row } j \text{ of } T_1,\\
u(s,j) &:= \text{number of } s\text{'s in row } j \text{ of } T_2.
\end{aligned}$$

What remains then is to calculate $\mathcal{R}(\phi_1(\{t_1\})\phi_2(\{t_2\}))$ for all appearing overlaps between $\{t_1\}$ and $\{t_2\}$ (to rewrite the problem in terms of coefficients in a basis of the symmetric polynomials $\mathbb{R}[x]^G$). This depends fully on the application, i.e., how S_n acts on the ring one works with. For example, as we will later see in Section 8.3.3, symmetrization for flag algebras will be equivalent to canonically labeling a flag.

8

Symmetry Reduced Flag SOS Hierarchies

In this chapter we fully exploit the symmetries of the Lasserre hierarchy for flag sums-of-squares, as introduced in Chapter 6. These are hierarchies limiting the number of true relations (e.g., edges), and thus the degree of the corresponding monomials, in the products. The next chapter, Chapter 9, instead considers the symmetry of the hierarchies obtained by limiting the number of vertices of the flags appearing in the hierarchies.

As seen in Section 6.5, we can see flags as limits of polynomials, where a fully labeled flag M corresponds to the monomial consisting of binary variables corresponding to $E(M)$. As a reminder, these variables correspond to predicate relations in flags, indexed by (potentially ordered/unordered/distinct) tuples in $[n]^{m_i}$, for each m_i denoting the number of arguments of a predicate P_i. We let S_n act on the variables by acting on the indices (i.e., vertices of the flags) simultaneously. Recently, it was observed in [Ray+18] that one can recover flag sums-of-squares by partially exploiting the symmetries of these polynomial hierarchies, and taking the limit.

We first make this step "backwards" to polynomials as well, and then fully exploit the symmetries of the polynomial hierarchies for each n. We will see here, too, that the sizes of the hierarchies become independent of n, if n is chosen big enough, which will allow us to take the limit of the hierarchy. Going backwards

to polynomials, while first increasing the size of the hierarchies, also increases the symmetries of the hierarchy and removes redundancies contained in the definition of flag algebras. This way we obtain a better reduction than we would have when starting out with flag algebras, and then reducing them further.

Outline and Contributions of this Chapter. We start by considering an easier case in Section 8.1, where we optimize over binary variables with a single index. Determining the decomposition into irreducible submodules is straightforward in this case, as we can rely fully on well known facts about the representation theory of S_n, as explained earlier in Section 7.2. In Section 8.2 we attempt to generalize the approach, but quickly conclude that permutation modules are not enough to determine the full decomposition of the modules; we will need to generalize them. We determine the reduction for the general case in Section 8.3. There we first show that the decomposition *stabilizes* when n reaches a certain threshold, in similar vein as results from [Ray+18], [Rie+13] and [DR20]. We then propose a novel algorithm to decompose generalized permutation modules into irreducible submodules, which computes the decomposition independently of n (and thus also independently of the dimension of the original representation). The next Section 8.4 focuses on the case of graphs, where we determine the full decomposition of the Lasserre hierarchy of graphs with up to two edges. We give an interpretation of the reduced limit-hierarchy in terms of flags with vertex groups instead of vertex labels, which we call *Specht flags*. We apply the reduced hierarchies to compute outer approximations of graph-profiles of pairs of small (harmonic) graphs in Section 8.5, visualizing the rich structures of densities of small subgraphs in limits of graphs. In Section 8.6 we provide a proof-of-concept example for an extension of flag algebras to the much less well-understood setting of degenerate extremal combinatorics, where the edge density approaches zero as the number of vertices grows. Finally, we give a short description of the software package developed for flag sums-of-squares in Section 8.7.

Main Symmetry Reduction Idea. To exploit the symmetry of the hierarchies, we now want to decompose $\mathbb{R}[x]$ into irreducible S_n-submodules, as that gives us a block-diagonalization of the optimization hierarchy, as explained in Chapter 7. To do this, we try to decompose $\mathbb{R}[x]$ into permutation modules first, because we know how to decompose them further into irreducible submodules.

8.1 Warm Up: One Predicate with One Argument.

In this section we consider binary variables x_i with just one index in $i \in [n]$, where S_n acts by permuting variables directly. Here $\mathbb{R}[x]^{S_n}$ consists of the fully symmetric polynomials in binary variables. The corresponding hierarchy was fully reduced before in [Rie+13] for non-binary variables. The first easy step to find the decomposition into irreducible submodules is to determine the orbits of the monomial basis of $\mathbb{R}[x]$. In this case we get one orbit for each degree:

$$\begin{aligned} S_n(1) &= \{1\}, \\ S_n(x_1) &= \{x_1, \dots, x_n\}, \\ S_n(x_1x_2) &= \{x_ix_j : 1 \le i < j \le n\}, \\ &\vdots \end{aligned}$$

which correspond directly to permutation modules:

$$\begin{aligned} \operatorname{span}(S_n(1)) &\simeq M^{(n)}, \\ \operatorname{span}(S_n(x_1)) &\simeq M^{(n-1,1)}, \\ \operatorname{span}(S_n(x_1x_2)) &\simeq M^{(n-2,2)}, \\ &\vdots \end{aligned}$$

by sending the monomial $x_{i_1} \cdots x_{i_d}$, $i_1 < \cdots < i_d$ to the tabloid with the i_j in one row, and all indices that do not appear in the other. For example for the degree 3 orbit we have

$$x_2x_4x_5 \longleftrightarrow \begin{array}{|c|c|c|c|c|} \hline 1 & 3 & 6 & 7 \\ \hline 2 & 4 & 5 \\ \cline{1-3} \end{array} \in M^{(n-3,3)}$$

in the case $n = 7$.

Thus, the ring of polynomials (seen as S_n-module) decomposes as

$$\mathbb{R}[x] \simeq \bigoplus_{d=0}^{n} M^{(n-d,d)}.$$

We described the decomposition of permutation modules M^λ into Specht modules in Theorem 7.12, i.e., it is straightforward to get the full decomposition from here.

In this case calculating $\mathcal{R}(\phi_1(\{t_1\})\phi_2(\{t_2\}))$ is easy as well. Let us assume that ϕ_1 corresponds to the orbit of degree 2, and ϕ_2 to the orbit of degree 3. Then, given tabloids $\{t_1\}$ of shape $(n-2,2)$ and $\{t_2\}$ of shape $(n-3,3)$, this corresponds

to multiplying two monomials of degree 2 and 3, cancelling potential squares, and then determining its orbit (which is uniquely given by the degree). E.g.

$$\begin{aligned}
&\mathcal{R}\left(\phi_1\left(\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 5 & 6 & 7 \\ \hline 2 & 3 \\ \cline{1-2}\end{array}\right)\phi_2\left(\begin{array}{|c|c|c|c|}\hline 1 & 2 & 4 & 7 \\ \hline 3 & 5 & 6 \\ \cline{1-3}\end{array}\right)\right) \\
&= \mathcal{R}(x_2x_3 \cdot x_3x_5x_6) \\
&= \mathcal{R}(x_2x_3x_5x_6) \\
&= \mathcal{R}(x_1x_2x_3x_4).
\end{aligned}$$

Here $\mathcal{R}(x_1x_2x_3x_4)$ is exactly the polynomial known as the (normalized) elementary symmetric polynomial $e_4 := \sum_{1\le i<j<k<l\le n} x_ix_jx_kx_l$ (see, for example, [Mac95]).

8.2 The (Hyper-) Graph Case

The case of (hyper)-graphs was only partially reduced before in [Ray+18]. While they did find a reduction of which the size is independent of n for n big enough, the reduction they found is generally far from optimal.

Here we consider binary variables x_{ij} which are indexed by unordered pairs i, j (resp. triples, quadruples, ...), which correspond to (hyper-)edges. We follow the same idea in the general case: First, we split all of $\mathbb{R}[x]$ into parts resembling permutation modules, then we decompose these further. While this was straightforward in Section 8.1, in the general case working with just permutation modules is not enough anymore.

Again, we first want to split $\mathbb{R}[x]$ into the spans of orbits of monomials. As we are working with binary variables, and the order of the indices does not matter, monomials correspond exactly to *labeled (hyper-)graphs*. For example, in the case of graphs, the monomial $x_{12}x_{13}x_{14}x_{34}$ corresponds to the graph with edges $(1,2),(1,3),(1,4)$ and $(3,4)$. Now the action of S_n on the monomial is the same as S_n acting on the vertices of the graph. Orbits thus correspond to (hyper-)graphs up to isomorphism (without isolated nodes).

Given a (hyper-)graph $G = (V, E)$ we introduce the notation

$$M^{G,n} := \operatorname{span}\left(S_n\left(\prod_{(i_1,\dots,i_k)\in E} x_{i_1\dots i_k}\right)\right)$$

for the S_n-module spanned by the orbit of monomials corresponding to G. We call this module the *graph module of* G (and more generally, the *flag module of a model* M). This gives us the first decomposition of the full module of polynomials. For example, we have

$$\mathbb{R}[x] = \bigoplus_{\text{graphs } G \text{ up to isomorphism without isolated vertices}} M^{G,n}$$
$$= M^{\emptyset,n} \oplus M^{\,\cdot\,\cdot,n} \oplus M^{\wedge,n} \oplus M^{\,\cdot\!-\!\cdot\!-\!\cdot,n} \oplus M^{\triangle,n} \oplus \dots$$

in the case of graphs. We do not need to consider modules corresponding to graphs with isolated vertices, as variables correspond to edges, and thus monomials correspond to graphs without isolated vertices.

Note that we could generalize this for non-binary variables by allowing graphs with multi-edges. This generalizes in the obvious way to arbitrary flags, if we take the sum over all flags up to isomorphism.

First, let us take a look at two simple cases of these modules.

The Graph Module $M^{\wedge,n}$

Decomposing this module is straightforward: We can find a bijection between the basis and tabloids of shape $(n-3,2,1)$ by

$$\begin{array}{ccc} & i & \\ j & & k \end{array} \longleftrightarrow x_{ij}x_{ik} \longleftrightarrow \begin{array}{|c|c|c|} \hline [n]\setminus\{i,j,k\} \\ \hline j & k \\ \cline{1-2} i \\ \cline{1-1} \end{array},$$

which immediately tells us that $M^{\wedge,n} \simeq M^{(n-3,2,1)}$. This works here since the row-stabilizer of the tabloid is exactly the automorphism group of the graph. We can then further decompose it using the known decomposition into Specht modules.

The Graph Module $M^{\cdot\!-\!\cdot\!-\!\cdot,n}$

Decomposing this module is more tricky. Here we can find a description of monomials using tabloids, too, but it is not bijective:

$$\begin{array}{ccc} i & - & j \\ k & - & l \end{array} \longleftrightarrow x_{ij}x_{kl} \longleftrightarrow \left\{ \begin{array}{|c|c|c|} \hline [n]\setminus\{i,j,k,l\} \\ \hline i & j \\ \cline{1-2} k & l \\ \cline{1-2} \end{array}, \begin{array}{|c|c|c|} \hline [n]\setminus\{i,j,k,l\} \\ \hline k & l \\ \cline{1-2} i & j \\ \cline{1-2} \end{array} \right\}.$$

I.e., switching the second and third row of a tabloid results in the same monomial. Thus

$$M^{\begin{smallmatrix}\bullet\\\bullet\\\bullet\end{smallmatrix},n} \simeq M^{(n-4,2,2)}/_{S_2} \subseteq M^{(n-4,2,2)}, \tag{8.1}$$

where S_2 is the group permuting the last two rows of $M^{(n-4,2,2)}$.

Schur's lemma tells us that the function from $M^{(n-4,2,2)}$ to $M^{(n-4,2,2)}/_{S_2}$ which symmetrizes elements over the row permutation group S_2, *respects* the decomposition of the permutation module, in the sense that it sends Specht modules in $M^{(n-4,2,2)}$ to equivalent Specht modules in $M^{(n-4,2,2)}/_{S_2}$, or to $\{0\}$. I.e., the decomposition of $M^{(n-4,2,2)}/_{S_2}$ is included in the one of $M^{(n-4,2,2)}$, but certain Specht modules may fall together, or may be cancelled out.

In the case $n = 8$ we can calculate the multiplicities of the Specht modules in both from their characters:

$$\begin{aligned} M^{(4,2,2)} \simeq & S^{(4,2,2)} \oplus S^{(4,3,1)} \oplus S^{(4,4)} \oplus 2S^{(5,2,1)} \oplus 2S^{(5,3)} \\ & \oplus S^{(6,1,1)} \oplus 3S^{(6,2)} \oplus 2S^{(7,1)} \oplus S^{(8)}, \end{aligned}$$

but

$$\begin{aligned} M^{\begin{smallmatrix}\bullet\\\bullet\\\bullet\end{smallmatrix},n} \simeq & S^{(4,2,2)} \oplus (\mathbf{1}-\mathbf{1})S^{(4,3,1)} \oplus S^{(4,4)} \oplus (\mathbf{2}-\mathbf{1})S^{(5,2,1)} \oplus (\mathbf{2}-\mathbf{1})S^{(5,3)} \\ & \oplus (\mathbf{1}-\mathbf{1})S^{(6,1,1)} \oplus (\mathbf{3}-\mathbf{1})S^{(6,2)} \oplus (\mathbf{2}-\mathbf{1})S^{(7,1)} \oplus S^{(8)} \\ = & S^{(4,2,2)} \oplus S^{(4,4)} \oplus S^{(5,2,1)} \oplus S^{(5,3)} \\ & \oplus 2S^{(6,2)} \oplus S^{(7,1)} \oplus S^{(8)}. \end{aligned}$$

While the full permutation module decomposes into 14 Specht modules, the quotient decomposes into 8. We now have to figure out how to determine the way the Specht modules are contained in $M^{\begin{smallmatrix}\bullet\\\bullet\\\bullet\end{smallmatrix},n}$, i.e. we need to determine an explicit basis of the S_n-homomorphisms in $\mathrm{Hom}\left(S^\lambda, M^{\begin{smallmatrix}\bullet\\\bullet\\\bullet\end{smallmatrix},n}\right)$ for each S^λ appearing in the decomposition.

8.3 The General Case: Decomposing the Module $M^{F,n}$

Let F be a flag of a given theory on m vertices. If we can determine the Reynolds operator of the automorphism group $\mathrm{Aut}(F)$ acting on $\mathrm{Hom}(S^\mu, M^\lambda)$, we can then choose homomorphisms which span the image of the operator.

We can always describe the module $M^{F,n}$ as the quotient of a *hook*-permutation module and the automorphism group of F

$$M^{F,n} \simeq M^{(n-m,1,\dots,1)}/_{\mathrm{Aut}(F)}, \tag{8.2}$$

where the part 1 is repeated m times, one time for each vertex of F, and $\mathrm{Aut}(F)$ acts on the module by permuting rows starting from the second.

We can immediately simplify this by combining freely interchangeable nodes of F into the same rows, as we have done for the two examples above.

Definition 8.1. Let F be a flag. Then the *shape* of F is the partition of the vertex set V of F, where two vertices i, j are in the same part if and only if swapping them results in the same flag, i.e., the permutation (i, j) is an automorphism of F. Since we can chain these automorphisms, this property of pairs of vertices defines an equivalence relation between vertices, and thus results in a well-defined partition of V. We assume that its parts are ordered from biggest to smallest (potentially non-unique). We may use them as partitions of $[m] = [|V|]$ by replacing the parts with their cardinalities.

Example 8.2.

$$\mathrm{shape}\left(\begin{array}{ccc} & \textcircled{1} & \\ \textcircled{2} & & \textcircled{3} \end{array}\right) = (\{2,3\},\{1\}),$$

$$\mathrm{shape}\left(\begin{array}{ccc} \textcircled{2} & - & \textcircled{1} \\ \textcircled{3} & - & \textcircled{4} \end{array}\right) = (\{1,2\},\{3,4\}).$$

Remark 8.3. We can calculate $\mathrm{shape}(F)$ of a flag in polynomial time (in $m = |V(F)|$), by checking which of the $\binom{m}{2}$ pairwise permutations between pairs of vertices leave F unchanged.

We use the notation $(c, v\dots)$, where c is a scalar and v a vector, to denote the vector $(c, v_1, v_2, \dots)$.

Theorem 8.4. *Let F be a flag,* $\mathrm{shape}(F)$ *its shape and m the number of its vertices. Then*

$$M^{F,n} \simeq M^{(n-m,\mathrm{shape}(F)\dots)}/_{G},$$

where G is the group of automorphisms of F acting on the vertices in the parts of $\mathrm{shape}(F)$*. Thus, G acts on the rows of the tabloids of shape* $(n-m, \mathrm{shape}(F)\dots)$.

Proof. We know by (8.2) that

$$M^{F,n} \simeq {}^{M^{(n-m,1,\dots,1)}}\!/_{\mathrm{Aut}(F)}.$$

Splitting the quotient into two, we see

$${}^{M^{(n-m,1,\dots,1)}}\!/_{\mathrm{Aut}(F)} \simeq \left({}^{M^{(n-m,1,\dots,1)}}\!/_{\mathrm{Aut}(\mathrm{shape}(F))}\right)/G,$$

where

$$\mathrm{Aut}(\mathrm{shape}(F)) = \bigoplus_{c \in \mathrm{shape}(F)} S_c$$

consists of the permutations in S_n which do not change shape(F). Since the parts of the partition shape(F) are pairwise disjoint, Aut(shape(F)) is a Young-group, and thus

$${}^{M^{(n-m,1,\dots,1)}}\!/_{\mathrm{Aut}(\mathrm{shape}(F))} \simeq M^{(n-m,\mathrm{shape}(F)\dots)}.$$

□

8.3.1 The Decomposition of $M^{F,n}$ *Stabilizes* for n Big Enough

While the modules M^{F,n_1} and M^{F,n_2} are non-isomorphic if $n_1 \neq n_2$, the decomposition into Specht modules does not change anymore once n is big enough. This was first noted in [Ray+18] to stabilize for k-uniform hypergraphs at $n = 2kd$, where d is the degree/level of the hierarchy (which corresponds to the maximum number of considered edges of the hyper-graphs F). Similar properties have also been investigated for variables with a single index and S_n symmetry in [Rie+13], and, more recently, for symmetries coming from more general reflection groups in [DR20]. We see the same stabilization property here, and can refine the result further.

Proposition 8.5. *Let $m < n_1 \leq n_2$, and fix partitions $\lambda \vdash m$ and $(n_1 - m, \lambda \dots) \lhd \mu_1 \vdash n_1$, as well as $(n_2 - m, \lambda \dots) \lhd \mu_2 \vdash n_2$, where μ_1 and μ_2 are identical partitions up to the first, and biggest, parts $(\mu_1)_1$ and $(\mu_2)_1$.*

If $n_1 \geq 2m$, then

$$\mathrm{Hom}(S^{\mu_1}, M^{(n_1-m,\lambda\dots)}) \simeq \mathrm{Hom}(S^{\mu_2}, M^{(n_2-m,\lambda\dots)}).$$

We get a vector space isomorphism by sending the homomorphism ϑ_{T_1}, where T_1 is a semistandard-tableau, to ϑ_{T_2}, where T_2 is the semistandard tableau we get by extending its first row by inserting $n_2 - n_1$ ones at the start of it.

Proof. Let T_1 be a semistandard tableau of shape μ_1 and content $(n_1 - m, \lambda \dots)$. Since T_1 is semistandard, it needs to have $n_1 - m \geq 2m - m = m$ ones in first row. Since μ_1 is a partition of m, the number of ones in the first row of T_1 is at least as big as the length of its second row. Hence, inserting additional ones at the start of the first row results in another semistandard tableau T_2.

Analogously, the number of ones in the first row of a semistandard tableau of shape μ_2 and content $(n_2 - m, \lambda \dots)$ exceeds the length of its second row by at least $n_2 - n_1$, i.e., removing one's results in another semistandard tableau.

Since both insertion and removal are clearly injective, and the semistandard tableaux form a basis of the multiplicity spaces, they are isomorphic. □

Corollary 8.6. *Let $G = (V, E)$ be a (hyper-)graph. Then the module $M^{G,n}$ decomposes as $M^{G,2|V|}$ if $n \geq 2|V|$ in the sense that if for some set of partitions I*

$$M^{G,2|V|} \simeq \bigoplus_{\mu \in I} c_\mu S^{(2|V| - |\mu|, \mu \dots)},$$

then

$$M^{G,n} \simeq \bigoplus_{\mu \in I} c_\mu S^{(n - |\mu|, \mu \dots)}.$$

The explicit homomorphisms are given by the tableaux one gets by extending the first rows of the semistandard tableaux defining the ones for $M^{G,2|V|}$ with ones.

In particular, we see that the decomposition of the module of polynomials $\mathbb{R}[x]_{\leq d}$ up to degree d does not change for $n \geq 2kd$. This implies that the size of the corresponding SDPs does not change anymore at that point, and we will see that, after some rescaling, even all coefficients of the constraints of the SDPs converge. This then allows us to consider the case where $n \to \infty$, and will result in a hierarchy equivalent to the (Lasserre) *flag SOS* hierarchy, while being more efficient to compute. Interestingly, the limit is not unique: One can determine additional, new limit hierarchies, which can then handle *degenerate* problems where the usual flag methods fail, since there all subgraph densities approach zero as n increases.

8.3.2 Decomposing the Modules $M^{F,n}$ Explicitly

We have seen that the modules $M^{F,n}$ are isomorphic to quotients of the form M^λ/G, where G is a group acting on the rows (starting from the second) of the tabloids, and that the decomposition does not change anymore once n is big enough. But we still have to determine how the module actually decomposes.

To do this, we need to determine a basis of the multiplicity space

$$\mathrm{Hom}(S^\mu, {}^{M^\lambda}\!/_G) = \mathcal{R}_G(\mathrm{Hom}(S^\mu, M^\lambda)),$$

where $\mathcal{R}_G$ is the Reynolds operator of the action of G on the multiplicity space of S^μ in the full permutation module M^λ, extended in the obvious way from the action of G on M^λ. As we have seen earlier in Section 7.2.3, a basis of $\mathrm{Hom}(S^\mu, M^\lambda)$ is given by the semistandard tableaux of shape μ and content λ. One could now either calculate $\mathcal{R}_G$ explicitly in this basis, and chose a set of columns generating the column space of the operator, or determine the intersection of the spaces of homomorphisms invariant under the actions of the elements of a generator of G.

The problem here is that the action of G is not very straightforward. That is, there is no known efficient algorithm to express a homomorphism given by a tableau that is not coming from a semistandard tableaux as linear combination of homomorphisms each given by a semistandard tableau. While there is such a "straightening algorithm" (involving Garnir elements) for polytabloids to express them as linear combination of polytabloids coming from standard tableaux (see, for example, [Sag13, Section 2.6]), the usual way to prove that the semistandard tableaux form a basis of the multiplicity space is based on an existence argument (Lemma 8.7).

Of course, one could calculate the basis explicitly, but due to the exponential number of terms in polytabloids one should avoid it. Instead, one can work on the level of semistandard tableaux and tabloids, if one generalizes tabloids further to allow for *partial* rows.

First, we can take a look at the image of a polytabloid e_t (an element of the basis of S^μ) under a homomorphism $\vartheta_T \in \mathrm{Hom}(S^\mu, {}^{M^\lambda}\!/_G)$, given by a standard tableau t and a generalized tableau T respectively.

$$\begin{aligned}
\vartheta_T(e_t) &= \vartheta_T\left(\sum_{\sigma\in C_t} \mathrm{sgn}(\sigma)\{\sigma t\}\right) \\
&= \sum_{\sigma\in C_t} \mathrm{sgn}(\sigma) \sum_{T'\in\{T\}} (\sigma t)[T'] \\
&= \sum_{\sigma\in C_t} \mathrm{sgn}(\sigma) \sum_{T'\in\{T\}} \{\text{tableau with entry } (\sigma t)(i,j) \text{ in row } T'(i,j)\} \\
&= \sum_{\sigma\in C_t} \mathrm{sgn}(\sigma) \sum_{T'\in\{T\}} \{\text{tableau with entry } t(\sigma^{-1}(i),j) \text{ in row } T'(i,j)\}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\sigma \in C_t} \operatorname{sgn}(\sigma) \sum_{T' \in \{T\}} \{\text{tableau with entry } t(i,j) \text{ in row } T'(\sigma(i), j)\} \\
&= \sum_{\sigma \in C_t} \operatorname{sgn}(\sigma) \sum_{T' \in \{T\}} \{\text{tableau with entry } t(i,j) \text{ in row } \sigma^{-1}(T'(i,j)\} \\
&= \sum_{\sigma \in C_t} \operatorname{sgn}(\sigma) \sum_{T' \in \{T\}} t[\sigma^{-1} T'].
\end{aligned}
$$

That is, we can decompose $\vartheta_T(e_t)$, earlier introduced in Section 7.2.3, as a sum over terms of the form $c_i t[T_i], c_i \in \mathbb{Z}$, where the T_i are generalized tableaux obtained by first permuting within rows and then within columns of T.

Lemma 8.7 (Lemma 2.10.7 in [Sag13])**.** *Let T be a (not necessarily semistandard) generalized tableau of shape μ, and we assume that the homomorphism decomposes in the form*

$$\vartheta_T(e_t) = \sum_i c_i t[T_i], \tag{8.3}$$

where $c_i \in \mathbb{Z}$ and T_i are generalized tableaux.

Then either $\vartheta_T(v) = 0$ for all $v \in S^\mu$, or there exists an i such that T_i is semistandard.

This leads to the constructive argument that one can repeatedly subtract $c_i \vartheta_{T_i}$ from ϑ_T, where T_i is semistandard, to eventually end up with the full decomposition in the basis. How to do this explicitly on the other hand is left open in the literature. Again, the problem is that the number of terms $c_i t[T_i]$ grows exponentially in both the row and columns size of the tableaux.

So, we are going to need two things: A way to identify semistandard tableaux appearing in the sum efficiently, and a way to work with the difference $\vartheta_T(e_t) - \vartheta_{T_i}(e_t)$.

The usual way to prove the existence of a semistandard tableau in the decomposition (8.3) is to start with an arbitrary term in the sum, and then apply *Garnir-elements* to it to move step by step towards a term with a semistandard tableau.

Now, while this is enough to get a constructive proof, in practice this result does not seem too useful, if one works with bigger tableaux. The main problem is this: Having identified a semistandard tableau T_i in the decomposition (8.3) this way, one then wants to repeat the same process with the difference $\vartheta_T(e_t) - \vartheta_{T_i}(e_t)$. It does not seem clear (assuming one does not work with the decomposition explicitly) how to determine a term in the difference, as there are

potentially many ways to arrive at the same term from T (resp. T_i), and terms may cancel out. Especially, it seems hard to decide when one is "done", i.e., one has found the full decomposition and the remaining difference is zero.

Instead, we are going to first determine *all* semistandard terms appearing in the decomposition of $\vartheta_T(e_t)$, and then describe the difference $\vartheta_T(e_t) - \sum_{T_i \text{ semistandard}} \vartheta_{T_i}(e_t)$ with as few "*partial*" homomorphisms as possible. For a set of generalized tableaux S we call

$$\overline{\vartheta_S}(t) := \sum_{T' \in S} t[T']$$

a *partial homomorphism* if S is a subset of a generalized tabloid $\{T\}$, here usually obtained by fixing certain entries to specific positions in a row of T. We have

$$\vartheta_T = \sum_{T' \in \{T\}} \overline{\vartheta_{\{T'\}}},$$

where the $\{T'\}$'s are sets with one element (instead of the tabloids) and

$$\overline{\vartheta_{S_1 \sqcup S_2}} = \overline{\vartheta_{S_1}} + \overline{\vartheta_{S_2}}.$$

Let us start with an example. Note that to permute rows in the image of ϑ_T, we apply the permutation to each entry of T simultaneously. Consider the homomorphism given by the permuted semistandard tableau

$$T = (12)\overline{\begin{array}{|c|c|c|c|}\hline 1&1&2&3\\\hline 2&3\\\cline{1-2}\end{array}} = \overline{\begin{array}{|c|c|c|c|}\hline 2&2&1&3\\\hline 1&3\\\cline{1-2}\end{array}} = \overline{\begin{array}{|c|c|c|c|}\hline 1&2&2&3\\\hline 1&3\\\cline{1-2}\end{array}},$$

which is obviously not semistandard. We now want to find all semistandard tableaux we can obtain by first permuting within rows, and then within columns of T. To do this, we first consider the partial homomorphisms where the position of the ones in T are fixed (written in **bold** font) in different ways:

$$= \overline{\begin{array}{|c|c|c|c|}\hline \mathbf{1}&2&2&3\\\hline \mathbf{1}&3\\\cline{1-2}\end{array}} \cup \overline{\begin{array}{|c|c|c|c|}\hline 1&2&2&3\\\hline \mathbf{3}&\mathbf{1}\\\cline{1-2}\end{array}} \cup \overline{\begin{array}{|c|c|c|c|}\hline 2&\mathbf{1}&2&3\\\hline \mathbf{1}&3\\\cline{1-2}\end{array}} \cup \overline{\begin{array}{|c|c|c|c|}\hline 2&\mathbf{1}&2&2\\\hline \mathbf{3}&\mathbf{1}\\\cline{1-2}\end{array}}$$
$$\cup \overline{\begin{array}{|c|c|c|c|}\hline 2&2&\mathbf{1}&3\\\hline \mathbf{1}&3\\\cline{1-2}\end{array}} \cup \overline{\begin{array}{|c|c|c|c|}\hline 2&2&\mathbf{1}&3\\\hline \mathbf{3}&\mathbf{1}\\\cline{1-2}\end{array}} \cup \overline{\begin{array}{|c|c|c|c|}\hline 2&2&3&\mathbf{1}\\\hline \mathbf{1}&3\\\cline{1-2}\end{array}} \cup \overline{\begin{array}{|c|c|c|c|}\hline 2&2&3&\mathbf{1}\\\hline \mathbf{3}&\mathbf{1}\\\cline{1-2}\end{array}}.$$

Now we can make use of the observation that given a semistandard tableau, the tableau obtained by restricting to the boxes with entries at most some i is again semistandard. Since there is only one semistandard tableau with just two ones (namely $\overline{\begin{array}{|c|c|}\hline 1&1\\\hline\end{array}}$), we do know that after sorting the columns every semistandard tableau must have two ones in the first row at the start. Thus, only the partial homomorphisms given by

$$\overline{\begin{array}{|c|c|c|c|}\hline \mathbf{1}&2&2&3\\\hline \mathbf{3}&\mathbf{1}\\\cline{1-2}\end{array}} \text{ and } \overline{\begin{array}{|c|c|c|c|}\hline 2&\mathbf{1}&2&3\\\hline \mathbf{1}&3\\\cline{1-2}\end{array}}$$

may contain semistandard tableaux. Next, we can repeat the same for the two's in both tableaux. Eliminating the sets which may not contain a semistandard tableaux (by sorting columns and checking if the tableaux restricted to the entries one and two are semistandard), we obtain the three sets (and corresponding partial homomorphisms)

$$\begin{array}{l}\mathbf{1}\ 3\ \mathbf{2}\ \mathbf{2}\\ 3\ \mathbf{1}\end{array}, \begin{array}{l}\mathbf{2}\ \mathbf{1}\ \mathbf{2}\ 3\\ \mathbf{1}\ 3\end{array} \text{ and } \begin{array}{l}3\ \mathbf{1}\ \mathbf{2}\ \mathbf{2}\\ \mathbf{1}\ 3\end{array}$$

This tells us that the decomposition of ϑ_T contains exactly the three column sorted semistandard tableaux

$$\begin{array}{|c|c|c|c|}\hline 1&1&2&2\\\hline 3&3\\\cline{1-2}\end{array}, \begin{array}{|c|c|c|c|}\hline 1&1&2&3\\\hline 2&3\\\cline{1-2}\end{array} \text{ and } \begin{array}{|c|c|c|c|}\hline 1&1&2&2\\\hline 3&3\\\cline{1-2}\end{array},$$

all appearing with sign -1, as we need to permute one pair of entries within one column of each to sort them.

Now we subtract the three corresponding homomorphisms, and take a look at the difference

$$\vartheta_T - \left(-2\vartheta_{\begin{array}{l}1\ 1\ 2\ 2\\ 3\ 3\end{array}} - \vartheta_{\begin{array}{l}1\ 1\ 2\ 3\\ 2\ 3\end{array}} \right).$$

Lemma 8.7 tells us that this difference is zero if and only if its decomposition (8.3) does not contain a semistandard tableau. Since we eliminate all semistandard tableaux appearing in the decomposition given by ϑ_T, remaining semistandard tableaux can only appear in the terms we subtract. While for any given semistandard tableau T' the decomposition of $\vartheta_{T'}(e_t)$ contains a term corresponding to the semistandard T' itself, it may not be the only one. In this case both tableaux are the unique semistandard tableaux appearing in both their corresponding decompositions, i.e., we have

$$\vartheta_{\begin{array}{l}1\ 2\ 2\ 3\\ 1\ 3\end{array}} = -2\vartheta_{\begin{array}{l}1\ 1\ 2\ 2\\ 3\ 3\end{array}} - \vartheta_{\begin{array}{l}1\ 1\ 2\ 3\\ 2\ 3\end{array}}.$$

$$(12)\begin{array}{l}1\ 1\ 2\ 3\\ 2\ 3\end{array} = \begin{array}{l}1\ 2\ 2\ 3\\ 1\ 3\end{array} = -2\begin{array}{l}1\ 1\ 2\ 2\\ 3\ 3\end{array} - \begin{array}{l}1\ 1\ 2\ 3\\ 2\ 3\end{array}$$

Generally there may be more semistandard tableaux appearing in the decomposition, for example the homomorphism corresponding to the semistandard tableau

$$\begin{array}{l}1\ 4\\ 2\ 5\\ 3\end{array}$$

contains terms corresponding to both this tableau itself, and to the semistandard tableau

$$\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline 5 \\ \cline{1-1}\end{array},$$

which is obtained by first permuting the second row, and then sorting columns. Thus, we generally have to recursively repeat the procedure of determining all semistandard tableaux appearing in the decompositions of the subtracted terms, excluding their main terms. This algorithm has to finish in a finite number of steps due to the *dominance lemma for generalized tableaux* (see e.g., Lemma 2.10.2 in [Sag13]), which tells us that all other semistandard tableaux appearing in the decomposition have to be *smaller* than the main one (according to the column dominance ordering of generalized tableaux). As there is a finite number of tableaux of a given size, this algorithm has to finish in a finite number of steps.

Thus, we obtain Algorithm 5 to write a homomorphism given by a tableau T, which is not semistandard, in the semistandard basis.

Algorithm 5: Decompose a generalized tableau T in terms of semistandard tableaux

```
// Determine all appearing semistandard tableaux:
PartialHomomorphisms ← {ϑ_T};
for i = 1, ..., m do
    Split all ϑ̄_S ∈ PartialHomomorphisms further by fixing all i-entries in
      all possible ways;
    Delete all partial homomorphisms which are not semistandard after
      sorting their columns and restricting to entries 1, ..., i;
// Determine their multiplicities:
c_{T'} ← 0 for all semistandard tableaux T';
for ϑ̄_S ∈ PartialHomomorphisms do
    p ← the permutation sorting the columns of S;
    T' ← column-sorted S;
    c_{T'} = c_{T'} + sign(p);
if not the highest level of the recursion then
    c_T = 0;
return Σ_{T'} c_{T'}(ϑ_{T'} − Decompose(T'))
```

Determining the Reynolds Operators. To determine the symmetry adapted basis of $M^\lambda/_G$, we need to determine a basis of the multiplicity space of S^μ in $M^\lambda/_G$ for each μ. As mentioned above, we let $\mathcal{R}_G$ be the Reynolds operator of G and see that

$$\mathrm{Hom}(S^\mu, M^\lambda/_G) = \mathcal{R}_G(\mathrm{Hom}(S^\mu, M^\lambda)).$$

If we can now determine a matrix representation of $\mathcal{R}_G$ explicitly in the semistandard basis of $\mathrm{Hom}(S^\mu, M^\lambda)$, we can simply choose a subset of columns which span the image of $\mathcal{R}_G$ to obtain a basis of $\mathrm{Hom}(S^\mu, M^\lambda/_G)$.

We can now use this Algorithm 5 to determine the matrix representation $M_{\mathcal{R}_G} \in \frac{1}{|G|}\mathbb{Z}^{\dim(\mathrm{Hom}(S^\mu,M^\lambda))\times\dim(\mathrm{Hom}(S^\mu,M^\lambda))}$ of the Reynolds operator $\mathcal{R}_G$ in the basis of homomorphisms given by semistandard tableaux. We do this by first decomposing $\sigma(\vartheta_T)$ for each semistandard T and element σ of a generator of F. Knowing the matrices corresponding to a generator, we can then generate the full matrices corresponding to the full group, and average them.

In some cases we may see that $\sigma(\vartheta_T) = \pm\sigma(\vartheta_{T'})$ for another semistandard tableau T'. In these cases we know that $\mathcal{R}_F(\vartheta_T) = \mathcal{R}_F(\vartheta_{T'})$, allowing us to eliminate one of the basis elements immediately. Generally this is not enough, but in some cases these eliminations alone already result in a basis, if it matches the dimension of $\mathrm{Hom}(S^\mu, M^\lambda/_F)$, in the case this dimension was known beforehand. This idea is used to determine a basis of $\mathrm{Hom}(S^{(n-2,1,1)}, M^{1,\dots,1}/_{\mathbb{Z}/n\mathbb{Z}})$ for an application to crossing numbers in [BP22].

Remark 8.8. We can determine $M_{\mathcal{R}_G}$ more efficiently by first determining a Stabilizer-Chain of G via the Schreier-Sims algorithm, and then iterating over a transversal of each stabilizer in the chain. We refer the interested reader to [Ser03; Sim70; Knu91] for details.

8.3.3 Multiplying and Symmetrizing the Elements of the Symmetry Adapted Basis

We can now apply the algorithm described in Section 7.2.4 to compute the symmetrized products of the symmetry adapted basis we obtained from computing the Reynolds operators. The algorithm returns a linear combination in terms of *overlaps* between two tabloids, which was defined as

$$\mathrm{overlap}(\{t_1\},\{t_2\}) := (|\mathrm{row}(\{t_1\},i) \cap \mathrm{row}(\{t_2\},j)|)_{i,j=1}^{m}.$$

Let us consider two graphs G_1 and G_2, such that the corresponding modules are isomorphic to the quotients $M^{G_1,n} \simeq M^{\lambda^1}/_{F_1}$ and $M^{G_2,n} \simeq M^{\lambda^2}/_{F_2}$. Especially we

know how to translate tabloids $\{t_1\}$ and $\{t_2\}$ of shapes λ^1 and λ^2 respectively to specific full labelings of G_1 and G_2.

Given an overlap matrix $O \in \mathbb{N}^{m\times m}$, where m is the maximum number of rows of λ^1 and λ^2, we can now build a representative of the equivalence class corresponding to the overlap. To do this, we simply take any fixed labeling of G_1, and then label G_2 greedily corresponding to the overlap O. We can then glue together the two fully labeled graphs obtained this way, and *label them canonically* to obtain the corresponding basis element of $\mathbb{R}[x]^{S_n}$. This is exactly the action of the *unlabeling operator* $[\![\cdot]\!]$ as introduced in Chapter 6.1.

While the problem of canonically labeling a graph is at least as hard as the graph isomorphism problem, in practice there exist algorithms, such as NAUTY and TRACES [MP14], that perform well, especially for the (very) small graphs appearing in the levels of the hierarchies considered in this work. The algorithms in [MP14] can easily be generalized to general flags, and a general version relying just on a function that can *distinguish* vertices from another has been implemented as part of the Julia package "FlagSOS.jl", described in Section 8.7.

8.4 Example and Interpretation: The Reduced Second Level of the Lasserre Hierarchy for Graphs

Let us take a look at a specific small example: The Lasserre hierarchy indexed by monomials up to degree two, i.e., graphs with at most two edges. We assume here that $n \geq 8$ so that the decomposition stabilizes by Corollary 8.6. The case $n < 8$ can be decomposed analogously, but some semistandard tableaux may not exist, or look a bit different as columns may need to be sorted, thus requiring separate cases for each small n.

Here the S_n module $\mathbb{R}[x]_{\leq 2}$ decomposes directly into the four graph modules

$$\mathbb{R}[x]_{\leq 2} = M^{\varnothing,n} \oplus M^{\bullet\!\bullet,n} \oplus M^{\wedge,n} \oplus M^{\square,n},$$

of which the first three are isomorphic to permutation modules:

$$M^{\varnothing,n} \simeq M^{(n)},$$
$$M^{\bullet\!\bullet,n} \simeq M^{(n-2,2)},$$
$$M^{\wedge,n} \simeq M^{(n-3,2,1)}.$$

We can then use the known decomposition of each permutation module, which was described in detail in Section 7.2, to determine a symmetry adapted basis of each of the three graph modules.

We saw earlier in (8.1) that the fourth module, which corresponds to the graph with two disjoint edges, is isomorphic to the quotient

$$M^{\bullet\!-\!\bullet\;\bullet\!-\!\bullet,n} \simeq M^{(n-4,2,2)}/_{S_2},$$

where S_2 permutes the last two rows of the tabloids in $M^{(n-4,2,2)}$. Next, we can use Algorithm 5 to compute the Reynolds operators for each of the multiplicity spaces $\mathrm{Hom}(S^\mu, M^{(n-4,2,2)})$ for the nine partitions $\mu \trianglerighteq (n-4,2,2)$ dominating $(n-4,2,2)$. We give all nine Reynolds operators in Table 8.1, where the semistandard basis is ordered by sorting their column-wise vectorization lexicographically.

μ	$\mathcal{R}_{S_2}$	Chosen basis
$(4,2,2)$	$\begin{pmatrix}1\end{pmatrix}$	$\begin{array}{|c|c|c|c|}\hline 1&1&\cdots&1\\\hline 2&2\\\cline{1-2} 3&3\\\cline{1-2}\end{array}$
$(4,3,1)$	$\begin{pmatrix}0\end{pmatrix}$	
$(4,4)$	$\begin{pmatrix}1\end{pmatrix}$	$\begin{array}{|c|c|c|c|c|c|}\hline 1&1&1&1&\cdots&1\\\hline 2&2&3&3\\\cline{1-4}\end{array}$
$(5,2,1)$	$\frac{1}{2}\begin{pmatrix}1&-1\\-1&1\end{pmatrix}$	$\begin{array}{|c|c|c|c|c|}\hline 1&1&\cdots&1&3\\\hline 2&2\\\cline{1-2} 3\\\cline{1-1}\end{array}$
$(5,3)$	$\frac{1}{2}\begin{pmatrix}1&1\\1&1\end{pmatrix}$	$\begin{array}{|c|c|c|c|c|}\hline 1&1&1&\cdots&1&3\\\hline 2&2&3\\\cline{1-3}\end{array}$
$(6,1,1)$	$\begin{pmatrix}0\end{pmatrix}$	
$(6,2)$	$\frac{1}{2}\begin{pmatrix}1&0&1\\0&2&0\\1&0&1\end{pmatrix}$	$\begin{array}{|c|c|c|c|c|c|}\hline 1&1&\cdots&1&3&3\\\hline 2&2\\\cline{1-2}\end{array}$, $\begin{array}{|c|c|c|c|c|c|}\hline 1&1&\cdots&1&2&3\\\hline 2&3\\\cline{1-2}\end{array}$
$(7,1)$	$\frac{1}{2}\begin{pmatrix}1&1\\1&1\end{pmatrix}$	$\begin{array}{|c|c|c|c|c|c|}\hline 1&\cdots&1&2&3&3\\\hline 2\\\cline{1-1}\end{array}$
(8)	$\begin{pmatrix}1\end{pmatrix}$	$\begin{array}{|c|c|c|c|c|c|c|}\hline 1&\cdots&1&2&2&3&3\\\hline\end{array}$

Table 8.1: Reynolds operators of S_2 on $\mathrm{Hom}(S^\mu, M^{(n-4,2,2)})$

As we can see in Table 8.1, the graph module $M^{\bullet\!-\!\bullet\;\bullet\!-\!\bullet,n}$ decomposes into 8 Specht modules, while the full permutation module $M^{(4,2,2)}$ decomposes into 14. Together with the decompositions of the first three modules, we obtain the full de-

composition

$$\begin{aligned}\mathbb{R}[x]_{\leq 2} =& M^{\emptyset,n} \oplus M^{\bullet,n} \oplus M^{\wedge,n} \oplus M^{\boxplus,n} \\ \simeq& S^{(n-4,2,2)} \oplus S^{(n-4,4)} \oplus 2S^{(n-3,2,1)} \oplus 2S^{(n-3,3)} \\ & \oplus S^{(n-2,1,1)} \oplus 5S^{(n-2,2)} \oplus 4S^{(n-1,1)} \oplus 4S^{(n)}.\end{aligned}$$

The multiplicities give us the block-sizes of the block-diagonalized SDP: one block of size 5, two each of sizes 4 and 2, and three blocks of size 1.

We can now compute the (polynomial) entries of the block-diagonalized SDP. Take, for example, the semistandard tableau

$$T_1 = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & \cdots & 1 & 2 \\ \hline 2 & 3 \\ \cline{1-2}\end{array}$$

for the graph $\wedge$, and the semistandard tableau

$$T_2 = \begin{array}{|c|c|c|c|c|c|}\hline 1 & 1 & \cdots & 1 & 3 & 3 \\ \hline 2 & 2 \\ \cline{1-2}\end{array}$$

of the same shape for the graph $\boxplus$. To compute the corresponding elements of the symmetry adapted basis, we first need to understand the two homomorphisms $\vartheta_{T_i} : M^{(n-2,2)} \to \mathbb{R}[x]_{\leq 2}$ for $i = 1, 2$.

Graphs with Grouped Vertices. In Section 7.2 we defined the homomorphisms corresponding to semistandard tableaux T to be given by

$$\vartheta_T(\{t\}) = \sum_{T' \in \{T\}} t[T'],$$

i.e., we here send a tabloid $\{t\}$ of shape $(n-2,2)$ to a *sum over the elements T' of the row equivalence class of the semistandard tableau T*, where $t[T']$ is the tabloid of shape $(n-3,2,1)$ respectively $(n-4,2,2)$ obtained by moving the entries of t to the rows determined by the entries of T'. Hence, the images of the tabloid

$$\{t\} = \overline{\begin{array}{c}[n]\setminus\{i,j\} \\ \hline i \quad j\end{array}}$$

are

$$\vartheta_{T_1}(\{t\}) = \sum_{k \in [n]\setminus\{i,j\}} \left(\overline{\begin{array}{c}[n]\setminus\{i,j,k\} \\ \hline i \quad k \\ \hline j\end{array}} + \overline{\begin{array}{c}[n]\setminus\{i,j,k\} \\ \hline j \quad k \\ \hline i\end{array}} \right)$$

and

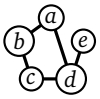
$$\vartheta_{T_2}(\{t\}) = \sum_{k<\ell\in[n]\setminus\{i,j\}} \begin{array}{|c|c|}\hline [n]\setminus\{i,j,k,\ell\} \\ \hline i \;\; j \\ \hline k \;\; \ell \\ \hline \end{array}\,.$$

We can now apply the homomorphisms relating the permutation and graph modules to first obtain linear combinations of monomials and then graphs:

$$\begin{aligned}
\vartheta_{T_1}(\{t\}) &= \sum_{k\in[n]\setminus\{i,j\}} \left(x_{ij}x_{jk} + x_{ij}x_{ik}\right) \\
&= \sum_{k\in[n]\setminus\{i,j\}} \left(\begin{array}{ccc} & i & \\ j & & k \end{array} + \begin{array}{ccc} & i & \\ j & & k \end{array} \right) \\
&= (n-2)\left(\begin{array}{cc} i & \\ j & \bullet \end{array} + \begin{array}{ccc} & i & \\ j & & \bullet \end{array} \right) \\
&= (n-2) \begin{array}{|c|}\hline i \\ j \\ \hline \end{array}\;\bullet \in M_{\vee,n}, \text{ and} \\
\vartheta_{T_2}(\{t\}) &= \sum_{k<\ell\in[n]\setminus\{i,j\}} x_{ij}x_{k\ell} \\
&= \sum_{k<\ell\in[n]\setminus\{i,j\}} \begin{array}{cc} i & j \\ k & \ell \end{array} \\
&= \binom{n-2}{2} \begin{array}{cc} i & j \\ \bullet & \bullet \end{array} \in M_{\vdots\vdots,n}.
\end{aligned}$$

Here we first use the notation where we *group* together vertices to indicate that we *sum over all the possible graphs we can obtain by permuting vertex labels within the group*. The edge connecting i resp., j to the unlabeled vertex (formerly vertex k) can be connected to either of the two vertices i or j. We can interpret the unlabeled (filled in) vertices to be part of a group together with all vertices that do not appear explicitly in the graph. No vertex group is necessary for $\vartheta_{T_2}(\{t\})$ since the graph already has the symmetries (which we can see in T_2 itself: the second row only contains twos).

This notation can be generalized to more complicated homomorphisms in a straightforward manner. Consider, for example, the graph

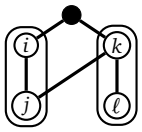
$$G = \quad$$


of shape $\text{shape}(G) = (\{a,c\},\{b\},\{d\},\{e\})$. Since the automorphism group of G is exactly the corresponding Young-group $S_{\{1,3\}} \times S_{\{2\}} \times S_{\{4\}} \times S_{\{5\}}$, we know that $M^{G,n} \simeq M^{(n-5,2,1,1,1)}$ for n big enough. We now obtain an element of $\text{Hom}(S^{(n-4,2,2)}, M^{G,n})$ by the semistandard tableau

$$T = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & \cdots & 1 & 2 \\ \hline 2 & 3 \\ \cline{1-2} 4 & 5 \\ \cline{1-2}\end{array}\,.$$

Here we need three groups: One group for unlabeled vertices, containing one of the two vertices in the first part $\{a,c\}$ of $\text{shape}(G)$, and two more groups of size two each:

$$\vartheta_T\left(\begin{array}{c}[n]\setminus\{i,j,k,\ell\} \\ \hline i \;\; j \\ \hline k \;\; l \\ \hline\end{array}\right) = (n-4)\sum_{\sigma\in S_{\{i,j\}}\times S_{\{k,\ell\}}} \sigma\left(\quad\right)$$


$$= \quad \in M^{G,n}.$$


Specht Flags. But what we are actually interested are the elements of the symmetry adapted basis, i.e., the images of *polytabloids* under ϑ_T. Specifically, we *fix* one polytabloid $e_t \in S^\mu$ for each shape μ, since we only need the image of one and the same polytabloid under all homomorphisms in $\text{Hom}(S^\mu, \mathbb{R}[x]_{\le d})$. Continuing the example from above, let us fix the standard tableaux

$$t = \begin{array}{|c|c|c|c|c|}\hline 1 & 3 & 5 & \cdots & n \\ \hline 2 & 4 \\ \cline{1-2}\end{array}$$

of shape $(n-2,2)$ obtained by filling it column-wise. The corresponding polytabloid is

$$e_t = \frac{1\;3\;5\cdots n}{2\;4} - \frac{2\;3\;5\cdots n}{1\;4} - \frac{1\;4\;5\cdots n}{2\;3} + \frac{2\;4\;5\cdots n}{1\;3} \in M^{(n-2,2)}.$$

We now obtain two elements of the symmetry adapted basis of $\mathbb{R}[x]_{\leq 2}$ by

$$\vartheta_{T_1}(e_t) = (n-2)\left(\quad - \quad - \quad + \quad \right) \in M$$

$$\vartheta_{T_2}(e_t) = \binom{n-2}{2}\left(\quad - \quad - \quad + \quad \right) \in M$$

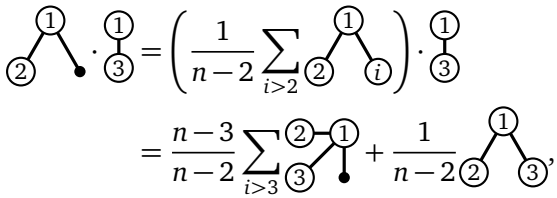
As these quantum flags/polynomials generalize *(higher) Specht polynomials*, which were first introduced in [ATY97] (see also [MRV21] for more recent results related to them), we call these quantum graphs *(higher) Specht graphs*, or more generally *Specht flags*.

Calculating the Entries of the Block-Diagonalized SDP. To obtain the block-diagonalization, we now need to compute the *symmetrized products* of the elements of the symmetry adapted basis. The symmetrized product of the two Specht graphs $\vartheta_{T_1}(e_t)$ and $\vartheta_{T_2}(e_t)$ is

$$\begin{aligned}
&\frac{1}{(n-2)\binom{n-2}{2}} [\![\vartheta_{T_1}(e_t)\vartheta_{T_2}(e_t)]\!] \\
=&[\![\left(\quad - \quad - \quad + \quad \right) \\
&\cdot \left(\quad - \quad - \quad + \quad \right)]\!] \\
=&4 \left[\!\!\left[\quad \cdot \quad \right]\!\!\right] - 8 \left[\!\!\left[\quad \cdot \quad \right]\!\!\right] \\
&+ 4 \left[\!\!\left[\quad \cdot \quad \right]\!\!\right],
\end{aligned} \tag{8.4}$$

where we used the fact that we symmetrize the products in the last step. Here we only need to know how the *labels overlap relative to each other* between the two quantum graphs of each of the three pairs. The algorithm explained in Section 8.3.3 does exactly return a decomposition corresponding to this step, split a bit further by the way the *unlabeled vertices can overlap with the other graph*.

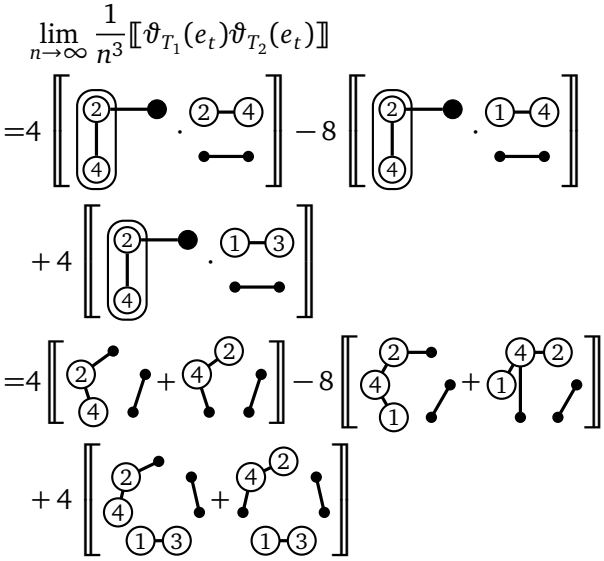
If n is finite we now still need to consider the fact that unlabeled vertices of a flag average over all *labels that do not appear in the graph*, which potentially includes vertices of the flag we glue the flag to. For example,

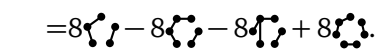


which results in a significant number of additional terms in $[\![\vartheta_{T_1}(e_t)\vartheta_{T_2}(e_t)]\!]$.

While we can calculate the exact coefficients as rational polynomials in n using the algorithm explained in Section 8.3.3, by working with *monomials with exponents that are polynomials in n*, we are mostly interested in the case where n approaches infinity. We can modify the algorithm to instead return the decomposition at step (8.4), and continue to compute the limit case from there.

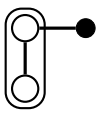
The Limit Case. To make the coefficients in $[\![\vartheta_{T_1}(e_t)\vartheta_{T_2}(e_t)]\!]$ converge as n approaches infinity, we rescale the basis: We simply divide each element $\vartheta_T(e_t)$ of the symmetry adapted basis by n^m, where m is the number of vertices appearing in the first row of T (i.e., unlabeled vertices). This then turns $\vartheta_T(e_t)$ into a finite linear combination of flags, a quantum flag, in the limit, which we can continue to work with to compute:





We can compute the other coefficients analogously to obtain the full reduced second level of the hierarchy, both for fixed n and the limit case.

A Product for Graphs with Grouped Vertices. Since we are only interested in unlabeled products of elements of the form $\vartheta_T(e_t)$ for a fixed polytabloid e_t, we can get rid of the labels fully, and simply write



$$:= \lim_{n\to\infty} \frac{1}{n}\vartheta_{T_1}(e_t)$$

for the limit Specht flags contained in the symmetry adapted basis. In general, it is enough to give a graph G together with a partition $V(G) = V_0 \sqcup V_1 \sqcup V_2 \sqcup \cdots \sqcup V_k$ of its vertices to fully determine the products $[\![\vartheta_{T_1}(e_t)\vartheta_{T_2}(e_t)]\!]$. The first vertex group V_0 is the special group of unlabeled vertices, i.e., it can be seen as the group containing $V_0 \cup [n] \setminus V(G)$, and can potentially be empty. We assume that the remaining groups are ordered by size, i.e., $|V_1| \geq |V_2| \geq \cdots \geq |V_k|$.

Definition 8.9. Let $\mathbb{G}_1 = (G_1, (V_i^1)_{i=0,\dots,m_1})$ and $\mathbb{G}_2 = (G_2, (V_i^2)_{i=0,\dots,m_2})$ be two graphs (Flags) with grouped vertices. Let

$$\lambda_i = (n - |V(G_i)| + |V_0^i|, |V_1^i|, |V_2^i|, \dots, |V_k^i|)$$

be the partitions giving the group sizes of the two graphs, where n is a variable considered to be big enough according to Proposition 8.5.

Then

$$\mathbb{G}_1 \cdot \mathbb{G}_2 := 0,$$

if $\lambda_1 \neq \lambda_2$, and

$$\mathbb{G}_1 \cdot \mathbb{G}_2 := \lim_{n\to\infty} \frac{1}{n^{|V_0^1|+|V_0^2|}} [\![\vartheta_{T_1}(e_t)\vartheta_{T_2}(e_t)]\!], \tag{8.5}$$

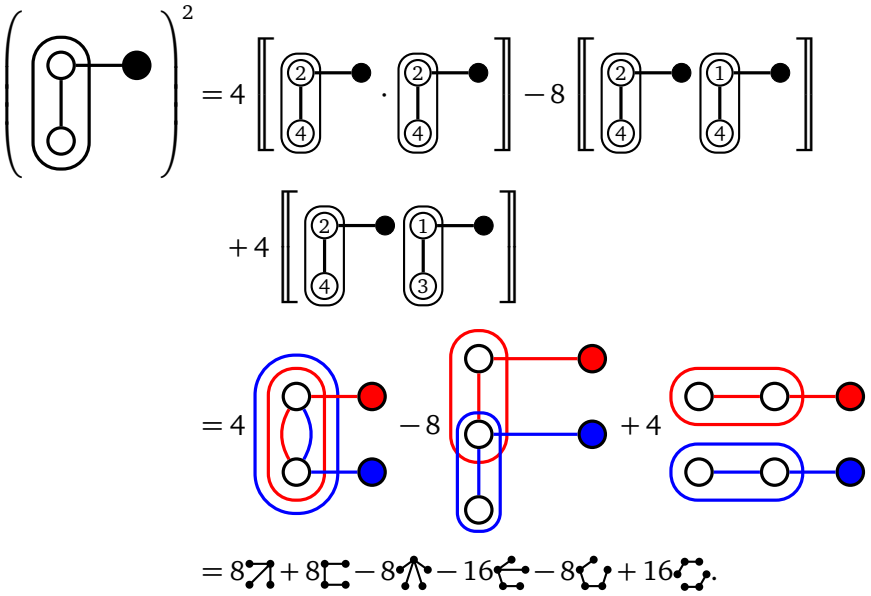
for a fixed tableau t of shape $\lambda = \lambda_i$. Here

$$\vartheta_{T_i} \colon S^\lambda \to M^{G_i,n}$$

are two homomorphisms embedding S^λ into the flag algebra, and the T_i are the generalized tableaux obtained by filling row j of T_i with $|V_j \cap \text{shape}(G_i)_\ell|$ times entry ℓ.

If we loose the specific order of the V_i's, other than V_0, we can still recover the product up to sign, because $\vartheta_T = \vartheta_{T'}$ if the tableaux are row-equivalent. And by definition of polytabloids, $e_t = e_{t'}$ if we can obtain t' from t by swapping two rows of the same *even* length, and $e_t = -e_{t'}$ if we can obtain t' from t by swapping two rows of same odd length.

We can also visualize the product without using labels, by *drawing the groups on top of each other*, and using, for example, two different colors for the symmetrization groups, which are symmetrized independently of each other:



The Symmetry Adapted Basis. This means that we can give a symmetry adapted basis of $\mathbb{R}[x]_{\leq d}$, uniquely up to sign, simply by giving a list of graphs with grouped vertices. We give the full basis for the case $d = 2$ in Table 8.2. While we computed the symmetry adapted basis for bigger d's, we only give the block sizes for d up to 5 in Table 8.3 due to the amount of data. Of course, one can also consider different hierarchies, such as one generated by graphs with few vertices if a problem is formulated in small dense graphs, or a hierarchy generated by bipartite graphs alone (for example for a problem such as Sidorenko's conjecture, see Section 6.6). The block-sizes for these two hierarchies, reduced in the same way, are given in Tables 8.4 and 8.5. The block sizes are given in the format $\text{size}^{\text{multiplicity}}$. All the hierarchies still grow very quickly despite symmetry reduction. In practice some *clever guessing* of orbits of graphs, blocks, or even elements of the symmetry

adapted basis can help for some problems to obtain better bounds than one can reach with the last fully computable level of the hierarchy.

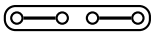
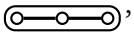
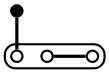
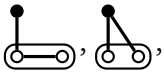
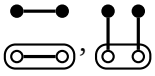
Block	Symmetry adapted basis $M^{\varnothing}$	M 	M 	M 
$(n-4,2,2)$				
$(n-4,4)$				
$(n-3,2,1)$				
$(n-3,3)$				
$(n-2,1,1)$				
$(n-2,2)$				
$(n-1,1)$				
(n)	$\varnothing$,			

Table 8.2: The symmetry adapted basis of the flag algebra of graphs up to two edges. The vertices have been arranged in order of vertex groups $V_0, \dots, V_k$ from top to bottom.

8.5 Applications: Graph Profiles

Let G and H be two (or more) graphs. The graph profile of G and H is the set of pairs of densities that can be attained simultaneously from the same sequence of graphs:

$$\text{profile}(G,H) \coloneqq \{(p(\mathcal{G},G), p(\mathcal{G},H)) : \mathcal{G} \text{ sequence of graphs}\}.$$

There is very little known about graph profiles to this date. They were first studied in [ELS79], where it was shown that the profile of s different connected graphs is a full dimensional subset of $[0,1]^s$. The only fully understood cases are the profiles of the edge and complete graph K_n, where first the case $n = 3$ was

d	block sizes	total size
0	1^1	1
1	$2^1 1^2$	4
2	$5^1 4^2 2^2 1^3$	20
3	$19^1 14^2 13^1 9^1 8^3 7^1 3^4 2^2 1^5$	121
4	$78^1 69^1 68^1 65^1 51^1 49^1 41^2 40^1 37^1 34^1 26^2 22^1 21^1 20^2 15^1 12^1 9^4$ $8^1 5^1 4^3 3^4 2^3 1^9$	823
5	$435^1 375^1 342^1 334^1 287^1 278^1 277^1 271^1 248^1 237^1 236^1 235^1$ $234^1 162^1 155^1 151^1 144^1 121^1 113^3 92^1 90^1 88^1 83^1 74^2 69^1 65^1$ $63^1 57^1 55^1 52^2 46^1 30^1 29^1 28^1 27^2 26^1 25^1 23^2 19^1 16^1 14^1 13^2 12^1$ $11^1 9^5 5^1 4^7 3^8 2^3 1^{13}$	6278

Table 8.3: The block-sizes of the symmetry reduced Lasserre hierarchy generated by graphs with up to $d = 5$ edges (i.e., graphs with up to $2d$ edges appear in the coefficients).

t	block sizes	total size
1	1^1	1
2	$2^1 1^2$	4
3	$4^3 2^1 1^2$	16
4	$22^1 16^1 14^2 11^1 8^1 7^1 6^1 5^1 3^1 2^1 1^1$	109
5	$154^1 128^1 120^1 116^1 79^1 78^1 76^1 72^1 70^1 44^1 43^1 40^1 34^1 29^1 28^1$ $23^1 19^1 13^1 5^1$	1171
6	$2116^1 1940^1 1594^1 1376^1 1350^1 1306^1 1260^1 1228^1 1076^1 1054^1$ $996^1 769^1 724^1 682^1 598^1 510^1 495^1 454^1 444^1 384^1 309^1 296^1$ $244^1 214^1 205^1 157^1 156^1 129^1 122^1 23^1$	22211

Table 8.4: The block-sizes of the symmetry reduced Lasserre hierarchy generated by graphs with up to $t = 6$ vertices (i.e., graphs with up to $2t$ vertices appear in the coefficients).

d	block sizes	total size
0	1^1	1
1	$2^1 1^2$	4
2	$5^1 4^2 2^2 1^3$	20
3	$18^1 14^1 13^1 12^1 8^4 7^1 3^4 2^2 1^5$	117
4	$72^1 65^1 60^1 58^1 49^1 46^1 40^2 36^1 34^1 30^1 26^2 21^2 19^1 17^1 15^1 12^1 9^4$ $8^1 5^1 4^3 3^4 2^3 1^9$	775
5	$380^1 314^1 313^1 306^1 260^1 252^1 242^1 228^1 227^1 226^1 219^1 215^1$ $189^1 154^1 150^1 135^1 116^1 112^1 108^1 98^1 93^1 88^2 87^1 81^1 73^1 70^1$ $68^1 62^1 59^1 56^1 55^1 52^2 35^1 30^1 29^1 28^1 27^2 26^1 24^1 23^1 21^1 19^1$ $16^1 14^1 13^2 12^1 11^1 9^5 5^1 4^7 3^8 2^3 1^{13}$	5717
t	block sizes	total size
1	1^1	1
2	$2^1 1^2$	4
3	$3^3 1^3$	12
4	$13^1 9^1 8^1 7^2 4^3 2^2 1^2$	62
5	$49^1 43^1 37^1 35^1 27^1 25^1 24^1 22^2 15^1 13^1 12^2 9^1 8^1 6^2 4^1 1^1$	370
6	$286^1 280^1 201^2 192^1 178^2 170^1 166^1 157^1 134^1 110^1 99^1 89^1 81^1$ $73^2 68^1 61^1 45^1 40^1 39^1 35^1 31^1 22^2 20^1 18^1 16^1 1^1$	3086
7	$2248^1 2017^1 1979^1 1820^1 1740^1 1684^1 1647^1 1595^1 1407^1 1260^1$ $1162^1 1049^1 1039^1 1004^1 875^1 832^1 742^1 738^1 641^1 632^1 615^1$ $575^1 573^1 495^1 475^1 422^1 353^1 329^1 324^1 312^1 302^1 283^1 281^1$ $261^1 257^1 223^1 182^1 162^1 153^1 132^1 88^1 61^1 53^1 52^1 8^1$	33082

Table 8.5: The block-sizes of the symmetry reduced Lasserre hierarchy generated by *bipartite* graphs up to $d = 5$ edges resp., $t = 7$ vertices.

shown by Razborov [Raz08] using an inductive flag SOS based proof involving *combinatoric derivatives*, followed by the case $n = 4$ in [Nik10], and the general case in [Rei16]. There is no fully understood profile of three or more graphs, but partial results for the profile of graphs with up to three vertices are in [Hua+13] and [Gle+16].

Graph profiles are inherently interesting objects, as the profile of graphs $G_1, \dots, G_m$ models all optimization problems involving linear combinations of the graphs $G_1, \dots, G_m$ and their powers, including (in-)equality constraints built with the same graphs. If one can find a description of the profile $P = \text{profile}(G_1, \dots, G_m)$, answering whether $f = \sum_{i=1}^{m} \sum_{k \in \mathbb{N}} c_{i,k} G_i^k$ is nonnegative, where $c_{i,k} \in \mathbb{R}$ are scalars, simplifies to asking whether f bounds the profile P. If we can describe (or at least bound) P using polynomials in m variables corresponding to the m graphs, this reduces the problem to a potentially much easier polynomial optimization problem in m variables, replacing the big flag SOS hierarchies. Similarly, we can add additional constraints to profiles to formulate problems from extremal graph theory with forbidden subgraphs, or more complicated constraints.

A multitude of open questions can be formulated this way. For example, Sidorenko's conjecture (6.16) would be answered if one can determine all graph profiles $\text{profile}(\vdots, H)$, where H is bijective, as the conjecture implies one of the functions bounding these profiles. Another longstanding open problem is related to crossing numbers: the rectilinear crossing number of a graph is the minimum number of edges that cross in any drawing of the graph. The minimum number of crossings is still open for complete and complete bipartite graphs (the bipartite setting is also known as Turan's brick factory problem, see [KST54]). If one works in the flag algebra of limits of point order types [Goa+18], one can formulate the limit of the (appropriately scaled) crossing number of the complete graph as a problem of order types with up to four vertices. If one additionally adds two colors to the theory, one can formulate the limit of the crossing number of the complete bipartite graph as well. Both of these questions can be seen as asking to optimize within a profile of small order types in a given direction, which corresponds to the number of crossings in the limit.

In [Ble+20b] the authors show that graph profiles are limits of sequences of feasible sets of *polynomial matrix inequalities*, which they call *SOS-profiles*, but they also show that the simple looking profile of the edge e and the path P_3 with three edges cannot be proven using a finite level of the flag SOS hierarchy. More specifically, the Blakeley-Roy inequality $P_3 - e^3 \geq 0$, which gives one of the sides

of the profile, cannot be proven using a finite flag sums-of-squares certificate.

Graph profiles are not necessarily convex or semi-algebraic, as can be seen from the easiest fully known profile, the edge-triangle graph profile, as shown in Figure 8.1. It is clearly non-convex, and one needs infinitely many polynomials to define its lower bound.

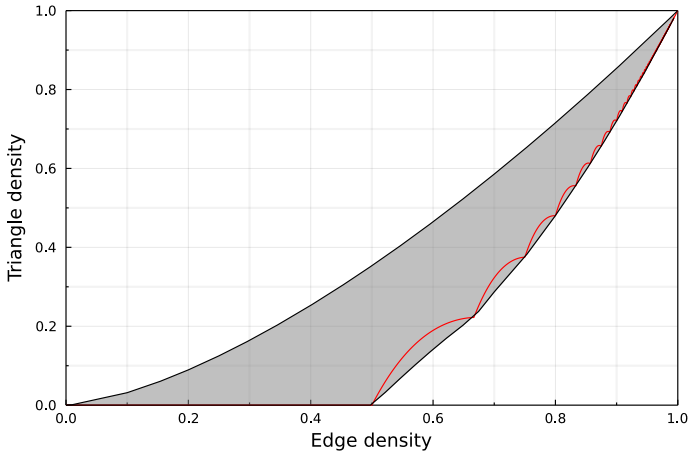


Figure 8.1: The edge-triangle graph-profile.
The graph profile from [Raz08] shown in gray. The red line is an exaggeration of its lower bound to make its features more distinct.

Bounding Graph Profiles. We are interested in *outer approximations* of graph profiles of given graphs G and H. While we can clearly say that a point (x, y) is not in the G-H-profile, if we can prove that -1 is a sum-of-squares in the variety determined by $G = x$ and $H = y$, this only allows us to exclude points from profile(G, H). Instead, we are going to try to find univariate polynomials $f_{\text{lower}} : \mathbb{R} \to \mathbb{R}$ and $f_{\text{upper}} : \mathbb{R} \to \mathbb{R}$ which *bound the profile as close as possible from below and above*. To this end, we approximate the problems

$$\max \int_0^1 f_{\text{lower}}(x)\,dx \qquad \min \int_0^1 f_{\text{upper}}(x)\,dx$$
$$\text{s.t. } H - f_{\text{lower}}(G) \geq 0 \qquad \text{s.t. } f_{\text{upper}}(G) - H \geq 0$$

using flag sums-of-squares. Given two feasible solutions f_{lower} and f_{upper}, we clearly have

$$\text{profile}(G,H) \subseteq \{(x,y) : f_{\text{lower}}(x) \leq y \leq f_{\text{upper}}(x), x \in [0,1]\}.$$

For example, we obtain the outer approximation in Figure 8.2 from the third level of the Lasserre hierarchy. While the upper bound is a good approximation of the exact bound $\sqrt{\vcenter{\hbox{\includegraphics[height=0.8em]{}}}^3}$, the lower bound is quite far from sharp.

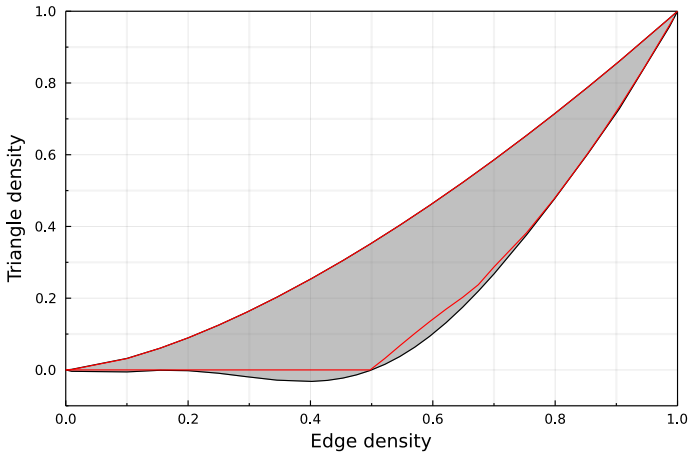


Figure 8.2: Outer approximation of the edge-triangle profile.
The outer approximation obtained from the third level of the Lasserre hierarchy, i.e., the univariate lower and upper bounds have degree 6. The exact profile from [Raz08] is shown in red.

Piecewise Approximations. We can obtain a significantly better approximation by cutting the interval $[0,1]$ into small steps $[0,1] = \bigcup_i [a_i, b_i]$, and solving sums-of-squares approximations of the problems

$$\begin{aligned} &\max \int_{a_i}^{b_i} f_{\text{lower},i}(x)\,dx && \min \int_{a_i}^{b_i} f_{\text{upper},i}(x)\,dx \\ &\text{s.t. } H - f_{\text{lower},i}(G) \geq 0 && \text{s.t. } f_{\text{upper},i}(G) - H \geq 0 \\ &\quad a_i \leq G \leq b_i && \quad a_i \leq G \leq b_i \end{aligned}$$

for each i. Note that due to the inequalities, "nonnegative" now means that $H - f_{\text{lower},i}(G)$ respectively $f_{\text{upper},i}(G) - H$ lie in the quadratic module (1.8) given by

the two inequalities, and are thus of the form

$$s_0 + (G - a_i)s_1 + (b_i - G)s_2,$$

where s_0, s_1 and s_2 are flag sums-of-squares.

By cutting the interval into just 20 equal parts we already obtain a much better approximation of the edge-triangle-profile, as shown in Figure 8.3. In total, computations took about 5 seconds. While the bound is not sharp (and never will be everywhere), this level of the hierarchy correctly captures some non-convexity in the interval $[\frac{1}{2}, \frac{2}{3}]$.

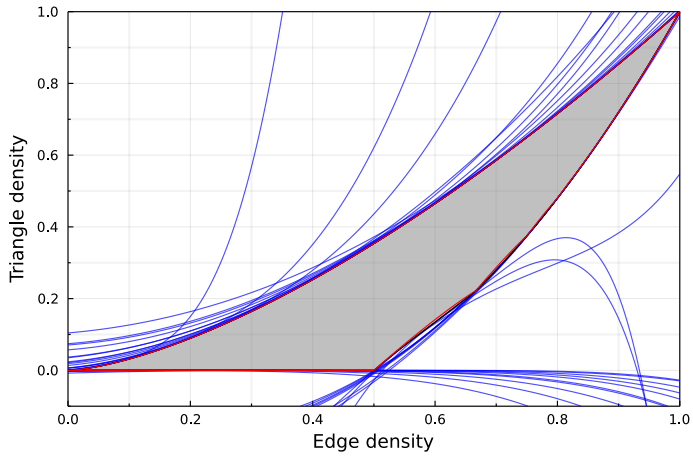


Figure 8.3: Piecewise outer approximation of the edge-triangle profile.
The outer approximation of the edge-triangle profile obtained from the third level of the Lasserre hierarchy, where the interval $[0, 1]$ has been cut into 20 equal parts. The exact profile from [Raz08] is shown in red, the functions computed approximating the profile in blue.

This way we can quite rapidly obtain (rough) approximations for several previously unknown graph-profiles. Approximations of all graph profiles between graphs with at most 4 vertices are shown in Figure 8.4. It took about 11 minutes to calculate all of them with Mosek on an i7-1185G7. Better approximations coming from higher levels of the hierarchies, or more precise ones, can easily be calculated with more time with the developed Julia software package described in Section 8.7.

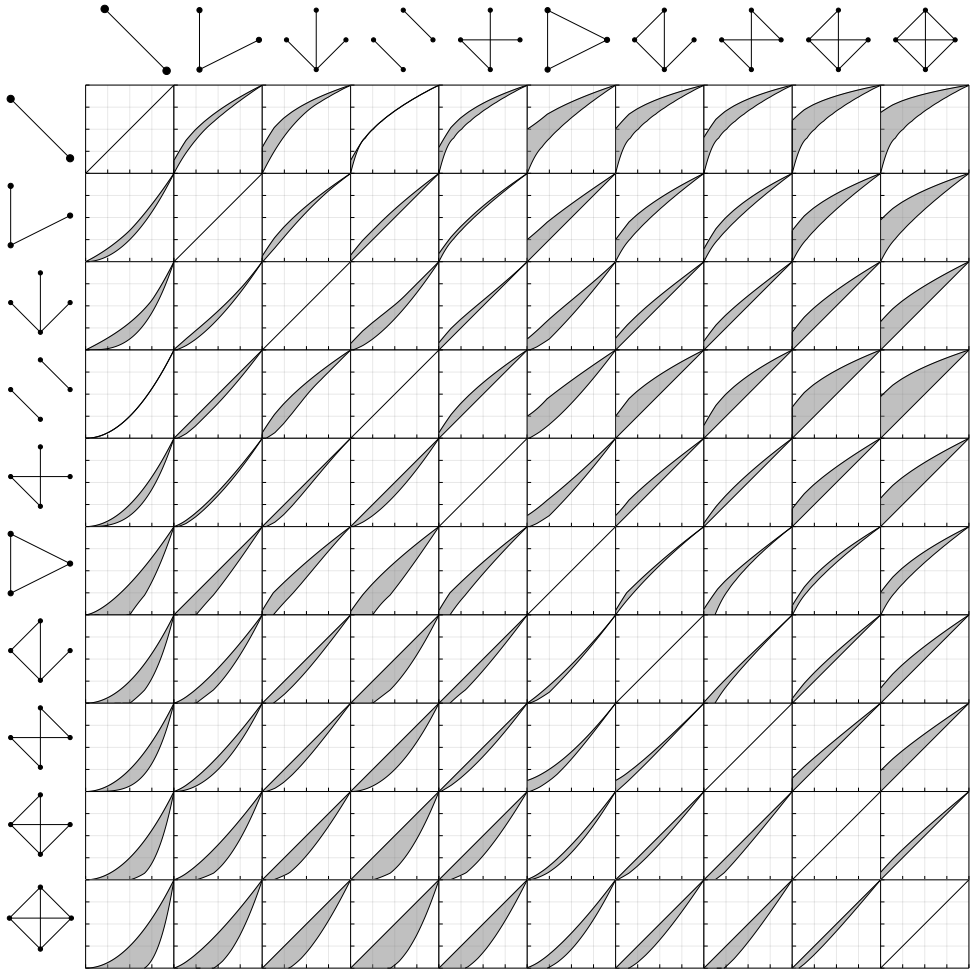


Figure 8.4: Outer approximations of graph profiles of pairs of graphs with at most 4 vertices.

The bounds were obtained from the Lasserre hierarchy generated by the orbits of graphs with up to 4 vertices (and thus containing products with up to 8 vertices). The interval $[0,1]$ has been cut into 10 equal parts for each of the squares, which each denote the area $[0,1]^2$.

Profiles of Harmonic Flags. In Section 6.6 we introduced *harmonic flags*, a different basis of the flag algebra, which comes from the characters of the binary hypercube. We can also compute their profiles, which are always contained in the squares $[-1,1]^2$, by either writing them as linear combinations of flags by equation (6.9), or by generating the SDP starting out in the harmonic basis. The symmetry adapted basis is found analogously, as we only need to adjust the gluing operation to delete edges glued on top of each other, as seen in Section 6.6.

Proposition (6.9) tells us that these *profiles of harmonic flags* are *slices* of the higher dimensional profiles involving all graphs appearing as subgraphs in the two harmonic flags. As for regular flags, we give outer approximations of all pairs of harmonic graphs with up to four vertices in Figure 8.5, as well as a bigger view at the profile of the harmonic graphs corresponding to the edge and the complete graph on four vertices in Figure 8.6. While we did not manage to prove that any of these approximations are sharp, these slices were fully unknown before, and, already for these very small cases, seem to exhibit much more varied behaviors than the profiles of pairs of flags in Figure 8.4. But the approximations in Figure 8.5 also visualize a weakness of this approach: Since we only approximate the profile with a function from above and below, we potentially miss big details. Of course the G-H-profile is exactly the same as the H-G-profile after swapping the coordinates. But we obtain, for example, very different approximations of the two profiles for the harmonic flags corresponding to  and  .

Remark 8.10. Using the harmonic basis as initial basis for the symmetry reduction seems to have some computational advantages. Despite solving the same amount of very similar SDPs, where the SDPs have the exact same block-sizes and structure, computing the approximations in Figure 8.4 took about 11 minutes, while computing Figure 8.5 took only 8.

One could now think that the reason for this could be the different gluing operation for harmonic flags: As seen in Section 6.6, the product between harmonic flags deletes edges that are glued on top of each other. While this does reduce the number of total non-zero coefficients in the data matrices of the SDP in some cases, as seen in Table 8.6, it actually increases the number of nonzero coefficients in the case used for the Figures ($t = 4$).

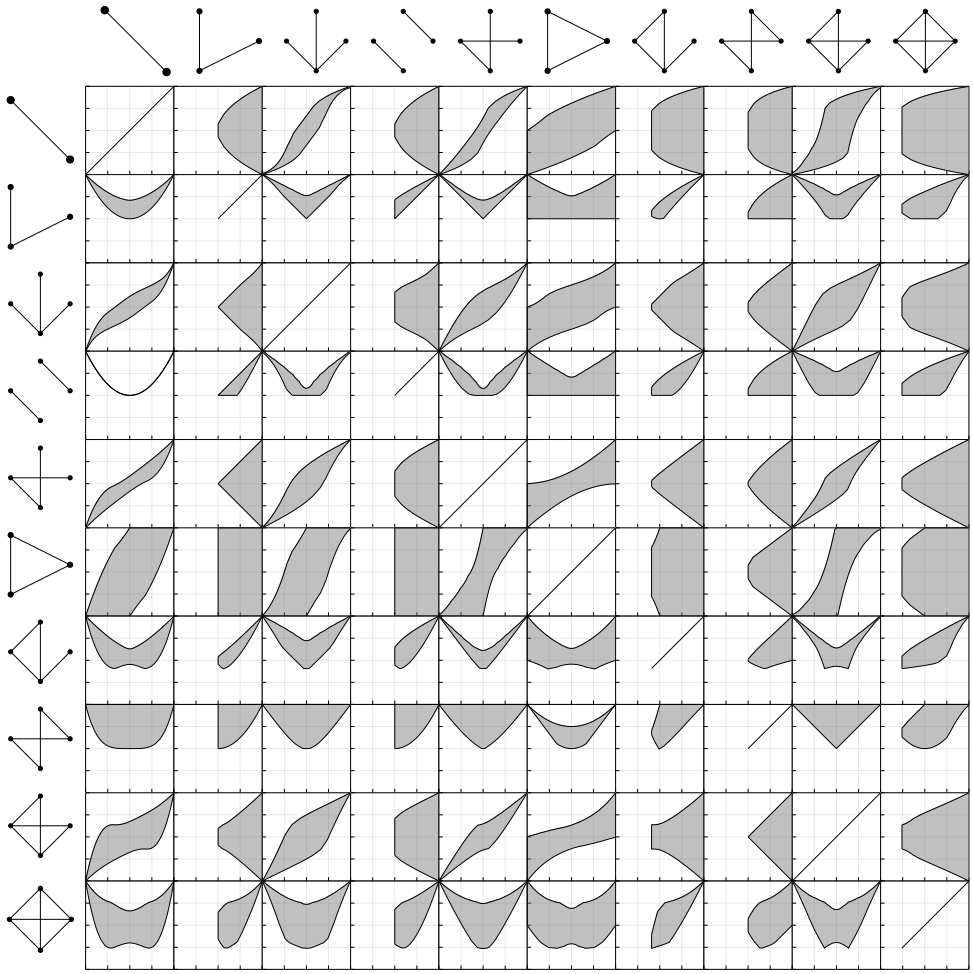


Figure 8.5: Outer approximations of graph profiles of pairs of harmonic graphs with at most 4 vertices.
The bounds were obtained from the Lasserre hierarchy generated by the orbits of *harmonic* graphs with up to 4 vertices. The interval $[-1,1]$ has been cut into 10 equal parts for each of the squares, which each denote the area $[-1,1]^2$.

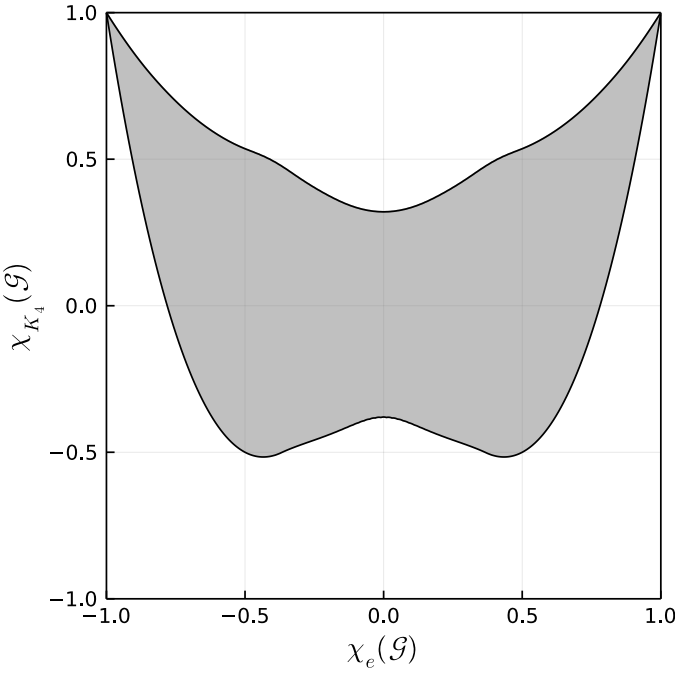


Figure 8.6: Outer approximation of the profile of the character flags corresponding to the edge and K_4.

A bigger view at one of the profiles shown in Figure 8.5. The same level of the hierarchy was used, but the interval was cut into 100 equal parts. Both the upper and lower bound were approximated piecewise by degree four polynomials. Using this profile, we can prove any bounds coming from functions bounding the gray area from the outside, where a sufficiently big gap should be left due to the numerical nature of the bounds.

	Number of nonzero coefficients	
d	flags	Harmonic flags
1	8	8
2	149	146
3	7062	6800
4	748123	713390
t	flags	Harmonic flags
1	1	1
2	8	8
3	100	97
4	5285	5339
5	1600307	1750120

Table 8.6: Total number of non-zero coefficients in the symmetry reduced Lasserre hierarchy for two different bases, generated by orbits of graphs with at most d edges respectively at most t vertices.

The better computational performance should thus be related to the orthogonality of the harmonic basis. Intuitively, comparing the two Figures 8.4 and 8.5 tells us that the feasible set of the dual (moment) side of the SDP is "thicker", i.e., we can fit a bigger ball in it, as the harmonic profiles generally *seem* wider than the graph profiles. Formalizing this remains a task for future work.

Since swapping to harmonic flags is just another change of basis for any of the here considered hierarchies, it remains to be seen whether reformulating problems in the harmonic basis can lead to computational advantages for general problems.

8.6 Flag Algebras for Degenerate Extremal Combinatorics

While flag algebras have proven a powerful tool in *dense* extremal combinatorics, they inherently fail if the considered extremal models are *sparse*, then called *degenerate*, in the sense that all sub-flag densities tend towards zero. In this section we give a small example, to be considered a "proof of concept", that shows that one can extend flag algebras to these degenerate cases, at the cost of significantly

more complicated product rules.

The easiest case to consider here is the case of graphs without 4-cycles, we can show using flag SOS that

$$\mathrm{ex}(\,\cdot\,;\square) = 0.$$

In cases such as these, we are instead interested in the rates at which the considered densities approach zero, especially the *leading term*. Here it is known ([KST54]) that, if the considered graph sequence is C_4-free, we have

$$\cdot\,\cdot \leq \frac{n+\sqrt{4n^3-3n^2}}{4\binom{n}{2}} = \frac{1}{\sqrt{n}} + o\left(\frac{1}{\sqrt{n}}\right) \to 0.$$

In [Ray+18] the authors translated the proof of [KST54] into the setting of sums-of-squares, building an SOS-certificate for every (big enough) n.

We are interested in building *a single SOS based certificate*, which proves the leading term $\frac{1}{\sqrt{n}}$ exactly. While this is a weaker result, it is of interest to have a better method for this setting, as degenerate extremal combinatorics is a lot less well-understood than dense extremal combinatorics. See, for example [FS13] for an extensive survey on the topic. Excluding any bipartite graph always results in a degenerate limit, and already for "easy" cases such as C_6-free graphs the leading coefficient, and for C_8-free graphs the leading exponent are not known.

A Sequence of Small SDPs. Let us consider sums-of-squares, where the squared polynomials can be written in just 6 elements of the symmetry adapted basis:

$$\mu_1 = (n): \quad B_1 = \{b_{11}, b_{12}, b_{13}\} = \{\varnothing, \cdot\,\cdot, \wedge\},$$
$$\mu_2 = (n-1,1): \quad B_2 = \{b_{21}, b_{22}\} = \left\{\cdot\,\cdot,\ \cdot\!-\!\cdot\right\},$$
$$\mu_3 = (n-2,2): \quad B_3 = \{b_{31}\} = \left\{\triangle\right\},$$

i.e., we here consider an SDP with three blocks of sizes 3, 2 and 1.

Normally, we would now compute the limits of the symmetrized products (8.5) to obtain the flag SOS hierarchy for the case that $n \to \infty$. Instead, let us calculate all products dependent on n (where n is big enough), *without rescaling by dividing trough n to the power of unlabeled vertices*. Normally we would avoid calculating these in-between steps by immediately sending unlabeled vertices to distinct vertices.

For example, to compute the unscaled non-limit product P^2_{11} coming from $b_{21}b_{21}$, we calculate

$$
\begin{aligned}
b_{21}b_{21} &= \lim_{n\to\infty} \frac{1}{n^2} \left[\!\!\left[\left(\sum_{i\neq 2} \overset{\textcircled{2}}{\underset{\textcircled{i}}{\bullet}} - \sum_{j\neq 1} \overset{\textcircled{1}}{\underset{\textcircled{j}}{\bullet}} \right)^2 \right]\!\!\right] \\
&= \lim_{n\to\infty} \frac{1}{n^2} \left(2 \left[\!\!\left[\left(\sum_{i\neq 2} \overset{\textcircled{2}}{\underset{\textcircled{i}}{\bullet}} \right)^2 \right]\!\!\right] - 2 \left[\!\!\left[\left(\sum_{i\neq 2} \overset{\textcircled{2}}{\underset{\textcircled{i}}{\bullet}} \sum_{j\neq 1} \overset{\textcircled{1}}{\underset{\textcircled{j}}{\bullet}} \right)^2 \right]\!\!\right] \right) \\
&= \lim_{n\to\infty} \frac{1}{n^2} \big(2(n-1) \, \overset{\bullet}{\bullet} + 2(n-1)(n-2) \, \wedge \\
&\qquad\qquad - 6(n-2) \, \wedge - 2 \, \overset{\bullet}{\bullet} - 2(n-2)(n-3) \, \square \big) \\
&= \lim_{n\to\infty} \frac{1}{n^2} \big((2n-4) \, \overset{\bullet}{\bullet} + (2n^2 - 12n + 16) \, \wedge + (-2n^2 + 10n - 12) \, \square \big) \\
&= \lim_{n\to\infty} \frac{1}{n^2} P^2_{11}(n).
\end{aligned}
$$

We can compute the other entries similarly to obtain the three matrices $P^1(n) = (P^1_{ij}(n))_{i,j=1,\dots,3}$, $P^2(n) = (P^2_{ij}(n))_{i,j=1,\dots,2}$ and $P^3(n) = (P^3_{ij}(n))_{i,j=1,\dots,1}$ containing linear combinations of flags with coefficients that lie in $\mathbb{R}[n]$.

If we now rescale the rows and columns of each block with the appropriate functions and take the limit, we obtain the normal limit hierarchy:

$$
\begin{aligned}
P^1_{\lim} &:= \lim_{n\to\infty} \left(1, n^{-2}, n^{-3}\right) P^1(n) \begin{pmatrix} 1 \\ n^{-2} \\ n^{-3} \end{pmatrix} = \begin{pmatrix} 1 & \overset{\bullet}{\bullet} & \wedge \\ \overset{\bullet}{\bullet} & \overset{\bullet}{\bullet}\!\!\overset{\bullet}{\bullet} & \wedge\!\!\bullet \\ \wedge & \wedge\!\!\bullet & \square \end{pmatrix} \\
P^2_{\lim} &:= \lim_{n\to\infty} \left(n^{-1}, n^{-2}\right) P^2(n) \begin{pmatrix} n^{-1} \\ n^{-2} \end{pmatrix} = \begin{pmatrix} 2\wedge - 2\square & 2\square - 2\bullet\!\!\wedge \\ 2\square - 2\bullet\!\!\wedge & 2\bullet\!\!\wedge - 2\bullet\!\!\square \end{pmatrix} \\
P^3_{\lim} &:= \lim_{n\to\infty} \left(n^{-1}\right) P^3(n) \left(n^{-1}\right) = \left(4\square - 8\bullet\!\!\wedge + 4\bullet\!\!\square\right).
\end{aligned}
$$

With this limit we can indeed prove that $\mathrm{ex}(\overset{\bullet}{\bullet}; \square) = 0$. This is easier to see in the dual: There we turn the graphs $\overset{\bullet}{\bullet}$, $\wedge$, $\square$, $\bullet\!\!\wedge$, $\bullet\!\!\square$, $\square$, $\bullet\!\!\square$ and $\square$ into real variables (moments) in $[0,1]$, and assume $P^i_{\lim} \succcurlyeq 0$ for $i = 1,2,3$. It is straightforward to see that $\square = 0$ and $P^i_{\lim} \succcurlyeq 0$ together imply that $\overset{\bullet}{\bullet} = 0$, proving $\mathrm{ex}(\overset{\bullet}{\bullet}; \square) = 0$.

The approach taken in [Ray+18], which is based on the proof in [KST54], now instead constructs an SOS-based proof for each n, i.e., in our notation we would equivalently give matrices $X_i(n)$ with coefficients that depend on n, which

are positive semidefinite for each n, such that

$$\langle P^1(n), X_1(n)\rangle + \langle P^2(n), X_2(n)\rangle + \langle P^3(n), X_3(n)\rangle = \; \vdots \; - \frac{n+\sqrt{4n^3-3n^2}}{4\binom{n}{2}} \geq 0.$$

Considering a Different Limit. We are interested in the leading term alone; we want to show that $\; \vdots \; \leq \frac{1}{\sqrt{n}} + o\left(\frac{1}{\sqrt{n}}\right)$. Multiplying the inequality with $\sqrt{n}$, we equivalently want to prove that

$$\lim_{n\to\infty} 1 - \sqrt{n} \; \vdots \; \geq 0.$$

This idea leads to the definition of *degenerate flags*. For a flag G and function $f(n)$, we define

$$G_{f(n)} \coloneqq \lim_{n\to\infty} f(n) p(\mathcal{G}, G),$$

which captures the *density of G at rate f*. We assume, without loss of generality, that $f(n)$ is positive for n big enough, since $G_{-f(n)} = -G_{f(n)}$. While degenerate flags not necessarily converge, we can still find product rules for them.

For example, consider the square of the density of $\; \overset{\textcircled{1}}{\vdots} \;$ at rate $\sqrt{n}$:

$$\begin{aligned}
\left[\!\left[\left(\overset{\textcircled{1}}{\vdots}_{\sqrt{n}} \right)^2 \right]\!\right] &= \lim_{n\to\infty} \sqrt{n}^2 \frac{1}{n^2} \left[\!\left[\left(\sum_{i\neq 2} \overset{\textcircled{2}}{\underset{\textcircled{i}}{\vdots}} \right)^2 \right]\!\right] \\
&= \lim_{n\to\infty} \frac{1}{n} \left((n-1) \; \vdots \; + (n-1)(n-2) \; \wedge \right) \\
&= \; \vdots \; + \; \wedge_{n-3} \\
&= \; \vdots \; + \; \wedge_{n} - 3 \; \wedge .
\end{aligned}$$

While these gluing rules are generally much more complicated than the gluing rules for regular flags, they are easier than using the full expressions for finite n, as we only catch some higher order terms. We can again consider sums-of-squares in this extended algebra. Here we need to make use of a key observation:

$$G_f - G_g \geq 0 \quad \text{if there is an } N > 0 \text{ such that } f(n) - g(n) \geq 0 \text{ for all } n > N. \tag{8.6}$$

This allows us to construct SOS certificates that *only hold for n big enough*. For example, we can say that

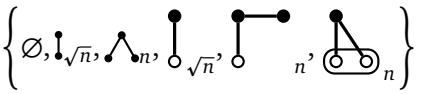
$$G_n - 5G_{\sqrt{n}} \geq 0,$$

despite it not being nonnegative in the finite setting. Adding all of these inequalities to the extended algebra (resp. as quadratic modules to the SDP) is not possible computationally, as there are infinitely many of them. For example

$$G_n - \alpha G \geq 0$$

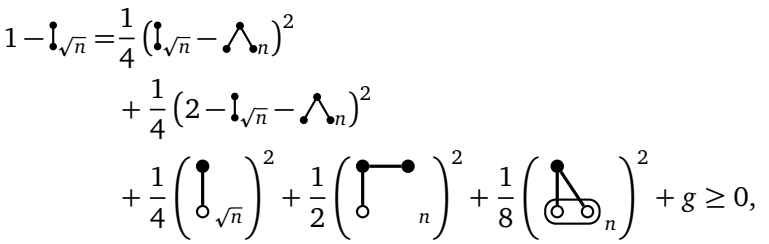
for every choice of α, requiring an inequality for every α. Still, for a given certificate, we can check whether it holds in the limit. What we can do practically is to add inequalities of the form $G_f - \alpha G_g \geq 0$ to the program for a very big constant α (since G_g is assumed to be nonnegative in the limit, this implies all inequalities with a smaller factor than α), for every pair of functions f, g appearing as rates of the same subgraph density with $g \in o(f)$.

To formulate a degenerate SOS proof, we thus want to rescale the rows and columns of the $P^i(n)$ with functions that depend on n, to not change the SDPs for finite n. In this case, where we want to maximize the edge density in a 4-cycle free graph, we choose the degenerate symmetry adapted basis



to generate our SDP.

We can now give a single SOS based proof for the first order of the edge density in triangle free graphs:



where we assumed that $\square_{f(n)} = 0$ for all functions f, and g is a positive linear combination of terms of the form (8.6).

While we can compute the products of degenerate flags with the algorithm from Section 8.3.3 by implementing it in a way to allow for univariate polynomial exponents of the polynomials, one still needs to guess all the rates of flags and inequalities (8.6) necessary to actually build the SDP. It remains to be seen if this can be done automatically, potentially by using an iterative approach, or by including the same columns at multiple different rates at the same time.

8.7 Software: FlagSOS.jl

Every hierarchy and reduction described and result given in this Chapter as well as Chapter 9 were computed using a custom Julia software package called "FlagSOS.jl", (soon to be) available at `https://github.com/DanielBrosch/FlagSOS.jl`. All algorithms were implemented in a way that allows for them to be generalized easily to any kind of flag, as well as translate between different bases such as the non-induced, induced and harmonic basis. While there is an existing software package called "Flagmatic" [FV13] for Razborov's vertex truncated hierarchies described in Chapter 9, it does not implement the Lasserre hierarchy for flags, and only supports (directed and undirected) graphs and 3-uniform hypergraphs.

Implementing Different Kinds of Flags. Depending on which features one requires, one needs to implement different black-box functions for the theory of models the flag algebra is based on. For the main reduction algorithms one requires a function that glues two flags together, as well as functions canonically labeling a flag, and determining its group of automorphisms. If one cannot provide a better alternative (such as NAUTY/TRACES [MP14]), the package contains a basic generalized version of the NAUTY/TRACES algorithm to canonically label flags and determine their automorphisms. While the speed of the implementation is not competitive with [MP14], it can easily be used for different kinds of flags, given a function that *distinguishes* a vertex v in some way from others relatively to a group M of other vertices. For example, for graphs (and hypergraphs) it would be enough to return the number of edges between v and the vertex group M. If one also wants to generate flags up to isomorphism or swap between the different bases, one needs to be able to provide (some abstract representation of) predicates in a flag, as well as potentially check the axioms of the theory.

9

An Alternative Hierarchy Intertwining the Lasserre Hierarchy for Binary Problems with S_n Symmetry

The Lasserre hierarchy grows very quickly as one increases the level d of the hierarchy, which contains polynomials up to degree $2d$. What if we want to compute bounds for problems of high degree? We saw earlier in Section 6.4 that induced subgraph densities correspond to (limits of) polynomials of degree $\binom{k}{2}$, where k is the number of vertices of the graph. To even be able to formulate problems formulated in induced subgraph densities, one would need a high level of the Lasserre hierarchy. In this chapter we investigate a different truncation of the hierarchy, introduced by Razborov [Raz07]: we truncate the hierarchy by the number of vertices of flags. While Razborov formulated the hierarchy directly in the setting of flags, we show how to obtain this hierarchy from the Lasserre hierarchy, give some ideas as to how it can be generalized to other problems, and how it can be reduced further. We obtain a very strong reduction in block-sizes without needing to rely on Artin-Wedderburn theory, which is often used for symmetry reductions, and it may be very fruitful if one attempts similar reductions for other highly symmetric combinatoric problems in the future. This chapter should be seen as a

"recipe" for similar hierarchies for highly symmetric problems in binary variables, such as constant weight error correcting codes, or potentially packing problems (under suitable modifications).

Main Contributions. We introduce a symmetry-reduction technique based on Möbius transformations for binary polynomial optimization problems, and construct a hierarchy, based on a different truncation of the Lasserre hierarchy, for which the reduction is very well suited. We show that the constructed hierarchy, after reduction, generalizes Razborov's hierarchy for flag algebras [Raz07]. This allows us to compare the strength of the (generalized) Razborov and Lasserre hierarchies in Theorem 9.16. We then show how to continue from Razborov's definition, exploiting further symmetries contained in the hierarchy.

Outline of this Chapter. As we will see in this chapter, the generalized Razborov hierarchy, while more complicated to define, is very well suited for a symmetry reduction based on Möbius transformations, without having to rely on complicated representation theoretic tools.

The generalization of Razborov's hierarchy will be defined through a series of relaxations of a high level of the Lasserre hierarchy: We start by truncating the hierarchy by the number of vertices in Section 9.1, where we enforce a strict sparsity pattern. This sparsity pattern will forbid monomials with more than T vertices from appearing, even when cancelled out.

Next, in Section 9.2, we will decompose the semidefinite variable into the maximal cliques of the sparsity graph, and determine the "biggest" maximal cliques. At this step we obtain a trivial symmetry reduction by only considering maximal cliques up to symmetry, eliminating most of them in the process. We then further relax the hierarchy to ignore all but the "biggest" maximal cliques, which have a lot of additional structures we can exploit.

The blocks corresponding to this special family of maximal cliques are each symmetric under the action of $S_k \times S_{n-k} \subseteq S_n$, for a $k \ll n$. Crucially, each of these cliques will be closed under multiplication with variables invariant under S_{n-k}, which makes it possible to apply a Möbius transformation to each of them, diagonalizing each orbit under S_k, resulting in a significantly stronger reduction than symmetry reduction coming from S_k.

The final relaxation, described in Section 9.3, is equivalent to the hierarchies defined by Razborov, and results, in his language, in sums-of-squares in each flag algebra $\mathcal{A}^\sigma$ of each type σ. It is here equivalently obtained by only considering

the trivial isotypic component of the action of S_{n-k}, i.e., we restrict ourselves to the subspace invariant under S_{n-k} in each block. Despite these quite significant relaxations it is still possible to compare the strength of the resulting hierarchy to the Lasserre hierarchy in both directions, which we will show using some consequences coming from the representation theory of S_n.

Finally, we can reduce the hierarchy further, by exploiting the remaining symmetries in the leftover blocks.

While most of the constructions in this chapter can be generalized for groups other than S_n, the case of S_n is easier to handle, and allows us to compare the hierarchies we end up with. We will here focus on the case of the flag algebras of graphs.

9.1 A Vertex Truncation of the Lasserre Hierarchy

Let $[X]$ be the monomial basis of the ring of polynomials $\mathbb{R}[x]$ (in binary variables), where S_n acts on $[X]$ via permutations. In the case of graphs, for example, the variables x_{ij} correspond to edges, as explained in detail in Section 6.5. Sums-of-squares in $\mathbb{R}[x]$ can be written as $[X]^T M[X]$, where M is a positive semidefinite matrix. The Lasserre hierarchy is now obtained by truncating the basis vectors to only contain monomials up to a degree $d \in \mathbb{N}$, that is we optimize over nonnegative polynomials of the form $[X]_{\leq d} M [X]_{\leq d}$.

We now want to truncate the hierarchy instead by the number of "vertices" in the appearing monomials. To do this, we first generalize the number of vertices of a flag to more arbitrary operations of S_n.

Definition 9.1. Let m be a monomial. The *vertex set* $V(m)$ of m is the smallest set $A \subseteq [n]$ such that the stabilizer of m

$$(S_n)_m := \{\sigma \in S_n : \sigma(m) = m\}$$

contains all of $S_{[n]\setminus A}$, i.e.

$$V(m) := \operatorname*{arg\,min}_{A \subseteq [n]} |A| \quad \text{s.t. } (S_n)_m \supseteq S_{[n]\setminus A}.$$

Example 9.2. Consider the setting of graphs, for $n \gg 3$ and $m = x_{13}x_{23}$. Then

$$(S_n)_{x_{12}x_{23}} = S_{\{2,3\}} \times S_{[n]\setminus\{1,2,3\}} \supseteq S_{[n]\setminus\{1,2,3\}},$$

and $V(m) = \{1,2,3\}$, since n is assumed to be big. If n was small, say $n = 4$, then we would instead have $V(m) = \{1,4\}$ since $(S_n)_{x_{12}x_{23}} \supseteq S_{[n]\setminus\{1,4\}} = S_{\{2,3\}}$. But if n

is big enough, the automorphism group of the considered graphs is small enough for this definition to exactly correspond to the number of vertices of the graph.

If a graph contains less than half of all vertices in $[n]$ (or does not have a very big automorphism group), then this definition generalizes the vertex set of a graph, when interpreting them as monomials as explained in Section 6.5. In the following we assume $T \ll n$, the upper limit on the vertices, is small enough to not have to worry about these edge cases.

Similar to the degree of monomials, this function behaves nicely when multiplying two monomials m_1, m_2, which both have small enough vertex counts. We have

$$V(m_1 m_2) = V(m_1) \cup V(m_2), \tag{9.1}$$

since $(S_n)_{pq} \supseteq (S_n)_p \cap (S_n)_q$, and for small vertex counts the automorphism group of pq will not be bigger than $S_{n-|V(m_1)\cup V(m_2)|}$.

We can now define a vertex truncated hierarchy Vert_T for each natural number T. As happens "for free" in the degree constrained hierarchy, we do not want any terms cancelling out that have more than T vertices here either. But first, analogously to the degree constrained hierarchy, we simply truncate the basis vector to only contain basis elements with up to T vertices:

$$\mathcal{B}_T := \{m \in [X] \colon |V(x)| \leq T\}.$$

Let $f = \sum_{m\in\mathcal{B}_T} c_m m$ be a polynomial. We define *the vertex count of its square* as

$$V^2(f) := \max_{m,m'\in\mathcal{B}_T \colon c_m, c_{m'}\neq 0} |V(mm')|,$$

which is not necessarily the same as the maximum vertex count of monomials appearing in $[\![f^2]\!]$, as this counts the vertices of monomials that get cancelled out as well. For example, the vertex count of the square of the quantum graph

$$f = \begin{smallmatrix}1\\2\end{smallmatrix} + \begin{smallmatrix}2\\3\end{smallmatrix} - \frac{1}{2}\begin{smallmatrix}1\\3\end{smallmatrix}$$

is $V^2(f) = 3$, since f^2 contains the product $\begin{smallmatrix}1\\2\end{smallmatrix} \cdot \begin{smallmatrix}2\\3\end{smallmatrix} = \begin{smallmatrix}&1&\\2&&3\end{smallmatrix}$, but $[\![f^2]\!] = \frac{9}{4}\,\overset{\bullet}{\underset{\bullet}{\bullet}}$ only contains graphs with up to two vertices.

Definition 9.3. The Tth level Vert_T of the *vertex truncated SOS-hierarchy* is given by optimizing over sums-of-squares of polynomials p_i with $V^2(p_i) \leq T$. I.e., we

optimize over $\mathcal{B}_T^T M \mathcal{B}_T$, where the positive semidefinite matrix M can be decomposed into rank-one matrices

$$M = M_1 + \cdots + M_k$$

for some k, such that each of the matrices M_i, for $i = 1, \ldots, k$, is positive semidefinite and follows the same sparsity pattern:

$$(M_i)_{m,m'} = 0 \quad \text{if} \quad |V(mm')| > T$$

for all $m, m' \in \mathcal{B}_T$.

To give explicit block-sizes, we will consider the specific case $T = 4$, i.e., we want at most 4 vertices in all appearing graphs. We assume n is big enough for the symmetry reduction to stabilize as seen in Proposition 8.5.

Remark 9.4. One could consider different definitions of $V(m)$ for different problems, but this seemed the most natural at the time. If one attempts different definitions of $V(m)$, one needs to make sure that (9.1) holds, and that there is a finite number of orbits of monomials with $|V(m)| \leq T$ for constants T. Another "natural" definition for problems on graphs (such as stable set), which does not rely on symmetries, would be simply the set of vertices appearing in the variables of the monomial. But one could also push things further, prioritizing "local" connected structures in graphs, by adding all vertices appearing on shortest paths between vertices in $V(m)$ to $V(m)$. This would then truncate the problem to prioritize small connected subgraphs. Note however, that without additional symmetries the resulting hierarchies will quickly outgrow the computational capabilities of today's solvers.

Naive Approach: Symmetry Reduction. Since we are interested in solving symmetric sums-of-squares problems, we can exploit the symmetry of the truncated Lasserre hierarchy as done in Chapter 8. Ignoring the sparsity pattern, the case $T = 4$ corresponds to the direct sum of graph modules of graphs with up to 4 vertices. We symmetry reduced this hierarchy in Table 8.4, where we obtain blocks of sizes $22, 16, 14, 14, 11, 8, 7, 6, 5, 3, 2$ and 1, totaling a block size of 109, and 775 variables/entries in the SDP variables (only counting upper-triangular entries).

The problem is now the sparsity pattern: The block-diagonalization and symmetrization changes the rank of the matrices, and it is unclear how to formulate the decomposition condition after symmetry reduction. But we can still obtain a

stronger hierarchy by setting the entries of the positive semidefinite matrix M to zero according to the sparsity pattern. Since $V(\sigma(m)) = \sigma(V(m))$ for monomials m and permutations σ, the sparsity pattern itself is symmetric, and can thus be translated into constraints for the block-diagonalized SDP.

But we can do much better, by realizing that we can assume that the rank-one matrices M_i correspond to the maximal cliques of the sparsity pattern.

9.2 Decomposition into Maximal Cliques

We can define the sparsity graph G with vertices $\mathcal{B}_T$, where m, m' are connected if the entry $M_{m,m'}$ does not have to be zero, i.e., whenever $|V(mm')| \leq T$.

While we generally cannot decompose a matrix M with the given sparsity pattern into a sum of matrices corresponding to the maximal cliques in the graph G, we can clearly do so for the positive semidefinite rank 1 matrices M_i. If a rank one matrix $M_i = vv^T$ has a zero in position (m, m'), then $v_m v_{m'} = 0$, implying that the nonzero entries of v are contained in a clique of G. So, our matrices M in Definition 9.3 can equivalently be written as

$$M = \sum_{C \text{ maximal clique of } G} M_C,$$

where $M_C \succcurlyeq 0$ is a positive semidefinite matrix which only has non-zero coefficients in rows and columns corresponding to the clique C.

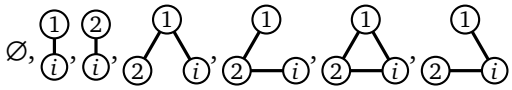
Proposition 9.5. *For each subset $S \subseteq [n]$ of cardinality up to T with $T \equiv |S| \mod 2$, the subset*

$$C_S := \left\{ m \in \mathcal{B}_T : |V(m) \setminus S| \leq \frac{T - |S|}{2} \right\} \subseteq \mathcal{B}_T$$

is a maximal clique of the sparsity graph G

Proof. By $V(m_1 m_2) = V(m_1) \cup V(m_2)$ any product between elements in C_S has at most $|S| + 2\frac{T-|S|}{2} = T$ vertices, i.e., every C_S is a clique. Since we assumed $T \ll n$ we can, for each m_1 not contained in C_S, find an element m_2 in C_S with $(V(m_1) \cap V(m_2)) \setminus S = \emptyset$ and $V(m_2) = |S| + \frac{T-|S|}{2}$. Then $|V(m_1 m_2)| = |V(m_1) \cup V(m_2)| > T$, and m_1 cannot be added to the clique C_S. □

Example 9.6. Let $T = 4$, $S = \{1, 2\}$ in the case of (undirected) graphs. Then C_S contains all graphs with at most one vertex not in S, i.e., it contains the elements



for all choices of $i \in [n]$. Products between any two elements of C_S result in a graph on at most 4 vertices, and adding any other would result in a graph with at least 4 vertices.

Sadly, this construction does not to give all maximal cliques of G. For example, in the case $T = 3$, we obtain another clique

$$C_{\text{missing}} = \left\{ \varnothing, \begin{smallmatrix} 1 \\ 2 \end{smallmatrix}, \begin{smallmatrix} 1 \\ 3 \end{smallmatrix}, \begin{smallmatrix} 2 \\ 3 \end{smallmatrix} \right\}$$

which is not one of the cliques in Proposition 9.5. However, C_{missing} is contained in the clique $C_{\{1,2\}}$ in the case $T' = 4$.

While this family of maximal cliques C_S are not all the maximal cliques in the sparsity graph G, they are the "biggest" in the sense that they have the most "freely choosable labels" such as the vertices labeled i in Example 9.6. Furthermore, as seen in Proposition 9.7 below, all other cliques are contained in cliques of this form if we increase T enough.

Proposition 9.7. *Let C be a clique of the sparsity graph G on $\mathcal{B}_T$. Then C is contained in a maximal clique of the sparsity graph on $\mathcal{B}_{2T}$, of the form C_S of Proposition 9.5.*

Proof. Let m be an element of C with the maximum number of vertices of elements in C. Any other element $m' \in C$ has at most $T - |V(m)|$ vertices outside $V(m)$, and is thus contained in the clique $C_{V(m)}$ of the case $T' = |V(m)| + 2(T - |V(m)|) = 2T - |V(m)| \leq 2T$. □

Generally this bound is quite weak, as we only used the fact that all monomials have to intersect the biggest monomial of each clique, even though they all have to intersect each other. For example, the clique C_{missing} of the case $T = 3$ already appears in $C_{\{1,2\}}$ of the case $T = 4$. We will now simply ignore all cliques that do not fit into this construction, making the relaxation somewhat weaker.

We can now immediately exploit the symmetry of the problem by restricting the hierarchy to only contain the blocks with $S = \{1, \ldots, |S|\}$, as blocks corresponding to sets S of the same cardinality are identical by the action of S_n. This

results in a strong symmetry-reduction, for which we did not have to think about the representation theory of S_n. In the case $T = 4$ we obtain 3 maximal cliques of type C_S:

$$C_\emptyset = \left\{\emptyset, \begin{smallmatrix}(i)\\(j)\end{smallmatrix} : i < j \in [n]\right\},$$

$$C_{\{1,2\}} = \left\{\emptyset, \begin{smallmatrix}(1)\\(2)\end{smallmatrix}, \begin{smallmatrix}(1)\\(i)\end{smallmatrix}, \begin{smallmatrix}(2)\\(i)\end{smallmatrix}, \begin{smallmatrix}&(1)&\\(2)&&(i)\end{smallmatrix}, \begin{smallmatrix}(1)\\(2)\text{—}(i)\end{smallmatrix}, \begin{smallmatrix}&(1)&\\(2)&&(i)\\&\text{—}&\end{smallmatrix}, \begin{smallmatrix}(1)\\(2)\text{—}(i)\end{smallmatrix} : i > 2\right\},$$

$$C_{\{1,2,3,4\}} = \left\{\text{The } 2^{\binom{4}{2}} = 64 \text{ graphs on } \{1,2,3,4\}\right\}.$$

Proposition 9.8. *Each clique C_S is invariant under action of $S_S \times S_{[n]\setminus S}$. Furthermore, if x is a variable with $V(x) \subseteq S$, then*

$$xC_S = \{xm : m \in C_S\} \subseteq C_S.$$

Proof. Immediate by definition of C_S. □

We could now try to exploit the $S_S \times S_{[n]\setminus S}$ symmetry of each block corresponding to each clique C_S, but this proposition tells us we can do much better. It implies that the Möbius transformation on the vertices in S, in the sense of Section 6.4, is a basis transformation on the span of C_S. Indeed, the elements in each C_S are linearly independent, and we obtain a better basis by applying a Möbius transformation on it. We can do this by multiplying each monomial m, for each variable x that does not appear in m with $V(x) \subseteq S$, with $(1 - x)$. Now, since we are working with binary variables, we know that $x(1-x) = 0$, which causes elements of the transformed basis to be orthogonal to each other (in the sense that $pq = 0$), if they differ on the vertices in S.

In the case $T = 4$ the first clique $C_\emptyset$ does not change since S is empty there, but the block corresponding to $C_{\{1,2\}}$ is block-diagonalized, and $C_{\{1,2,3,4\}}$ is linearized by the transformation. The block $C_{\{1,2\}}$ splits into two, one for each graph on two vertices up to isomorphism:

$$\mathcal{B}_{\{1,2\}}\left(\begin{smallmatrix}(1)\\(2)\end{smallmatrix}\right) := \left\{\begin{smallmatrix}(1)\\(2)\end{smallmatrix}, \begin{smallmatrix}&(1)&\\(2)&&(i)\end{smallmatrix}, \begin{smallmatrix}(1)\\(2)\text{—}(i)\end{smallmatrix}, \begin{smallmatrix}&(1)&\\(2)&&(i)\\&\text{—}&\end{smallmatrix} : i > 2\right\},$$

$$\mathcal{B}_{\{1,2\}}\left(\begin{smallmatrix}(1)\\(2)\end{smallmatrix}\right) := \left\{\emptyset - \begin{smallmatrix}(1)\\(2)\end{smallmatrix}, \begin{smallmatrix}(1)\\(i)\end{smallmatrix} - \begin{smallmatrix}&(1)&\\(2)&&(i)\end{smallmatrix}, \begin{smallmatrix}(2)\\(i)\end{smallmatrix} - \begin{smallmatrix}(1)\\(2)\text{—}(i)\end{smallmatrix}, \begin{smallmatrix}(1)\\(2)\text{—}(i)\end{smallmatrix} - \begin{smallmatrix}&(1)&\\(2)&&(i)\\&\text{—}&\end{smallmatrix} : i > 2\right\},$$

where $\emptyset$ is the empty graph, which always has density one (i.e. $\emptyset$ corresponds to the constant 1 polynomial).

In general, each block corresponding to a clique C_S splits into one block corresponding to $\mathcal{B}(m)$ for each monomial m (graph H) with $V(m) \subseteq S$ (vertices in S), where

$$\mathcal{B}_S(m) := (m_{\text{ind}} C_S \setminus \{0\}) \subset \text{span}(C_S).$$

Here m_{ind} is the linear combination of monomials corresponding to the induced density of the fully labeled graph on S corresponding to m, as explained in Section 6.4, i.e.

$$m_{\text{ind}} = m \prod_{\substack{x : V(x) \subseteq S \\ x \text{ does not appear in } m}} (1 - x).$$

Thus, Proposition 9.8 directly implies

$$\text{span}(C_S) = \text{span}\left(\bigcup_{m \text{ monomial: } V(m) \subseteq S} \mathcal{B}_S(m) \right),$$

as we can use the formulas introduced in Section 6.4 to transform between elements of C_S and the union of the $\mathcal{B}_S(m)$. Since our variables are binary, we have $x(1-x) = 0$ for each variable x, and thus

$$pq = 0 \quad \text{for all } p \in \mathcal{B}_S(m), q \in \mathcal{B}_S(m'),$$

if $m \neq m'$ are monomials with vertices in S. Hence, this block-diagonalizes the block corresponding to C_S, and we can then delete copies of blocks by the symmetries of S_n, further reducing the hierarchy to have one block for each monomial (graph) up to isomorphism.

Remark 9.9. The reduction obtained from the Möbius transformation is strictly better than exploiting the symmetry S_k in each block, as it fully removes some orbits of pairs of basis elements. While it does "break" the symmetries in the sense that the resulting blocks $\mathcal{B}_S(H)$ each have a smaller symmetry group $\text{Aut}(H) \times S_{n-|S|}$ instead of $S_{|S|} \times S_{n-|S|}$, the orbits that got cut into smaller orbits have all but one of the new orbits deleted simply by the process of deleting copies of blocks (corresponding to isomorphic copies of H).

Remark 9.10. The Möbius transformation allows us to define *equivalent submodules between any product is zero*, in seeming contradiction to Theorem 7.8 stating that none of the block-diagonalized SDP entries are the zero polynomial. For example, consider the two submodules

$$M_1 = \operatorname{span}\left\{ \begin{array}{c}1\\2\end{array}\!\!\diagup\!\!\diagdown\!\!\bullet\,,\ \begin{array}{c}1\\2\end{array}\!\!\bullet\!\!-\!\!\bullet \right\} \subset \mathcal{B}_{\{1,2\}}\left(\begin{array}{c}1\\2\end{array}\right),$$

$$M_2 = \operatorname{span}\left\{ \begin{array}{c}1\\2\end{array}\!\!\diagup\!\!\diagdown\!\!\bullet \;-\; \begin{array}{c}1\\2\end{array}\!\!\diagdown\!\!\diagup\!\!\bullet\,,\ \begin{array}{c}1\\2\end{array}\!\!\bullet\!\!-\!\!\bullet \;-\; \begin{array}{c}1\\2\end{array}\!\!\bullet\!\!-\!\!\bullet \right\} \subset \mathcal{B}_{\{1,2\}}\left(\begin{array}{c}1\\2\end{array}\right).$$

The two submodules are equivalent, as they are both isomorphic to the S_2-permutation module $M^{(1,1)}$, but any product between elements of M_1 and M_2 is zero, since $x_{1,2}(1-x_{1,2})=0$:

$$M_1 \simeq M_2 \quad \text{and} \quad (pq = 0 \ \ \forall p \in M_1, q \in M_2).$$

The reason this is not truly a contradiction to Theorem 7.8, is that we are optimizing over the algebraic variety of the *binary hypercube* instead of polynomials in complex variables. The field $\{0,1\} = \mathbb{F}_2$ is not algebraically closed, which is a requirement for the second part of Schur's Lemma 7.5 to hold. (While $\mathbb{R}$ is not algebraically closed either, we can always block-diagonalize a problem with S_n symmetry over the reals.)

9.3 Razborov's Hierarchy

We can now define the hierarchy Razborov uses, which is, for example, implemented in the software package Flagmatic [FV13]. It is obtained by relaxing the hierarchy again, this time by only considering elements in the span of the C_m which are invariant under $S_{[n]\setminus S}$ (i.e., the vertices outside S are unlabeled/symmetrized over all vertices outside S).

Definition 9.11. Let T be an integer. The Tth level Raz_T of the *Razborov hierarchy* is obtained by optimizing over sums-of-squares of the form

$$\sum_{\substack{k:k\le T,\\ k\equiv T \mod 2}} \ \sum_{\substack{m \text{ monomial up to isomorphism,}\\ V(m)\subseteq[k]}} [\![p_{m,k}]\!],$$

where $p_{m,k}$ is a sum-of-squares of polynomials in

$$\mathcal{A}^{m,k} := \operatorname{span}\{f \in \mathcal{B}_{[k]}(m) : \sigma(f) = f \quad \text{for all } \sigma \in S_{[n]\setminus[k]}\},$$

which is the trivial isotypic component of the action of $S_{[n]\setminus[k]}$ on $\mathcal{B}_{[k]}(m)$.

Remark 9.12. The vector spaces $\mathcal{A}^{m,k}$ are exactly the (truncations of the) *Flag algebras of type* σ Razborov defined in [Raz13], where σ is the induced graph on $[k]$ corresponding to m (if we let n go towards infinity). These are given by the spans of the induced flags which are, when restricted to $[k]$, exactly σ, and are unlabeled otherwise. These form algebras themselves, but the unit of $\mathcal{A}^{m,k}$ is the fully labeled induced flag σ_{ind} instead of the empty flag. Note however, that our basis is slightly different from his, as we only consider *partially induced graph densities*, where only the fully labeled part on $[k]$ is induced.

Example 9.13. Let $T = 4$. Here the hierarchy Raz_4 optimizes over 3 semidefinite matrices. We have one 2×2 block given by

$$\mathcal{A}^{\varnothing,0} = \text{span}\left\{\varnothing, \right\},$$

two 4×4 blocks given by

$$\mathcal{A}^{\bullet,2} = \text{span}\left\{ \right\},$$

$$\mathcal{A}^{\varnothing,2} = \text{span}\left\{ \varnothing - \right\}.$$

The block coming from $C_{\{1,2,3,4\}}$ diagonalizes into 11 blocks of size 1×1 corresponding to the fully labeled graphs on four vertices up to isomorphism.

This is the hierarchy used to obtain most of the results in the flag SOS literature (see the beginning of Chapter 6 for an overview). It may seem easy to suggest that the Razborov hierarchy is weaker than the Lasserre hierarchy, as we obtained it through a sequence of relaxations of it. But, it turns out that the Razborov hierarchy can still be compared to the Lasserre hierarchy in both directions, as we will see in the following section.

Nearly Symmetric Sums-of-Squares. First, we make an important observation about sums-of-squares representations of polynomials invariant under some action of S_n on the set of variables: If we can write a symmetric polynomial as a sum-of-squares, then we can also write it as a sum-of-squares of *polynomials that are nearly symmetric*. More formally, we can show the following proposition, generalizing an idea of [Ray+18].

Proposition 9.14. *Let p be a nonnegative polynomial invariant under some action of S_n on the variables. If p has an SOS-certificate of degree d (i.e., we are considering the bound Las_{2d}), then we can find an SOS-certificate of p in the form*

$$p = \sum_{i=1}^{k} \mathcal{R}_{S_n}(q_i^2),$$

where $\deg(q_i) \leq d$ such that each q_i is invariant under the action of $S_{n-d\tau}$, where τ is a parameter that depends on the action of S_n, and is given by

$$\tau := \max_x \min_{T \subseteq [n]} |T| \quad \textit{s.t. } x \textit{ is invariant under action of } S_{[n]\setminus T} = \max_x |V(x)|. \quad (9.2)$$

In the case of flags, τ is exactly the maximum number of arguments a predicate takes.

Proof. The definition of τ tells us that, given the orbit of a monomial m, we can identify an element of the orbit by giving an ordered set of at most $\tau \deg(m)$ integers in $[n]$. Hence, we can embed the linear span of the orbit of m in the permutation module M^λ where $\lambda = (n - d\tau, 1, \ldots, 1)$ where 1 appears $d\tau$ times. Theorem 7.12 allows us to further decompose the M^λ for each monomial orbit in terms of Specht modules S^μ where each μ is a partition with the biggest part μ_1 at least $n - d\tau$.

As seen in Section 7.2 we now need to fix a nonzero basis element of S^μ for each μ to obtain the symmetry adapted basis for this problem. Earlier we fixed this element to be a polytabloid e_t given by a fixed standard-tableau t of shape μ. Here, we instead make a more symmetric choice. Let

$$e_\mu = \sum_{\hat{t} \sim t} e_{\hat{t}} \in S^\mu,$$

where $a \sim b$ denotes row-equivalency and t is a fixed tableau of shape μ. Since e_μ is sum of polytabloids it lies in S^μ, and by definition it is invariant under the action of S_T, where T is the set of indices appearing in the first row of t. We can fix t to the tableau we obtain by filling it row-wise, so $T \subseteq [n - d\tau]$. The vector e_μ is non-zero, since the Young-tabloid (row equivalency class) $\{t\}$ appears in each $e_{\hat{t}}$ with positive coefficient. The proposition now follows immediately by Theorem 7.8 with this choice of element. □

Remark 9.15. While this basis looks different and maybe "nicer" than when working with basis given by polytabloids, by Schur's Lemma (and the proof of Theorem 7.8) we know that for any two copies of S^μ given by ϕ_1 and ϕ_2 we have

$\mathcal{R}(\phi_1(e_\mu)\phi_2(e_\mu)) = c\mathcal{R}(\phi_1(e_t)\phi_2(e_t))$ for a positive factor $c \in \mathbb{R}$, i.e., the resulting hierarchy only differs by a constant factor.

If we now were to index our SDP by all polynomials symmetric under S_T for a fixed $T \subseteq [n]$ with $n - d\tau$ elements, we obtain the partially symmetry reduced SDP investigated in [Ray+18], which is equivalent to the Lasserre hierarchy.

Comparing the Razborov Hierarchy with the Lasserre Hierarchy. We can now use Proposition 9.14 to compare the two different hierarchies Las_{2d} (prioritizing a low degree) and Raz_T (prioritizing low number of vertices). Let $\Sigma_{\text{Las}_{2d}}$ and Σ_{Raz_d} be the sums-of-squares contained in the dth level of the Lasserre resp. Razborov hierarchy for flags.

Theorem 9.16. *We have*

$$\Sigma_{\text{Las}_{2d}} \subseteq \Sigma_{\text{Raz}_{2d\tau}}$$

and

$$\Sigma_{\text{Raz}_T} \subseteq \Sigma_{\text{Las}_{2k}},$$

where $k = \max_{m:|V(m)|\leq T} \deg(m)$ *is the maximum degree of a monomial with at most* T *vertices, and* $\tau = \max_x |V(x)|$ *as in* (9.2).

Proof. We show that the hierarchies appear as sub-blocks in each other, if one considers a high enough level of the hierarchy.

The level Las_{2d} contains polynomials of up to degree $2d$. As each variable has at most τ vertices, the appearing monomials have at most $2d\tau$ vertices. By Proposition 9.14 we can obtain the bound Las_{2d} by considering products of polynomials with at most $d\tau$ different labels. By definition, level $2d\tau$ of the Razborov hierarchy contains all products between flags with labels in $[2d\tau]$ resulting in flags with at most $2d\tau$ vertices, hence, it contains Las_{2d} as sub-block.

To see the other way, note that by choice of k, the dth level of the Razborov hierarchy contains products between polynomials of at most degree k. By definition, the level Las_{2k} contains all products between polynomials with up to degree k, so Raz_d is a sub-block of Las_{2k}. □

Corollary 9.17. *In the setting of (undirected) graphs we have*

$$\Sigma_{\text{Las}_{2d}} \subseteq \Sigma_{\text{Raz}_{4d}}$$

and

$$\Sigma_{\text{Raz}_T} \subseteq \Sigma_{\text{Las}_{2\binom{T}{2}}}.$$

Example 9.18. If we want to find minimum $p = \inf f$ of a linear combination of graph densities

$$f = \sum_i c_i G_i$$

without further constraints, we would solve

$$p \geq \mathrm{Las}_{2d} = \sup \ \lambda \quad \text{s.t. } f - \lambda \in \Sigma_{\mathrm{Las}_{2d}}$$

or

$$p \geq \mathrm{Raz}_T = \sup \ \lambda \quad \text{s.t. } f - \lambda \in \Sigma_{\mathrm{Raz}_T}.$$

Then

$$\mathrm{Las}_{2d} \leq \mathrm{Raz}_{4d} \leq p$$

and

$$\mathrm{Raz}_T \leq \mathrm{Las}_{2\binom{T}{2}} \leq p,$$

where bigger means we obtain a better bound on the minimum of f. If we had additional constraints of the form $g \geq 0$, where g is again a linear combination of flags, we would add quadratic modules as in (1.8), but the different truncations by the number of edges respectively vertices in the products gs, where s is a sum-of-squares, makes the resulting hierarchies less straightforward to compare.

Remark 9.19. Generally the inequalities are strict: $\mathrm{Raz}_{2d\tau}$ contains significantly more products between polynomials than Las_{2d}, and Las_{2k} contains significantly more than Raz_T. While the choice of hierarchy does not matter much in theory, we cannot reach high levels of either computationally, i.e., both may be a better choice in practice than the other, depending on the problem.

Breaking Symmetry Comes at a Cost. While we always had $S_k \times S_{n-k}$ symmetry for some k up the point we applied the Möbius transformation, at that point the symmetries broke. The final blocks appearing in the Razborov hierarchy have symmetry group $\mathrm{Aut}(G)$ acting on the blocks, where G is the graph (or monomial) defining a block. Eventually every graph G will show up in the hierarchy as T increases. As it is possible to construct a (finite, directed) graph with automorphism group isomorphic to any finite group by Frucht's Theorem [Fru39], the full symmetry reduction of the hierarchy involves the representation theory of every finite group already in the case of directed graphs. Hence, an analytical reduction is here out of the question even further than for the reduction in Chapter

T	block sizes	total size	variables	Las
4	$4^2 2^1 1^{11}$	21	34	109
	$3^2 2^1 1^{13}$	21	28	
5	$8^4 6^1 1^{34}$	72	199	1171
	$6^3 4^2 2^4 1^{34}$	68	129	
6	$20^2 16^{11} 4^1 1^{156}$	376	2082	22211
	$14^2 12^2 10^1 9^2 8^2 6^5 5^2 4^5 3^{10} 1^{162}$	348	990	
7	$72^4 32^{34} 20^1 1^{1044}$	2440	29718	
	$48^2 26^2 24^8 22^2 20^6 18^6 16^2 12^{15} 10^2 8^9 6^{32} 4^8 2^{22} 1^{1044}$	2228	11984	
8	$272^{11} 120^2 64^{156} 11^1 1^{12346}$	25573	759820	
	$176^2 152^1 120^1 116^2 96^2 92^2 84^2 74^2 68^2 64^8 60^6 48^{30}$ $46^2 44^2 40^{16} 36^{28} 32^6 30^{12} 28^4 27^4 24^{42} 21^6 20^6 18^{18}$ $16^{38} 15^{12} 13^2 12^{92} 11^3 10^{16} 9^{32} 8^{26} 7^4 6^{22} 5^8 4^{50} 3^{58}$ 1^{12378}	23733	259220	

Table 9.1: Comparing the block sizes of the Razborov hierarchy before and after additional symmetry reduction.
The block sizes of the (full) Razborov hierarchy (rows with gray background) compared with the block sizes of the same hierarchy, after exploiting additional symmetries computationally (white background). The last column gives the total size of the hierarchy obtained from the naive symmetry reduction approach of the Lasserre hierarchy generated by graphs with up to T vertices (see Table 8.4 for more details).

8. But what we can do, knowing the symmetry of a block, is a numerical block-diagonalization using algorithms described in [Mur+10]. A basic implementation of the numerical block-diagonalization algorithm is available as part of the Julia software package "SDPSymmetryReduction.jl", described in [BK22a], which we make use of in the package "FlagSOS.jl" mentioned in Chapter 8. Depending on the group, this will not always be possible over the real numbers, but may require blocks over the quaternions or complex numbers.

We give a comparison of block-sizes before and after this additional reduction in Table 9.1. Since the total size of the SDP does not change much after this additional reduction, we also give the total number of variables in the matrix variables (only counting upper-triangular entries). For additional comparison, we also give the total size of the hierarchy obtained from the naive approach mentioned in Section 9.1, where already the case $T = 6$ is far out of reach of today's solvers with a total block size of 22211.

A

Characterizing Nonnegative Projections

The following Theorem, originally due to Belitskii and Lyubich (cf. [BL88, p. 108]), has a typo in the proof in [BL88]. Here we include a cleaner, more detailed version of the proof.

Theorem A.1. *Let P be a nonnegative projection, i.e., $P \geq 0$ and $P^2 = P$. Then P is of the form*

$$Px = \sum_{i=1}^{m} \langle x, a_i^* \rangle (a_i + b_i),$$

where $m = \operatorname{rank}(P)$, $\{a_i\}$ are orthogonal vectors, the set $\{a_i^*\}$ is biorthogonal to the set $\{a_i\}$, i.e., $\langle a_i, a_j^* \rangle = \delta_{ij}$, and $\{b_i\}$ is orthogonal to $\{a_i\} \cup \{a_i^*\}$. All those vectors a_i, a_i^*, b_i are element wise nonnegative.

Proof. The projection P is uniquely defined by the images of the elements of the canonical basis $\{e_i\}_{i=1}^n$, which we call $c_i := Pe_i \geq 0$. Thus, we can define a cone

$$K_P := \operatorname{cone}\{c_1, \ldots, c_n\} = \left\{ \sum_{i=1}^{n} \lambda_i c_i \ : \ \lambda_i \geq 0 \text{ for } i = 1, \ldots, n \right\},$$

which has a non-empty interior when seen as subset of $\operatorname{Im}(P)$. We call a point $p \in K_P$ *extreme*, if for all $a, b \in K_P \setminus \{0\}$ with $p = \frac{1}{2}(a + b)$ follows that a, b and p are collinear, i.e., are scaled versions of each other.

Let m now be the cardinality of a maximum set of extreme, pairwise non-collinear points in K_p. Then we can relabel the e_i (and accordingly c_i), such that

$$K_P = \text{cone}\{c_1, \ldots, c_m\},$$

i.e., $c_1, \ldots, c_m$ span the image of P. We see the *support* of a vector p as the set $\text{supp}(p) \coloneqq \{e_i \ : \ p_i > 0\}$.

Lemma A.2. $\text{supp}(c_i) \cap \text{supp}(c_j) \subset \ker(P)$ *for all* $1 \leq i < j \leq m$.

Proof. Let $e_k \in \text{supp}(c_i)$. Because P is element wise nonnegative (and thus $c_i \geq 0$), and $c_i = Pe_i = P^2 c_i = Pc_i$, we have, for $\varepsilon > 0$ small enough, that

$$c_i \pm \varepsilon P e_k = P(c_i \pm \varepsilon e_k) = ((c_i)_k \pm \varepsilon) P e_k + \sum_{\substack{l=1,\ldots,n \\ l \neq k}} (c_i)_l P e_l \in K_P.$$

Since c_i is extreme, we get that $Pe_k = \lambda c_i$ for a $\lambda \geq 0$.

Let now $e_k \in \text{supp}(c_i) \cap \text{supp}(c_j)$ for $1 \leq i < j \leq m$. Then $Pe_k = \lambda_1 c_i = \lambda_2 c_j$, and thus, by choice of the c_i, that $\lambda = 0$, and $e_k \in \ker(P)$.

□

But now back to the proof of Theorem A.1. We can decompose the $c_i \geq 0$ for $i = 1, \ldots, m$ by splitting their support depending on whether the coordinate is in $\ker(P)$:

$$c_i = a_i + b_i, \text{ where } \text{supp}(a_i) \cap \ker(P) = \emptyset, \text{supp}(b_i) \subset \ker(P).$$

Of course, we also have $a_i \geq 0$ and $b_i \geq 0$. Now, Lemma A.2 tells us that

$$\text{supp}(a_i) \cap \text{supp}(a_j) = \emptyset \text{ for } i \neq j,$$

and by definition we have

$$\text{supp}(a_i) \cap \text{supp}(b_j) = \emptyset \text{ for all } i, j,$$

and thus

$$\langle a_i, a_j \rangle = 0 \text{ for } i \neq j \text{ and } \langle a_i, b_j \rangle = 0 \text{ for all } i, j.$$

Because $Pc_i = c_i = a_i + b_i \notin \ker(P)$ we know that $a_i \neq 0$ for all $i = 1, \ldots, m$, and all c_i are linearly independent, since their a_i-parts all have disjoint supports. Thus, we get that $m \leq \text{rank}(P)$, and we already knew that $m \geq \text{rank}(P)$ because the cone K_P does have non-empty interior in the image space of P.

We now want to determine the coordinates $\gamma(x)$ of Px in the basis given by the c_i, i.e.

$$Px = \sum_{i=1}^{m} \gamma(x)_i c_i = \sum_{i=1}^{m} \gamma(x)_i (a_i + b_i).$$

Because the a_i are all nonzero, and have supports disjoint from each others and those of the b_i, these coordinates are exactly

$$\gamma(x)_i = \frac{\langle Px, a_i \rangle}{\langle a_i, a_i \rangle} = \left\langle x, \frac{P^* a_i}{\langle a_i, a_i \rangle} \right\rangle = \langle x, a_i^* \rangle, \text{ where } a_i^* := \frac{P^* a_i}{\langle a_i, a_i \rangle} \geq 0.$$

We see that $Pa_i = Pa_i + Pb_i = Pc_i = c_i$, which tells us that $\langle a_i, a_i^* \rangle = \delta_{ij}$, since

$$c_i = Pa_i = \sum_{i=1}^{m} \langle a_i, a_i^* \rangle c_i.$$

Finally, we see that $\langle a_i^*, b_j \rangle = 0$ for all i, j, because $b_i \in \ker(P)$. Conversely, every such function is a nonnegative projection, as it is obviously nonnegative, and idempotent:

$$\begin{aligned} PPx &= \sum_{i=1}^{m} \langle Px, a_i^* \rangle (a_i + b_i) \\ &= \sum_{i,j=1}^{m} \langle x, a_j^* \rangle \langle a_j + b_j, a_i^* \rangle (a_i + b_i) \\ &= \sum_{i,j=1}^{m} \langle x, a_j^* \rangle \delta_{ij} (a_i + b_i) = Px. \end{aligned}$$

□

Corollary A.3. *If P is a nonnegative orthogonal projection, then it is of the form*

$$Px = \sum_{i=1}^{m} \langle x, a_i \rangle a_i,$$

where $\{a_i\}$ is a set of orthonormal, nonnegative vectors and $m = \operatorname{rank}(P)$.

Proof. We have seen that $Pa_i = a_i + b_i$, and $b_i \in \ker P$. Since P is orthogonal, we have $0 = \langle a_i, Pb_i \rangle = \langle a_i + b_i, b_i \rangle = \langle b_i, b_i \rangle$ and thus $b_i = 0$. We earlier fixed $a_i^* = \frac{P^* a_i}{\langle a_i, a_i \rangle} = Pa_i = a_i + b_i = a_i$. □

Index

Bochner's theorem, 147
  For flags, 152
Bose-Mesner algebra of the Johnson scheme, 106

Configuration
  Coherent, 28, 54
    Weisfeiler-Leman algorithm, 29
  Jordan, 28, 32, 54
    For the relaxation of the QAP, 53

Energy minimization, 59
  As a QAP, 61
  Bound comparison, 64
  Checkerboard conjecture, 76
  Fractional total energy, 62
  In the continuous setting, 60
  Inverting an arrangement of particles, 62
  Jordan reduction of the SDP bound, 69
  Linear bound based on the reduced SDP bound, 77
  Numerical results, 80
  On a toric grid, 60
  Simulated annealing heuristic, 80
  Symmetry reduced SDP bound, 75

First-order theory, 136
  Axioms, 137
  Predicate, 136
Flag, 133
  Algebra, 134
    As abelian group, 148
    Characters, 148
    Non-induced, 130
    Razborov's, 137
  Connection to polynomial optimization, 143
  Convergent sequence, 131
  Empty flag, 134
  Flag module, 175
  For degenerate extremal combinatorics, 206
  Fourier decomposition, 151
  Gluing, 132, 133, 138, 150
  Harmonic, 147
  Labeling canonically, 211
  Lasserre hierarchy, 144
    Symmetry reduction, 171
  Möbius transformation, 139
  Notation, xviii
  Quantum flag, 134

Razborov's hierarchy, 222
Symmetry reduction, 226
Specht flags, 191
Subgraph density
Induced, 137
Non-induced, 130
Swapping, 139
Sums-of-squares, 132, 135
Symmetries, 143
Unlabeling, 135, 138
Fourier analysis, 147

Gatermann Parrilo reduction, 156, 160
Graph
Hypergraph, 93, 174
Module, 175
Notation, xviii
Profile of, 195
Specht graph, 191
Triangle free, 130

Hilbert Cube, 131
Hilbert's trick, 111

Lasserre hierarchy, 10
Comparison with Razborov's hierarchy, 225
Vertex truncated, 216

Möbius transformation, 139
For symmetry reducton, 220
Mantel's theorem, 130
Lower bound, 132
Upper bound, 135
Module, 15, *see also* Representation
Flag module, 175
Decomposition, 179
Irreducible, 158
Permutation module, 164
Specht, 165

Optimization
Conic, 3, 6, 23
Convex, 1
Doubly nonnegative, 21, 30, 35, 46
Dual problem, 3, 76
Linear, 4, 77
Polynomial, 12, 93, 143, 156, *see also* Sums-of-squares
Primal problem, 3, 6
Quadratic, convex, 45
Second order cone, 75
Semidefinite, 5, 6, 8, 13
Weak duality, 4

Projection, 23
For symmetry reduction, 23
Nonnegative projection matrices, 31
Characterization of, 229
orthogonal, 25
Preserving doubly nonnegativity, 30
Preserving positive semidefiniteness, 33
Proper convex cone, 3

Quadratic assignment problem, 22, 43
Relaxations, 44
Symmetry reduction, 52
Application to QAPLib, 54
Queuing theory, 98

Connection to the polynomial optimization problem, 99
Redundancy scheduling, 99

Recursion, 115, 153, 184, *see* Recursion
Relaxation, 7
By adding dual constraints, 77
Of stable set, 7
Of the quadratic assignment problem, 44
Comparison, 47
Convex quadratic, 45
Projected eigenvalue, 44
Semidefinite, 46
Representation, 15, *see also* Module
Maschke's Theorem, 159
Module, 158
Multiplicity space, 167
Of the symmetric group, 163
Permutation module, 164
Specht-module, 165
Young-tableau, 164
Young-tableau, generalized, semistandard, 166
Young-tabloid, 164
Schur's Lemma, 159
Symmetry adapted basis, 160
Multiplying basis elements, 168
Of permutation modules, 166
Representation theory
Notation, xvii
Reynolds operator, 156, 180

Schur complement, 48, 120
Sidorenko's conjecture, 153
Simulated annealing, 80
Software
For flag sums-of-squares, 211
For symmetry reduction, 35
Sums-of-squares, 12, 132, 156
For flags, 144
Symmetric circulant matrices, 68
Symmetric circulant matrix
Block-diagonalization, 73
Symmetry reduction, 1, 14, 95
As a proving tool, 101
Block-diagonalization, 17, 159
Analytical, 73
Computational, 37
Of the Lasserre hierarchy for flags, 191
Of the Terwilliger algebra, 104
By Möbius transformation, 220
Combination with facial reduction, 39
For polynomial optimization, 156
Group symmetry reduction, 14
Jordan reduction, 23–29
Analytical, 69
Application to the QAP, 52
Constraint Set Invariance, 23
Extension to the doubly nonnegative cone, 30
Lasserre hierarchy for flags, 171
Lasserre hierarchy for stable set, 11
Of the relaxation of the QAP, 52
Of the Theta function, 9
By Jordan reduction, 33
Software, 35

Code example, 39

Terwilliger algebra, 95, 103
Block-diagonalization, 104

Theta function, 7, 21
Definition, 8
Of Erdős-Rényi graphs, 34
Symmetries, 9

Bibliography

[AB01a] Kurt M. Anstreicher and Nathan W. Brixius. “A new bound for the quadratic assignment problem based on convex quadratic programming”. In: *Mathematical Programming* 89.3 (Feb. 2001), pp. 341–357. DOI: 10.1007/pl00011402 (cit. on pp. 44–46, 52).

[AB01b] Kurt M. Anstreicher and Nathan W. Brixius. “Solving quadratic assignment problems using convex quadratic programming relaxations”. In: *Optimization Methods and Software* 16.1-4 (Jan. 2001), pp. 49–68. DOI: 10.1080/10556780108805828 (cit. on p. 44).

[Ans+02] Kurt Anstreicher, Nathan Brixius, Jean-Pierre Goux, and Jeff Linderoth. “Solving large quadratic assignment problems on computational grids”. In: *Mathematical Programming* 91.3 (Feb. 2002), pp. 563–588. DOI: 10.1007/s101070100255 (cit. on p. 44).

[Art27] Emil Artin. “Zur Theorie der hyperkomplexen Zahlen”. In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 5.1 (Dec. 1927), pp. 251–260. DOI: 10.1007/bf02952526 (cit. on p. 17).

[ATY97] Susumu Ariki, Tomohide Terasoma, and Hiro-Fumi Yamada. “Higher Specht polynomials”. In: *Hiroshima Mathematical Journal* 27.1 (1997), pp. 177–188 (cit. on p. 191).

[Bab+97] Luitpold Babel, Irina V. Chuvaeva, Mikhail Klin, and Dmitrii V. Pasechnik. “Algebraic Combinatorics in Mathematical Chemistry. Methods and Algorithms. II. Program Implementation of the Weisfeiler-Leman Algorithm”. In: *arXiv preprint arXiv:1002.1921* (1997) (cit. on p. 29).

[Bac+11] Christine Bachoc, Dion C. Gijswijt, Alexander Schrijver, and Frank Vallentin. “Invariant Semidefinite Programs”. In: *Handbook on semidefinite, conic and polynomial optimization*. Springer US, Sept. 2011,

pp. 219–269. DOI: 10.1007/978-1-4614-0769-0_9 (cit. on pp. 22, 96).

[Bar02] Alexander Barvinok. *A Course in Convexity*. American Mathematical Society, Nov. 2002. DOI: 10.1090/gsm/054 (cit. on p. 3).

[BDL13] Niek Bouman, Jan Draisma, and Johan S. H. van Leeuwaarden. "Energy Minimization of Repelling Particles on a Toric Grid". In: *SIAM Journal on Discrete Mathematics* 27.3 (2013), pp. 1295–1312. DOI: 10.1137/120869067 (cit. on pp. 18, 59, 60, 62–64, 66, 76, 80, 81, 83).

[BDM12] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, Jan. 2012. DOI: 10.1137/1.9781611972238 (cit. on pp. 43–45).

[BEK78] G.P. Barker, L.Q. Eifler, and T.P. Kezlan. "A non-commutative spectral theorem". In: *Linear Algebra and its Applications* 20.2 (1978), pp. 95–100. DOI: 10.1016/0024-3795(78)90043-5 (cit. on p. 103).

[BK22a] Daniel Brosch and Etienne de Klerk. "Jordan symmetry reduction for conic optimization over the doubly nonnegative cone: theory and software". In: *Optimization Methods and Software* (2022), pp. 1–20. DOI: 10.1080/10556788.2021.2022146 (cit. on pp. 20, 227).

[BK22b] Daniel Brosch and Etienne de Klerk. "Minimum energy configurations on a toric lattice as a quadratic assignment problem". In: *Discrete Optimization* 44 (May 2022), p. 100612. DOI: 10.1016/j.disopt.2020.100612 (cit. on pp. 20, 53).

[BKR97] Rainer E. Burkard, Stefan E. Karisch, and Franz Rendl. "QAPLib – A Quadratic Assignment Problem Library". In: *Journal of Global Optimization* 10.4 (1997), pp. 391–403. DOI: 10.1023/a:1008293323270 (cit. on pp. 17, 54, 60).

[BL88] G. R. Belitskii and Yurii I. Lyubich. *Matrix Norms and their Applications*. Volume 36 of Operator Theory: Advances and Applications. Birkhäuser Basel, 1988. DOI: 10.1007/978-3-0348-7400-7 (cit. on pp. 31, 229).

[Ble+20a] Grigoriy Blekherman, Annie Raymond, Mohit Singh, and Rekha R Thomas. "Simple graph density inequalities with no sum-of-squares proofs". In: *Combinatorica* 40.4 (2020), pp. 455–471 (cit. on p. 153).

[Ble+20b] Grigoriy Blekherman, Annie Raymond, Mohit Singh, and Rekha R Thomas. “Tropicalization of Graph Profiles”. In: *arXiv preprint arXiv:2004.05207* (2020) (cit. on p. 198).

[BLS21] Daniel Brosch, Monique Laurent, and Andries Steenkamp. “Optimizing Hypergraph-Based Polynomials Modeling Job-Occupancy in Queuing with Redundancy Scheduling”. In: *SIAM Journal on Optimization* 31.3 (2021), pp. 2227–2254. DOI: 10.1137/20m1369592 (cit. on p. 20).

[Boc32] Salomon Bochner. *Vorlesungen über Fouriersche integrale*. Vol. 12. Akademische Verlagsgesellschaft, 1932 (cit. on pp. 147, 152).

[BP22] Daniel Brosch and Sven Polak. “New lower bounds on crossing numbers of $K_{m,n}$ from permutation modules and semidefinite programming”. In: *arXiv preprint arXiv:2206.02755* (June 2022) (cit. on p. 185).

[Cam03] Peter J. Cameron. “Coherent configurations, association schemes and permutation groups”. In: *Groups, combinatorics and geometry*. World Scientific, Mar. 2003, pp. 55–71. DOI: 10.1142/9789812564481_0004 (cit. on p. 28).

[CBL20] Ellen Cardinaels, Sem Borst, and Johan S. H. van Leeuwaarden. “Heavy-Traffic Universality of Redundancy Systems with Assignment Constraints”. In: *arXiv preprint arXiv:2005.14566* (May 2020) (cit. on p. 99).

[CBL21] Ellen Cardinaels, Sem Borst, and Johan S. H. van Leeuwaarden. “Power-of-two Policies in Redundancy Systems: the Impact of Assignment Constraints”. In: *arXiv preprint arXiv:2111.05777* (Nov. 2021) (cit. on pp. 18, 93, 94, 98–100).

[CE03] Henry Cohn and Noam Elkies. “New upper bounds on sphere packings I”. In: *Annals of Mathematics* 157.2 (Mar. 2003), pp. 689–714. DOI: 10.4007/annals.2003.157.689 (cit. on p. 147).

[CK06] Henry Cohn and Abhinav Kumar. “Universally optimal distribution of points on spheres”. In: *Journal of the American Mathematical Society* 20.1 (Sept. 2006), pp. 99–148. DOI: 10.1090/s0894-0347-06-00546-7 (cit. on pp. 60, 83, 84, 147).

[CL18] David Conlon and Joonkyung Lee. “Sidorenko’s conjecture for blow-ups”. In: *arXiv preprint arXiv:1809.01259* (2018) (cit. on p. 153).

[Coh+17] Henry Cohn, Abhinav Kumar, Stephen Miller, Danylo Radchenko, and Maryna Viazovska. “The sphere packing problem in dimension 24”. In: *Annals of Mathematics* 185.3 (May 2017). DOI: 10.4007/annals.2017.185.3.8 (cit. on p. 147).

[DAn02] John D’Angelo. *Inequalities from Complex Analysis*. American Mathematical Society, 2002. DOI: 10.5948/upo9780883859704 (cit. on p. 163).

[Del73] Philippe Delsarte. “An algebraic approach to the association schemes of coding theory”. In: *Philips Res. Rep. Suppl.* 10 (1973) (cit. on pp. 35, 96).

[DL98] Philippe Delsarte and Vladimir I. Levenshtein. “Association schemes and coding theory”. In: *IEEE Transactions on Information Theory* 44.6 (1998), pp. 2477–2504 (cit. on pp. 96, 106).

[Dos+17] Maria Dostert, Cristóbal Guzmán, Fernando Mário de Oliveira Filho, and Frank Vallentin. “New Upper Bounds for the Density of Translative Packings of Three-Dimensional Convex Bodies with Tetrahedral Symmetry”. In: *Discrete & Computational Geometry* 58.2 (Mar. 2017), pp. 449–481. DOI: 10.1007/s00454-017-9882-y (cit. on p. 163).

[DR20] Sebastian Debus and Cordian Riener. “Reflection groups and cones of sums-of-squares”. In: *arXiv preprint arXiv:2011.09997* (2020) (cit. on pp. 172, 178).

[DR21] Mirjam Dür and Franz Rendl. “Conic optimization: A survey with special focus on copositive optimization and binary quadratic problems”. In: *EURO Journal on Computational Optimization* 9 (2021), p. 100021. ISSN: 2192-4406. DOI: https://doi.org/10.1016/j.ejco.2021.100021 (cit. on p. 6).

[ELS79] Paul Erdős, László Lovász, and Joel Spencer. “Strong independence of graphcopy functions”. In: *Graph theory and related topics* (1979), pp. 165–172 (cit. on p. 195).

[FA94] Jacques Faraut and A. Korányi. “Analysis on symmetric cones”. In: *Oxford mathematical monographs* (1994) (cit. on p. 26).

[FBR87] Gerd Finke, Rainer E. Burkard, and Franz Rendl. “Quadratic Assignment Problems”. In: *Surveys in Combinatorial Optimization*. Elsevier, 1987, pp. 61–82. DOI: 10.1016/s0304-0208(08)73232-8 (cit. on p. 44).

[FK69] L. M. Falicov and J. C. Kimball. “Simple Model for Semiconductor-Metal Transitions: SmB6 and Transition-Metal Oxides”. In: *Physical Review Letters* 22.19 (1969), pp. 997–999. DOI: 10.1103/physrevlett.22.997 (cit. on p. 59).

[Fru39] Robert Frucht. “Herstellung von Graphen mit vorgegebener abstrakter Gruppe”. In: *Compositio Mathematica* 6 (1939), pp. 239–250 (cit. on p. 226).

[FS13] Zoltán Füredi and Miklós Simonovits. “The History of Degenerate (Bipartite) Extremal Graph Problems”. In: *Erdős Centennial*. Ed. by László Lovász, Imre Z. Ruzsa, and Vera T. Sós. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 169–264. ISBN: 978-3-642-39286-3. DOI: 10.1007/978-3-642-39286-3_7 (cit. on p. 207).

[FV13] Victor Falgas-Ravry and Emil R. Vaughan. “Applications of the Semi-Definite Method to the Turán Density Problem for 3-Graphs”. In: *Combinatorics, Probability and Computing* 22.1 (2013), pp. 21–54. DOI: 10.1017/S0963548312000508 (cit. on pp. 136, 211, 222).

[Gal04] Aurél Galántai. *Projectors and Projection Methods*. Vol. 6. Advances in Mathematics. Springer US, 2004. DOI: 10.1007/978-1-4419-9180-5 (cit. on p. 31).

[Gar+16] Kristen Gardner, Samuel Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytiä, and Alan Scheller-Wolf. “Queueing with redundant requests: exact analysis”. In: *Queueing Systems* 83.3-4 (July 2016), pp. 227–259. DOI: 10.1007/s11134-016-9485-y (cit. on p. 100).

[Gij09] Dion Gijswijt. “Block-diagonalization for algebra’s associated with block codes”. In: *arXiv preprint arXiv:0910.4515* (Oct. 2009) (cit. on pp. 17, 168, 169).

[Gil+19] Shoni Gilboa, Roman Glebov, Dan Hefetz, Nati Linial, and Avraham Morgenstern. “On the local structure of oriented graphs–a case study in flag algebras”. In: *arXiv preprint arXiv:1908.06480* (2019) (cit. on p. 129).

[GJ78] M. R. Garey and D. S. Johnson. “"Strong" NP-Completeness Results: Motivation, Examples and Implications”. In: *Journal of the ACM* 25.3 (July 1978), pp. 499–508. DOI: 10.1145/322077.322090 (cit. on p. 7).

[Gle+16] Roman Glebov, Andrzej Grzesik, Ping Hu, Tamas Hubai, Daniel Kral, and Jan Volec. “Densities of 3-vertex graphs”. In: *arXiv preprint arXiv:1610.02446* (2016) (cit. on p. 198).

[GMS12] Dion C. Gijswijt, Hans D. Mittelmann, and Alexander Schrijver. “Semidefinite Code Bounds Based on Quadruple Distances”. In: *IEEE Transactions on Information Theory* 58.5 (May 2012), pp. 2697–2705. DOI: 10.1109/tit.2012.2184845 (cit. on p. 96).

[GN08] C.D. Godsil and M.W. Newman. “Eigenvalue bounds for independent sets”. In: *Journal of Combinatorial Theory, Series B* 98.4 (July 2008), pp. 721–734. DOI: 10.1016/j.jctb.2007.10.007 (cit. on pp. 34, 35).

[Goa+18] Xavier Goaoc, Alfredo Hubard, Rémi De Joannis De Verclos, Jean-Sébastien Sereni, and Jan Volec. “Limits of Order Types *”. working paper or preprint. Nov. 2018. URL: https://hal.archives-ouvertes.fr/hal-01913144 (cit. on pp. 129, 198).

[GP04] Karin Gatermann and Pablo A. Parrilo. “Symmetry groups, semidefinite programs, and sums-of-squares”. In: *Journal of Pure and Applied Algebra* 192.1-3 (Sept. 2004), pp. 95–128. DOI: 10.1016/j.jpaa.2003.12.011 (cit. on pp. 19, 96, 155, 160, 163).

[Gra06] Robert M. Gray. “Toeplitz and Circulant Matrices: A Review”. In: *Foundations and Trends® in Communications and Information Theory* 2.3 (2006), pp. 155–239. DOI: 10.1561/0100000006 (cit. on pp. 68, 73).

[GST08] Dion Gijswijt, Alexander Schrijver, and Hajime Tanaka. “New upper bounds for nonbinary codes based on the Terwilliger algebra and semidefinite programming”. In: *Journal of Combinatorial Theory, Series A* 113.8 (2008), pp. 1719–1731. DOI: 10.1016/j.jcta.2006.03.010 (cit. on p. 96).

[Hil94] David Hilbert. “Ein Beitrag zur Theorie des Legendre’schen Polynoms”. In: *Acta Mathematica*. Springer Berlin Heidelberg, 1894, pp. 367–370. DOI: 10.1007/978-3-662-26737-0_21 (cit. on p. 111).

[HRW90] Scott W Hadley, Franz Rendl, and Henry Wolkowicz. “Bounds for the Quadratic Assignment Problems Using Continuous Optimization Techniques.” In: *IPCO*. 1990, pp. 237–248 (cit. on pp. 44, 45).

[HRW92] S. W. Hadley, F. Rendl, and H. Wolkowicz. “A New Lower Bound Via Projection for the Quadratic Assignment Problem”. In: *Mathematics of Operations Research* 17.3 (Aug. 1992), pp. 727–739. DOI: 10.1287/moor.17.3.727 (cit. on pp. 44, 45).

[HS18] Wilfrid Hodges and Thomas Scanlon. “First-order Model Theory”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2018. Metaphysics Research Lab, Stanford University, 2018 (cit. on p. 136).

[HSW19] Hao Hu, Renata Sotirov, and Henry Wolkowicz. “Facial Reduction for Symmetry Reduced Semidefinite Doubly Nonnegative Programs”. In: *arXiv preprint arXiv:1912.10245* (Dec. 2019) (cit. on p. 39).

[Hua+13] Hao Huang, Nati Linial, Humberto Naves, Yuval Peled, and Benny Sudakov. “On the 3-Local Profiles of Graphs”. In: *Journal of Graph Theory* 76.3 (Aug. 2013), pp. 236–248. DOI: 10.1002/jgt.21762 (cit. on p. 198).

[Kan+01] Yoshihiro Kanno, Makoto Ohsaki, Kazuo Murota, and Naoki Katoh. In: *Optimization and Engineering* 2.3 (2001), pp. 293–320. DOI: 10.1023/a:1015366416311 (cit. on p. 8).

[KDP11] Etienne de Klerk, Cristian Dobre, and Dmitrii V. Pasechnik. “Numerical block-diagonalization of matrix $*$-algebras with application to semidefinite programming”. In: *Mathematical programming* 129.1 (May 2011). DOI: 10.1007/s10107-011-0461-3 (cit. on p. 37).

[Ken94] Tom Kennedy. “Some rigorous results on the ground states of the Falicov-Kimball model”. In: *The State Of Matter: A Volume Dedicated to EH Lieb* (1994), pp. 42–80 (cit. on p. 59).

[Kle+09a] E. de Klerk, M.W. Newman, D.V. Pasechnik, and R. Sotirov. “On the Lovász ϑ-number of almost regular graphs with application to Erdős–Rényi graphs”. In: *European Journal of Combinatorics* 30.4 (2009), pp. 879–888. ISSN: 0195-6698. DOI: https://doi.org/10.1016/j.ejc.2008.07.022 (cit. on pp. 34, 35).

[Kle+09b] Etienne de Klerk, Mike W Newman, Dmitrii V Pasechnik, and Renata Sotirov. “On the Lovász ϑ-number of almost regular graphs with application to Erdős–Rényi graphs”. In: *European Journal of Combinatorics* 30.4 (2009), pp. 879–888 (cit. on p. 17).

[KLL15] Jeong Han Kim, Choongbum Lee, and Joonkyung Lee. “Two approaches to Sidorenko’s conjecture”. In: *Transactions of the American Mathematical Society* 368.7 (Dec. 2015), pp. 5057–5074. DOI: 10.1090/tran/6487 (cit. on p. 153).

[Knu91] Donald E. Knuth. “Efficient representation of perm groups”. In: *Combinatorica* 11.1 (Mar. 1991), pp. 33–43. DOI: 10.1007/bf01375471 (cit. on p. 185).

[KOP11] Etienne de Klerk, Fernando M. de Oliveira Filho, and Dmitrii V. Pasechnik. “Relaxations of Combinatorial Problems Via Association Schemes”. In: *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer US, Sept. 2011, pp. 171–199. DOI: 10.1007/978-1-4614-0769-0_7 (cit. on p. 74).

[KPS07] Etienne de Klerk, Dmitrii V. Pasechnik, and Alexander Schrijver. “Reduction of symmetric semidefinite programs using the regular $*$-representation”. In: *Mathematical Programming* 109.2-3 (Oct. 2007), pp. 613–624. DOI: 10.1007/s10107-006-0039-7 (cit. on p. 96).

[KPS08] Etienne de Klerk, Dmitrii V. Pasechnik, and Renata Sotirov. “On Semidefinite Programming Relaxations of the Traveling Salesman Problem”. In: *SIAM Journal on Optimization* 19.4 (Jan. 2008), pp. 1559–1573. DOI: 10.1137/070711141 (cit. on p. 73).

[KS10] Etienne de Klerk and Renata Sotirov. “Exploiting group symmetry in semidefinite programming relaxations of the quadratic assignment problem”. In: *Mathematical Programming* 122.2 (2010). DOI: 10.1007/s10107-008-0246-5 (cit. on pp. 22, 54, 66, 74).

[KS12] Etienne de Klerk and Renata Sotirov. “Improved semidefinite programming bounds for quadratic assignment problems with suitable symmetry”. In: *Mathematical Programming* 133.1-2 (2012), pp. 75–91. DOI: 10.1007/s10107-010-0411-5 (cit. on pp. 22, 54, 66).

[KST54] Tamás Kővári, Vera T. Sós, and Paul Turán. “On a problem of K. Zarankiewicz”. In: *Colloquium Mathematicum* 3 (1954), pp. 50–57 (cit. on pp. 198, 207, 208).

[KV16] Etienne de Klerk and Frank Vallentin. “On the Turing Model Complexity of Interior Point Methods for Semidefinite Programming”. In: *SIAM Journal on Optimization* 26.3 (Jan. 2016), pp. 1944–1961. DOI: 10.1137/15m103114x (cit. on p. 7).

[Las01] Jean B. Lasserre. “An Explicit Exact SDP Relaxation for Nonlinear 0-1 Programs”. In: *Integer Programming and Combinatorial Optimization*. Ed. by Karen Aardal and Bert Gerards. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 293–303. ISBN: 978-3-540-45535-6 (cit. on p. 11).

[Lau03] Monique Laurent. “A Comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre Relaxations for 0–1 Programming”. In: *Mathematics of Operations Research* 28.3 (Aug. 2003), pp. 470–496. DOI: 10.1287/moor.28.3.470.16391 (cit. on p. 139).

[Lau07] Monique Laurent. “Strengthened semidefinite programming bounds for codes”. In: *Mathematical Programming* 109.2-3 (2007), pp. 239–261. DOI: 10.1007/s10107-006-0030-3 (cit. on p. 96).

[Lob+98] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. “Applications of second-order cone programming”. In: *Linear Algebra and its Applications* 284.1-3 (Nov. 1998), pp. 193–228. DOI: 10.1016/s0024-3795(98)10032-0 (cit. on p. 75).

[Lof09] Johan Lofberg. “Pre- and Post-Processing Sum-of-Squares Programs in Practice”. In: *IEEE Transactions on Automatic Control* 54.5 (2009), pp. 1007–1011. DOI: 10.1109/TAC.2009.2017144 (cit. on p. 23).

[Loi+07] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. “A survey for the quadratic assignment problem”. In: *European Journal of Operational Research* 176.2 (Jan. 2007), pp. 657–690. DOI: 10.1016/j.ejor.2005.09.032 (cit. on p. 44).

[Lov12] László Lovász. *Large Networks and Graph Limits*. American Mathematical Society, Dec. 2012. DOI: 10.1090/coll/060 (cit. on pp. 134, 139, 144).

[Lov79] L. Lovasz. "On the Shannon capacity of a graph". In: *IEEE Transactions on Information Theory* 25.1 (1979), pp. 1–7. DOI: 10.1109/TIT.1979.1055985 (cit. on p. 8).

[LPS17] Bart Litjens, Sven Polak, and Alexander Schrijver. "Semidefinite bounds for nonbinary codes based on quadruples". In: *Designs, Codes and Cryptography* 84.1-2 (2017), pp. 87–100. DOI: 10.1007/s10623-016-0216-5 (cit. on p. 96).

[LS91] L. Lovász and A. Schrijver. "Cones of Matrices and Set-Functions and 0–1 Optimization". In: *SIAM Journal on Optimization* 1.2 (May 1991), pp. 166–190. DOI: 10.1137/0801013 (cit. on p. 11).

[LW68] AA Leman and B Weisfeiler. "A reduction of a graph to a canonical form and an algebra arising during this reduction". In: *Nauchno-Technicheskaya Informatsiya* 2.9 (1968), pp. 12–16 (cit. on pp. 29, 54).

[Mac95] Ian Grant Macdonald. *Symmetric functions and Hall polynomials*. Oxford university press, 1995 (cit. on p. 174).

[Man07] Willem Mantel. "Problem 28". In: *Wiskundige Opgaven* 10.60-61 (1907), p. 320 (cit. on p. 130).

[MM10] Takanori Maehara and Kazuo Murota. "A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with general irreducible components". In: *Japan J. Indust. Appl. Math.* 27.2 (May 2010), pp. 263–293. DOI: 10.1007/s13160-010-0007-8 (cit. on p. 38).

[MP14] Brendan D. McKay and Adolfo Piperno. "Practical graph isomorphism, II". In: *Journal of Symbolic Computation* 60 (Jan. 2014), pp. 94–112. DOI: 10.1016/j.jsc.2013.09.003 (cit. on pp. 186, 211).

[MR12] Jérôme Malick and Frédéric Roupin. "Solving k-cluster problems to optimality with semidefinite programming". In: *Mathematical Programming* 136.2 (Oct. 2012), pp. 279–300. DOI: 10.1007/s10107-012-0604-1 (cit. on p. 63).

[MRK20] Mikhail Muzychuk, Sven Reichard, and Mikhail Klin. “On Jordan schemes”. In: *arXiv preprint arXiv:1912.04551* (2020) (cit. on p. 29).

[MRV21] Philippe Moustrou, Cordian Riener, and Hugues Verdure. “Symmetric ideals, Specht polynomials and solutions to symmetric systems of equations”. In: *Journal of Symbolic Computation* 107 (2021), pp. 106–121 (cit. on p. 191).

[Mur+10] Kazuo Murota, Yoshihiro Kanno, Masakazu Kojima, and Sadayoshi Kojima. “A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with application to semidefinite programming”. In: *Japan Journal of Industrial and Applied Mathematics* 27.1 (May 2010), pp. 125–160. DOI: 10.1007/s13160-010-0006-9 (cit. on pp. 37, 55, 227).

[Nik10] V. Nikiforov. “The number of cliques in graphs of given order and size”. In: *Transactions of the American Mathematical Society* 363.3 (Oct. 2010), pp. 1599–1618. DOI: 10.1090/s0002-9947-2010-05189-x (cit. on p. 198).

[OEI99a] OEIS. *The On-Line Encyclopedia of Integer Sequences*. Published electronically at https://oeis.org. 1999. URL: https://oeis.org/A050535 (cit. on pp. 96, 122, 123).

[OEI99b] OEIS. *The On-Line Encyclopedia of Integer Sequences*. Published electronically at https://oeis.org. 1999. URL: https://oeis.org/A000088 (cit. on p. 142).

[Per17] Frank Noble Permenter. “Reduction methods in semidefinite and conic optimization”. PhD thesis. Massachusetts Institute of Technology, 2017. URL: https://dspace.mit.edu/handle/1721.1/114005 (cit. on pp. xiii, 24–30, 54, 67).

[Pol19] S.C. Polak. “New methods in coding theory: error-correcting codes and the Shannon capacity”. PhD thesis. University of Amsterdam, 2019 (cit. on pp. 160, 168, 169).

[Pol22] Sven C. Polak. “Symmetry Reduction to Optimize a Graph-based Polynomial From Queueing Theory”. In: *SIAM Journal on Applied Algebra and Geometry* 6.2 (May 2022), pp. 243–266. DOI: 10.1137/21m1413298 (cit. on pp. 114, 122).

[PP19] Frank Permenter and Pablo A. Parrilo. “Dimension reduction for semidefinite programs via Jordan algebras”. In: *Mathematical Programming* 181.1 (Mar. 2019), pp. 51–84. DOI: 10.1007/s10107-019-01372-5 (cit. on pp. 17, 22, 23, 25, 27, 60, 66).

[PR09] Janez Povh and Franz Rendl. “Copositive and semidefinite relaxations of the quadratic assignment problem”. In: *Discrete Optimization* 6.3 (Aug. 2009), pp. 231–241. DOI: 10.1016/j.disopt.2009.01.002 (cit. on pp. 22, 44, 46, 49).

[Put93] Mihai Putinar. “Positive Polynomials on Compact Semi-algebraic Sets”. In: *Indiana University Mathematics Journal* 42.3 (1993), pp. 969–984. ISSN: 00222518, 19435258 (cit. on p. 14).

[Ray+18] Annie Raymond, James Saunderson, Mohit Singh, and Rekha R. Thomas. “Symmetric sums-of-squares over k-subset hypercubes”. In: *Mathematical Programming* 167.2 (2018), pp. 315–354. DOI: 10.1007/s10107-017-1127-6 (cit. on pp. 19, 96, 129–131, 139, 143, 144, 171, 172, 174, 178, 207, 208, 223, 225).

[Raz07] Alexander A. Razborov. “Flag algebras”. In: *Journal of Symbolic Logic* 72.4 (Dec. 2007), pp. 1239–1282. DOI: 10.2178/jsl/1203350785 (cit. on pp. 18, 19, 96, 129, 130, 137, 138, 142, 144, 213, 214).

[Raz08] Alexander A. Razborov. “On the Minimal Density of Triangles in Graphs”. In: *Combinatorics, Probability and Computing* 17.4 (July 2008), pp. 603–618. DOI: 10.1017/s0963548308009085 (cit. on pp. 198–201).

[Raz13] Alexander A. Razborov. “Flag algebras: An Interim Report”. In: *The Mathematics of Paul Erdős II*. Springer New York, 2013, pp. 207–232. DOI: 10.1007/978-1-4614-7254-4_16 (cit. on pp. 129, 223).

[Rei16] Christian Reiher. “The clique density theorem”. In: *Annals of Mathematics* (2016), pp. 683–707 (cit. on p. 198).

[Ren01] James Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. Society for Industrial and Applied Mathematics, Jan. 2001. DOI: 10.1137/1.9780898718812 (cit. on p. 3).

[Rie+13] Cordian Riener, Thorsten Theobald, Lina Jansson Andrén, and Jean B. Lasserre. “Exploiting Symmetries in SDP-Relaxations for Polynomial Optimization”. In: *Mathematics of Operations Research* 38.1 (Feb. 2013), pp. 122–141. DOI: 10.1287/moor.1120.0558 (cit. on pp. 96, 172, 173, 178).

[Rie12] Cordian Riener. “On the degree and half-degree principle for symmetric polynomials”. In: *Journal of Pure and Applied Algebra* 216.4 (Apr. 2012), pp. 850–856. DOI: 10.1016/j.jpaa.2011.08.012 (cit. on p. 96).

[RST18] Annie Raymond, Mohit Singh, and Rekha R. Thomas. “Symmetry in Turán sums-of-squares polynomials from flag algebras”. In: *Algebraic Combinatorics* 1.2 (2018), pp. 249–274. DOI: 10.5802/alco.5 (cit. on p. 96).

[Sag13] Bruce E Sagan. *The symmetric group: representations, combinatorial algorithms, and symmetric functions*. Vol. 203. Springer Science & Business Media, 2013 (cit. on pp. 163, 180, 181, 184).

[Sch05] A. Schrijver. “New Code Upper Bounds From the Terwilliger Algebra and Semidefinite Programming”. In: *IEEE Transactions on Information Theory* 51.8 (Aug. 2005), pp. 2859–2866. DOI: 10.1109/tit.2005.851748 (cit. on pp. 96, 104).

[Sch79] A. Schrijver. “A comparison of the Delsarte and Lovász bounds”. In: *IEEE Transactions on Information Theory* 25.4 (July 1979), pp. 425–429. DOI: 10.1109/tit.1979.1056072 (cit. on pp. 21, 22, 96).

[Ser03] Ákos Seress. *Permutation Group Algorithms*. Cambridge Tracts in Mathematics. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511546549 (cit. on p. 185).

[Ser77] Jean-Pierre Serre. *Linear Representations of Finite Groups*. Springer New York, 1977. DOI: 10.1007/978-1-4684-9458-7 (cit. on p. 157).

[Sid93] Alexander Sidorenko. “A correlation inequality for bipartite graphs”. In: *Graphs and Combinatorics* 9.2-4 (June 1993), pp. 201–204. DOI: 10.1007/bf02988307 (cit. on p. 153).

[Sim70] Charles C. Sims. “Computational methods in the study of permutation groups††This research was supported in part by the National Science Foundation.” In: *Computational Problems in Abstract Algebra*. Ed. by John Leech. Pergamon, 1970, pp. 169–183. ISBN: 978-0-08-012975-4. DOI: https://doi.org/10.1016/B978-0-08-012975-4.50020-5 (cit. on p. 185).

[SL21] Lucas Slot and Monique Laurent. “Sum-of-Squares Hierarchies for Binary Polynomial Optimization”. In: *Integer Programming and Combinatorial Optimization*. Ed. by Mohit Singh and David P. Williamson. Cham: Springer International Publishing, 2021, pp. 43–57. ISBN: 978-3-030-73879-2 (cit. on pp. 147, 148).

[SS18] Jakub Sliacan and Walter Stromquist. “Improving bounds on packing densities of 4-point permutations”. In: *Discrete Mathematics & Theoretical Computer Science* Vol. 19 no. 2, Permutation Patterns 2016 (Feb. 2018). DOI: 10.23638/DMTCS-19-2-3. URL: https://dmtcs.episciences.org/4263 (cit. on p. 129).

[SS78] Lynn Arthur Steen and J Arthur Seebach. *Counterexamples in topology*. Vol. 7. Springer, 1978 (cit. on p. 131).

[Tai95] Éric D. Taillard. “Comparison of iterative searches for the quadratic assignment problem”. In: *Location Science* 3.2 (Aug. 1995), pp. 87–105. DOI: 10.1016/0966-8349(95)00008-6 (cit. on pp. 59, 60).

[TRR19] Armin Tavakoli, Denis Rosset, and Marc-Olivier Renou. “Enabling Computation of Correlation Bounds for Finite-Dimensional Quantum Systems via Symmetrization”. In: *Physical review letters* 122.7 (Feb. 2019), p. 070501. DOI: 10.1103/physrevlett.122.070501 (cit. on p. 23).

[Tru14] U. Truetsch. “A semidefinite programming based branch-and-bound framework for the quadratic assignment problem”. English. PhD thesis. Tilburg University, Oct. 2014. ISBN: 9789056684068. URL: https://research.tilburguniversity.edu/en/publications/a-semidefinite-programming-based-branch-and-bound-framework-for-t (cit. on p. 50).

[Tyc35] A. Tychonoff. “Über einen Funktionenraum”. In: *Mathematische Annalen* 111.1 (Dec. 1935), pp. 762–766. DOI: 10.1007/bf01472255 (cit. on p. 131).

[Val08] Frank Vallentin. “Lecture notes: Semidefinite programs and harmonic analysis”. In: *arXiv preprint arXiv:0809.2017* (Sept. 2008) (cit. on p. 148).

[Via17] Maryna Viazovska. “The sphere packing problem in dimension 8”. In: *Annals of Mathematics* 185.3 (May 2017). DOI: 10.4007/annals.2017.185.3.7 (cit. on p. 147).

[Wat97] G.I. Watson. “Repulsive particles on a two-dimensional lattice”. In: *Physica A: Statistical Mechanics and its Applications* 246.1-2 (Nov. 1997), pp. 253–274. DOI: 10.1016/s0378-4371(97)00349-x (cit. on p. 59).

[Wed08] JH Wedderburn. “On hypercomplex numbers”. In: *Proceedings of the London Mathematical Society* 2.1 (1908), pp. 77–118 (cit. on p. 17).

[Wed34] J. Wedderburn. *Lectures on Matrices*. American Mathematical Society, Dec. 1934. DOI: 10.1090/coll/017 (cit. on p. 103).

[Zha+98] Qing Zhao, Stefan E. Karisch, Franz Rendl, and Henry Wolkowicz. “Semidefinite programming relaxations for the quadratic assignment problem.” In: *Journal of Combinatorial Optimization* 2.1 (1998), pp. 71–109. DOI: 10.1023/a:1009795911987 (cit. on pp. 22, 44, 46).

CENTER DISSERTATION SERIES

CentER for Economic Research, Tilburg University, the Netherlands

No.	Author	Title	ISBN	Published
638	Pranav Desai	Essays in Corporate Finance and Innovation	978 90 5668 639 0	January 2021
639	Kristy Jansen	Essays on Institutional Investors, Asset Allocation Decisions, and Asset Prices	978 90 5668 640 6	January 2021
640	Riley Badenbroek	Interior Point Methods and Simulated Annealing for Nonsymmetric Conic Optimization	978 90 5668 641 3	February 2021
641	Stephanie Koornneef	It's about time: Essays on temporal anchoring devices	978 90 5668 642 0	February 2021
642	Vilma Chila	Knowledge Dynamics in Employee Entrepreneurship: Implications for parents and offspring	978 90 5668 643 7	March 2021
643	Minke Remmerswaal	Essays on Financial Incentives in the Dutch Healthcare System	978 90 5668 644 4	July 2021
644	Tse-Min Wang	Voluntary Contributions to Public Goods: A multi-disciplinary examination of prosocial behavior and its antecedents	978 90 5668 645 1	March 2021
645	Manwei Liu	Interdependent individuals: how aggregation, observation, and persuasion affect economic behavior and judgment	978 90 5668 646 8	March 2021
646	Nick Bombaij	Effectiveness of Loyalty Programs	978 90 5668 647 5	April 2021
647	Xiaoyu Wang	Essays in Microeconomics Theory	978 90 5668 648 2	April 2021
648	Thijs Brouwer	Essays on Behavioral Responses to Dishonest and Anti-Social Decision-Making	978 90 5668 649 9	May 2021
649	Yadi Yang	Experiments on hold-up problem and delegation	978 90 5668 650 5	May 2021
650	Tao Han	Imperfect information in firm growth strategy: Three essays on M&A and FDI activities	978 90 5668 651 2	June 2021

No.	Author	Title	ISBN	Published
651	Johan Bonekamp	Studies on labour supply, spending and saving before and after retirement	978 90 5668 652 9	June 2021
652	Hugo van Buggenum	Banks and Financial Markets in Microfounded Models of Money	978 90 5668 653 6	August 2021
653	Arthur Beddock	Asset Pricing with Heterogeneous Agents and Non-normal Return Distributions	978 90 5668 654 3	September 2021
654	Mirron Adriana Boomsma	On the transition to a sustainable economy: Field experimental evidence on behavioral interventions	978 90 5668 655 0	September 2021
655	Roweno Heijmans	On Environmental Externalities and Global Games	978 90 5668 656 7	August 2021
656	Lenka Fiala	Essays in the economics of education	978 90 5668 657 4	September 2021
657	Yuexin Li	Pricing Art: Returns, Trust, and Crises	978 90 5668 658 1	September 2021
658	Ernst Roos	Robust Approaches for Optimization Problems with Convex Uncertainty	978 90 5668 659 8	September 2021
659	Joren Koëter	Essays on asset pricing, investor preferences and derivative markets	978 90 5668 660 4	September 2021
660	Ricardo Barahona	Investor Behavior and Financial Markets	978 90 5668 661 1	October 2021
660	Stefan ten Eikelder	Biologically-based radiation therapy planning and adjustable robust optimization	978 90 5668 662 8	October 2021
661	Maciej Husiatyński	Three essays on Individual Behavior and New Technologies	978 90 5668 663 5	October 2021
662	Hasan Apakan	Essays on Two-Dimensional Signaling Games	978 90 5668 664 2	October 2021
663	Ana Moura	Essays in Health Economics	978 90 5668 665 9	November 2021
664	Frederik Verplancke	Essays on Corporate Finance: Insights on Aspects of the General Business Environment	978 90 5668 666 6	October 2021

No.	Author	Title	ISBN	Published
665	Zhaneta Tancheva	Essays on Macro-Finance and Market Anomalies	978 90 5668 667 3	November 2021
666	Claudio Baccianti	Essays in Economic Growth and Climate Policy	978 90 5668 668 0	November 2021
667	Hongwei Zhang	Empirical Asset Pricing and Ensemble Machine Learning	978 90 5668 669 7	November 2021
668	Bart van der Burgt	Splitsing in de Wet op de vennootschapsbelasting 1969 Een evaluatie van de Nederlandse winstbelastingregels voor splitsingen ten aanzien van lichamen	978 90 5668 670 3	December 2021
669	Martin Kapons	Essays on Capital Markets Research in Accounting	978 90 5668 671 0	December 2021
670	Xolani Nghona	From one dominant growth mode to another: Switching between strategic expansion modes	978 90 5668 672 7	December 2021
671	Yang Ding	Antecedents and Implications of Legacy Divestitures	978 90 5668 673 4	December 2021
672	Joobin Ordoobody	The Interplay of Structural and Individual Characteristics	978 90 5668 674 1	February 2022
673	Lucas Avezum	Essays on Bank Regulation and Supervision	978 90 5668 675 8	March 2022
674	Oliver Wichert	Unit-Root Tests in High-Dimensional Panels	978 90 5668 676 5	April 2022
675	Martijn de Vries	Theoretical Asset Pricing under Behavioral Decision Making	978 90 5668 677 2	June 2022
676	Hanan Ahmed	Extreme Value Statistics using Related Variables	978 90 5668 678 9	June 2022
677	Jan Paulick	Financial Market Information Infrastructures: Essays on Liquidity, Participant Behavior, and Information Extraction	978 90 5668 679 6	June 2022
678	Freek van Gils	Essays on Social Media and Democracy	978 90 5668 680 2	June 2022

No.	Author	Title	ISBN	Published
679	Suzanne Bies	Examining the Effectiveness of Activation Techniques on Consumer Behavior in Temporary Loyalty Programs	978 90 5668 681 9	July 2022
680	Qinnan Ruan	Management Control Systems and Ethical Decision Making	978 90 5668 682 6	June 2022
681	Lingbo Shen	Essays on Behavioral Finance and Corporate Finance	978 90 5668 683 3	August 2022
682	Joshua Eckblad	Mind the Gales: An Attention-Based View of Startup Investment Arms	978 90 5668 684 0	August 2022
683	Rafael Greminger	Essays on Consumer Search	978 90 5668 685 7	August 2022
684	Suraj Upadhyay	Essay on policies to curb rising healthcare expenditures	978 90 5668 686 4	September 2022
685	Bert-Jan Butijn	From Legal Contracts to Smart Contracts and Back Again: An Automated Approach	978 90 5668 687 1	September 2022
686	Sytse Duiverman	Four essays on the quality of auditing: Causes and consequences	978 90 5668 688 8	October 2022
687	Lucas Slot	Asymptotic Analysis of Semidefinite Bounds for Polynomial Optimization and Independent Sets in Geometric Hypergraphs	978 90 5668 689 5	September 2022
688	Daniel Brosch	Symmetry reduction in convex optimization with applications in combinatorics	978 90 5668 690 1	October 2022

Daniel Brosch (Leverkusen, Germany, 1996) received his bachelor's and master's degrees in Mathematics with honors at the University of Cologne (2017 & 2018). After that he started as PhD candidate at the department of econometrics & OR at Tilburg University, as part of the Marie-Curie innovative training network MINOA.

This dissertation explores different approaches to and applications of symmetry reduction in convex optimization. Using tools from semidefinite programming, representation theory and algebraic combinatorics, hard combinatorial problems are solved or bounded. The first chapters consider the Jordan reduction method, extend the method to optimization over the doubly nonnegative cone, and apply it to quadratic assignment problems and energy minimization on a discrete torus. The following chapter uses symmetry reduction as a proving tool, to approach a problem from queuing theory with redundancy scheduling. The final chapters propose generalizations and reductions of flag algebras, a powerful tool for problems coming from extremal combinatorics.

ISBN: 978 905668 690 1
DOI: 10.26116/12v4-k024

TILBURG SCHOOL OF ECONOMICS
AND MANAGEMENT

CentER Dissertation Series